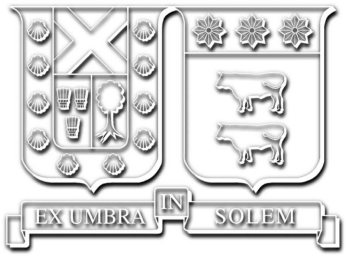
Universidad Técnica Federico Santa María
Departamento de Ingeniería Eléctrica
Valparaíso, Chile



Implementación de Inteligencia Artificial para mejorar el Control Predictivo de Máquina Asincrónica

Francis Llanca Flores

2026

Requisito parcial para obtener el título de:
Ingeniero Civil Electricista
Profesora Guía:
Dra.-Ing. Margarita Norambuena (UTFSM)

Comisión:
Dra.-Ing. Margarita Norambuena (UTFSM)

Valparaíso, octubre, 2026.

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Implementación de Inteligencia Artificial para mejorar el Control Predictivo de Máquina Asincrónica

Nombre del candidato(a): Francis Nicolás Llanca Flores

Carrera / Grado: Ingeniería Civil Eléctrica

Campus: Casa Central Valparaíso **Departamento:** Departamento de Electricidad

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Margarita Norambuena Valdivia, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

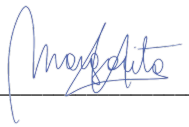
☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción)**:

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

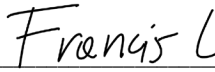
Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

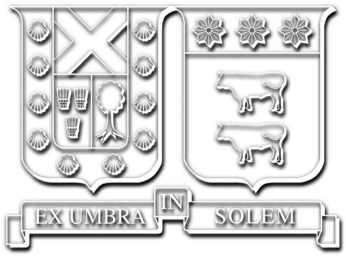
Fecha: 23.10.2026 **Firma:** 

Estudiante o Candidato(a):

Fecha: 23/10/2026 **Firma:** 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Universidad Técnica Federico Santa María
Departamento de Ingeniería Eléctrica
Valparaíso, Chile



Implementación de Inteligencia Artificial para mejorar el Control Predictivo de Máquina Asincrónica

Francis Llanca Flores

2026

"Y quién es tu enemigo?"
"Escúchame bien Thorfinn; tú no tienes enemigos.
Nadie tiene enemigos.
No hay nadie a quien debas lastimar."
"Un verdadero guerrero no necesita espada."
— Thors, Vinland Saga.

Agradecimientos

Hace siete años no sabía realmente qué llegaría a ser ni tenía certeza de qué quería estudiar. En los primeros años llegué a pensar que quizás había cometido un error al elegir esta carrera y que mi paso por la universidad sería únicamente una experiencia académica, sin vínculos importantes con las personas ni con la comunidad que me rodeaba. Hoy, cuando este período termina, puedo mirar hacia atrás y reconocer que estaba equivocado.

En primer lugar, agradezco a mis padres, Mabel Flores y Francis Llanca, su cariño y apoyo incondicional hicieron posible que llegara hasta aquí. Este logro también les pertenece, porque nada de esto habría sido posible sin todo lo que me han entregado a lo largo de mi vida.

Quiero agradecer a todos los amigos que conocí durante la carrera, ellos me demostraron que existen personas genuinas y dispuestas a apoyarse para alcanzar sus objetivos. Con ellos compartí jornadas de desvelo, trabajos interminables, certámenes, derrotas y victorias. Gracias por acompañarme durante estos años y por hacer que este camino fuera mucho más significativo.

Extiendo mi agradecimiento a todos los profesores y docentes que participaron en mi formación como ingeniero civil electricista, así como al personal de pañol. Cada uno, desde su labor y su manera de enseñar, contribuyó a desarrollar habilidades y valores profesionales con los que hoy termino esta etapa. De manera especial, agradezco a la profesora Margarita Norambuena, mi profesora guía, por su orientación, apoyo y paciencia durante el desarrollo de esta memoria.

Antes de finalizar, quiero también agradecer a quienes me acompañaron durante uno de los momentos más difíciles de mi etapa universitaria. Un año antes de concluir este trabajo, un siniestro inesperado me atacó profundamente y me hizo pensar directamente en congelar. Sin embargo, frente a esa situación, muchas personas me demostraron que no estaba solo.

Agradezco profundamente al Centro de Alumnos, a mis profesores, al personal del pañol, a la red de exAlumnos y a todas las personas que colaboraron conmigo. Su apoyo económico, académico y emocional me permitió continuar y terminar un año que, en algún momento, consideré perdido. Más allá de la ayuda recibida, siempre conservaré la certeza de ser parte de una comunidad capaz de ayudar y sostener a uno de sus integrantes cuando más lo necesita.

Hoy cierro esta etapa con gratitud y con un profundo orgullo por haberme formado en la Universidad Técnica Federico Santa María.

Índice de Contenidos

Índice de Contenidos I

Índice de Figuras VII

Índice de Tablas X

Nomenclatura XIII

Resumen 1

1. Introducción 2
1.1. Contexto, problema y propuesta del trabajo . . . 2
1.2. Objetivos del trabajo . . . 3
1.2.1. Objetivo general . . . 3
1.2.2. Objetivos específicos . . . 3
1.2.3. Objetivo adicional . . . 4
1.3. Estructura del documento . . . 4

2. Marco Teórico 6
2.1. Modelo dinámico de la máquina de inducción . . . 6
2.1.1. Hipótesis de modelado . . . 6
2.1.2. Fasores espaciales y representación de la máquina trifásica . . . 6
2.1.3. Sistema estacionario: coordenadas $\alpha\beta$ . . . 8
2.1.4. Transformación a sistemas rotatorios: coordenadas dq . . . 9
2.1.5. Ecuaciones eléctricas de la máquina de inducción . . . 10
2.1.5.a. Ecuaciones eléctricas de la IM: coordenadas dq . . . 10
2.1.6. Ecuación mecánica y torque electromagnético . . . 11
2.1.7. Modelo en coordenadas de campo orientado . . . 12
2.1.7.a. Coordenadas de campo: corriente de estator $\boldsymbol{i}_s$ y flujo de estator $\boldsymbol{\psi}_s$ . . . 14
2.1.8. Estimadores del flujo de rotor $\boldsymbol{\psi}_r$ . . . 14
2.1.8.a. Modelo de tensión . . . 15
2.1.8.b. Modelo de corriente . . . 15
2.2. Principio del control predictivo basado en modelo . . . 16
2.2.1. Control predictivo basado en modelo con conjunto de control finito . 17
2.3. Estructura de control realimentado . . . 17
2.3.1. El lazo cerrado . . . 18
2.3.2. El controlador PI . . . 18

2.3.3. Saturación de la actuación: *windup* y anti-*windup* 19
2.4. Redes neuronales artificiales (ANN) 20
2.4.1. Previo a una red: la neurona artificial 20
2.4.1.a. Neurona perceptrón 21
2.4.1.b. Neurona sigmoide 22
2.4.2. Arquitectura de una red 22
2.4.2.a. Redes multicapa hacia adelante (*Multilayer*) 23
2.4.3. Entrenamiento: cómo aprende una red? 24
2.4.3.a. Ajuste de pesos y sesgos: descenso de *gradiente* ∇C 24
2.4.3.b. Minilotes: por qué no se usa todo el conjunto? 25
2.4.3.c. El optimizador 25
2.4.4. Entrenamiento: cómo se calculan los gradientes por minilote? 26
2.4.4.a. Etapa 1: propagación hacia adelante 26
2.4.4.b. Etapa 2: el error en la capa de salida 26
2.4.4.c. Etapa 3: propagación del error hacia atrás 27
2.4.4.d. Etapa 4: obtención de las derivadas pesos y sesgos 27
2.4.4.e. Síntesis: algoritmo de entrenamiento completo 28
2.4.5. Limitaciones de las redes puramente basadas en datos 28
2.5. Redes neuronales informadas por física (PINN) 29
2.5.1. Qué hace a una red "informada por física" 29
2.5.2. Familias de PINN 29
2.5.3. La PINN inversa canónica 30
2.5.3.a. Un ejemplo: identificación de un circuito RLC 30
2.5.4. Propuestas de capas correctivas basadas en redes neuronales 32
2.5.4.a. Sistema de varias redes: des-sesgar un estimador afín al error . 32
2.5.4.b. Una sola red: estimador inverso amortizado 34

3. Implementación del control predictivo de torque y flujo (PTC) 36
3.1. Arquitectura del control predictivo 36
3.2. El inversor 2L-VSI como conjunto finito de actuaciones 37
3.2.1. Zona lineal y sobremodulación: la cota que impone el hexágono . . . 38
3.3. Lazo externo de velocidad: diseño del controlador PI 39
3.4. Flujo de trabajo: control predictivo (PTC) 40
3.4.1. Etapa de estimación 41
3.4.2. Etapa de predicción a $k+1$ 42
3.4.2.a. Compensación del retardo computacional: predicción a k+2 . . 42
3.4.3. Etapa de optimización: función de costo sobre $k+2$ 43
3.4.3.a. Referencias adaptadas al punto de operación 44
3.4.4. Etapa adicional: régimen de premagnetización de la máquina 46
3.4.4.a. Protección contra sobre-corriente 46
3.5. Naturaleza del esquema: predictivo en coordenadas de campo 47
3.5.1. Similitudes y diferencias: FOC y FCS-MPC/PTC 47
3.5.2. Debilitamiento de campo: referencia de flujo dependiente de la velocidad 48
3.5.3. Límite de torque eléctrico impuesto por el flujo de estator 48
3.6. Sensibilidad paramétrica del control predictivo 49
3.6.1. Propagación del desajuste de R_r en la cadena de control 49
3.6.2. Soluciones clásicas y sus limitaciones 49

4. Análisis del efecto del desajuste paramétrico en el PTC 51
4.1. Implementación del PTC en la plataforma PLECS . . . 51
4.1.1. Parámetros de la máquina y del controlador . . . 52
4.1.2. Modelado físico de la carga mecánica . . . 53
4.1.3. Separación entre la resistencia de la máquina y la del controlador . . 53
4.2. Caracterización del comportamiento bajo desajuste: máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$) . . . 54
4.2.1. Barridos con carga de cinta . . . 54
4.2.1.a. Arranque en vacío . . . 54
4.2.1.b. Arranque con carga de $0{,}5\,T_{nom}$ . . . 59
4.2.1.c. Escalón de carga sobre la máquina en marcha . . . 64
4.2.2. Barrido con carga de bomba . . . 69
4.2.3. Síntesis de comportamientos frente al desajuste . . . 74
4.2.3.a. El caso ajustado como patrón de comparación . . . 74
4.2.3.b. Arranque en vacío . . . 75
4.2.3.c. Arranque con carga de $0{,}5\,T_{nom}$ . . . 76
4.2.3.d. Escalón de carga sobre la máquina en marcha . . . 77
4.2.3.e. Carga de bomba . . . 79
4.2.4. Resumen de características frente al desajuste . . . 80
4.3. Caracterización del comportamiento bajo desajuste: distintos valores de $R_{r,\mathrm{MAS}}$ . . . 82
4.3.1. Qué cambia al mover $R_{r,\mathrm{MAS}}$: la curva de torque–deslizamiento . . . 82
4.3.2. Casos considerados . . . 83
4.3.3. Resultados . . . 84
4.4. Resumen del efecto del desajuste de $R_{r,\mathrm{MPC}}$ . . . 90
4.4.1. Comportamiento con subestimación ($\Delta R_r < 0$) . . . 90
4.4.1.a. Velocidad . . . 90
4.4.1.b. Corrientes . . . 90
4.4.1.c. Torque estimado . . . 90
4.4.1.d. Flujo de estator estimado . . . 91
4.4.2. Comportamiento con sobreestimación ($\Delta R_r > 0$) . . . 91
4.4.2.a. Velocidad . . . 91
4.4.2.b. Corrientes . . . 92
4.4.2.c. Torque estimado . . . 92
4.4.2.d. Flujo de estator estimado . . . 92
4.4.3. Dependencia del punto de operación . . . 92
4.4.4. Propagación analítica del error . . . 93

5. Propuesta 1: Expertos PINN para la Corrección del Modelo Predictivo 95
5.1. Construcción de un residuo físico universal . . . 95
5.1.1. Construcción del núcleo físico: el residuo $\boldsymbol{\varepsilon}$ afín del predictor . . . 95
5.1.1.a. Predictor auxiliar de corriente en coordenadas $\alpha\beta$ . . . 96
5.1.1.b. El residuo de predicción $\boldsymbol{\varepsilon}[k]$ como indicador del desajuste . . . 96
5.1.1.c. Afinidad del predictor en R_r . . . 97
5.1.1.d. Sensibilidad del residuo afín . . . 97
5.1.2. Limitación del residuo y proyección sobre un ciclo de corriente $\hat{d}$ . . . 98
5.2. Consideraciones de diseño: salidas, expertos por maniobra y entradas . . . 100
5.2.1. Las salidas: corrección, confianza y dominio . . . 100

5.2.1.a. Determinación del dominio: etiquetas, rangos y estados 102
5.2.2. Expertos independientes por maniobra 103
5.2.3. La asimetría de observabilidad: por qué se dividen dos redes por maniobra 104
5.2.4. El vector de entrada de los expertos 105
5.2.4.a. El vector canónico de 141 componentes 105
5.2.4.b. Las 27 características base 106
5.2.4.c. Ajuste de la entrada por sentido de giro 109
5.2.5. Estructura preliminar: cuatro redes y dos coordinadores 111
5.2.5.a. Diferencias: maniobra de carga vs el arranque 112
5.2.5.b. Coordinador de arranque y algoritmo físico: *breakaway* 112
5.2.5.c. Coordinador dual de carga y vía de emergencia 114
5.2.6. La escritura del parámetro: guardián 116
5.2.6.a. Algoritmo de escritura: guardián. 116
5.2.6.b. Las dos puertas de escritura 117
5.2.6.c. Cierre del lazo sobre el controlador 118
5.2.7. Estructura final: comparativa de los dos expertos 118
5.3. Redes MLP: arquitectura y entrenamiento de las redes PINN 119
5.3.1. Arquitectura y salidas de las cuatro redes 119
5.3.2. La función de pérdida 120
5.3.3. Entrenamiento de las redes PINN 121
5.3.3.a. Entrenamiento del experto de arranque 122
5.3.3.b. Entrenamiento del experto de carga 123
5.4. Implementación en línea en la plataforma PLECS 125
5.4.1. Ubicación en el lazo del PTC y despliegue en C 125
5.4.2. Flujo 1: mediciones de un ciclo hasta el primer vector de entrada . . 125
5.4.2.a. Memoria de un ciclo válido 126
5.4.2.b. Memorias entre ciclos y construcción de los vectores de entrada 127
5.4.3. Flujo 2: trabajo de las redes neuronales 128
5.4.4. Flujo 3: capa correctiva online 129
5.4.5. Cierre del lazo sobre el controlador 129
5.4.5.a. Sintesis: secuencia completa de la operación en línea 129
5.4.6. Configuración final de entrenamiento y ejecución 130

6. Propuesta 2: Una Sola Red como Estimador Inverso Amortizado 132
6.1. Construcción de una red neuronal como estimador de desajuste 132
6.1.1. Mismo principio de corrección 132
6.1.2. Estimación directa de la red desde la forma de onda 132
6.2. Diseño de las entradas y salidas 133
6.2.1. Las salidas de la capa y el techo de la corrección 133
6.2.1.a. Zona muerta 134
6.2.2. Entrada: ventana de tiempo con seis canales 134
6.2.2.a. De la ventana al descriptor: una red convolucional 135
6.2.3. Adicional a la ventana: vector de contexto 136
6.2.4. Estructura final: red más algoritmo de escritura 137
6.2.4.a. Bloqueo en marcha invertida 137
6.3. Arquitectura y entrenamiento 138
6.3.1. Arquitectura de la red 138

6.3.2. La función de pérdida . . . 140
6.3.3. Los escenarios del conjunto de datos . . . 142
6.4. La capa correctora en línea . . . 145
6.4.1. Cadencia y aplicación de la corrección . . . 145
6.4.2. Configuración final de entrenamiento y ejecución . . . 146

7. Resultados y Análisis 148
7.1. Resultados de la Propuesta 1: el sistema de expertos . . . 148
7.1.1. Organización de la evaluación . . . 148
7.1.2. Rutina cinta a velocidad nominal: resistencia de rotor nominal 0.83 Ω150
7.1.3. Rutina completa a velocidad nominal, para cada resistencia de rotor real . . . 154
7.1.3.a. Máquina de 0,10 Ω . . . 158
7.1.3.b. Máquina de 0,50 Ω . . . 159
7.1.3.c. Máquina de 3,00 Ω . . . 160
7.1.3.d. Máquina de 5,00 Ω . . . 161
7.1.4. Carga de bomba . . . 162
7.1.5. Modo cinta: diferente velocidad de operación . . . 166
7.1.6. Análisis de *sucesión*: quién corrige el parámetro . . . 168
7.2. Resultados de la Propuesta 2: estimador inverso de una sola red . . . 170
7.2.1. Rutina cinta a velocidad nominal . . . 170
7.2.2. Carga de bomba . . . 174
7.3. Comparación: evaluación sobre una rutina completa . . . 178
7.3.1. Rutinas con resistencia nominal . . . 180
7.3.1.a. Sin corrección . . . 181
7.3.1.b. Propuesta 1 . . . 183
7.3.1.c. Propuesta 2 . . . 185
7.3.2. Rutinas con resistencia de rotor de 3,00 Ω . . . 186
7.3.2.a. Sin corrección . . . 187
7.3.2.b. Propuesta 1 . . . 188
7.3.2.c. Propuesta 2 . . . 191
7.3.3. Lectura conjunta . . . 192
7.4. Síntesis comparativa de las dos propuestas . . . 193

8. Conclusiones 196
8.1. Síntesis y discusión . . . 196
8.2. Ventajas y desventajas de cada propuesta . . . 197
8.3. Contribuciones . . . 199
8.4. Trabajo futuro . . . 199

Apéndices 201

A. Parámetros e Inicialización de la Simulación 201

B. Código de los Bloques C-Script 205
B.1. C-Script principal: Code Declarations . . . 205
B.2. C-Script principal: Start Function . . . 211
B.3. C-Script principal: Output Function . . . 213

B.4. C-Script principal: Update Function 216
B.5. C-Script de carga física 227
B.6. Integración de los expertos en el C-Script principal 229
B.7. Vías de respaldo físicas 233

C. Implementación en Python: Núcleo Físico y Redes 239
C.1. Núcleo físico compartido 239
C.2. Arquitectura compartida de los expertos 240
C.3. Pérdida híbrida compartida 240
C.4. Flujo de reproducción 241

D. Figuras Complementarias de la Campaña de Evaluación 243
D.1. Rutina completa por resistencia real de rotor 243
D.1.1. Máquina de 0,10 Ω 244
D.1.2. Máquina de 0,50 Ω 246
D.1.3. Máquina de 3,00 Ω 248
D.1.4. Máquina de 5,00 Ω 250
D.2. Carga de bomba 252
D.3. Velocidad reducida 253

E. Tasa de Corrección frente a Ventana de Arranque 256
E.1. El síntoma 256
E.2. Las dos restricciones 256
E.3. La estimación era correcta 257
E.4. Por qué se manifiesta en una sola faceta 258

F. Comportamiento Fuera del Rango de Entrenamiento 259
F.1. Un obstáculo previo: el techo de escritura 259
F.2. Por encima del rango, con el techo elevado 260
F.3. Por debajo del rango 261
F.4. Qué se concluye y qué no 261

G. Verificaciones de la Implementación de la Propuesta 1 263
G.1. Vector de entrada: despliegue frente a entrenamiento 263
G.2. Instantes del predictor y de la sensibilidad 265
G.3. Marcha en sentido inverso 266
G.4. Aplicación de la reversión del coordinador de carga 268
G.5. El sesgo de la proyección y el alcance de rho2 268

Bibliografía 270

Índice de Figuras

2.1. Construcción del fasor espacial de corriente 7
2.2. El plano estacionario $\alpha\beta$ 8
2.3. Referencia de un fasor a un marco rotatorio 9
2.4. Modelo de eje rígido máquina–carga 12
2.5. Diagrama del modelo dinámico de la máquina 13
2.6. Estimador de flujo por modelo de tensión 15
2.7. Estimador de flujo por modelo de corriente 16
2.8. Principio de funcionamiento del FCS-MPC 16
2.9. Principio de funcionamiento del FCS-MPC 17
2.10. Estructura básica de control realimentado 18
2.11. Estructura paralela del controlador PID 19
2.12. Lazo de control con anti-*windup* 20
2.13. Modelo no lineal de una neurona artificial 21
2.14. Función de activación umbral 21
2.15. Función de activación sigmoide 22
2.16. Red multicapa hacia adelante 23

3.1. Esquema del control predictivo de torque y flujo implementado 36
3.2. El inversor 2L-VSI 37
3.3. Vectores de tensión del inversor 2L-VSI 37
3.4. Zona lineal y región de sobremodulación 38
3.5. Efecto de la compensación del retardo computacional 43

4.1. Esquema de la plataforma implementada en PLECS 52
4.2. Velocidad mecánica en arranque en vacío bajo desajuste 55
4.3. Corrientes de estator en arranque en vacío bajo desajuste 56
4.4. Torque de referencia, estimado y real en arranque en vacío 57
4.5. Magnitud del flujo de estator en arranque en vacío 58
4.6. Velocidad mecánica en arranque con carga bajo desajuste 60
4.7. Corrientes de estator en arranque con carga bajo desajuste 61
4.8. Torque de referencia, estimado y real en arranque con carga 62
4.9. Magnitud del flujo de estator en arranque con carga 63
4.10. Velocidad ante escalón de carga sobre la máquina en marcha 65
4.11. Corrientes de estator ante escalón de carga 66
4.12. Torque ante escalón de carga sobre la máquina en marcha 67
4.13. Magnitud del flujo de estator ante escalón de carga 68
4.14. Velocidad con carga de bomba bajo desajuste 70
4.15. Corrientes de estator con carga de bomba 71
4.16. Torque con carga de bomba bajo desajuste 72

4.17. Magnitud del flujo de estator con carga de bomba 73
4.18. Velocidad con carga de bomba para distintos valores de $R_{r,\mathrm{MAS}}$ 85
4.19. Corrientes de estator con carga de bomba para distintos valores de $R_{r,\mathrm{MAS}}$ 86
4.20. Corrientes de estator en equilibrio para distintos valores de $R_{r,\mathrm{MAS}}$ 87
4.21. Torque para distintos valores de $R_{r,\mathrm{MAS}}$ con carga de bomba 88
4.22. Magnitud de flujo de estator para distintos valores de $R_{r,\mathrm{MAS}}$ 89

5.1. Flujo 1: de la medición del ciclo a la escritura del parámetro 126
5.2. Flujo 2: la plantilla de una red 128
5.3. El accionamiento completo con la capa de expertos 130

6.1. Construcción de la ventana de entrada 135
6.2. Arquitectura de la red 139
6.3. El accionamiento completo con la capa iPINN 145
6.4. Capa correctora: geometría temporal y cadena de decisión 146

7.1. Residuo de la resistencia de rotor, rutina completa 151
7.2. Error de seguimiento de velocidad, rutina completa 152
7.3. Distorsión armónica de corriente, rutina completa 152
7.4. Error de estimación de torque, rutina completa 153
7.5. *Ripple* de torque, rutina completa 153
7.6. Error de estimación de flujo, rutina completa 154
7.7. Residuo de la resistencia de rotor, máquina de 0,10 Ω 158
7.8. Error de seguimiento de velocidad, máquina de 0,10 Ω 158
7.9. Residuo de la resistencia de rotor, máquina de 0,50 Ω 159
7.10. Error de seguimiento de velocidad, máquina de 0,50 Ω 159
7.11. Residuo de la resistencia de rotor, máquina de 3,00 Ω 160
7.12. Error de seguimiento de velocidad, máquina de 3,00 Ω 160
7.13. Residuo de la resistencia de rotor, máquina de 5,00 Ω 161
7.14. Error de seguimiento de velocidad, máquina de 5,00 Ω 161
7.15. Residuo de la resistencia de rotor, carga de bomba 164
7.16. Error de seguimiento de velocidad, carga de bomba 165
7.17. Error de estimación de torque, carga de bomba 165
7.18. Error de estimación de flujo, carga de bomba 165
7.19. Residuo de la resistencia de rotor, rutina completa, 1000 r/min 167
7.20. Error de seguimiento de velocidad, rutina completa, 1000 r/min 168
7.21. Residuo de la resistencia de rotor, iPINN, rutina completa 171
7.22. Error de seguimiento de velocidad, iPINN, rutina completa 172
7.23. Distorsión armónica de corriente, iPINN, rutina completa 172
7.24. Error de estimación de torque, iPINN, rutina completa 173
7.25. Ripple de torque, iPINN, rutina completa 173
7.26. Error de estimación de flujo, iPINN, rutina completa 174
7.27. Residuo de la resistencia de rotor, iPINN, carga de bomba 176
7.28. Error de seguimiento de velocidad, iPINN, carga de bomba 177
7.29. Error de estimación de torque, iPINN, carga de bomba 177
7.30. Error de estimación de flujo, iPINN, carga de bomba 177
7.31. Rutina completa con el modelo ajustado, máquina nominal 180
7.32. Rutina completa sin corrección, máquina nominal 181
7.33. Rutina completa con la Propuesta 1, máquina nominal 183

7.34. Rutina completa con la Propuesta 2, máquina nominal 185
7.35. Rutina completa sin corrección, máquina de 3,00 Ω 187
7.36. Rutina completa con la Propuesta 1, máquina de 3,00 Ω 189
7.37. Rutina completa con la Propuesta 2, máquina de 3,00 Ω 191

D.1. Distorsión armónica de corriente, máquina de 0,10 Ω 244
D.2. Error de estimación de torque, máquina de 0,10 Ω 244
D.3. *Ripple* de torque, máquina de 0,10 Ω 245
D.4. Error de estimación de flujo, máquina de 0,10 Ω 245
D.5. Distorsión armónica de corriente, máquina de 0,50 Ω 246
D.6. Error de estimación de torque, máquina de 0,50 Ω 246
D.7. *Ripple* de torque, máquina de 0,50 Ω 247
D.8. Error de estimación de flujo, máquina de 0,50 Ω 247
D.9. Distorsión armónica de corriente, máquina de 3,00 Ω 248
D.10. Error de estimación de torque, máquina de 3,00 Ω 248
D.11. *Ripple* de torque, máquina de 3,00 Ω 249
D.12. Error de estimación de flujo, máquina de 3,00 Ω 249
D.13. Distorsión armónica de corriente, máquina de 5,00 Ω 250
D.14. Error de estimación de torque, máquina de 5,00 Ω 250
D.15. *Ripple* de torque, máquina de 5,00 Ω 251
D.16. Error de estimación de flujo, máquina de 5,00 Ω 251
D.17. Distorsión armónica de corriente, carga de bomba 252
D.18. *Ripple* de torque, carga de bomba 252
D.19. *Ripple* de flujo, carga de bomba 253
D.20. Distorsión armónica de corriente, rutina completa, 1000 r/min 253
D.21. Error de estimación de torque, rutina completa, 1000 r/min 254
D.22. *Ripple* de torque, rutina completa, 1000 r/min 254
D.23. Error de estimación de flujo, rutina completa, 1000 r/min 255

Índice de Tablas

3.1. Verificación del lazo de velocidad en pequeña señal 40
3.2. Efecto de las dos adaptaciones sobre una máquina de $R_r = 5\ \Omega$ con el modelo ajustado, con carga nominal y referencia de 1440 rpm. 45

4.1. Parámetros de la máquina de inducción y del controlador FCS-MPC/PTC. 52
4.2. Escalera de $R_{r,\mathrm{MPC}}$ empleada en el barrido, con el desajuste resultante frente a la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$). 54
4.3. Arranque con carga de $0{,}5\,T_{nom}$: velocidad final (promedio de los últimos 200 ms) y error de regulación respecto de la referencia de 1440 rpm, a los ocho casos del barrido. 59
4.4. Respuesta al escalón de $0{,}75\,T_{nom}$ aplicado a los 1,6 s sobre la máquina en estado estacionario: velocidad previa, hundimiento máximo en la ventana 1,6–2,1 s, velocidad final (promedio de los últimos 200 ms de los 4,1 s de rutina) y error de regulación resultante. 64
4.5. Arranque con carga de bomba: punto de equilibrio alcanzado por cada caso del barrido. El torque de carga en el equilibrio se contrasta con el valor teórico $T_{nom}(n/n_{nom})^2$. 69
4.6. Duración de la pre-magnetización, tiempo de asentamiento de la velocidad y torque electromagnético real de *peak*, para los ocho casos del barrido con arranque en vacío. 75
4.7. Resumen de la degradación de cada magnitud graficada frente al caso ajustado, por maniobra y por sentido del desajuste. 81
4.8. Estiramiento de la característica torque–deslizamiento con $R_{r,\mathrm{MAS}}$. La última columna es la velocidad que resultaría de alimentar la máquina a frecuencia nominal fija, como haría la red. 83
4.9. Combinaciones empleadas en la exploración sobre distintos valores de $R_{r,\mathrm{MAS}}$. La fila del valor nominal se incluye como referencia. 83
4.10. Velocidad de equilibrio alcanzada con carga de bomba y error de regulación resultante, para las veinte combinaciones de la Tabla 4.9. Entre paréntesis, el desajuste relativo ε_R de cada combinación. 84

5.1. Las tres salidas de cada red PINN 101
5.2. Los dos expertos y sus cuatro redes direccionales: papel, dominio y criterio de diseño. 104
5.3. Asimetría de observabilidad entre los dos sentidos del desajuste, medida sobre el barrido factorial piloto (12 848 ciclos). Las medianas de la Tabla 5.4 proceden de otro conjunto y no son directamente comparables con estas. 105

5.4. Las 27 características base. Medianas típicas sobre 80 corridas, población distinta de la de la Tabla 5.3. 108
5.5. Referencia de las características al sentido de giro canónico 110
5.6. Parámetros del coordinador de arranque. La columna reforzada rige cuando $\rho^2 \leq 0{,}50$, esto es, cuando el residuo no respalda la estimación de la red. 113
5.7. Comparación entre el experto de arranque y el experto de carga: mismo núcleo físico, distinta maniobra. 119
5.8. Los siete términos de la pérdida: qué compara cada uno, sobre qué ciclos se ajustan y con qué peso entra, y el origen de ese peso. El conjunto `fuera` es la unión del caso ajustado, el sentido contrario, el fuera-de-dominio por magnitud y la región ciega, que solo existe en el conjunto de arranque. Cuando el peso difiere entre expertos se indica como arranque/carga. . . 120
5.9. Campañas de entrenamiento del experto de arranque 122
5.10. Campañas de entrenamiento del experto de carga 124
5.11. Perfiles de par de las rutinas de escalones 125
5.12. Configuración final de las cuatro redes de la Propuesta 1 131

6.1. Las salidas de la capa correctora 133
6.2. Canales de entrada de la red 134
6.3. La red única ante las ventanas de la marcha inversa 138
6.4. Arquitectura de la red 139
6.5. Los pesos de la función de pérdida de la red única 141
6.6. Campañas de entrenamiento de la iPINN 143
6.7. Configuración final de la red de la Propuesta 2 147

7.1. Indicadores de desempeño 149
7.2. Síntesis de la escalera, máquina de 0,83 Ω 151
7.3. Síntesis de la escalera, máquina de 0,10 Ω 155
7.4. Síntesis de la escalera, máquina de 0,50 Ω 156
7.5. Síntesis de la escalera, máquina de 3,00 Ω 156
7.6. Síntesis de la escalera, máquina de 5,00 Ω 157
7.7. Bomba a 750 r/min, Propuesta 1 162
7.8. Bomba a 1000 r/min, Propuesta 1 163
7.9. Bomba a 1250 r/min, Propuesta 1 163
7.10. Bomba a 1440 r/min, Propuesta 1 164
7.11. Síntesis de la escalera a 1440 r/min 166
7.12. Síntesis de la escalera a 1000 r/min 167
7.13. Quién mueve $R^{\mathrm{eff}}_{r,\mathrm{MPC}}$ en cada rutina: pasos del *breakaway* y de la vía de emergencia, aplicaciones de la red de arranque y compromisos de la red de carga 169
7.14. Síntesis de la escalera, Propuesta 2 171
7.15. Bomba a 750 r/min, Propuesta 2 175
7.16. Bomba a 1000 r/min, Propuesta 2 175
7.17. Bomba a 1250 r/min, Propuesta 2 176
7.18. Bomba a 1440 r/min, Propuesta 2 176
7.19. Rutina de evaluación en el dominio del tiempo 178
7.20. Resultado de la rutina sobre las dos máquinas 192

7.21. Las dos propuestas sobre el subconjunto nominal común 193

8.1. Ventajas y desventajas de cada propuesta 197

E.1. Residuo del experto de arranque a lo largo de la escalera 256
E.2. Decisiones necesarias frente a decisiones disponibles 257
E.3. Corrección propuesta frente a corrección aplicada 257

F.1. Límites de escritura de ambas propuestas 259
F.2. Comportamiento por encima del rango de entrenamiento 260
F.3. Comportamiento por debajo del rango de entrenamiento 261

G.1. Diferencia entre el vector de entrenamiento y el desplegado 264
G.2. Secuencia temporal del predictor auxiliar y de la sensibilidad en el código 265
G.3. Efecto de alinear la sensibilidad sobre los indicadores en lazo cerrado . . . 266
G.4. Condiciones de los coordinadores que suponen el sentido de giro positivo 267
G.5. Rutina de cinco segundos en los dos sentidos de giro 268
G.6. Reversiones del coordinador de carga en la campaña 268
G.7. El sesgo de la proyección en la corrida de referencia 269

Nomenclatura

Símbolos

$\boldsymbol{v}_s,\ \boldsymbol{i}_s,\ \boldsymbol{\psi}_s$	:	Fasores espaciales de tensión, corriente y enlace de flujo de estator.
$\boldsymbol{i}_r,\ \boldsymbol{\psi}_r$	:	Fasores espaciales de corriente y enlace de flujo de rotor.
$R_s,\ R_r$	:	Resistencias de estator y de rotor.
$L_s,\ L_r,\ L_m$	:	Inductancias propia de estator, propia de rotor y mutua.
σ	:	Coeficiente de dispersión, $\sigma = 1 - L_m^2/(L_s L_r)$.
L_σ	:	Inductancia transitoria de estator, $L_\sigma = \sigma L_s$.
k_r	:	Constante de acoplamiento de rotor, $k_r = L_m/L_r$.
τ_r	:	Constante de tiempo del rotor, $\tau_r = L_r/R_r$.
τ_σ	:	Constante de tiempo transitoria, $\tau_\sigma = L_\sigma/R_\sigma$, con $R_\sigma = R_s + k_r^2 R_r$.
J	:	Momento de inercia del conjunto rotatorio.
p	:	Número de pares de polos.
$T_e,\ T_L$	:	Torque electromagnético y torque de carga.
$\omega_m,\ \omega_{em}$	:	Velocidad mecánica y velocidad eléctrica del rotor.
$\omega_s,\ \omega_r$	:	Velocidad sincrónica y frecuencia de deslizamiento.
$\theta,\ \theta_k$	:	Ángulo eléctrico del rotor y ángulo de un marco de referencia genérico k.
γ_r	:	Ángulo de orientación del flujo de rotor.
T_s	:	Período de muestreo del controlador.
λ	:	Factor de ponderación de la función de costo del PTC.
$R_{r,\mathrm{MAS}}$	:	Resistencia de rotor de la máquina simulada (planta).
$R_{r,\mathrm{MPC}}$	:	Resistencia de rotor utilizada por el modelo del controlador.
$R_{r,\mathrm{MPC}}^{\mathrm{eff}}$	:	Resistencia de rotor *efectiva* del controlador, corregida en línea.
k_{rr}	:	Factor de reescalamiento de la resistencia de rotor del controlador.
$\boldsymbol{\varepsilon}$	:	Residuo de predicción de corriente en coordenadas $\alpha\beta$.
$\boldsymbol{\varphi}$	:	Sensibilidad del predictor de corriente respecto de R_r.
$\hat{d}$	:	Proyección del residuo sobre la sensibilidad, $\hat{d} \approx R_{r,\mathrm{MAS}} - R_{r,\mathrm{MPC}}^{\mathrm{eff}}$.
e_{reg}	:	Error de regulación de velocidad normalizado.

Acrónimos

MAS	:	Máquina asincrónica (máquina de inducción).
SCIM	:	*Squirrel-Cage Induction Machine* (máquina de inducción jaula de ardilla).
MPC	:	*Model Predictive Control* (control predictivo basado en modelo).
FCS-MPC	:	*Finite Control Set MPC* (MPC de conjunto finito de actuaciones).
PTC	:	*Predictive Torque Control* (control predictivo de torque).
PCC	:	*Predictive Current Control* (control predictivo de corriente).
MPDTC	:	*Model Predictive Direct Torque Control*.
FOC	:	*Field Oriented Control* (control por campo orientado).
DTC	:	*Direct Torque Control* (control directo de torque).
2L-VSI	:	*Two-Level Voltage Source Inverter* (inversor fuente de tensión de dos niveles).
PI	:	Controlador proporcional-integral.
ANN	:	*Artificial Neural Network* (red neuronal artificial).
MLP	:	*Multi-Layer Perceptron* (perceptrón multicapa).
PINN	:	*Physics-Informed Neural Network* (red neuronal informada por física).
MRAS	:	*Model Reference Adaptive System*.
EKF	:	*Extended Kalman Filter* (filtro de Kalman extendido).
THD	:	*Total Harmonic Distortion* (distorsión armónica total).
RMS	:	*Root Mean Square* (valor eficaz).
MAE	:	*Mean Absolute Error* (error absoluto medio).
FSM	:	*Finite State Machine* (máquina de estados finitos).
OOD	:	*Out Of Domain* (fuera del dominio de acción).
RPC	:	*Remote Procedure Call* (llamada a procedimiento remoto).
DFT	:	*Discrete Fourier Transform* (transformada discreta de Fourier).

Resumen

El control predictivo de conjunto finito (FCS-MPC) se ha consolidado como una alternativa competitiva para el accionamiento de máquinas de inducción, gracias a su rápida respuesta dinámica y a la incorporación natural de múltiples objetivos en una única función de costo. Sin embargo, su desempeño depende críticamente de la fidelidad del modelo interno con que predice el comportamiento futuro de la máquina. Entre los parámetros de ese modelo, la resistencia de rotor R_r es el más problemático: no es medible en forma directa en una máquina jaula de ardilla. Cuando el valor utilizado por el controlador difiere del valor real de la máquina, las predicciones se degradan, aumenta la distorsión de las corrientes, se deteriora la regulación de velocidad y, en casos severos, se puede perder el control del accionamiento.

Esta memoria desarrolla y evalúa **dos propuestas** de compensación basadas en redes neuronales informadas por física (PINN, *Physics-Informed Neural Networks*) que estiman en línea el desajuste de la resistencia de rotor y actualizan el valor efectivo que emplea el control predictivo, sin conocer el valor real del parámetro. Ambas conservan íntegramente la estructura del control predictivo: no intervienen la función de costo ni la selección de los estados de conmutación, y actúan únicamente sobre la resistencia de rotor del modelo interno.

La **primera propuesta** parte de un observable construido con la física del mismo control predictivo. Se demuestra que la predicción de corriente en coordenadas estacionarias es una función afín de la resistencia de rotor, de modo que el residuo entre la corriente medida y la predicha es aproximadamente proporcional al error de resistencia con una sensibilidad calculable analíticamente; su proyección sobre esa sensibilidad entrega un estimador físico sesgado del desajuste. Sobre él operan cuatro redes especialistas, organizadas por maniobra y por dirección del error, que corrigen el sesgo del estimador y cuantifican la confianza y la pertenencia al dominio de cada inferencia.

La **segunda propuesta** concentra el trabajo en una única red, que recibe una ventana temporal de las señales de control y entrega directamente una propuesta de corrección del parámetro, junto con una medida de confianza y otra de pertenencia a su dominio. Es un estimador inverso amortizado: aprende de las trayectorias una relación entre las señales del lazo y el desajuste que las produce.

En ambas propuestas, la salida neuronal pasa por una lógica de autorización antes de modificar la resistencia efectiva del modelo.

Palabras clave: máquina de inducción, control predictivo, MPC, FCS-MPC, control predictivo de torque, PTC, redes neuronales informadas por física, PINN, estimación de parámetros, resistencia de rotor, PLECS.

Capítulo 1

Introducción

1.1. Contexto, problema y propuesta del trabajo

Las máquinas de inducción jaula de ardilla (SCIM, *Squirrel-Cage Induction Machine*) se utilizan ampliamente en accionamientos eléctricos industriales debido a su robustez constructiva, bajo requerimiento de mantenimiento y alta confiabilidad [1]. Para aplicaciones que requieren un elevado desempeño dinámico, su control ha evolucionado desde estrategias escalares hacia esquemas vectoriales como el control por campo orientado (FOC, *Field Oriented Control*) y el control directo de torque (DTC, *Direct Torque Control*) [2–4]. En este contexto, el control predictivo basado en modelo (MPC, *Model Predictive Control*) se ha consolidado como una alternativa de interés para convertidores de potencia y accionamientos eléctricos [5, 6].

En particular, el control predictivo de conjunto finito (FCS-MPC, *Finite Control Set Model Predictive Control*) aprovecha directamente el conjunto discreto de estados de conmutación del convertidor. Para cada actuación posible, el control utiliza un modelo matemático de la planta para predecir su comportamiento futuro y selecciona el estado que minimiza una función de costo [7]. En el control predictivo de torque (PTC, *Predictive Torque Control*) aplicado a una máquina de inducción, permite controlar directamente variables como el torque electromagnético y el flujo, sin requerir un modulador convencional [5, 8].

La principal dificultad de esta estrategia es, al mismo tiempo, una consecuencia de su principio de funcionamiento: la calidad de las predicciones depende de la igualdad entre el modelo utilizado por el control y la máquina real. El modelo interno emplea parámetros eléctricos como las resistencias de estator y rotor (R_s, R_r) y las inductancias (L_s, L_r y L_m), cuyos valores pueden diferir de los considerados en el modelo o variar con las condiciones de operación [3, 9]. Entre ellos, la resistencia de rotor resulta ser la más sensible, ya que interviene directamente en la constante de tiempo del rotor

$$\tau_r = \frac{L_r}{R_r}, \tag{1.1}$$

y, por lo tanto, afecta tanto la estimación del flujo de rotor como el modelo utilizado para predecir las corrientes y el torque [4, 10]. Debido a que el rotor de una máquina jaula de ardilla no posee terminales accesibles, su resistencia tampoco puede ser medida directamente durante la operación.

Cuando la resistencia de rotor considerada por el controlador no coincide con la resistencia efectiva de la máquina, los errores del modelo se propagan hacia las variables estimadas y predichas. Dependiendo de la magnitud y del sentido del desajuste, esta discrepancia puede aumentar el *ripple* de las variables eléctricas y mecánicas, deteriorar el seguimiento de velocidad y, en condiciones suficientemente severas, comprometer el funcionamiento del control predictivo.

A partir de esta problemática, el presente trabajo propone incorporar sobre el PTC **capas correctivas basadas en redes neuronales informadas por física** (PINN, *Physics-Informed Neural Networks*) [11]. Las estrategias conservan la estructura original del control predictivo y utilizan información disponible durante la operación, incluidas variables eléctricas y mecánicas. En esta memoria, la denominación PINN se aplica a arquitecturas híbridas de identificación paramétrica que incorporan información del modelo en sus entradas o en su función de pérdida; no a redes que resuelven en línea una ecuación diferencial mediante puntos de colocación.

Las dos propuestas se desarrollan y evalúan sobre una plataforma de simulación en PLECS que reproduce el accionamiento completo de una máquina de inducción alimentada por un inversor fuente de tensión de dos niveles.

1.2. Objetivos del trabajo

1.2.1. Objetivo general

Desarrollar y evaluar estrategias de compensación basadas en redes neuronales informadas por física para asistir el control predictivo de una máquina asincrónica, mediante la estimación del desajuste de resistencia de rotor y la actualización supervisada de la resistencia efectiva utilizada por el modelo interno del predictor, sin intervenir directamente en la función de costo ni en las señales de conmutación.

1.2.2. Objetivos específicos

1. Estudiar las estrategias de control predictivo aplicadas a máquinas de inducción, tales como *Finite Control Set – Model Predictive Control* (FCS-MPC), *Predictive Torque Control* (PTC) y *Model Predictive Direct Torque Control* (MPDTC), incluyendo estimación de flujo en coordenadas de campo orientado, compensación de retardo computacional y diseño de la función de costo.

2. Desarrollar un escenario controlado de **desajuste de modelo**, denominado *model-free* en el planteamiento original del trabajo, y caracterizar mediante simulación el comportamiento del FCS-MPC cuando la resistencia de rotor del modelo interno del predictor difiere de la utilizada por la planta. El término no implica ausencia de modelo, pues tanto el PTC como los observables físicos dependen de él.

3. Implementar y comparar dos capas correctivas basadas en redes neuronales informadas por física: un sistema de expertos apoyado en un estimador físico explícito del desajuste y un estimador inverso amortizado de red única. En ambos casos, la corrección debe actuar sobre la resistencia efectiva del modelo interno del PTC mediante una lógica de autorización, y no sobre las salidas de conmutación del controlador.

4. Evaluar comparativamente el desempeño del FCS-MPC estándar frente al FCS-MPC asistido por inteligencia artificial, utilizando indicadores cuantitativos (error de seguimiento de velocidad, *ripple* de torque, distorsión armónica de corriente y error de flujo) bajo distintos grados de desajuste paramétrico.

1.2.3. Objetivo adicional

Este objetivo es **adicional** en sentido estricto: no sustituye ni modifica ninguno de los cuatro objetivos específicos.

> *Verificar el alcance de una red neuronal para corregir distintos valores de resistencia de rotor desde diferentes puntos de partida y condiciones de operación, utilizando solo señales disponibles en el lazo de control.*

La pregunta que este objetivo plantea es hasta qué punto se mantiene la **cobertura y precisión** en un corrector de este tipo. Una arquitectura ajustada a un único punto de operación puede alcanzar alta precisión local, mientras que otra diseñada para cubrir un dominio amplio debe conservar utilidad frente a variaciones de resistencia, carga, velocidad y sentido del desajuste. La evaluación separa ambos atributos y evita interpretar la cobertura paramétrica como portabilidad automática hacia máquinas con parámetros eléctricos distintos.

1.3. Estructura del documento

El resto del documento se organiza de la siguiente manera:

El Capítulo 2 presenta el marco teórico necesario para el desarrollo del trabajo, incluyendo el modelo dinámico de la máquina de inducción, los sistemas de coordenadas utilizados en accionamientos eléctricos, la estimación de flujo de rotor, los fundamentos del control predictivo basado en modelo y las bases de las redes neuronales informadas por física.

El Capítulo 3 desarrolla la implementación del control predictivo de torque utilizado como sistema de referencia. Se describen el inversor como conjunto finito de actuaciones, las etapas de estimación y predicción, la compensación del retardo computacional, la función de costo, la selección del vector óptimo y el lazo externo de velocidad. El capítulo finaliza identificando la dependencia del controlador respecto de los parámetros utilizados por su modelo interno.

El Capítulo 4 presenta la plataforma de simulación implementada en PLECS y caracteriza el efecto del desajuste de resistencia de rotor sobre el comportamiento del accionamiento. A partir de distintos escenarios de operación se determina cómo la discrepancia entre la máquina y el modelo predictivo se manifiesta en las variables disponibles para el controlador, estableciendo la base para el diseño de la estrategia correctiva.

Los Capítulos 5 y 6 desarrollan las dos propuestas de capa correctiva basadas en redes neuronales informadas por física. Ambas persiguen el mismo objetivo: corregir en línea la resistencia de rotor con que predice el controlador, pero se distinguen en qué se le pide aprender a la red.

El Capítulo 5 presenta la **Propuesta 1**, un sistema de cuatro redes especialistas construido sobre un estimador auxiliar deducido de la física del propio predictor. Las redes corrigen el sesgo de esa estimación y entregan indicadores de confianza y dominio; los árbitros deciden si sus propuestas pueden escribirse en el modelo interno. Se describen la construcción del residuo y su proyección, las variables utilizadas por las redes, el procedimiento de generación de datos y entrenamiento, el diseño de los expertos asociados a las maniobras estudiadas, los criterios de activación y protección, y su integración con el controlador predictivo.

El Capítulo 6 presenta la **Propuesta 2**, en la que una única red recibe el muestreo temporal de las señales del accionamiento y entrega directamente la corrección del parámetro. Se describen el diseño de sus entradas y salidas, su arquitectura y entrenamiento, la capa que traduce su salida en una acción sobre el controlador, y el alcance de validez que su formulación impone.

El Capítulo 7 evalúa el desempeño del sistema completo mediante simulaciones comparativas entre el PTC convencional y el PTC asistido por la capa correctiva neuronal, considerando diferentes condiciones de operación y grados de desajuste paramétrico. Finalmente, el Capítulo 8 resume las principales conclusiones obtenidas, analiza el cumplimiento de los objetivos y presenta las líneas de trabajo futuro.

Capítulo 2

Marco Teórico

Este capítulo reúne los fundamentos generales sobre los que se construye el trabajo: el modelo dinámico de la máquina de inducción, los sistemas de coordenadas utilizados en accionamientos eléctricos, la estimación de flujo de rotor, el control predictivo basado en modelo y las redes neuronales informadas por física. La realización concreta del controlador predictivo de torque y flujo de estator (PTC) se desarrolla en el Capítulo 3.

2.1. Modelo dinámico de la máquina de inducción

Esta sección construye, de lo general a lo particular, el modelo dinámico de la máquina de inducción que el control predictivo utilizará como modelo interno.

2.1.1. Hipótesis de modelado

El modelo dinámico clásico de la máquina de inducción se obtiene bajo un conjunto de hipótesis habituales en la teoría de máquinas de corriente alterna con fines de control [9,12]:

- La máquina es trifásica y simétrica, con devanados distribuidos de manera que la fuerza magnetomotriz presenta una distribución espacial sinusoidal a lo largo del entrehierro.
- El entrehierro es uniforme y se desprecian los efectos de ranurado y de borde.
- No circula corriente de secuencia cero, pues el neutro se encuentra aislado.
- El circuito magnético no satura o su nivel de saturación se considera constante en el punto de operación analizado, y se desprecian las pérdidas de hierro.
- Los parámetros eléctricos (R_s, R_r, L_r, L_s y L_m) del modelo se consideran constantes.

2.1.2. Fasores espaciales y representación de la máquina trifásica

El devanado del estator de una máquina trifásica está constituido por tres grupos de bobinas distribuidos simétricamente a lo largo de la circunferencia de la sección transversal, un grupo de bobinas por cada fase (a, b y c), con sus ejes magnéticos separados

120° eléctricos. Al circular corriente por cada fase se genera una distribución de fuerza magnetomotriz aproximadamente sinusoidal a lo largo del entrehierro, cuya amplitud es proporcional a la corriente instantánea de esa fase y cuya orientación coincide con el eje magnético de la bobina correspondiente [12, 13]. Cada contribución puede representarse entonces como un vector en un plano complejo perpendicular al eje de la máquina: la de la fase a como un vector colineal con su eje magnético, y las de las fases b y c como vectores girados 120° y 240°, los que se expresan multiplicando por el operador de rotación $\boldsymbol{a} = e^{j2\pi/3}$.

La suma de las tres contribuciones produce un único vector resultante que representa el efecto conjunto de las tres corrientes instantáneas sobre el entrehierro. La Figura 2.1 ilustra esta construcción sobre la máquina.

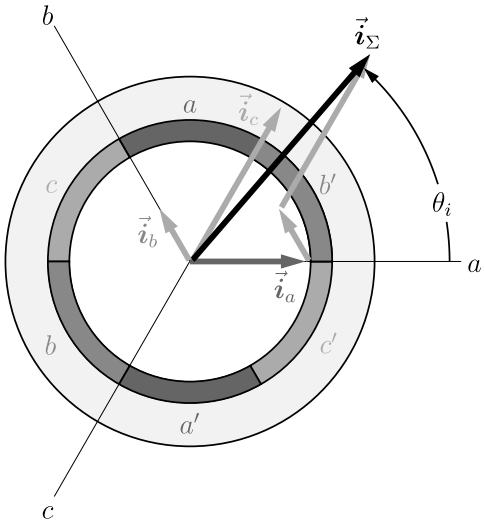


Figura 2.1: Construcción del fasor espacial de corriente sobre la sección transversal de una máquina trifásica. La suma $\boldsymbol{i}_\Sigma = i_a + \boldsymbol{a}\, i_b + \boldsymbol{a}^2 i_c$ de las tres contribuciones define un único vector resultante de ángulo θ_i, del cual el fasor espacial (2.1) es la versión **escalada** por 2/3. Adaptado de [4].

Esta construcción motiva la herramienta central de todo el desarrollo posterior, el **fasor espacial** [3, 9]: dada una magnitud trifásica ($x_a(t)$, $x_b(t)$ y $x_c(t)$) de corriente, tensión o enlace de flujo, se define la variable compleja

$$\boldsymbol{x} = \frac{2}{3}\left(x_a + \boldsymbol{a}\, x_b + \boldsymbol{a}^2 x_c\right), \qquad \boldsymbol{a} = e^{j2\pi/3}, \tag{2.1}$$

donde el factor de escala 2/3 se elige de modo que, en ausencia de secuencia cero, la parte real del fasor coincida con el valor instantáneo de la fase a y, en régimen sinusoidal balanceado, el módulo del fasor coincida con la amplitud de las variables de fase (convención *invariante en amplitud*). El fasor espacial admite una interpretación física directa: su módulo representa la magnitud de la distribución sinusoidal correspondiente en el entrehierro (por ejemplo, de la fuerza magnetomotriz creada por las tres corrientes) y su ángulo la orientación espacial instantánea de esa distribución [12, 14].

Conviene precisar la terminología para evitar una confusión frecuente. El *fasor sinusoidal* convencional del análisis de circuitos es un número complejo *constante* que resume amplitud y fase de una señal sinusoidal de **frecuencia fija**, y solo tiene sentido en régimen

permanente. El fasor espacial de (2.1), en cambio, es un vector *en función del tiempo*: se construye con los valores instantáneos de las tres fases, es válido en cualquier condición de operación, incluidos los transitorios y su carácter "espacial" alude a que describe la orientación de una distribución física en el entrehierro.

2.1.3. Sistema estacionario: coordenadas $\alpha\beta$

El plano complejo en que vive el fasor espacial define el **sistema estacionario de coordenadas** $\alpha\beta$, fijo al estator: su eje real α es colineal con el eje magnético de la fase a y su eje imaginario β está en cuadratura. Todo fasor espacial se descompone entonces como $\boldsymbol{x} = x_\alpha + j\, x_\beta$, tal como muestra la Figura 2.2 para la corriente de estator.

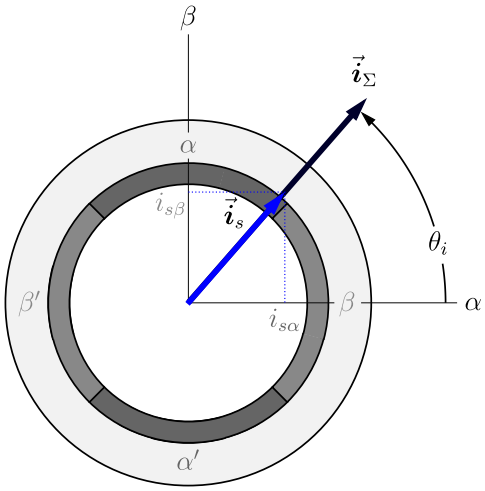


Figura 2.2: Plano estacionario $\alpha\beta$: El fasor espacial de corriente de estator $\boldsymbol{i}_s$, de ángulo θ_i, se descompone en las componentes $i_{s\alpha}$ e $i_{s\beta}$, que pueden interpretarse como las corrientes de dos devanados equivalentes en cuadratura alimentados por las tensiones $v_{s\alpha}$ y $v_{s\beta}$. Adaptado de [4].

Separando la parte real e imaginaria de (2.1) se obtiene la *transformación de Clarke* (trifásica–$\alpha\beta$) [14]:

$$x_\alpha = \frac{2}{3}\left(x_a - \frac{1}{2}\,x_b - \frac{1}{2}\,x_c\right), \qquad x_\beta = \frac{1}{\sqrt{3}}\,(x_b - x_c)\,, \tag{2.2}$$

que en forma matricial corresponde a la operación:

$$\begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix} = \frac{2}{3}\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & \frac{\sqrt{3}}{2} & -\frac{\sqrt{3}}{2} \end{bmatrix}\begin{bmatrix} x_a \\ x_b \\ x_c \end{bmatrix}. \tag{2.3}$$

En el plano $\alpha\beta$, las variables de una máquina operando en régimen sinusoidal balanceado describen circunferencias: cada fasor espacial gira a la frecuencia eléctrica ($2\pi\, f_{el} = \omega_{el}$) de alimentación con módulo constante. Las variables siguen siendo señales sinusoidales en el tiempo en este marco y la representación compacta es la introducción a los sistemas rotatorios.

2.1.4. Transformación a sistemas rotatorios: coordenadas dq

La necesidad de esta transformación nace de un hecho estructural del modelo: las ecuaciones del estator se formulan naturalmente en el marco estacionario con devanados fijos, mientras que las del rotor se formulan en un marco situado en el rotor que gira con él. Para analizar y controlar la máquina es necesario expresar ambas en un *sistema de coordenadas común*; de lo contrario, el acoplamiento magnético entre estator y rotor queda descrito por inductancias que dependen del ángulo instantáneo entre ambos marcos [12,13].

Un fasor espacial $\boldsymbol{x}_s$ expresado en el marco estacionario puede referirse a un sistema de coordenadas genérico k que gira con velocidad angular ω_k y cuyo eje real forma un ángulo $\theta_k(t)$ con el eje α, como se ilustra en la Figura 2.3:

$$\boldsymbol{x}^{(k)} = \boldsymbol{x}_s\, e^{-j\theta_k}, \qquad \theta_k(t) = \int_0^t \omega_k\, d\tau + \theta_k(0). \tag{2.4}$$

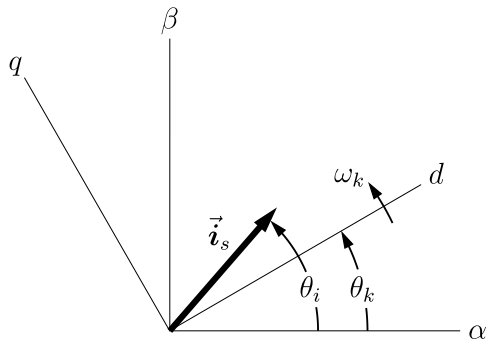


Figura 2.3: Referencia de un fasor espacial a un sistema de coordenadas rotatorio dq. Los ejes dq giran a velocidad ω_k y forman un ángulo θ_k con el eje α del marco estacionario. El fasor de corriente de estator $\boldsymbol{i}_s$, cuyo ángulo respecto de α es θ_i, conserva su módulo al cambiar de marco y pasa a verse con ángulo $\theta_i - \theta_k$ desde los ejes dq. Adaptado de [13].

La transformación no altera el módulo del fasor, solo su ángulo: puede interpretarse como el giro del plano de referencia visto por un observador situado en el estator [13]. En componentes, la proyección del fasor sobre un marco de ángulo θ_k (transformada de Park) es:

$$\begin{bmatrix} x_d \\ x_q \end{bmatrix} = \begin{bmatrix} \cos\theta_k & \sin\theta_k \\ -\sin\theta_k & \cos\theta_k \end{bmatrix} \begin{bmatrix} x_\alpha \\ x_\beta \end{bmatrix}. \tag{2.5}$$

La elección de ω_k define el sistema de coordenadas de trabajo, y cada elección tiene un significado físico preciso: con $\omega_k = 0$ el marco queda fijo al estator (el propio plano $\alpha\beta$, donde las variables de régimen son sinusoidales de frecuencia estatórica); con $\omega_k = \omega_{em}$ el marco gira solidario al rotor; y con ω_k igual a la velocidad sincrónica ω_s las variables se vuelven cantidades *constantes* en régimen permanente, lo que facilita el análisis y el control. La diferencia conceptual entre ambos marcos conviene tenerla siempre presente: $\alpha\beta$ es un marco *estacionario* en que las variables oscilan, y dq es un marco *giratorio sincrónico* en que las mismas variables aparecen como magnitudes continuas.

Además, al derivar respecto del tiempo una variable referida al marco giratorio aparece

un término adicional:

$$\frac{d\boldsymbol{x}}{dt} = e^{j\theta_k}\left(\frac{d\boldsymbol{x}^{(k)}}{dt} + j\,\omega_k\,\boldsymbol{x}^{(k)}\right), \tag{2.6}$$

donde $j\omega_k\boldsymbol{x}^{(k)}$ es el efecto de observar el vector desde un marco *giratorio*; este término produce el acoplamiento entre los ejes ortogonales del marco giratorio y aparecerá de forma natural en las ecuaciones de la máquina [4, 14].

2.1.5. Ecuaciones eléctricas de la máquina de inducción

La ecuación eléctrica de cada fase del estator relaciona su tensión con la caída resistiva y la variación de su enlace de flujo:

$$v_a = R_s\,i_a + \frac{d\psi_a}{dt}, \qquad v_b = R_s\,i_b + \frac{d\psi_b}{dt}, \qquad v_c = R_s\,i_c + \frac{d\psi_c}{dt},$$

donde R_s es la resistencia por fase del devanado. Aplicando la definición (2.1) a las tres ecuaciones, el conjunto se condensa en una única expresión compleja en el marco estacionario [3, 4, 9]:

$$\boldsymbol{v}_s = R_s\boldsymbol{i}_s + \frac{d\boldsymbol{\psi}_s}{dt}, \tag{2.7}$$

con $\boldsymbol{v}_s$, $\boldsymbol{i}_s$ y $\boldsymbol{\psi}_s$ los fasores espaciales de tensión, corriente y enlace de flujo de estator.

El rotor jaula de ardilla se representa mediante un devanado trifásico equivalente, referido al estator, que produce la misma distribución de fuerza magnetomotriz que las barras reales [12, 13]. Como las barras están cortocircuitadas por los anillos extremos, la tensión de rotor es nula, y su ecuación eléctrica en su marco natural *fijo al rotor*, es:

$$0 = R_r\boldsymbol{i}_r + \frac{d\boldsymbol{\psi}_r}{dt}, \tag{2.8}$$

con R_r la resistencia de rotor referida al estator e $\boldsymbol{i}_r$, $\boldsymbol{\psi}_r$ los fasores de corriente y enlace de flujo de rotor. Los enlaces de flujo acoplan magnéticamente ambos devanados, con cada fasor expresado en su marco natural la relación es:

$$\boldsymbol{\psi}_s = L_s\boldsymbol{i}_s + L_m\boldsymbol{i}_r\,e^{j\theta_{sr}}, \tag{2.9}$$

$$\boldsymbol{\psi}_r = L_r\boldsymbol{i}_r + L_m\boldsymbol{i}_s\,e^{-j\theta_{sr}}, \tag{2.10}$$

donde θ_{sr} es el ángulo eléctrico entre los ejes de referencia de estator y rotor. Las inductancias propias se descomponen conceptualmente en una componente magnetizante común y una de dispersión propia de cada devanado, $L_s = L_{s\sigma} + L_m$ y $L_r = L_{r\sigma} + L_m$, donde la inductancia mutua L_m representa el flujo que efectivamente cruza el entrehierro y enlaza ambos devanados, mientras que $L_{s\sigma}$ y $L_{r\sigma}$ representan el flujo que enlaza solo a su propio devanado [9].

2.1.5.a. Ecuaciones eléctricas de la IM: coordenadas dq

Las relaciones (2.9)–(2.10) exhiben el problema estructural anticipado en (2.6): los factores $e^{\pm j\theta_{sr}}$ aparecen precisamente porque cada distribución de flujo está referida a un marco distinto, y θ_{sr} varía continuamente con el giro del rotor. La transformación

(2.4) elimina esta dependencia al referir todas las variables a un único marco que gira a velocidad ω_k. Aplicándola a (2.7)–(2.10) junto con la regla de derivación (2.6), el modelo en el marco común resulta [4, 14]:

$$\boldsymbol{v}_s^{(k)} = R_s\, \boldsymbol{i}_s^{(k)} + \frac{d\boldsymbol{\psi}_s^{(k)}}{dt} + j\,\omega_k\, \boldsymbol{\psi}_s^{(k)}, \tag{2.11}$$

$$0 = R_r\, \boldsymbol{i}_r^{(k)} + \frac{d\boldsymbol{\psi}_r^{(k)}}{dt} + j\,(\omega_k - \omega_{em})\, \boldsymbol{\psi}_r^{(k)}, \tag{2.12}$$

con los enlaces de flujo, ahora sin dependencia angular explícita,

$$\boldsymbol{\psi}_s^{(k)} = L_s\, \boldsymbol{i}_s^{(k)} + L_m\, \boldsymbol{i}_r^{(k)}, \tag{2.13}$$

$$\boldsymbol{\psi}_r^{(k)} = L_r\, \boldsymbol{i}_r^{(k)} + L_m\, \boldsymbol{i}_s^{(k)}, \tag{2.14}$$

donde ω_{el} es la velocidad angular eléctrica del rotor, definida en la subsección siguiente.

En las ecuaciones de tensión se distinguen tres términos con nombre propio en la literatura [12, 13]: la *caída resistiva* ($R\,\boldsymbol{i}$); la *tensión inducida* ($d\boldsymbol{\psi}/dt$) por la variación temporal del enlace de flujo; y la *tensión rotacional* ($j\omega_k\boldsymbol{\psi}_s^{(k)}$ en el estator, $j(\omega_k-\omega_{em})\boldsymbol{\psi}_r^{(k)}$ en el rotor), inducida por el movimiento relativo entre el devanado y el marco de referencia.

Y la velocidad que multiplica cada término rotacional es, en efecto, la velocidad *relativa* entre el marco k y el devanado correspondiente: ω_k para el estator (fijo) y $\omega_k - \omega_{em}$ para el rotor (que gira a ω_{em}).

Como verificación de consistencia, con $\omega_k = 0$ se recupera (2.7), y con $\omega_k = \omega_{em}$ (marco solidario al rotor) se recupera (2.8), es decir, las formulaciones son equivalentes.

Además, sobre un marco de coordenadas orientado a *velocidad sincrónica* ω_s, existe la *frecuencia de deslizamiento* $\omega_r = \omega_s - \omega_{em}$, que mide el retraso del rotor respecto del campo giratorio; el deslizamiento "s" clásico de la teoría en régimen permanente es la razón $s = \omega_r/\omega_s$.

2.1.6. Ecuación mecánica y torque electromagnético

Antes de formular la dinámica mecánica conviene fijar las magnitudes angulares que intervienen en el análisis y que no deben confundirse. La posición angular y velocidad *mecánicas* del eje, θ_m y ω_m, se relacionan con sus equivalentes *eléctricos* a través del número de pares de polos p:

$$\theta_{em} = p\,\theta_m, \qquad \omega_{em} = p\,\omega_m. \tag{2.15}$$

En una máquina de p pares de polos, un giro mecánico completo del rotor corresponde a p ciclos eléctricos, por lo que las variables eléctricas "ven" el rotor girar p veces más rápido; todas las ecuaciones eléctricas anteriores están escritas en radianes eléctricos.

Sobre la dinámica mecánica se obtiene aplicando la segunda ley de Newton al conjunto rotatorio, considerado como cuerpo rígido y accionando una carga a través de un eje común (Figura 2.4).

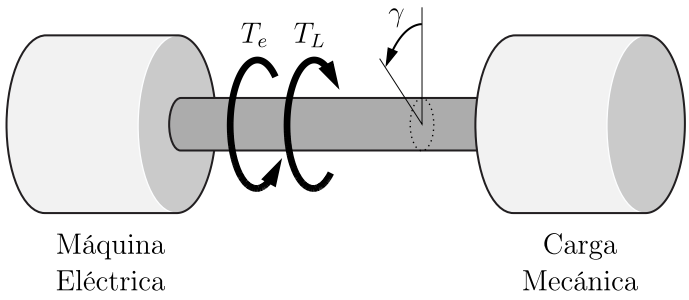


Figura 2.4: Modelo de eje rígido que vincula la máquina eléctrica con un sistema mecánico. Sobre el eje actúan el torque electromagnético T_e y el torque mecánico de la carga T_L, que en operación como motor se opone al primero; γ denota la posición angular del rotor. Adaptado de [15].

En su formulación general, expresada en la velocidad mecánica mediante (2.15), incluye el roce viscoso [13]:

$$\frac{J}{p}\,\frac{d\omega_{em}}{dt} + \frac{D}{p}\,\omega_{em} = T_e - T_L, \tag{2.16}$$

donde J es el momento de inercia del conjunto rotatorio, D el coeficiente de amortiguación rotacional (roce viscoso) y T_L el torque de carga, que el lazo de velocidad trata como perturbación externa.

El torque electromagnético se expresa en función de los fasores de flujo y corriente de estator como [4, 9]:

$$T_e = \frac{3}{2}\,p\,\mathrm{Im}\{\boldsymbol{\psi}_s^*\,\boldsymbol{i}_s\} = \frac{3}{2}\,p\,(\psi_{s\alpha}\,i_{s\beta} - \psi_{s\beta}\,i_{s\alpha})\,, \tag{2.17}$$

donde $(\cdot)^*$ denota el conjugado complejo. El factor $3/2$ es consecuencia directa de la convención invariante en amplitud de la transformación (2.2): los fasores espaciales conservan las amplitudes de fase, de modo que las magnitudes de potencia y torque (trifásicas) deben reescalarse. Y la fórmula $\mathrm{Im}\{\boldsymbol{\psi}_s^*\boldsymbol{i}_s\}$ es invariante ante el marco de referencia, porque la rotación (2.4) multiplica ambos fasores por factores que se cancelan en el producto, por lo que el torque puede evaluarse en cualquier sistema de coordenadas. Físicamente, (2.17) indica que el torque depende de las magnitudes de flujo y corriente y del seno del ángulo entre ambos fasores: modificar la orientación relativa entre ellos es el mecanismo por el cual todo control de alto desempeño (vectorial o predictivo) actúa sobre el torque.

2.1.7. Modelo en coordenadas de campo orientado

El modelo (2.11)–(2.14) queda en función de cuatro fasores ($\boldsymbol{i}_s$, $\boldsymbol{i}_r$, $\boldsymbol{\psi}_s$, $\boldsymbol{\psi}_r$), de los cuales pueden elegirse dos como variables de estado; las relaciones de flujo (2.13)–(2.14) permiten expresar los otros dos en función de los elegidos. La elección conveniente para el control es la corriente de estator $\boldsymbol{i}_s$ (la única de las cuatro directamente medible) y el flujo de rotor $\boldsymbol{\psi}_r$, cuya dinámica es la más lenta del sistema; esta elección evita además usar la corriente de rotor, inaccesible en una máquina jaula de ardilla [12–14]. La Figura 2.5 resume el modelo completo bajo esta elección de variables, todavía en un marco de referencia genérico.

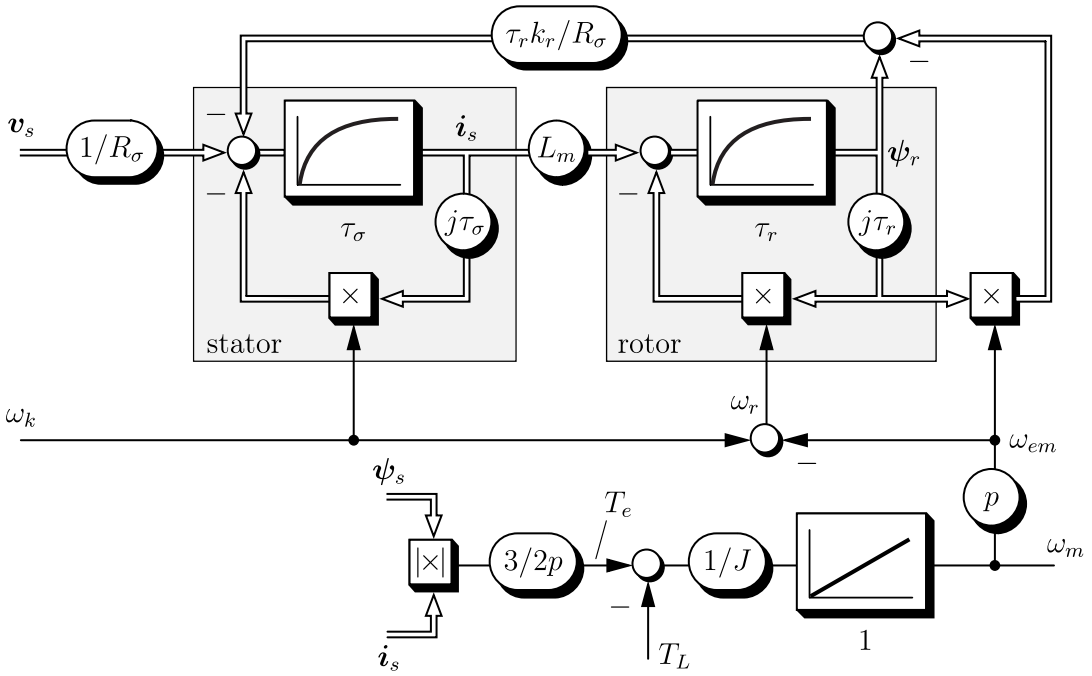


Figura 2.5: Diagrama del modelo dinámico de la máquina de inducción en un sistema de referencia genérico de velocidad ω_k, con la corriente de estator $\boldsymbol{i}_s$ y el flujo de rotor $\boldsymbol{\psi}_r$ como variables de estado. Particularizando $\omega_k = \omega_s$ y alineando el eje d con $\boldsymbol{\psi}_r$ se obtienen las ecuaciones de campo orientado. Adaptado de [14].

En coordenadas de campo orientado, se adopta un sistema de coordenadas dq rotatorio cuyo eje real d se alinea con el fasor de flujo de rotor (*orientación al flujo de rotor*); el ángulo de ese marco respecto del eje α se denota γ_r y su velocidad es la sincrónica $\omega_k = \omega_s$. Bajo la condición de orientación perfecta, la componente q del flujo de rotor es nula por definición, $\psi_{rq} = 0$ y $\psi_{rd} = \psi_r$, y la expresión del torque se simplifica a:

$$T_e = \frac{3}{2}\, p\, k_r\, \psi_r\, i_{sq}, \tag{2.18}$$

donde $k_r = L_m/L_r$ es el factor de acoplamiento de rotor. Separando la ecuación de rotor (2.12) en sus partes real e imaginaria bajo esta orientación, la dinámica del rotor se reduce a dos relaciones escalares:

$$\tau_r\, \frac{d\psi_r}{dt} + \psi_r = L_m\, i_{sd}, \tag{2.19}$$

$$\omega_r = \frac{L_m\, i_{sq}}{\tau_r\, \psi_r}, \tag{2.20}$$

donde $\tau_r = L_r/R_r$ es la constante de tiempo del rotor. La primera es la dinámica (ahora escalar) del flujo de rotor; la segunda se conoce como *condición de orientación* [12]: determina la frecuencia de deslizamiento que el marco debe respetar para permanecer alineado con el flujo. Nótese que ambas dependen de τ_r y, por tanto, de R_r.

Las ecuaciones (2.18)–(2.20) exhiben la estructura desacoplada que motiva la orientación de campo: el flujo de rotor es gobernado exclusivamente por la componente i_{sd}, con la constante de tiempo lenta τ_r; y en condición de flujo de rotor constante, el torque

es proporcional a i_{sq}, que puede modificarse tan rápido como lo permita la dinámica de corriente de estator.

Como la corriente de estator es mucho más rápida que el flujo ($\tau_\sigma \ll \tau_r$, con τ_σ definida más abajo), la máquina se comporta de manera análoga a una máquina de corriente continua con excitación independiente: i_{sd} juega el papel de la corriente de campo e i_{sq} el de la corriente de armadura [2, 4].

2.1.7.a. Coordenadas de campo: corriente de estator i_s y flujo de estator ψ_s

El modelo del estator en coordenadas dq, expresado en las variables de estado elegidas es [4]:

$$\frac{di_{sd}}{dt} = -\frac{1}{\tau_\sigma}\, i_{sd} + \omega_s\, i_{sq} + \frac{k_r}{\tau_r L_\sigma}\, \psi_r + \frac{1}{L_\sigma}\, v_{sd}, \tag{2.21}$$

$$\frac{di_{sq}}{dt} = -\frac{1}{\tau_\sigma}\, i_{sq} - \omega_s\, i_{sd} - \frac{k_r\, \omega_{em}}{L_\sigma}\, \psi_r + \frac{1}{L_\sigma}\, v_{sq}, \tag{2.22}$$

donde $\sigma = 1 - L_m^2/(L_s L_r)$ es el coeficiente de dispersión total, $L_\sigma = \sigma L_s$ la inductancia transitoria de estator, $R_\sigma = R_s + k_r^2 R_r$ la resistencia equivalente del modelo, $\tau_\sigma = L_\sigma / R_\sigma$ la constante de tiempo transitoria y $\omega_s = \omega_{em} + \omega_r$ la velocidad sincrónica. Los términos en ω_s son los acoplamientos cruzados entre ejes heredados de la rotación del marco (el término $j\omega_k \boldsymbol{x}^{(k)}$ de (2.6)), y el término en $k_r \psi_r$ representa la tensión inducida por el flujo de rotor sobre el estator.

Y el enlace de flujo de estator en coordenadas dq, bajo orientación de campo, se calcula como:

$$\psi_{sd} = k_r\, \psi_r + L_\sigma\, i_{sd}, \tag{2.23}$$

$$\psi_{sq} = L_\sigma\, i_{sq}, \tag{2.24}$$

y su forma dinámica es:

$$\frac{d\psi_{sd}}{dt} = v_{sd} - R_s\, i_{sd} + \omega_s\, \psi_{sq}, \tag{2.25}$$

$$\frac{d\psi_{sq}}{dt} = v_{sq} - R_s\, i_{sq} - \omega_s\, \psi_{sd}. \tag{2.26}$$

2.1.8. Estimadores del flujo de rotor ψ_r

Toda la construcción anterior descansa sobre una variable que no es medible: el flujo de rotor. La determinación directa exigiría instalar sensores de flujo en el entrehierro, lo que se descarta por costo y complejidad. Por lo tanto el flujo debe *estimarse* a partir de las variables disponibles: corrientes, tensiones, velocidad, y de los parámetros del modelo [10, 12]. Dos familias clásicas de estimadores en lazo abierto y de primer orden dominan la práctica, y conviene contrastarlas porque sus sensibilidades paramétricas son complementarias [12, 14].

2.1.8.a. Modelo de tensión

El estimador integra la ecuación de estator (2.7) en coordenadas estacionarias,

$$\hat{\boldsymbol{\psi}}_s = \int (\boldsymbol{v}_s - R_s\, \boldsymbol{i}_s)\, dt,$$

y obtiene el flujo de rotor desde la relación de enlaces de flujo, descontando el flujo de dispersión, según la estructura de la Figura 2.6.

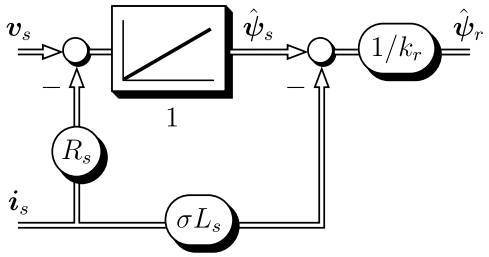


Figura 2.6: Estimador de flujo por modelo de tensión, en coordenadas estacionarias. Nótese que la única constante involucrada del lado del rotor es el factor de acoplamiento k_r. Adaptado de [14].

Sus ventajas son estructurales: no requiere la medición de velocidad ni la constante de tiempo del rotor, lo que lo hace atractivo en esquemas sin sensor mecánico.

Sus limitaciones también lo son. Primero, se basa en una integración en lazo abierto: cualquier componente de valor medio no nulo en las señales, como ruido de medición, se acumula sin cota y lleva la salida del integrador a la deriva, por pequeño que sea el sesgo; las técnicas que lo mitigan limitan en la práctica la frecuencia de operación [12]. Segundo, a baja velocidad la tensión aplicada es pequeña y la caída resistiva $R_s\boldsymbol{i}_s$ domina el integrador, de modo que el estimador se vuelve muy sensible al error en R_s precisamente en la región de arranque y baja frecuencia; su desempeño mejora a alta velocidad, donde la influencia relativa de R_s disminuye [10, 14].

2.1.8.b. Modelo de corriente

El estimador en coordenadas de campo integra la propia ecuación dinámica del rotor (2.19)-(2.20), reemplazando las cantidades reales por sus estimaciones:

$$\tau_r \frac{d\hat{\psi}_r}{dt} + \hat{\psi}_r = L_m\, i_{sd}, \tag{2.27}$$

$$\hat{\omega}_r = \frac{L_m\, i_{sq}}{\tau_r\, \hat{\psi}_r}, \tag{2.28}$$

La posición angular del flujo de rotor se obtiene integrando la velocidad sincrónica; suma de la velocidad mecánica medida y de la frecuencia de deslizamiento dada por la condición de orientación (2.20):

$$\gamma_r(t) = \int_0^t \hat{\omega}_s\, d\tau = \int_0^t \left(\omega_{em} + \frac{L_m\, i_{sq}}{\tau_r\, \hat{\psi}_r}\right) d\tau. \tag{2.29}$$

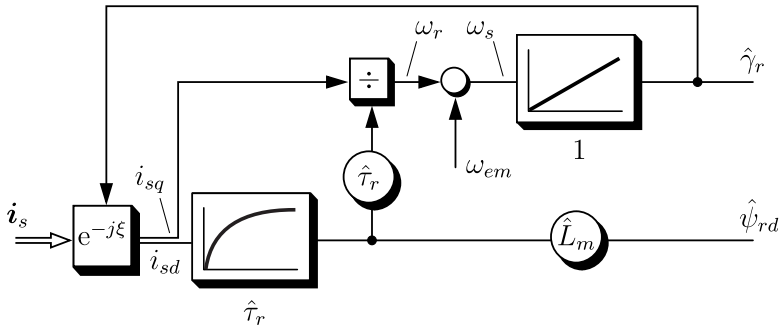


Figura 2.7: Estimador de flujo por modelo de corriente, en coordenadas de campo. $\hat{\omega}_r$ según la condición de orientación (2.20) sumada a la velocidad eléctrica medida ω, esta produce la velocidad sincrónica $\hat{\omega}_s$ cuya integral es el ángulo del flujo denotado como γ_r, que a su vez orienta la propia transformación del lazo. Adaptado de [4].

Este estimador requiere la medición de velocidad y, a cambio, opera correctamente desde velocidad cero, lo que resulta esencial para las maniobras de arranque. Su limitación es la contraparte exacta de su ventaja: la calidad de la estimación depende de manera crítica de la exactitud de la medición de velocidad y de la constante de tiempo del rotor τ_r, un problema mayor, porque este parámetro es cociente de parámetros no medibles directamente. Para deslizamientos altos la magnitud del flujo estimado es muy sensible a R_r; para deslizamientos bajos domina la sensibilidad a L_m; y el ángulo γ_r del flujo estimado que orienta todo el marco de control es sensible a ambos [9, 10, 12].

2.2. Principio del control predictivo basado en modelo

El control predictivo basado en modelo (MPC, *Model Predictive Control*) designa una familia de estrategias cuyo elemento común es el uso explícito de un **modelo** del sistema para **anticipar** el efecto futuro de las actuaciones posibles, y la selección de la actuación óptima mediante la minimización de una función de costo [5, 16]. La Figura 2.8 resume la estructura de un MPC.

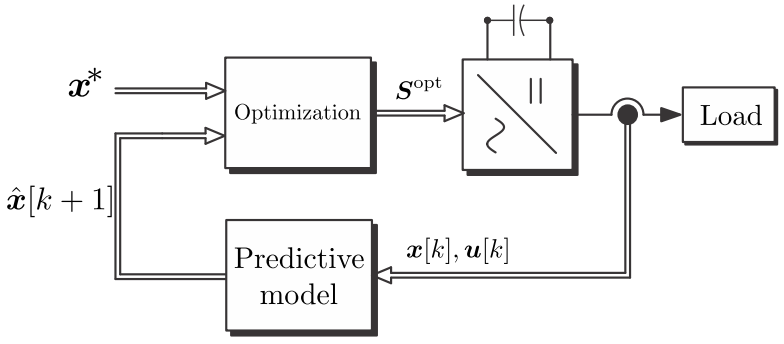


Figura 2.8: Estructura genérica de un MPC aplicado a un inversor que controla una carga. Extraído de [14].

En una formulación discreta general, el modelo de predicción entrega el estado futuro a partir del estado medido y de la actuación candidata,

$$\mathbf{x}[k+1] = f\left(\mathbf{x}[k],\, \mathbf{u}[k]\right), \tag{2.30}$$

y la función de costo g cuantifica la desviación entre las variables predichas y sus referencias, ponderando la importancia relativa de cada objetivo. La actuación aplicada es la que minimiza g dentro del conjunto admisible $\mathcal{U}$:

$$\mathbf{u}^{\text{opt}}[k] = \arg\min_{\mathbf{u}\in\mathcal{U}} \; g. \tag{2.31}$$

El procedimiento se repite dentro de cada período de muestreo con las mediciones actualizadas, aplicando solo actuación óptima de la secuencia (estrategia de *horizonte deslizante*); esta realimentación implícita es la que confiere robustez al esquema [5]. Entre sus ventajas prácticas destacan la inclusión natural de no linealidades y restricciones en el modelo, el tratamiento directo del caso multivariable y una estructura conceptualmente simple. Su costo es una carga de cálculo elevada respecto de los controladores lineales y, de manera central para esta memoria, una dependencia directa de la *calidad del modelo*: las predicciones, y con ellas la actuación elegida, heredan cualquier error paramétrico del **modelo** interno [5, 6].

2.2.1. Control predictivo basado en modelo con conjunto de control finito

En los convertidores de potencia, el conjunto de actuaciones no es continuo: el convertidor solo puede adoptar un número finito de estados de conmutación. La variante que explota esta característica es el MPC de conjunto finito (FCS-MPC, *Finite Control Set MPC*), que evalúa la función de costo exhaustivamente para cada estado admisible y aplica el estado óptimo durante el período de muestreo completo, sin etapa de modulación intermedia [16, 17].

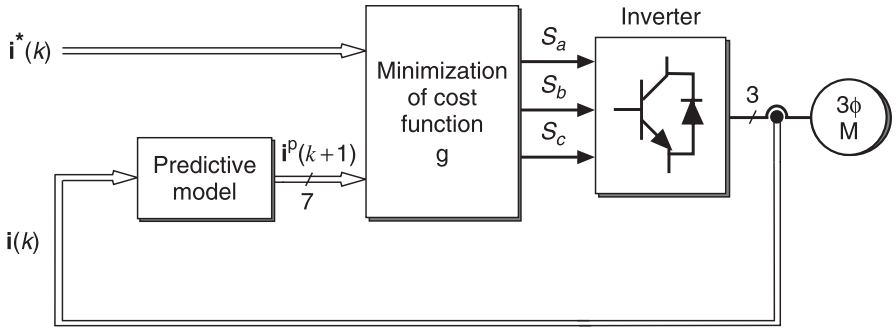


Figura 2.9: Estructura genérica de un FCS-MPC aplicado a un inversor trifásico, ilustrada aquí sobre el control predictivo de corriente. Extraído de [5].

Dentro de esta familia, el *control predictivo de torque* (PTC, *Predictive Torque Control*) es la variante cuya función de costo controla conjuntamente el torque electromagnético y la magnitud del flujo de estator [7]. Conceptualmente, el PTC hereda la idea del control directo de torque (DTC): gobernar torque y flujo actuando directamente sobre los vectores de tensión del inversor. Pero reemplaza las tablas de conmutación por histéresis por una selección óptima basada en las predicciones del modelo [5, 18].

2.3. Estructura de control realimentado

El controlador predictivo de la sección anterior gobierna las variables eléctricas rápidas de la máquina; la velocidad mecánica, en cambio, se regula en esta memoria mediante un

lazo externo con un controlador PI clásico. Esta sección presenta los fundamentos generales de esa estructura: el lazo realimentado, las acciones proporcional e integral, y el fenómeno de *windup* con su corrección, dejando el diseño y los valores concretos del lazo de velocidad implementado para el Capítulo 3.

2.3.1. El lazo cerrado

En la estructura de control realimentado, mostrada en la Figura 2.10.

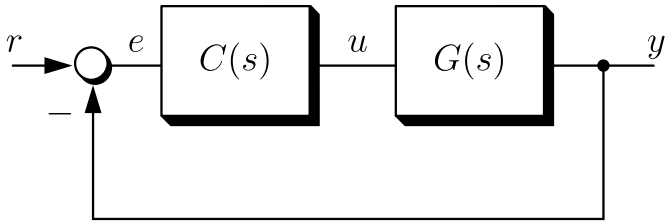


Figura 2.10: Estructura básica de control realimentado. Adaptado de [19].

la variable de salida de la planta $y(t)$ se mide y se compara continuamente con una referencia deseada $r(t)$; su diferencia define el *error*:

$$e(t) = r(t) - y(t), \tag{2.32}$$

y el controlador calcula, en función de ese error, una *actuación* $u(t)$ que se aplica a la planta para reducirlo [13, 20]. La realimentación es lo que confiere al esquema sus dos propiedades esenciales: corrige el efecto de las *perturbaciones* (entradas no controladas que desvían la salida, como el torque de carga en el eje) y tolera cierta *incertidumbre* en el modelo de la planta, pues el controlador no actúa sobre lo que el modelo predice sino sobre lo que la medición informa. El costo es la posible inestabilidad del lazo si el controlador se diseña sin respetar la dinámica, lo que hace del ajuste de sus parámetros un paso de diseño.

2.3.2. El controlador PI

Dentro de la familia de controladores PID, el controlador *proporcional-integral* (PI) es el adecuado para plantas cuya dinámica dominante es de primer orden o integradora, como los lazos de corriente y velocidad de los accionamientos eléctricos [4, 13]. La Figura 2.11 muestra la estructura paralela de un controlador PID, en la que el PI corresponde a prescindir de la rama derivativa.

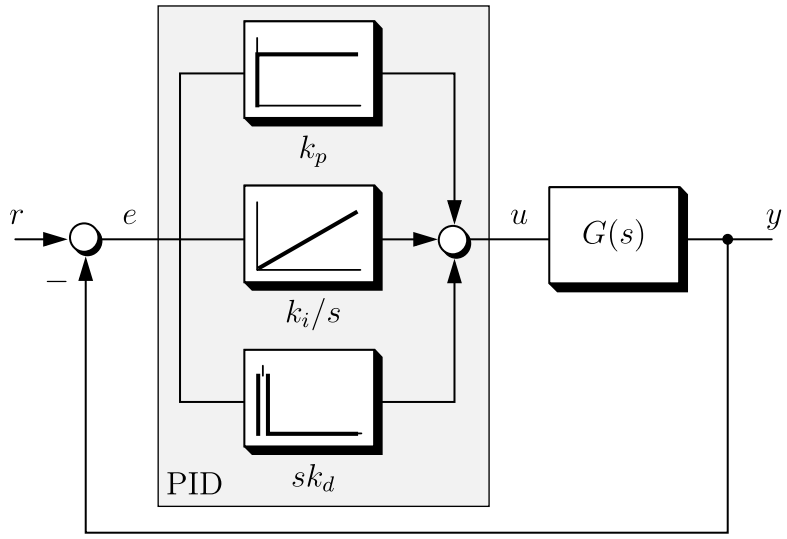


Figura 2.11: Estructura paralela del controlador PID: las acciones proporcional (k_p), integral (k_i/s) y derivativa ($k_d s$) se calculan por separado a partir del error $e(t)$ y se suman para formar la actuación $u(t)$ sobre la planta $G(s)$. Adaptado de [19, 20].

El controlador PI combina dos acciones complementarias. La acción *proporcional* ($u_P = k_p\, e(t)$) produce una actuación inmediata y proporcional al error presente: aporta rapidez de respuesta, pero por sí sola deja en general un error estacionario, pues requiere un error no nulo para sostener una actuación no nula. La acción *integral* ($u_I = (1/\tau_{k_i}) \int e(t)\, dt$) acumula el error en el tiempo: mientras exista error la actuación sigue creciendo, lo que fuerza al sistema a anular el error estacionario frente a referencias y perturbaciones constantes; su costo es una dinámica adicional que, mal dimensionada, lentifica o desestabiliza el lazo. La combinación de ambas se escribe en tiempo continuo como

$$C_{PI}(s) = k_p\, \frac{\tau_{k_i}\, s + 1}{\tau_{k_i}\, s}, \tag{2.33}$$

donde k_p es la ganancia del controlador y τ_{k_i} su constante de tiempo, que fija el peso relativo de la acción integral.

2.3.3. Saturación de la actuación: windup y anti-windup

En cualquier sistema real la actuación está acotada, por ejemplo, un accionamiento no puede solicitar torque ilimitado, y la referencia que un lazo de velocidad entrega debe saturarse a los límites admisibles de la máquina y del convertidor. Esta saturación interactúa de manera cíclica con la acción integral. Si el error persiste mientras la actuación está saturada, por ejemplo, durante un arranque largo o ante una perturbación grande, el integrador *sigue acumulando* un error que la actuación, ya en su tope, no puede corregir: es el fenómeno de *windup* (enrollamiento). Cuando el error finalmente cambia de signo, el integrador cargado debe primero "descargarse" antes de que la actuación abandone la saturación, lo que produce sobrepasos grandes, respuestas lentas e incluso inestabilidad [13, 20].

Los esquemas *anti-windup* corrigen el problema realimentando al controlador la información de la actuación real (saturada), como muestra la Figura 2.12:

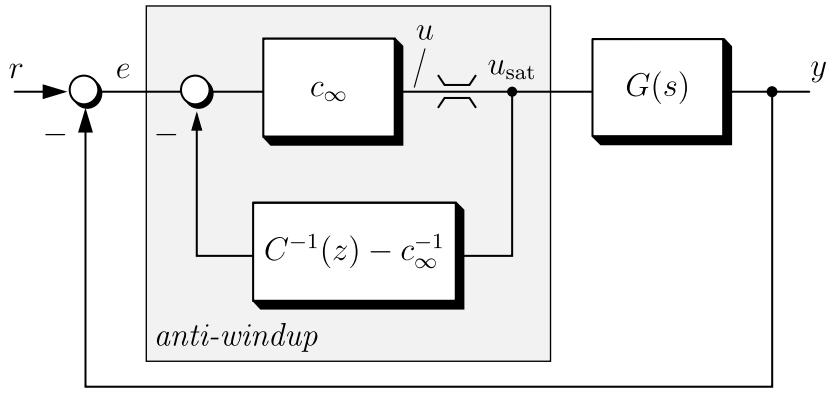


Figura 2.12: Lazo de control con anti-*windup*. Adaptado de [19].

Cuando la salida $u(t)$ calculada excede el límite, se detiene o se compensa la integración del error, de modo que el estado del integrador se mantenga consistente con la actuación efectivamente aplicada [19, 20]. Existen múltiples variantes: integración condicional, realimentación de la diferencia entre actuación calculada y saturada. Todas con el mismo principio: el integrador no debe acumular lo que la planta no está recibiendo.

2.4. Redes neuronales artificiales (ANN)

Una red neuronal artificial es, vista desde fuera, una función paramétrica: transforma un vector de entrada en un vector de salida mediante una composición de operaciones elementales cuyos coeficientes no se programan, sino que se ajustan a partir de ejemplos. Su interés para un problema de identificación no está en que aproxime cualquier función (eso lo hacen otros métodos), sino en que ese ajuste se realiza mediante un procedimiento sistemático y barato, aplicable cuando la relación entre lo medido y lo buscado es conocida sólo de forma implícita. La explicación que sigue toma como referencia principal los textos de Haykin [21], y en Nielsen [22].

2.4.1. Previo a una red: la neurona artificial

La unidad de cómputo elemental es la *neurona artificial*, cuyo modelo se compone de tres elementos [21]. Un *peso* w_{kj} que pondera la señal x_j que llega por ella; un *sumador* $\sum$ que combina linealmente las entradas ponderadas; y una *función de activación* $\varphi(\cdot)$ que acota la amplitud de la salida, típicamente al intervalo $[0, 1]$ o $[-1, 1]$. Estos tres elementos generan la *sinapsis* de una *neurona artificial*.

A esa combinación lineal se añade un término externo, el *sesgo* b_k, que eleva o rebaja la entrada neta de la activación según su signo. La neurona k queda descrita entonces por

$$v_k = \sum_{j=1}^{m} w_{kj}\, x_j + b_k, \qquad y_k = \varphi(v_k)\,, \tag{2.34}$$

donde v_k recibe el nombre de *campo local inducido* o potencial de activación [21]. El efecto del sesgo es aplicar una transformación afín a la salida del sumador: la relación entre v_k y la combinación lineal de las entradas deja de pasar por el origen, lo que permite desplazar

el punto en que la neurona se activa sin alterar la importancia relativa de sus entradas. La Figura 2.13 muestra esa estructura.

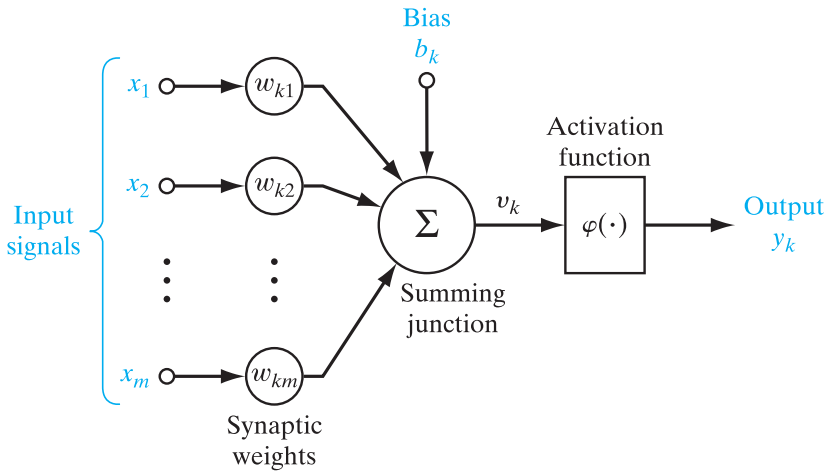


Figura 2.13: Modelo no lineal de una neurona artificial. Extraído de [21].

Lo que distingue a un tipo de neurona de otro es únicamente la elección de su *función de activación* $\varphi(\cdot)$, y esa elección tiene consecuencias que van mucho más allá de la forma de la curva: determina si la red puede entrenarse por gradiente.

2.4.1.a. Neurona perceptrón

La elección más simple es la *función umbral*, que produce una salida binaria según el signo del campo local inducido [21]:

$$\varphi(v) = \begin{cases} 1, & v \geq 0, \\ 0, & v < 0. \end{cases} \tag{2.35}$$

En terminología de ingeniería es la función de Heaviside, y la neurona que la emplea se conoce como modelo de McCulloch–Pitts: su salida vale 1 cuando el campo local inducido es no negativo y 0 en caso contrario, sin valores **intermedios**. Esta propiedad de *todo o nada* es la que da nombre al *perceptrón*.

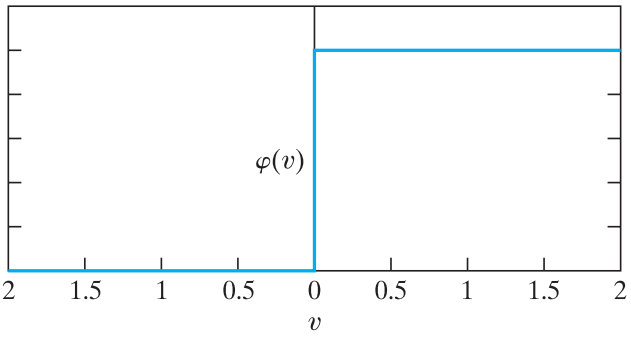


Figura 2.14: Función de activación umbral. La salida conmuta de 0 a 1 al cruzar el origen, sin valores intermedios: la neurona decide, no gradúa. Extraído de [21].

La Figura 2.14 deja ver el problema. Una neurona con salida binaria decide, pero no gradúa: un cambio arbitrariamente pequeño en un peso puede hacer que la salida salte de 0 a 1, y ese salto se propaga y altera por completo el comportamiento del resto de la red. La discontinuidad en $v = 0$ impide derivar la salida respecto de los pesos, es decir, el perceptrón sirve para razonar sobre qué puede representar una red; no sirve para entrenarla por gradiente.

2.4.1.b. Neurona sigmoide

La alternativa que resuelve esa limitación es la *función sigmoide*, definida como una función estrictamente creciente que mantiene un equilibrio entre comportamiento lineal y no lineal. Su forma más habitual es la función logística [21]:

$$\varphi(v) = \frac{1}{1 + e^{-av}}, \tag{2.36}$$

donde a es el *parámetro de pendiente*, cuyo valor fija la inclinación de la curva en el origen, igual a $a/4$.

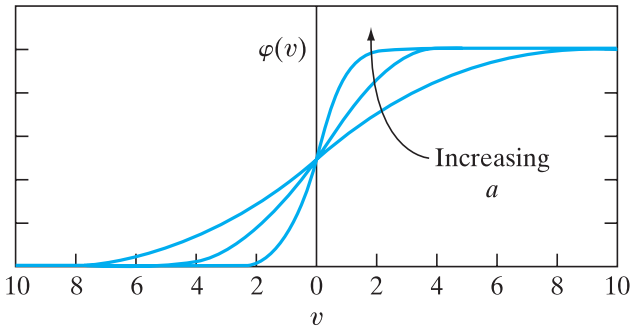


Figura 2.15: Función de activación sigmoide para distintos valores del parámetro de pendiente a. Extraído de [21].

La Figura 2.15 ilustra el efecto de ese parámetro y, con él, la relación entre ambas activaciones: al crecer a la transición se estrecha, y en el límite $a \to \infty$ la sigmoide degenera en la función umbral [21]. La diferencia decisiva es de otra naturaleza. Mientras el umbral sólo toma los valores 0 o 1, la sigmoide recorre de forma continua todo el intervalo entre ambos, y es derivable en todo su dominio. Esa derivabilidad es la que permite calcular cuánto cambia la salida ante una variación pequeña de cada peso, y por tanto la que hace entrenable a la red.

Las activaciones (2.35) y (2.36) recorren el intervalo $[0, 1]$. Cuando conviene que la salida tome también valores negativos se emplean sus versiones impares: la función signo en el caso del umbral, y la tangente hiperbólica en el de la sigmoide [21].

2.4.2. Arquitectura de una red

La forma en que se organizan las neuronas está ligada al algoritmo con que la red se entrena [21]. Entre las arquitecturas posibles, dos concentran la mayor parte del uso práctico y se distinguen por una sola característica: la presencia o ausencia de realimentación.

- **Redes hacia adelante** (*feedforward*): la información recorre la red en un único sentido, de la entrada a la salida, sin que ninguna salida vuelva sobre una entrada previa. La consecuencia es que la salida depende exclusivamente de la entrada presente: la red no tiene memoria, y evaluarla es una operación determinista de costo fijo
- **Redes recurrentes**: existe al menos un lazo de realimentación, de modo que la salida depende también de las entradas anteriores. Esa memoria interna las hace naturales para series temporales, a cambio de un entrenamiento más difícil y de un costo de evaluación que ya no es constante.

2.4.2.a. Redes multicapa hacia adelante (Multilayer)

Dentro de las redes hacia adelante, la variante *Multilayer* es la que incorpora una o más *capas ocultas*, cuyos nodos se denominan *neuronas ocultas*. El término alude a que esa parte de la red no se observa ni desde la entrada ni desde la salida: su función es intervenir entre ambas de modo útil [21]. Al agregar capas ocultas la red queda habilitada para extraer estadísticos de orden superior de su entrada, lo que en términos prácticos significa que puede representar relaciones que una sola capa no alcanza.

La Figura 2.16 muestra la disposición. Los nodos fuente de la capa de entrada suministran los elementos del vector de entrada a las neuronas de la primera capa oculta; las salidas de ésta alimentan a la capa siguiente, y así sucesivamente hasta que el conjunto de salidas de la capa final constituye la respuesta de la red. Al nombrar la arquitectura no se cuenta la capa de entrada, porque en ella no se realiza cómputo alguno.

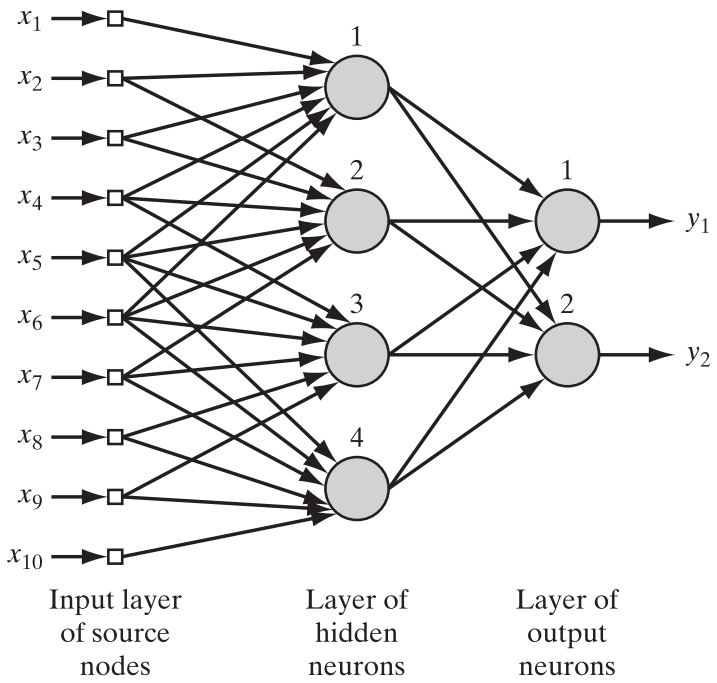


Figura 2.16: Red multicapa hacia adelante con una capa oculta. Extraído de [21].

Agrupando los pesos de una capa en la matriz $\mathbf{W}$ y los sesgos en el vector $\mathbf{b}$, la

operación de una capa completa se escribe de forma compacta como

$$\mathbf{h} = \varphi(\mathbf{W}\mathbf{x} + \mathbf{b}), \tag{2.37}$$

con la activación aplicada elemento a elemento. El apilamiento de capas de este tipo es lo que se conoce como *perceptrón multicapa* (MLP), denominación heredada por razones históricas: estas redes se construyen con neuronas sigmoides y no con perceptrones [22].

2.4.3. Entrenamiento: cómo aprende una red?

Los pesos y sesgos no se programan: se *aprenden* a partir de ejemplos. En el paradigma de *aprendizaje supervisado* se dispone de un conjunto de pares formados por una entrada $\mathbf{x}$ y la respuesta correcta asociada $\mathbf{y}(\mathbf{x})$, y se busca que la salida de la red $\mathbf{a}(\mathbf{x})$ se aproxime a esa respuesta sobre todo el conjunto.

Y el desacuerdo se cuantifica mediante una *función de costo*. La elección habitual en problemas de regresión es el costo cuadrático [22]

$$C(\mathbf{w}, \mathbf{b}) = \frac{1}{2n} \sum_{\mathbf{x}} \|\mathbf{y}(\mathbf{x}) - \mathbf{a}(\mathbf{x})\|^2, \tag{2.38}$$

donde $\mathbf{a}(\mathbf{x})$ es la salida de la red ante la entrada $\mathbf{x}$ y n el número de ejemplos. Entrenar consiste en minimizar C respecto de todos los pesos y sesgos.

2.4.3.a. Ajuste de pesos y sesgos: descenso de gradiente ∇C

El costo (2.38) es una función de *todos* los parámetros de la red a la vez. Conviene apilarlos en una única lista $\mathbf{v} = (w_1, w_2, \ldots, b_1, b_2, \ldots)$, de modo que entrenar sea buscar el punto de esa lista, un punto en un espacio de tantas dimensiones como parámetros tenga la red, donde C es mínimo. Con decenas de miles de parámetros, ese mínimo no puede buscarse por inspección ni resolverse en forma cerrada.

Lo que sí puede calcularse, estando parado en un punto cualquiera, es hacia dónde *crece* el costo. Ésa es la información que contiene el *gradiente* ∇C: un vector con una componente por cada parámetro, donde la componente k-ésima, $\partial C/\partial w_k$, responde a la pregunta *"si muevo un poco este peso y dejo todos los demás quietos, ¿cuánto sube el costo?"*. Un valor positivo grande significa que ese peso está empujando el costo hacia arriba con fuerza; uno cercano a cero, que ese peso apenas influye.

Con esa información, la estrategia es inmediata: mover cada parámetro un poco en el sentido *contrario* al de su propia componente. Formalmente, una variación pequeña $\Delta\mathbf{v}$ de los parámetros modifica el costo según $\Delta C \approx \nabla C \cdot \Delta\mathbf{v}$; eligiendo

$$\Delta\mathbf{v} = -\eta\,\nabla C \qquad \Longrightarrow \qquad \Delta C \approx -\eta\,\|\nabla C\|^2 \leq 0, \tag{2.39}$$

el costo no puede aumentar, porque $\|\nabla C\|^2$ es no negativo y η es positivo [22]. El escalar η se denomina *tasa de aprendizaje* y fija cuán largo es el paso: demasiado pequeño y el descenso es lento; demasiado grande y la aproximación lineal deja de valer, con lo que el paso puede aumentar el costo en lugar de reducirlo.

Escrita parámetro por parámetro, la ecuación (2.39) es la *regla de ajuste*:

$$w_k \;\leftarrow\; w_k - \eta\,\frac{\partial C}{\partial w_k}, \qquad b_l \;\leftarrow\; b_l - \eta\,\frac{\partial C}{\partial b_l}. \tag{2.40}$$

La flecha indica reemplazo: el valor nuevo de cada peso es el viejo menos una fracción de cuánto crece el costo con él. El signo es lo que hace el trabajo. Si aumentar w_k aumenta el costo, la derivada es positiva y la regla *resta*, con lo que w_k disminuye; si aumentar w_k lo reduce, la derivada es negativa y la resta de un número negativo hace que w_k crezca. En ambos casos el parámetro se mueve hacia donde el costo baja, y la magnitud del movimiento es proporcional a cuánto influye ese parámetro. Todos los pesos y sesgos se corrigen simultáneamente, cada uno por su propia cantidad, y el procedimiento se repite.

2.4.3.b. Minilotes: por qué no se usa todo el conjunto?

Aplicar (2.40) tal cual exige conocer el *gradiente* ∇C, y ahí aparece un problema de costo computacional. El costo (2.38) es un *promedio* sobre los n ejemplos, de modo que su gradiente también lo es: para dar un solo paso de ajuste habría que evaluar la red sobre el conjunto completo, calcular el *gradiente* $\nabla C_{\mathbf{n}}$ de cada ejemplo y promediarlos. Con conjuntos de miles de ejemplos, cada paso cuesta miles de evaluaciones, y el entrenamiento se vuelve impracticable.

La salida es no calcular el gradiente exacto, sino *estimarlo* sobre una muestra pequeña. Se escogen al azar m ejemplos $\mathbf{X}_1, \ldots, \mathbf{X}_m$, un *minilote* y se aproxima

$$\nabla C \;\approx\; \frac{1}{m} \sum_{j=1}^{m} \nabla C_{\mathbf{X}_j}, \tag{2.41}$$

promediando únicamente sobre esos m ejemplos. Con esa estimación se aplica la regla de ajuste, los pesos cambian, y sólo entonces se toma el minilote siguiente. Es importante notar que ese minilote nuevo exige un *gradiente nuevo*: como los pesos ya se movieron, las derivadas parciales cambiaron y hay que volver a calcularlas.

El balance es favorable. Cada paso cuesta n/m veces menos, a cambio de una estimación ruidosa del gradiente; en lugar de un paso exacto se dan muchos pasos aproximados, y en conjunto se avanza más rápido. Cuando se han recorrido todos los ejemplos del conjunto, minilote a minilote, se dice que se ha completado una *época* de entrenamiento [22]. Conviene retener que una época no es un ajuste sino *muchos*: con n ejemplos y minilotes de m, cada época aplica n/m veces la regla (2.40).

2.4.3.c. El optimizador

Hasta aquí la regla de ajuste usa la misma tasa de aprendizaje η para todos los parámetros y en todo momento. Eso es lo que se conoce como descenso de gradiente estocástico simple, y funciona, pero desperdicia información: parámetros distintos tienen gradientes de magnitudes muy distintas, y un η único obliga a elegirlo lo bastante pequeño para el parámetro más sensible, con lo que todos los demás avanzan más lento de lo que podrían.

Un *optimizador* es el algoritmo que se interpone entre el gradiente y el ajuste: recibe las derivadas parciales y decide con ellas cuánto mover cada parámetro. La retropropagación de la Subsección 2.4.4 entrega la *dirección*; el optimizador determina el *paso*.

En este trabajo se emplea Adam [23]. Su idea es mantener, para cada parámetro por separado, una memoria de sus gradientes recientes: el promedio de sus valores y el de sus cuadrados. Con el primero suaviza el ruido que introduce el muestreo por minilotes, de

modo que la dirección efectiva no salte con cada lote; con el segundo mide cuán grandes vienen siendo los gradientes de ese parámetro en particular y escala su paso en consecuencia. El efecto es que cada parámetro recibe su propia tasa de aprendizaje, ajustada sola: los que reciben gradientes consistentemente pequeños siguen avanzando, y los que reciben gradientes grandes y erráticos se frenan.

El entrenamiento recorre múltiples épocas vigilando el costo sobre un subconjunto de *validación* que no interviene en el ajuste. Si el costo de entrenamiento sigue bajando mientras el de validación empeora, la red está *sobreajustando*: memorizando los ejemplos en lugar de generalizar y el entrenamiento debe detenerse.

Un punto clave de diseño que gobierna todo el Capítulo 5: la función de costo define *qué* aprende la red. Todo aquello que no esté penalizado en C es invisible para el entrenamiento.

2.4.4. Entrenamiento: cómo se calculan los gradientes por minilote?

La regla (2.40) supone conocidas las derivadas parciales del costo respecto de cada peso y cada sesgo, y calcularlas de forma directa es inviable. La aproximación evidente consiste en perturbar un peso y medir el cambio del costo, $\partial C/\partial w_j \approx [C(\mathbf{w} + \epsilon \mathbf{e}_j) - C(\mathbf{w})]/\epsilon$, pero eso exige una evaluación completa de la red por cada parámetro: una red con un millón de pesos requeriría un millón de pasadas para obtener un solo gradiente [22]. La *retropropagación* [24] obtiene *todas* las derivadas parciales con una única pasada hacia adelante y una hacia atrás, a un costo total comparable al de dos pasadas hacia adelante. Es esa eficiencia la que hace practicable el entrenamiento de redes.

El procedimiento se aplica a cada ejemplo del minilote y consta de cuatro etapas. Las cuatro se describen a continuación.

2.4.4.a. Etapa 1: propagación hacia adelante

La primera etapa es evaluar la red con la entrada del ejemplo, exactamente igual que en operación. Partiendo de $\mathbf{a}^0 = \mathbf{x}$, cada capa calcula primero su campo local inducido y después aplica la activación:

$$\mathbf{z}^l = \mathbf{W}^l \mathbf{a}^{l-1} + \mathbf{b}^l, \qquad \mathbf{a}^l = \varphi\Big(\mathbf{z}^l\Big), \qquad l = 1, \ldots, L. \tag{2.42}$$

La diferencia con una evaluación ordinaria es que aquí **no se descarta nada**: se conservan en memoria todos los $\mathbf{z}^l$ y todos los $\mathbf{a}^l$ de todas las capas. La razón se verá en la etapa 4: las derivadas respecto de los pesos se expresan en función de esas activaciones intermedias, de modo que tirarlas obligaría a recalcularlas.

Al terminar esta etapa se dispone de la salida $\mathbf{a}^L$ que la red produjo para ese ejemplo.

2.4.4.b. Etapa 2: el error en la capa de salida

La segunda etapa compara esa salida con la respuesta correcta $\mathbf{y}$ del ejemplo. La comparación no se usa directamente: se convierte en la magnitud sobre la que trabaja todo el algoritmo, el *error* de una neurona, definido como la sensibilidad del costo respecto de

su campo local inducido [22],

$$\delta_j^l \equiv \frac{\partial C}{\partial z_j^l}. \tag{2.43}$$

Es decir, cuánto cambiaría el costo si esa neurona en particular recibiera un empujón pequeño justo antes de aplicar su activación. La utilidad de definirlo así está en tres propiedades que se aprovechan en cadena: en la capa de salida se calcula de inmediato, en cualquier otra capa se obtiene del de la capa siguiente, y las derivadas que busca (2.40) se escriben directamente en función de él.

En la última capa el error resulta

$$\boldsymbol{\delta}^L = \nabla_{\mathbf{a}} C \odot \varphi'\left(\mathbf{z}^L\right), \tag{2.44}$$

donde $\odot$ denota el producto de Hadamard, esto es, el producto elemento a elemento de dos vectores de igual dimensión [22]. La expresión tiene dos factores con lecturas distintas. Para el costo cuadrático (2.38) el primero se reduce a $\mathbf{a}^L - \mathbf{y}$: la diferencia entre lo que la red produjo y lo que debía producir, es decir, cuánto se equivocó. El segundo, $\varphi'(\mathbf{z}^L)$, atenúa esa diferencia allí donde la activación está saturada, porque en esa zona mover la neurona apenas cambia su salida: equivocarse mucho importa poco si el parámetro no puede corregirlo.

2.4.4.c. Etapa 3: propagación del error hacia atrás

Conocido el error de la capa de salida, el de cualquier capa anterior se obtiene del de la capa que le sigue:

$$\boldsymbol{\delta}^l = \left[\left(\mathbf{W}^{l+1}\right)^\top \boldsymbol{\delta}^{l+1}\right] \odot \varphi'\left(\mathbf{z}^l\right), \qquad l = L-1, \ldots, 1. \tag{2.45}$$

Multiplicar por la traspuesta de la matriz de pesos es repartir el error de cada neurona de la capa siguiente entre las neuronas que la alimentaron, en proporción al peso de cada conexión: una neurona que contribuyó mucho a un error recibe una porción mayor de la responsabilidad. El segundo factor cumple aquí el mismo papel que en (2.44).

Aplicando (2.44) y luego (2.45) de forma recursiva desde la salida hacia la entrada se obtiene el error de *todas* las neuronas de la red en una sola pasada. De ahí el nombre del algoritmo: el error recorre la red en sentido contrario al de la inferencia. Nótese que esta etapa consume los $\mathbf{z}^l$ guardados en la etapa 1.

2.4.4.d. Etapa 4: obtención de las derivadas pesos y sesgos

Con el error de cada neurona, las derivadas que necesita la regla de ajuste son inmediatas. Respecto del sesgo, la derivada *es* el error de la neurona:

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l, \tag{2.46}$$

lo cual es coherente con la definición (2.43), pues el sesgo entra sumando directamente en el campo local inducido: empujar el sesgo y empujar el campo local son la misma cosa.

Y respecto de un peso, es el producto del error de la neurona de destino por la activación de la neurona de origen:

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \, \delta^l_j. \tag{2.47}$$

Aquí se ve por qué había que guardar las activaciones: a^{l-1}_k es un valor calculado en la etapa 1. La expresión admite además una lectura útil: un peso sólo importa en la medida en que por él circule señal. Cuando la activación de entrada es pequeña, el gradiente también lo es, y ese peso aprende con lentitud.

2.4.4.e. Síntesis: algoritmo de entrenamiento completo

Reuniendo las piezas de esta sección, el entrenamiento de una red procede así:

1. **Inicializar** todos los pesos y sesgos con valores aleatorios pequeños.
2. **Tomar un minilote** de m ejemplos escogidos al azar del conjunto de entrenamiento.
3. Para cada ejemplo del minilote:
   a) **Propagar hacia adelante:** desde $\mathbf{a}^0 = \mathbf{x}$ hasta la salida $\mathbf{a}^L$, calcular capa por capa $\mathbf{z}^l = \mathbf{W}^l\mathbf{a}^{l-1} + \mathbf{b}^l$ y $\mathbf{a}^l = \varphi(\mathbf{z}^l)$, guardando todos los $\mathbf{z}^l$ y $\mathbf{a}^l$.
   b) **Evaluar el error de salida** con (2.44), comparando $\mathbf{a}^L$ con la respuesta correcta.
   c) **Retropropagar** con (2.45) desde $l = L-1$ hasta $l = 1$, obteniendo el error de cada capa a partir del de la siguiente, hasta la primera capa oculta.
   d) **Acumular las derivadas** de ese ejemplo con (2.46) y (2.47).
4. **Promediar** las derivadas sobre los m ejemplos, según (2.41), para obtener la estimación del gradiente.
5. **Ajustar** pesos y sesgos con la regla (2.40), con el paso que decida el optimizador.
6. **Repetir** desde el paso 2 con el minilote siguiente. Agotados todos los ejemplos se completa una *época*, y se comienza otra reordenando el conjunto al azar.
7. **Detener** cuando el costo de validación deje de mejorar.

2.4.5. Limitaciones de las redes puramente basadas en datos

Una red entrenada exclusivamente con datos aprende la correspondencia entre entrada y salida sin conocimiento alguno del sistema físico subyacente, como una *caja negra*. En problemas donde ese sistema es bien conocido, como el de esta memoria, la aproximación presenta dos limitaciones estructurales [11, 25]. Exige volúmenes de datos gigantes, porque toda la física que la red necesita debe estar implícita en los ejemplos y cubrir con muestras un espacio de operación de alta dimensión resulta costoso. No ofrece garantías de consistencia física: nada impide que la red produzca salidas que violen las ecuaciones del sistema, y fuera del dominio de entrenamiento la extrapolación de un MLP es esencialmente arbitraria.

2.5. Redes neuronales informadas por física (PINN)

2.5.1. Qué hace a una red "informada por física"

En su formulación canónica, una red neuronal informada por física (PINN, *Physics-Informed Neural Network*) incorpora las ecuaciones físicas en el entrenamiento mediante un residuo físico, según la formulación difundida por Raissi et al. [11]. En un sentido más amplio, el aprendizaje automático informado por física también puede incorporar conocimiento del modelo en la construcción de las entradas o en términos de regularización. Estas vías deben distinguirse:

1. En una PINN canónica, se penalizan en la función de pérdida las violaciones de la ecuación diferencial que gobierna el sistema. La red representa una solución y el residuo se obtiene mediante derivación respecto de las coordenadas de entrada.
2. En una arquitectura híbrida informada por física, se combinan objetivos supervisados con cantidades físicamente interpretables derivadas del modelo, ya sea como entradas o como términos de consistencia en la pérdida.

En ambos casos la física guía el aprendizaje, pero las arquitecturas resultantes no son equivalentes. En particular, utilizar un residuo o características físicas no convierte a una red en solucionadora de una ecuación diferencial ni garantiza por sí solo extrapolación fuera del dominio de entrenamiento.

2.5.2. Familias de PINN

Conviene distinguir las dos familias clásicas, porque determinan por completo el diseño de la red [11]:

1. **PINN directa (*forward*):** se conocen la ecuación diferencial y sus parámetros, y se desea hallar su *solución*. La red representa la solución $u(x,t)$ y se entrena minimizando el residuo de la ecuación junto con las condiciones iniciales y de borde, que anclan la solución única. Es el caso típico de "resolver una EDP con una red" en lugar de un método numérico.
2. **PINN inversa o de estimación de parámetros:** se conoce la forma de la ecuación, pero uno de sus parámetros es desconocido. A partir de mediciones del sistema y de la restricción física, la red *estima ese parámetro*. Aquí el papel de las condiciones se invierte: las mediciones anclan el problema y la ecuación restringe las explicaciones admisibles.

La física aporta estructura y los datos permiten corregir desviaciones del modelo idealizado, pero los términos de pérdida deben balancearse con cuidado: si sus magnitudes son muy diferentes, uno puede dominar el gradiente y hacer que el otro pierda influencia [11, 25]. Esos pesos no se conocen a priori: en la práctica se fijan a uno o por ensayo y error [26], y existen también esquemas que los adaptan durante el entrenamiento [27].

El problema de esta memoria es un problema inverso de identificación de la resistencia de rotor. Sin embargo, las dos propuestas implementadas no son PINN inversas canónicas: ninguna representa la solución de una ecuación diferencial ni reentrena un parámetro

desconocido durante la operación. Las subsecciones siguientes presentan primero la formulación canónica como referencia y luego explican las dos arquitecturas híbridas que se adoptaron para hacer posible la inferencia en línea.

2.5.3. La PINN inversa canónica

En la formulación canónica [11,27] el parámetro físico desconocido λ **no es entrada ni salida de la red**: es una variable entrenable más del optimizador, que aparece *únicamente* dentro del residuo de la ecuación. La red sigue representando la *solución* u, igual que en el caso directo, y el entrenamiento resuelve un problema de optimización conjunta sobre los pesos θ y el parámetro:

$$(\theta^{\dagger}, \lambda^{\dagger}) = \arg\min_{\theta,\lambda} \underbrace{\left\| u_{NN}(\theta) - u_{\text{obs}} \right\|^2}_{\text{ajuste a mediciones}} + \underbrace{\left\| \mathcal{D}\big(u_{NN}(\theta);\, \lambda\big) \right\|^2}_{\text{residuo físico}}. \tag{2.48}$$

Conviene detenerse en qué es cada término, porque la notación compacta esconde el mecanismo.

El símbolo $\mathcal{D}(\cdot\,;\lambda)$ denota el **operador diferencial de la ecuación**, es decir, la ecuación reescrita de modo que su lado derecho sea cero. Si la ecuación es $L\,\ddot{i}+R\,\dot{i}+i/C=0$, entonces $\mathcal{D}(i;R,L,C) = L\,\ddot{i} + R\,\dot{i} + i/C$, y $\mathcal{D}$ evaluado sobre la solución verdadera con los parámetros verdaderos vale exactamente cero. El *residuo* es lo que resulta de evaluar ese operador sobre la salida de la red y con los parámetros que el optimizador está probando: mide cuánto incumple la ecuación la función que la red representa. Las derivadas $\dot{i}$, $\ddot{i}$ no se aproximan por diferencias finitas sino que se obtienen por diferenciación automática, con el mecanismo descrito en la Subsección 2.4.4: se deriva la salida de la red respecto de su entrada, que aquí es el tiempo.

El residuo se evalúa en un conjunto de *puntos de colocación*: instantes escogidos dentro del dominio, que no necesitan tener medición asociada y su promedio cuadrático es el término físico. El primer término, en cambio, se evalúa sólo donde hay mediciones.

Esa asimetría es la que reparte los papeles: las mediciones anclan la solución y la ecuación restringe qué soluciones son admisibles.

2.5.3.a. Un ejemplo: identificación de un circuito RLC

Por su cercanía con el problema eléctrico de esta memoria conviene desarrollar un caso concreto: la identificación de los parámetros de un circuito RLC serie a partir de una única respuesta medida [28, 29]. La ecuación gobernante es de segundo orden,

$$L\,\frac{d^2 i}{dt^2} + R\,\frac{di}{dt} + \frac{1}{C}\,i = 0, \tag{2.49}$$

con dos condiciones iniciales que fijan la solución: la corriente parte de cero y la tensión inicial del condensador se reparte íntegra sobre la inductancia,

$$i(0) = 0, \qquad L\,\frac{di}{dt}\bigg|_{t=0} = V_0. \tag{2.50}$$

La red recibe como entrada el tiempo t y entrega la corriente $i_{NN}(t)$; es una red de dos capas ocultas de 128 neuronas [28].

En la versión *directa*, con R, L y C conocidos, la pérdida suma tres términos: dos de condición inicial y uno de residuo.

$$\mathcal{L}_1 = \big(i_{NN}(0) - 0\big)^2, \tag{2.51}$$

$$\mathcal{L}_3 = \left(L \left.\frac{di_{NN}}{dt}\right|_{t=0} - V_0\right)^2, \tag{2.52}$$

$$\mathcal{L}_2 = \frac{1}{N_c}\sum_{k=1}^{N_c}\left(L \left.\frac{d^2 i_{NN}}{dt^2}\right|_{t_k} + R \left.\frac{di_{NN}}{dt}\right|_{t_k} + \frac{1}{C}\, i_{NN}(t_k)\right)^2, \tag{2.53}$$

donde $t_1, \ldots, t_{N_c}$ son los puntos de colocación y $\mathcal{L}_2$ es el residuo físico de (2.49), la particularización de $\mathcal{D}$ para este problema.

La versión *inversa* cambia dos cosas. Primero, R, L y C dejan de ser constantes y pasan a ser **variables entrenables**: son los λ de (2.48), y el optimizador los ajusta junto con los pesos de la red. Segundo, y puesto que ya no basta con satisfacer la ecuación, ya que cualquier respuesta (R, L, C) admite alguna solución, se añade un cuarto término que obliga a la red a reproducir la respuesta *medida* del circuito bajo estudio [28]:

$$\mathcal{L}_4 = \frac{1}{N_m}\sum_{k=1}^{N_m}\big(i_{NN}(t_k) - i_{\mathrm{med}}(t_k)\big)^2, \tag{2.54}$$

y la pérdida total pondera ese ajuste a datos por encima del resto,

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 + 10\,\mathcal{L}_4. \tag{2.55}$$

Al converger el **entrenamiento**, el valor que tomaron las variables R, L y C es la **estimación** de los parámetros del circuito. **No hay una salida de la red que los entregue: se leen del optimizador.**

En ese ejemplo se conocen la tensión inicial, $V_0 = 12$ V, y la corriente inicial nula, y la respuesta a identificar son 60 muestras sin ruido generadas con la PINN directa; el entrenamiento parte de $R = L = C = 0{,}1$ y se detiene a las 1 000 épocas. Las estimaciones publicadas se apartan de los valores verdaderos hasta un 2 % en R, un 10,7 % en L y un 6,7 % en C, aunque la red reproduce la medida de forma visualmente indistinguible [28]. Esa semejanza no indica falta de unicidad: con V_0 conocido, la pendiente inicial fija L y la dinámica fija R y C, de modo que el problema es **identificable** y está bien condicionado; ajustada a las mismas muestras, la solución analítica recupera los tres parámetros. Lo que el ejemplo ilustra es una dificultad de **optimización**: la ecuación entra solo como penalización y la red puede reproducir los datos con parámetros que todavía no convergieron. Otra implementación del mismo ejemplo, con otro optimizador, los recupera con errores de a lo sumo un 0,5 % [29].

Por lo cual, la PINN inversa convencional está condicionada a: **el resolver el problema inverso por optimización, una vez por caso**. **Cada nueva respuesta desconocida exige un entrenamiento completo, y el resultado no se transfiere al caso siguiente**. En los trabajos citados eso es irrelevante, porque la identificación se hace fuera de línea sobre una trayectoria ya registrada. En esta memoria no lo es: la corrección debe producirse *dentro* de un lazo de control que decide cada 100 μs. Un procedimiento que exige reentrenar una red por cada valor a identificar no puede ocupar ese lugar.

2.5.4. Propuestas de capas correctivas basadas en redes neuronales

La PINN inversa de la subsección anterior no es trasladable tal cual al lazo de control, pero sí lo son sus dos ideas de fondo: que una red puede aprender a identificar un parámetro físico, y que el conocimiento del modelo puede incorporarse al aprendizaje en lugar de dejar que la red lo redescubra. A partir de ellas se plantearon y llevaron a simulación **dos propuestas de capa correctiva**, con filosofías opuestas respecto de cuánto trabajo hace la red.

Ambas comparten la misma interfaz con el controlador: no tocan la función de costo del MPC, ni las señales de conmutación, ni la estructura del control predictivo. Lo único que modifican es el valor de resistencia de rotor $R_{r,\mathrm{MPC}}^{\mathrm{eff}}$ con que el PTC trabaja. El controlador conserva íntegra su estructura y la capa sólo corrige el parámetro con que predice.

La diferencia entre ambas no está en la arquitectura de las redes ni en el objetivo, sino en **qué se le pide a la red que aprenda**.

1. La primera construye, con la física del propio predictor, un estimador auxiliar del desajuste, y emplea las redes para *juzgarlo*: no estiman la resistencia, sino la **fidelidad de esa estimación**, esto es, cuánto se desvía y cuánto merece confiarse en ella.
2. La segunda entrega a la red el muestreo temporal de las corrientes y le pide directamente el parámetro: la red aprende de la forma de onda, junto con el resto del vector de entrada, y propone una corrección.

La primera propuesta es, en ese sentido, la *menos* neuronal de las dos, porque reparte el trabajo entre un estimador auxiliar y redes neuronales para aplicar la corrección; la segunda lo concentra por completo en una red que estima el error paramétrico y aplicación. Se presentan en ese orden.

Terminología empleada en esta memoria. Ninguna de las dos propuestas es una PINN canónica de la formulación de Raissi et al. [11]: ninguna aproxima la solución de una ecuación diferencial ni construye el residuo derivando la salida neuronal respecto de las coordenadas. La Propuesta 1 combina un estimador físico analítico con redes que corrigen su sesgo y califican la inferencia; la Propuesta 2 es un estimador inverso amortizado cuya pérdida incluye consistencia física. Se emplea el término *informadas por física* para destacar esa integración entre modelo y aprendizaje, no para afirmar que las redes resuelvan una ecuación diferencial.

2.5.4.a. Sistema de varias redes: des-sesgar un estimador afín al error

La **primera propuesta** se construye sobre una observación acerca del propio control predictivo: el MPC ya calcula, en cada período de muestreo, una **predicción de la corriente de estator** en coordenadas α-β para el instante siguiente. Esa predicción utiliza el valor $R_{r,\mathrm{MPC}}^{\mathrm{eff}}$ del modelo interno, de modo que la diferencia entre la corriente medida y la predicha, denominada *residuo de predicción*, contiene la firma del desajuste.

Lo que eleva esa observación de heurística a fundamento es una propiedad estructural: el predictor es una función **afín** de la resistencia de rotor. En consecuencia, si la máquina

se comportara como el predictor evaluado con su resistencia verdadera y con los mismos estados y entradas, el residuo resultaría proporcional al error de resistencia,

$$\boldsymbol{\varepsilon} \;=\; \boldsymbol{\phi}\,\big(R_{r,\mathrm{MAS}} - R^{\mathrm{eff}}_{r,\mathrm{MPC}}\big), \tag{2.56}$$

donde $\boldsymbol{\phi}$ es la sensibilidad del predictor respecto de la resistencia, calculable analíticamente a partir de las mismas señales del controlador. Lo exacto es la identidad entre dos predicciones calculadas con los mismos estados; su traslado a la corriente medida es una hipótesis que la planta cumple solo de forma aproximada, porque los estados que usa el predictor son estimados, el predictor es un paso discreto y ni el modelo ni la medición son ideales (Subsección 5.1.1.d). Proyectando el residuo sobre esa sensibilidad a lo largo de una ventana de muestras se obtiene un estimador del error de resistencia,

$$\hat{d} \;=\; \frac{\langle \boldsymbol{\varepsilon},\, \boldsymbol{\phi}\rangle}{\langle \boldsymbol{\phi},\, \boldsymbol{\phi}\rangle} \;\approx\; R_{r,\mathrm{MAS}} - R^{\mathrm{eff}}_{r,\mathrm{MPC}}, \tag{2.57}$$

expresado directamente en ohm y con signo interpretable.

El punto de partida de esta propuesta es que $\hat{d}$ **no requiere ninguna red**. Se calcula con el modelo del predictor, sin ganancias ajustadas. En el caso ideal (2.56) el residuo escala con la sensibilidad, de modo que el tamaño de $\boldsymbol{\phi}$, que cambia con el punto de operación, se cancela en el cociente (2.57); por eso no incorpora una ganancia empírica calibrada para un único valor de R_r. Conviene precisar el alcance de esa invariancia: con el residuo fijo, multiplicar solo la sensibilidad por c divide la proyección por c; el cociente se conserva cuando residuo y sensibilidad se escalan juntos. Su traslado a otro accionamiento exige, no obstante, disponer de los parámetros R_s, L_s, L_r y L_m correspondientes y verificar nuevamente las capas neuronales y sus normalizaciones; la identidad analítica por sí sola no demuestra portabilidad de todo el sistema.

Por qué siguen haciendo falta redes. Si $\hat{d}$ fuese exacto, la física bastaría: leerlo y corregir. En la práctica está **sesgado** y contaminado por efectos que el modelo idealizado no representa por completo, entre ellos la estimación del flujo de rotor con una resistencia desajustada y el *ripple* de conmutación. Las redes aprenden a corregir el **sesgo** de $\hat{d}$ y a valorar la confiabilidad de la inferencia, tomando como entradas descriptores agregados por ciclo en lugar de la forma de onda cruda.

Por qué varias redes y no una. El sesgo de $\hat{d}$ no es uniforme, y ésa es la razón de la especialización. Dos limitaciones lo determinan:

- **Asimetría de observabilidad frente al sentido del desajuste.** Cuando el modelo *sobreestima* la resistencia, el residuo es informativo y $\hat{d}$ tiene el signo correcto, aunque su magnitud está sesgada; cuando la *subestima*, el residuo deja de informar y la estimación se degrada. No son dos casos simétricos del mismo problema, sino dos problemas de identificación con canales de información distintos.
- **Dependencia del régimen.** La observabilidad es mucho mayor bajo carga que en vacío, y el desajuste severo se manifiesta sobre todo durante el arranque, mientras que el moderado lo hace bajo carga sostenida.

En esta memoria, *observabilidad* se usa en un sentido práctico: en qué medida las señales disponibles en una maniobra contienen información sobre R_r. No designa la observabilidad ni la identificabilidad formales de la teoría de sistemas, que no se demuestran aquí. Se caracteriza por simulación (Capítulo 4) y se acompaña con el indicador de alineación ρ^2 (Subsección 5.2.3), que la refleja pero no la certifica.

Por estas limitaciones, surge la organización de redes especialistas, cada una entrenada sobre la región donde su canal de información existe, coordinadas por una lógica que decide a cuál hacer caso en cada momento.

2.5.4.b. Una sola red: estimador inverso amortizado

La **segunda propuesta** lleva la idea del problema inverso a una forma directa para inferencia en línea: una única red recibe las señales del accionamiento y entrega una propuesta de desajuste.

Las entradas son las mediciones que cualquier accionamiento tiene disponibles en línea: las **corrientes de estator**, las **tensiones aplicadas por el inversor** y la **velocidad medida** junto con su referencia. No se le entrega un instante aislado sino una ventana de historia reciente, porque el desajuste de R_r no se manifiesta en un punto sino en la forma de onda: en cómo evoluciona la corriente ante la tensión aplicada. La salida es el desajuste relativo del modelo respecto de la máquina real.

El replanteamiento respecto de (2.48) consiste en **invertir qué entrega la red y qué se entrena**. En lugar de resolver el problema inverso en línea, se entrena una única vez una red que aprende a *predecir la solución* de ese problema a partir de un tramo de mediciones; durante la operación solo se evalúa esa red, mediante una pasada hacia adelante de costo fijo. El costo de la inversión se asume durante el entrenamiento y se amortiza entre los casos posteriores. El resultado se denomina en esta memoria **estimador inverso amortizado con regularización física**.

La diferencia con la formulación canónica es estructural:

- En la PINN inversa canónica, la red presenta la *solución* de la ecuación y el parámetro desconocido es una *variable entrenable* del optimizador. Un caso nuevo exige reentrenar.
- En el estimador inverso amortizado, la red no representa la solución: actúa como un operador que recibe una ventana de mediciones y *entrega el parámetro como salida*. Un caso nuevo se resuelve con una evaluación.

Formalmente, si $\mathcal{I}$ denota el operador que determina un parámetro a partir de mediciones mediante un procedimiento inverso como (2.48), lo que se entrena es una aproximación paramétrica $\mathcal{I}_\theta$ de ese operador,

$$\hat{\lambda} = \mathcal{I}_\theta(\mathbf{u}_{\text{obs}})\,, \qquad \theta^\dagger = \arg\min_\theta \sum_i \Big[\underbrace{\big\|\mathcal{I}_\theta(\mathbf{u}_i) - \lambda_i\big\|^2}_{\text{supervisado}} + \gamma \underbrace{E_i\big(\mathcal{I}_\theta(\mathbf{u}_i)\big)}_{\text{residuo físico}} \Big], \tag{2.58}$$

donde el índice i recorre un conjunto de trayectorias simuladas, λ_i es el parámetro verdadero de cada una, conocido porque se generan en simulación, y $E_i(\cdot)$ es el residuo de la ecuación física de la máquina evaluado con el parámetro que la red propone.

La red no solo estima: también debe indicar cuándo abstenerse. Hay una segunda función, tan importante como la primera, que esta arquitectura debe cubrir. El desajuste de R_r **no es observable en todo régimen**: existen situaciones en que valores distintos de R_r producen señales semejantes, porque la sensibilidad de las señales a R_r casi se anula, como ocurre en vacío y en régimen. Un estimador que se limite a producir un número puede entregar allí una inferencia sin respaldo suficiente.

Si la red interpreta una ventana de vacío como si perteneciera a un régimen parecido a los que vio con carga, propondrá una corrección sin fundamento, y *alejará* $R^{\text{eff}}_{r,\text{MPC}}$ del valor real en el momento en que menos información hay para volver. Por eso la red entrega, además del desajuste estimado, indicadores de **en qué medida el régimen presente permite estimar**, y la capa correctiva los consulta antes de escribir. Estimar y saber cuándo no estimar son, en este problema, dos requisitos igual de exigentes.

Limitación. Este estimador aprende una relación entre formas de onda y desajuste que está atada a *esta* máquina: sus entradas se normalizan con las bases de corriente, tensión y velocidad del accionamiento concreto, y su referencia es la resistencia nominal de la máquina. Sirve, por tanto, para seguir los cambios de R_r frente a la temperatura o a la degradación respecto de ese valor de placa, que es el caso de uso previsto. No es un estimador universal: llevarlo a otra máquina exige regenerar el conjunto de datos y reentrenar por completo.

El Capítulo 5 desarrolla primero el sistema de expertos basado en el estimador afín, correspondiente a la Propuesta 1. A continuación, el Capítulo 6 presenta el estimador inverso amortizado de red única, correspondiente a la Propuesta 2.

Capítulo 3

Implementación del control predictivo de torque y flujo (PTC)

El Capítulo 2 estableció los fundamentos generales: el modelo dinámico de la máquina en coordenadas de campo orientado, el principio del control predictivo de conjunto finito y la estructura del control realimentado con PI. Este capítulo los ensambla en el controlador concreto del trabajo. El capítulo cierra con el análisis de la sensibilidad paramétrica del controlador frente al desajuste de la resistencia de rotor, preparando la caracterización cuantitativa del Capítulo 4.

3.1. Arquitectura del control predictivo

El esquema implementado corresponde al control predictivo de torque propuesto por Rodríguez et al. [7], organizado en dos lazos en cascada: un controlador PI externo de velocidad entrega la referencia de torque T^* al algoritmo predictivo interno, que selecciona directamente los estados de conmutación del inversor. La Figura 3.1 muestra la arquitectura general.

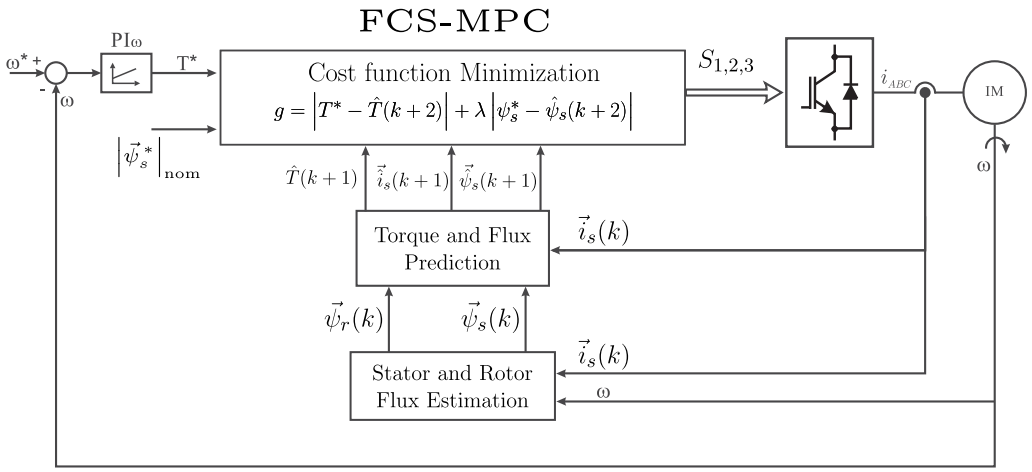


Figura 3.1: Esquema del control predictivo de torque y flujo (PTC) implementado en esta memoria, para una máquina de inducción alimentada por un inversor 2L-VSI.

3.2. El inversor 2L-VSI como conjunto finito de actuaciones

El accionamiento se compone de una máquina de inducción alimentada por un inversor trifásico de dos niveles. El convertidor toma la energía de un enlace de continua y la entrega a través de tres piernas, una por fase, cada una formada por dos interruptores IGBT que operan de forma complementaria; en consecuencia, la tensión que cada pierna aplica a su fase solo puede tomar dos valores discretos. La Figura 3.2 muestra la topología.

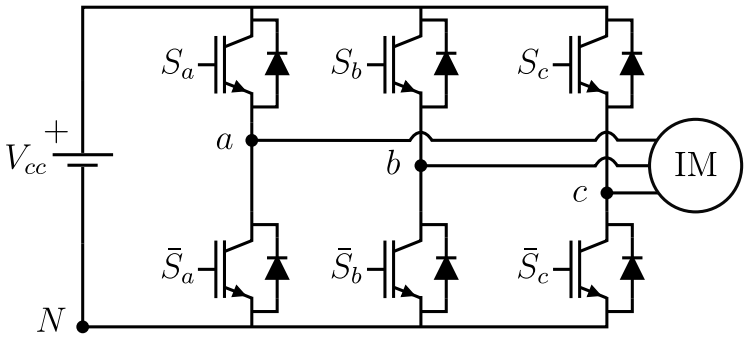
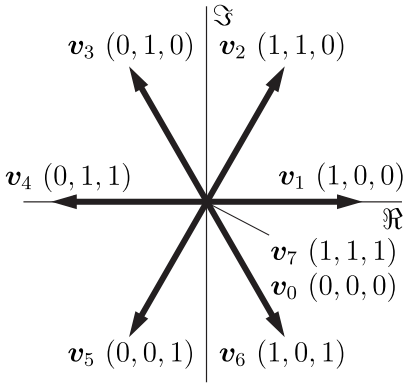


Figura 3.2: El inversor 2L-VSI. Adaptado de [5, 8].

Para una pierna cualquiera, como la de la fase a, por ejemplo, la tensión respecto al neutro vale $v_{aN} \in \{0,\, V_{dc}\}$ según cuál de sus dos interruptores conduzca, de modo que las tres piernas generan $2^3 = 8$ combinaciones de conmutación (S_a, S_b, S_c) [5, 30]. Aplicando la definición del fasor espacial (2.1) a las tensiones de fase que produce cada combinación se obtienen los vectores de tensión disponibles , que recoge la Figura 3.3.



$$\boldsymbol{v}_i = \frac{2}{3} V_{dc} \left(S_a + \boldsymbol{a}\, S_b + \boldsymbol{a}^2\, S_c \right), \quad i = 0, \ldots, 7, \quad (3.1)$$

Figura 3.3: Vectores de tensión del inversor 2L-VSI. Adaptado de [5, 8].

Donde V_{dc} es la tensión del enlace de continua. Los ocho estados de conmutación producen seis vectores activos $\boldsymbol{V}_1$–$\boldsymbol{V}_6$, de igual magnitud y separados 60° en el plano complejo $\alpha\beta$, más los dos vectores nulos $\boldsymbol{V}_0$ y $\boldsymbol{V}_7$ situados en el origen, que corresponden a las combinaciones $(0, 0, 0)$ y $(1, 1, 1)$ y cortocircuitan la carga a un mismo riel.

Este conjunto discreto y finito es exactamente el espacio de búsqueda del algoritmo predictivo: en lugar de sintetizar una tensión de referencia continua mediante modulación, el algoritmo evalúa el efecto futuro de cada uno de los ocho vectores y aplica directamente el mejor durante el período de muestreo completo

3.2.1. Zona lineal y sobremodulación: la cota que impone el hexágono

El conjunto de tensiones que el inversor puede sintetizar *en promedio* no es el hexágono completo. Combinando en el tiempo los seis vectores activos y los dos nulos se alcanza cualquier punto **interior** a él, pero solo dentro del **círculo inscrito** (de radio $V_{dc}/\sqrt{3}$) es posible hacerlo *en cualquier dirección*, se la denomina zona lineal. Más allá, la magnitud alcanzable depende de los vértices (llega a $2V_{dc}/3$).

Operar **fuera** del círculo inscrito es lo que se conoce como **sobremodulación**, y tiene una consecuencia que importa a esta memoria. Para sostener un promedio mayor que el círculo, el algoritmo debe recurrir cada vez menos a los vectores nulos, hasta dejar de usarlos: en el límite recorre el contorno del hexágono saltando entre vértices.

Lo negativo de recorrer el hexágono es que aparecen armónicos de *bajo orden* que se manifiestan como distorsión de la corriente de fase, mientras en zona lineal la combinación entre vectores activos y nulos empuja el contenido armónico hacia frecuencias altas, que la propia inductancia de la máquina filtra

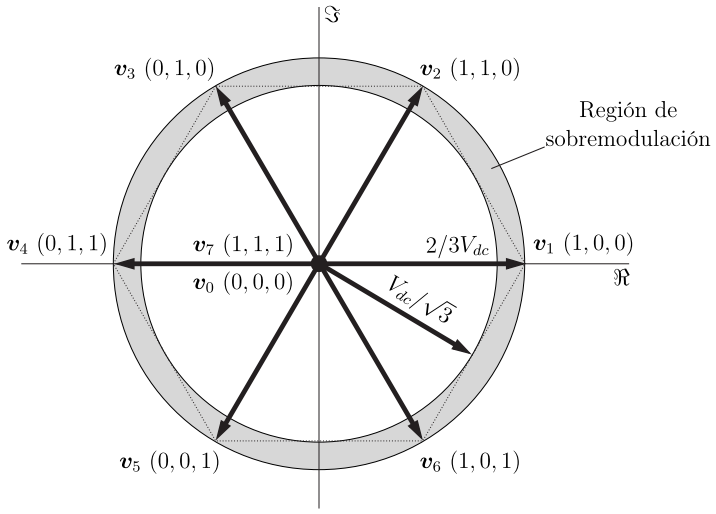


Figura 3.4: Zona lineal y región de sobremodulación en el plano $\alpha\beta$. Adaptado de [31].

En un esquema con *modulador* esta frontera se gestiona de forma **explícita**. En un control predictivo de estados finitos **no hay modulador**: la función de costo elige cada período uno de los ocho vectores, y la sobremodulación no es una estrategia programada sino *lo que ocurre* cuando el objetivo solicitado resulta inalcanzable, donde el vector nulo pierde sistemáticamente la comparación y deja de seleccionarse.

De ahí se sigue una consecuencia de diseño: la única forma de **evitarla** es actuar sobre la *referencia*, no sobre los candidatos, cuya magnitud está cuantizada ($\{0,\ 2V_{dc}/3\}$) y no admite acotación intermedia.

Esta observación caracteriza de la *referencia* de flujo. Como la tensión que la máquina exige es aproximadamente $|\boldsymbol{v}_s| \simeq \omega_s|\boldsymbol{\psi}_s|$, mantener la operación dentro de la zona lineal equivale a limitar la referencia como:

$$|\boldsymbol{\psi}_s| \ \leq \ \frac{V_{dc}}{\sqrt{3}\,\omega_s}, \tag{3.2}$$

que la Subsección 3.5.2 convierte en la *referencia* de flujo efectiva que emplea el controlador.

3.3. Lazo externo de velocidad: diseño del controlador PI

El lazo externo emplea un controlador PI discreto con anti-*windup*(Sección 2.3) y su salida es la *referencia* de torque T^* del algoritmo predictivo. Para su diseño, el lazo interno MPC se aproxima como un retardo de primer orden con constante de tiempo $2T_s$, correspondiente a la predicción a dos pasos. La planta vista por el PI es la ecuación mecánica (2.16) sin roce viscoso, escrita en la velocidad eléctrica $\omega_{em} = p\,\omega_m$ con que trabaja el controlador, de modo que la función de transferencia total resulta

$$G(s) = \frac{1}{1 + 2T_s\, s} \cdot \frac{p}{J\, s}. \tag{3.3}$$

Conforme al criterio de separación de escalas de la cascada (Subsección 2.3.2), el lazo de velocidad debe quedar al menos un orden de magnitud por debajo de la frecuencia de muestreo del lazo interno. Inicialmente se evaluaron los criterios clásicos del óptimo de módulo y del óptimo simétrico [19, 32]: el óptimo simétrico produjo una respuesta excesivamente lenta para este esquema, y el óptimo de módulo, si bien funcional, no explotaba la rapidez del lazo interno predictivo, cuya respuesta se completa en uno a dos períodos de muestreo. Se optó, en consecuencia, por un ajuste directo por respuesta temporal mediante la herramienta `rltool` de MATLAB® [20], con las siguientes especificaciones: amortiguamiento $\zeta \approx 0{,}83$ (sobrepaso inferior al 2 %), tiempo de asentamiento del lazo de velocidad de aproximadamente 0,7 ms ($\approx 7\,T_s$) y un ancho de banda de lazo cerrado del orden de 985 Hz, que cumple la separación de $\sim \times 10$ pasos respecto de la frecuencia de muestreo del algoritmo predictivo ($f_s = 10$ kHz). Ese ajuste se hizo con la inercia del primer modelo de la plataforma, $J = 0{,}0055$ kg·m^2, y sus coeficientes se conservaron al pasar a la máquina simulada, de $J = 0{,}06$ kg·m^2: son los que emplean todas las simulaciones de las dos propuestas. Con esa inercia mayor, el mismo controlador da un lazo más lento, que se verifica a continuación.

El controlador resultante se implementa en la estructura discreta con anti-*windup* [4,33]

$$x_w[k] = c_{w1}\, x_w[k-1] + c_{w2}\, T^*[k-1], \tag{3.4}$$
$$T^*[k] = k_w\,(e_\omega[k] - x_w[k])\,, \tag{3.5}$$

donde $e_\omega = p\,\omega^* - \omega_{em}$ es el error de velocidad *eléctrica*, en rad/s eléctricos, con ω^* la referencia de velocidad mecánica; T^* se expresa en N·m y k_w, en N·m·s/rad referidos a la velocidad eléctrica, y la referencia de torque se satura a $\pm 2\,T_{nom}$.

El controlador empleado. Sin saturación, la recurrencia (3.4)–(3.5) equivale a la transferencia discreta

$$C(z) = \frac{T^*(z)}{E_\omega(z)} = k_w\,\frac{z - c_{w1}}{z - (c_{w1} - k_w\, c_{w2})}. \tag{3.6}$$

Con los coeficientes empleados, $k_w = 16{,}1013$ N·m·s/rad, $c_{w1} = 0{,}9999999993$ y $c_{w2} = -4{,}3475 \cdot 10^{-11}$, el polo queda en $z = 1$ y el cero en $z = 1 - 7 \cdot 10^{-10}$, prácticamente encima de él. La constante de tiempo integral, $T_s/(1 - c_{w1})$, es de unas 40 horas, de modo que

en corridas de pocos segundos el controlador actúa como un **proporcional** de ganancia k_w. Sobre la máquina simulada, el lazo resultante cruza a 85 Hz con 82° de margen de fase y tiene un ancho de banda de 99 Hz, sin sobrepaso (Tabla 3.1): queda dos órdenes de magnitud por debajo de los 10 kHz del lazo interno, más separado aún que el orden de magnitud que pide el criterio de diseño.

Tabla 3.1: Lazo de velocidad en pequeña señal con los coeficientes empleados y la planta (3.3) discretizada a $T_s = 100$ μs, con el error en velocidad eléctrica ($p = 2$) y la inercia J en kg·m^2. MF: margen de fase; BW: ancho de banda del lazo cerrado a -3 dB; t_s: asentamiento al 2 %.

Máquina	**Cruce**	**MF**	**BW**	**Sobrepaso**	$\boldsymbol{t_s}$
Simulada, $J = 0{,}06$	85 Hz	82°	99 Hz	0 %	6,5 ms
Primer ajuste, $J = 0{,}0055$	696 Hz	36°	1196 Hz	34 %	2,2 ms

Consecuencias. Como no integra, el controlador deja en régimen un error de velocidad proporcional al torque de carga, $T_L/(k_w\,p)$: 5,4 r/min a medio par nominal y 10,8 r/min al par nominal, que la simulación confirma. Es la pequeña caída de velocidad, menor que el 0,8 % de la nominal, que muestra el modelo ajustado en los capítulos siguientes. Ese análisis de pequeña señal vale fuera de la saturación: mientras el error supere 21,6 r/min, la referencia queda saturada en $\pm 2\,T_{nom}$ y el arranque transcurre a torque máximo, fijado por el techo de torque del PTC y no por el controlador. El ajuste se conservó por su suavidad: los dos controladores con integral efectiva ajustados para $J = 0{,}06$ kg·m^2 eliminan esa caída, pero multiplican el rizado de la referencia de torque por 15 a 20 y el del torque y la velocidad por 2 a 3, con el mismo arranque. Como todas las simulaciones de las dos propuestas lo emplean, las comparaciones entre ellas no dependen de esa elección.

3.4. Flujo de trabajo: control predictivo (PTC)

Descritos ya el conjunto de actuaciones disponibles y el lazo que fija la *referencia* de torque T^*, puede explicarse con precisión qué *hace* el controlador en cada período de muestreo. El objetivo es siempre el mismo: **evaluar los ocho vectores de tensión del inversor y aplicar aquel que minimiza la función de costo del PTC**:

$$\begin{aligned} g_i = {} & \Big| T^* - \hat{T}_e[k+2] \Big| + \lambda \Big| \Psi_s^* - \big| \hat{\boldsymbol{\psi}}_s[k+2] \big| \Big| \\ & + \mu[k]\, k_i \big[\, |\boldsymbol{i}_s[k+2]| - i_s^{\text{lím}} \big]^+, \end{aligned} \tag{3.7}$$

Sus dos primeros términos miden dos apartamientos a la vez:

- El torque eléctrico predicho respecto de la referencia T^* que entrega el PI de velocidad de la Sección 3.3.
- La magnitud del flujo de estator respecto de la referencia Ψ_s^* que mantiene magnetizada la máquina.

De modo que el vector ganador es el que mejor concilia ambos objetivos en el instante evaluado.

El último término es una penalización blanda, con $[x]^+ = \max(0, x)$: no descarta ningún candidato, solo encarece al que excede el límite, de modo que siempre existe un vector ganador. Acota la magnitud de la corriente, con $i_s^{\text{lím}} = 33{,}4$ A, y actúa solo durante la premagnetización: $\mu[k] = 1$ mientras la máquina no se haya declarado magnetizada y $\mu[k] = 0$ desde entonces (Subsección 3.4.4). Su ganancia se calcula con (3.29), evaluada con el techo vigente de la referencia de torque (3.24).

El código implementa además una penalización de la misma forma sobre la corriente de eje q, para los candidatos cuya $|i_{sq}[k+2]|$ supera el valor óptimo i_{sq}^{opt} de (3.34). No se incluye en (3.7) porque no llega a actuar: durante la premagnetización la penalización de magnitud ya acota la corriente, y fuera de ella el techo de la referencia de torque mantiene la demanda de i_{sq} lejos de ese umbral, con a lo sumo 30,4 A frente a 53,2 A en los arranques con el modelo ajustado. Se comprobó en las campañas de las dos propuestas: en la de la Propuesta 2 ningún candidato supera el umbral, y en la de la Propuesta 1, en los casos de menor margen, las corridas repetidas con el término y sin él son idénticas.

3.4.1. Etapa de estimación

1. En cada período se miden las corrientes trifásicas i_a, i_b, i_c y la velocidad mecánica, ω_m, de la que se obtiene la eléctrica $\omega_{em} = p\,\omega_m$. Las corrientes se llevan al plano estacionario con la transformada de Clarke (2.2) y coordenadas de campo con la proyección (2.5), evaluada con $\theta_k = \gamma_r[k]$, el ángulo **estimado** del flujo de rotor, por integración rectangular de (2.29):

$$\gamma_r[k+1] = \gamma_r[k] + T_s\,\omega_s[k], \tag{3.8}$$

   En el instante k se utiliza $\gamma_r[k] = z^{-1}\{\gamma_r[k+1|k]\}$: el ángulo predicho durante el ciclo anterior se conserva como estimación del ángulo actual.

2. El flujo de rotor se estima mediante la discretización explícita (*Forward Euler*) del modelo de corriente (2.27) [4, 34]:

$$\hat{\psi}_r[k] = \frac{L_m T_s}{\tau_r}\, i_{sd}[k-1] + \left(1 - \frac{T_s}{\tau_r}\right) \hat{\psi}_r[k-1], \tag{3.9}$$

   donde el término $(1 - T_s/\tau_r)$, característico de la discretización explícita, evalúa la dinámica con las variables del instante anterior; la aproximación es adecuada porque $T_s \ll \tau_r$. Se aplica un piso mínimo $\hat{\psi}_r \geq \psi_{r,\text{mín}}$ para evitar la singularidad en el cálculo de la frecuencia de deslizamiento (2.20) durante el arranque, cuando el flujo aún no se ha establecido.

3. La velocidad sincrónica del campo se calcula con la velocidad mecánica y la frecuencia de deslizamiento estimada:

$$\omega_s = \omega_{em} + \frac{L_m\, i_{sq}[k]}{\tau_r\, \hat{\psi}_r[k]}. \tag{3.10}$$

4. El flujo de estator en el instante k se calcula con (2.23)–(2.24),

$$\hat{\psi}_{sd}[k] = k_r\, \hat{\psi}_r[k] + L_\sigma\, i_{sd}[k], \tag{3.11}$$

$$\hat{\psi}_{sq}[k] = L_\sigma\, i_{sq}[k], \tag{3.12}$$

y el torque electromagnético estimado con

$$\hat{T}_e[k] = \frac{3}{2}\, p\, (\psi_{sd}[k]\, i_{sq}[k] - \psi_{sq}[k]\, i_{sd}[k])\,. \tag{3.13}$$

3.4.2. Etapa de predicción a $k+1$

1. Se identifican los voltajes v_s en dq por el vector ganador anterior $\boldsymbol{v}_i^{\text{opt}}[k-1]$ y el ángulo **estimado** del flujo de rotor $\gamma_r[k]$:

$$\boldsymbol{v}_i^{\text{opt}}[k-1] = (v_{sa}, v_{sb}, v_{sc})[k] \xrightarrow{\text{Clarke}} (v_{s\alpha}[k], v_{s\beta}[k]) \xrightarrow{\text{Park}\,\gamma_r[k]} (v_{sd}[k], v_{sq}[k]) \tag{3.14}$$

2. Las ecuaciones de predicción a $k+1$ se obtienen por discretización explícita (*Forward Euler*) de (2.21)–(2.22) y (2.25)–(2.26):

$$i_{sd}[k+1] = \left(1 - \frac{T_s}{\tau_\sigma}\right) i_{sd}[k] + \frac{T_s}{L_\sigma}\left(v_{sd}[k] + \frac{k_r}{\tau_r}\,\psi_r[k]\right) + T_s\,\omega_s\, i_{sq}[k], \tag{3.15}$$

$$i_{sq}[k+1] = \left(1 - \frac{T_s}{\tau_\sigma}\right) i_{sq}[k] + \frac{T_s}{L_\sigma}\left(v_{sq}[k] - k_r\,\omega_{em}\,\psi_r[k]\right) - T_s\,\omega_s\, i_{sd}[k], \tag{3.16}$$

$$\psi_{sd}[k+1] = \psi_{sd}[k] + T_s\,(v_{sd}[k] - R_s\, i_{sd}[k] + \omega_s\,\psi_{sq}[k])\,, \tag{3.17}$$

$$\psi_{sq}[k+1] = \psi_{sq}[k] + T_s\,(v_{sq}[k] - R_s\, i_{sq}[k] - \omega_s\,\psi_{sd}[k])\,, \tag{3.18}$$

$$\psi_r[k+1] = \frac{L_m T_s}{\tau_r}\, i_{sd}[k] + \left(1 - \frac{T_s}{\tau_r}\right)\psi_r[k]. \tag{3.19}$$

3.4.2.a. Compensación del retardo computacional: predicción a k+2

En una implementación digital real, **entre** la medición de las corrientes y la aplicación efectiva del nuevo estado de conmutación, en un FCS-MPC: con sus ocho predicciones y evaluaciones de costo por ciclo, consume una fracción de tiempo apreciable del período de muestreo. En consecuencia, el vector calculado con las mediciones del instante k solo puede aplicarse en el instante $k+1$, y durante el intervalo intermedio sigue actuando el **vector anterior**. Como documentan Cortés et al. [35], **si este retardo se ignora el vector elegido opera sobre un estado que ya no es el supuesto**: la corriente oscila alrededor de su referencia y el *ripple* de corriente y torque aumenta de manera notoria.

La *solución* es considerar ese retardo en la propia formulación: en el instante k se aplica el **vector calculado en el ciclo anterior** $V_{k|k-1}$ y al mismo tiempo, se **predicen los estados** del instante $k+1$ usando el mismo vector $V_{k|k-1}$ que se encuentra efectivamente aplicado en el instante k, **pero la optimización se realiza un paso más adelante** sobre el instante $k+2$, tomando la predicción de los futuros estados $k+1$ como punto de partida. Es decir, se asume que el siguiente vector $V_{k+1|k}$ va a ser aplicado en el instante $k+1$ y no en el mismo instante k.

La Figura 3.5 contrasta ambas situaciones sobre la corriente de una fase: sin compensar, la corriente medida se aparta de su referencia con un rizado que domina la forma de onda; con la predicción desplazada a $k+2$, el seguimiento se ajusta y el rizado se reduce a la banda propia del control de conjunto finito.

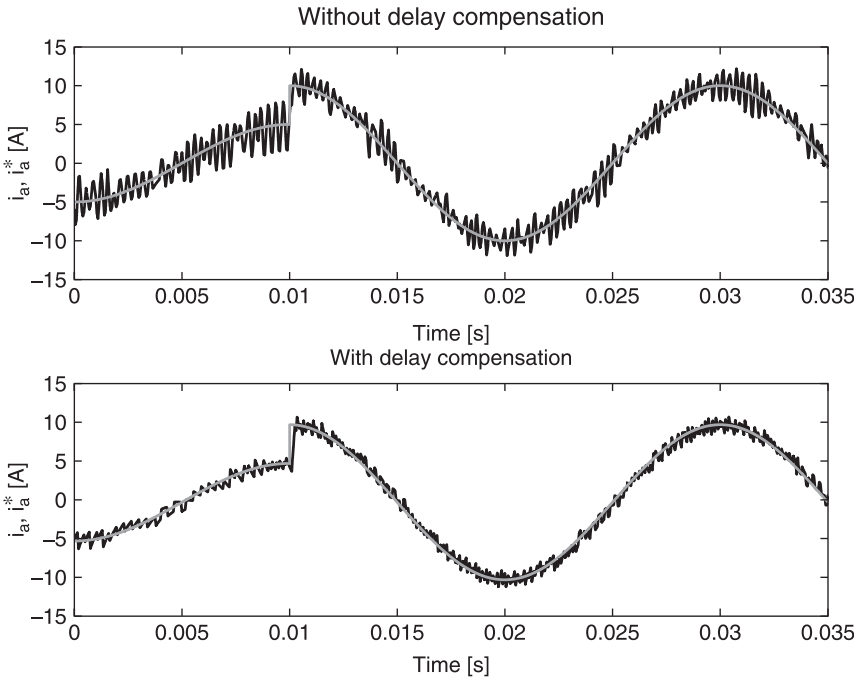


Figura 3.5: Efecto de la compensación del retardo computacional sobre la corriente de fase i_a y su referencia i_a^*. Figura extraída de [5].

El costo adicional es marginal, una sola predicción extra por ciclo, pues el desplazamiento del horizonte reutiliza las mismas ecuaciones y el comportamiento recupera el del caso ideal sin retardo [35].

3.4.3. Etapa de optimización: función de costo sobre $k+2$

Con los estados predichos en $k+1$ como punto de partida, se evalúan los ocho vectores de tensión del inversor. Para cada vector candidato $\boldsymbol{v}_i[k+1]$ de (3.1):

1. Cada vector de tensión se transforma a dq usando el ángulo predicho $\gamma_r[k+1]$ (3.8):

$$\boldsymbol{v}_i[k+1] = (v_{sa}, v_{sb}, v_{sc})[k+1] \xrightarrow{\text{Clarke–Park}\gamma_r[k+1]} (v_{sd}[k+1], v_{sq}[k+1]), \quad i = 0, \ldots, 7, \tag{3.20}$$

2. Se predicen corrientes en $k+2$, con las mismas ecuaciones (3.15)–(3.16) evaluadas en $k+1$

3. Se predicen los flujos de estator a $k+2$, con las mismas ecuaciones (3.17)–(3.18) evaluadas en $k+1$

4. Y se calcula el torque en base a las predicciones en $k+2$:

$$\hat{T}_e[k+2] = \frac{3}{2}\, p\, (\psi_{sd}[k+2]\, i_{sq}[k+2] - \psi_{sq}[k+2]\, i_{sd}[k+2])\,; \tag{3.21}$$

5. Se evalúa la función de costo (3.7) [7, 8]

$$g_i = \left| T^* - \hat{T}_e[k+2] \right| + \lambda \left| \Psi_s^* - \left| \hat{\boldsymbol{\psi}}_s[k+2] \right| \right|,$$

donde T^* es la referencia de torque entregada por el PI de velocidad y Ψ_s^* la referencia de magnitud de flujo de estator. Aquí se muestran solo los dos términos de seguimiento; la penalización de (3.7) se suma en este mismo paso, con la corriente predicha en $k+2$.

6. El vector de tensión que minimiza g_i se selecciona como actuación óptima $\boldsymbol{v}_i^{\text{opt}}[k]$ y sus señales de compuerta se almacenan para aplicarse en el ciclo siguiente, completando la compensación de retardo.

Acerca de la función (3.7), sus dos primeros términos comparan magnitudes de naturaleza y unidades distintas: torque en Nm y flujo en Wb. Por lo que el factor de ponderación λ cumple el doble papel de homogeneizar unidades y de fijar la importancia relativa de ambos objetivos; es el único parámetro de ajuste del lazo predictivo. Como criterio de partida clásico, cuando se desea igual peso para ambos objetivos se adopta la razón entre las magnitudes nominales, $\lambda \approx T_{nom}/\Psi_{s,nom}$ [5]; en este trabajo se emplea $\lambda = 2\,T_{nom}/\Psi_s^*$.

Adicional al MPC del trabajo, **ninguna de las dos referencias es constante en esta plataforma**, y la Subsección 3.4.3.a explica por qué: ambas se adaptan al punto de operación, y hacerlo resulta ser la diferencia entre que la función de costo pondere sus dos términos o que uno de ellos **domine** permanentemente al otro.

3.4.3.a. Referencias adaptadas al punto de operación

Las dos referencias de (3.7) se fijaron inicialmente en valores constantes: Ψ_s^* en el flujo nominal y T^* acotada a $2\,T_{nom}$. Esa elección resulta insuficiente en cuanto la resistencia de rotor se aparta de su valor de diseño, y el motivo no es el torque disponible sino la *frecuencia* sincrónica de trabajo.

1. Primera adaptación: la referencia de flujo Ψ_s^*. Como la frecuencia sincrónica es $\omega_s = p\,\omega_m + \omega_{slip}$, una jaula más resistiva (mayor R_r) exige una ω_s mayor para la misma velocidad ω_m; y por (3.2) esa frecuencia ω_s mayor solo puede trabajar en la zona lineal con un flujo *menor*.

   De (3.2) se sigue directamente:

$$
\begin{aligned}
\Psi_s^*[k] &= \text{máx}\big(0{,}3\,\Psi_{s,nom},\ \Psi_{\text{lím}}[k]\big),\\
\Psi_{\text{lím}}[k] &= \begin{cases} \text{mín}\left(\Psi_{s,nom},\ \dfrac{k_v\,V_{dc}[k]}{\sqrt{3}\,\bar{\omega}_s[k-1]}\right) & \bar{\omega}_s[k-1] > 1 \text{ rad/s},\\ \Psi_{s,nom} & \bar{\omega}_s[k-1] \le 1 \text{ rad/s}, \end{cases}\\
\bar{\omega}_s[k] &= \bar{\omega}_s[k-1] + \frac{T_s}{\tau_f}\Big(\big|\hat{\omega}_s[k]\big| - \bar{\omega}_s[k-1]\Big), \qquad \hat{\omega}_s = p\,\omega_m + \hat{\omega}_{slip},
\end{aligned}
\tag{3.22}
$$

   Donde $\hat{\omega}_s$ es la frecuencia sincrónica *estimada*: velocidad mecánica medida más el deslizamiento que entrega el estimador, y que por tanto depende de $R_{r,\text{MPC}}$. Mientras que k_v es un margen que evita sobredimensionar el flujo frente a la caída resistiva R_s; vale 0,95, y $V_{dc}[k]$ es la tensión del enlace medida.

   La ley acota el flujo con el mínimo de dos valores y le aplica cuatro precisiones de implementación:

(a) Usa el **valor absoluto** de $\hat{\omega}_s$, de modo que vale igual en los dos sentidos de giro.

(b) Lo **filtra** con un primer orden de constante $\tau_f = 10$ ms, lento frente a T_s y rápido frente a la dinámica mecánica. Sin el filtro, la referencia seguiría el rizado de i_{sq} y competiría con su propio lazo.

(c) Emplea la frecuencia filtrada del **período anterior**, $\bar{\omega}_s[k-1]$, porque la del período actual se calcula después, en la etapa de estimación (Subsección 3.4.1).

(d) Tiene dos saturaciones. Mientras $\bar{\omega}_s$ no supera 1 rad/s, como ocurre al arrancar desde el reposo, se usa el flujo nominal. Y la referencia nunca baja del 30 **% del nominal**: por debajo, la máquina quedaría desmagnetizada y el deslizamiento estimado, que divide por $\hat{\psi}_r$, se dispararía.

2. Segunda adaptación: el límite de la referencia de torque T^*. Con una jaula más resistiva, la aplicación aislada de (3.22) *empeora* el resultado: la velocidad cae y la corriente sube. La causa es una realimentación positiva en la función de costo. Con T^* saturada el 100 % del tiempo, el primer término de (3.7) mantiene un error grande que el segundo no puede equilibrar. La optimización deja de repartir esfuerzo entre torque T^* y flujo Ψ_s^* y prioriza la demanda de torque, eligiendo vectores que aumentan i_{sq} mientras disminuye la magnetización asociada a i_{sd}.

$$\Psi_s = \underbrace{(k_r\,\psi_r + L_\sigma\,i_{sd})}_{\text{magnetización}} + j\,\underbrace{L_\sigma\,i_{sq}}_{\text{torque}}, \tag{3.23}$$

La solución es cambiar la referencia máxima del controlador PI, lo que la máquina **puede dar** con el flujo que en ese instante tiene. Así, el torque máximo es regulado a escala con el flujo Ψ_s^{*2} (Subsección 3.5.3),

$$T^*_{\text{máx,eff}} = T^*_{\text{máx}} \left(\frac{\Psi_s^*}{\Psi_{s,nom}} \right)^2 . \tag{3.24}$$

Nótese que por debajo de la velocidad nominal el cociente vale uno y la expresión no tiene efecto alguno. Su función es doble. Equilibra los dos términos de (3.7), devolviendo peso relativo al de flujo; y sobre todo **corta la realimentación positiva**: al topar la referencia, i_{sq} deja de inflarse, el deslizamiento estimado deja de inflarse y el flujo deja de hundirse.

Tabla 3.2: Efecto de las dos adaptaciones sobre una máquina de $R_r = 5\ \Omega$ con el modelo ajustado, con carga nominal y referencia de 1440 rpm.

Referencias	n_{eq} [rpm]	error	$\Psi_s^*/\Psi_{s,nom}$	THD	$I_{\text{máx}}$ [A]
Ambas fijas	1185	17,7 %	1,01	0,14	16,9
Solo flujo adaptado	804	44,2 %	0,60	0,04	30,8
Ambas adaptadas	**1271**	**11,8 %**	**0,72**	**0,05**	21,0

La corriente aumenta de 16,9 a 21,0 A respecto de la configuración original porque, al reducirse el flujo, se requiere una corriente mayor para producir el mismo torque. La segunda adaptación interviene únicamente cuando la primera hace que uno de los términos de la función de costo **domine** sobre el otro.

3.4.4. Etapa adicional: régimen de premagnetización de la máquina

La máquina de inducción no puede producir torque sin un flujo de *rotor* previamente establecido:

$$T_e = \tfrac{3}{2} p\, k_r\, \psi_r\, i_{sq} \tag{3.25}$$

Por lo cual, previo a pedir torque, se necesita magnetizar la máquina con la referencia del flujo de estator Ψ_s^*, ya que:

$$\Psi_s = \underbrace{(k_r\, \psi_r + L_\sigma\, i_{sd})}_{\text{magnetización}} + j\, \underbrace{L_\sigma\, i_{sq}}_{\text{torque}}, \tag{3.26}$$

Se diseña un criterio de pre-magnetización sobre la corriente de magnetización de rotor $i_{mr} = \psi_r / L_m$, cuya dinámica es el retardo de primer orden $\tau_r\, \dot{i}_{mr} + i_{mr} = i_{sd}$.

Mientras i_{mr} no alcance el 95 % de su valor nominal se fuerza $T^* = 0$ en la función de costo, de modo que el algoritmo predictivo inyecta i_{sd} para construir el flujo sin solicitar torque. Alcanzado el valor de i_{mr}, recién el controlador PI de velocidad comienza a entregar referencias de torque.

La etapa termina, y con ella la bandera $\mu[k]$ de (3.7) pasa a cero, por la primera de dos vías:

- cuando la estimación $\hat{i}_{mr} = \hat{\psi}_r / L_m$ alcanza el 95 % de $i_{mr}^* = \Psi_s^* / L_s$;
- o cuando el tiempo alcanza la cota $-\tau_r \ln(1 - 0{,}95) \simeq 3\,\tau_r$, que es lo que tardaría el retardo de primer orden si i_{sd} se mantuviera exactamente en su valor final. Con cualquier corriente mayor tarda menos, de modo que rebasar la cota indica que el lazo no consigue sostener i_{sd}, y conviene continuar antes que bloquear el arranque.

Ambas vías usan $\tau_r = L_r / R_{r,\mathrm{MPC}}$, de modo que la duración de la etapa depende del desajuste. Una vez declarada magnetizada, la máquina no vuelve a esta etapa.

3.4.4.a. Protección contra sobre-corriente

Esta etapa requiere además una protección que el resto de la operación no necesita. Mientras el rotor está desmagnetizado, el término $k_r \psi_r$ del flujo de estator es nulo y, como se demuestra en 3.26, la regulación de $|\boldsymbol{\psi}_s|$ la corriente en d: sin nada que lo impida el algoritmo inyecta cuanta i_{sd} le permita el convertidor, con máximos medidos de 69,1 A.

Se añade por ello a la función de costo una penalización sobre la magnitud de corriente predicha, *únicamente* mientras se esté en esta etapa (el último término de (3.7), con $\mu[k] = 1$),

$$g_{\text{actual}_i} \leftarrow g_{\text{anterior}_i} + k_i \left(|\boldsymbol{i}_s[k+2]| - i_s^{\text{lím}} \right) \quad \text{si, y sólo si} \quad |\boldsymbol{i}_s[k+2]| > i_s^{\text{lím}}, \tag{3.27}$$

El límite se fija al doble del valor de *peak* nominal de placa, $i_s^{\text{lím}} = 2\sqrt{2}\, I_N = 33{,}4$ A con $I_N = 11{,}81$ A eficaces de línea: se admite el doble porque en esta fase no se produce par y conviene establecer el flujo cuanto antes. Además, es una penalización blanda: si ningún vector cumpliera el límite, gana el que menos lo viola y el algoritmo nunca se queda sin actuación. Alcanzada la magnetización el término se retira.

Determinación de la ganancia de penalización. La ganancia k_i es un grado de libertad de diseño. Fijarla con un valor arbitrario sería insatisfactorio; se deriva, en cambio, de una condición sobre la resolución del propio accionamiento.

La corriente no puede variar de forma continua: entre dos decisiones consecutivas, el mayor cambio que un vector activo puede imponerle es

$$\Delta i_s^{\text{máx}} = \frac{2}{3}\,\frac{V_{dc}\,T_s}{L_\sigma}, \tag{3.28}$$

que para la plataforma vale 2,73 A y es el mismo para cualquier componente, pues el incremento lo fija la tensión aplicada sobre la inductancia transitoria. Ese es el paso discreto con que el algoritmo puede aproximarse al límite. La condición adoptada es que **excederlo en un paso completo cueste tanto como el rango entero de la referencia de torque**, $k_i\,\Delta i_s^{\text{máx}} = T^*_{\text{máx}}$, de donde

$$k_i = \frac{3\,T^*_{\text{máx}}\,L_\sigma}{2\,V_{dc}\,T_s}. \tag{3.29}$$

La expresión no contiene constantes libres: todas sus magnitudes son parámetros del accionamiento, por lo que la ganancia se adapta por sí sola a otra máquina o a otro punto de diseño. Para los valores de la Tabla 4.1 resulta $k_i = 26{,}7$ N·m/A. En el código, $T^*_{\text{máx}}$ es el techo vigente $T^*_{\text{máx,eff}}$ de (3.24), que coincide con $T^*_{\text{máx}}$ durante la premagnetización porque allí la referencia de flujo es la nominal.

3.5. Naturaleza del esquema: predictivo en coordenadas de campo

3.5.1. Similitudes y diferencias: FOC y FCS-MPC/PTC

Dado que el algoritmo descrito opera en coordenadas de campo orientado con el flujo de rotor, conviene precisar qué lo mantiene como un FCS-MPC/PTC, para evitar confundirlo con un control por campo orientado FOC convencional.

1. *Coordenadas de campo orientado al flujo de rotor*: el estimador de flujo por modelo de corriente y su ángulo γ_r (Subsección 2.1.8), la descomposición de la corriente i_s con su desacoplamiento en los ejes dq: i_{sd} asociada al flujo, i_{sq} al torque. Y las ecuaciones de estado (2.21)–(2.22) son utilizadas como modelo de predicción.

2. *Ley de control predictiva*: no existen controladores PI internos de corriente ni etapa de modulación, y no se genera una referencia continua de tensión que un modulador deba sintetizar. La actuación se decide en cada período evaluando los ocho estados discretos del inversor con la función de costo (3.7) y aplicando el vector ganador durante el período completo; el único controlador lineal del esquema es el PI externo de velocidad, que genera la referencia T^* pero no interviene en la selección de vectores. En un FOC convencional, por contraste, los lazos internos de corriente calculan una tensión de referencia continua que un modulador traduce a conmutaciones. El esquema implementado es, por tanto, un FCS-MPC/PTC en sentido estricto.

3.5.2. Debilitamiento de campo: referencia de flujo dependiente de la velocidad

En cualquier accionamiento de velocidad en lazo cerrado (FOC o PTC) la frecuencia de estator ω_s deja de ser un dato y pasa a ser una *consecuencia*. La condición de orientación (2.20) calcula el deslizamiento necesario, $\omega_r = L_m i_{sq}/(\tau_r \psi_r)$, y el ángulo de campo γ_r avanza a $\omega_s = \omega_{em} + \omega_r$: el deslizamiento se suma a la frecuencia de alimentación en lugar de restarse a la velocidad. El deslizamiento sigue existiendo y conserva su valor físico, pero la velocidad mecánica queda donde el lazo la mande.

Queda por resolver el problema recíproco: qué ocurre cuando la velocidad pedida supera la nominal. La tensión que la máquina necesita es $|\boldsymbol{v}_s| \simeq \omega_s |\boldsymbol{\psi}_s|$, de modo que sostener el flujo nominal por encima de la frecuencia sincrónica nominal exigiría una tensión que el convertidor **no puede entregar** (Subsección 3.2.1). La solución clásica es *debilitar el campo*: reducir la referencia de flujo en proporción inversa a la velocidad [4], según

$$\Psi_s^*(\omega) = \begin{cases} \Psi_{s,\mathrm{nom}} & p\,|\omega^*| \le \omega_{s,\mathrm{nom}} \\ \Psi_{s,\mathrm{nom}}\,\dfrac{\omega_{s,\mathrm{nom}}}{p\,|\omega^*|} & p\,|\omega^*| > \omega_{s,\mathrm{nom}} \end{cases} \tag{3.30}$$

con lo que el producto $\omega_s|\boldsymbol{\psi}_s|$ permanece acotado y la tensión requerida no crece.

En esta simulación (3.30) no se implementa de esa forma, sino la que ya se introdujo como primera adaptación de las referencias (3.22): es el mismo principio, acotar el producto $\omega_s|\boldsymbol{\psi}_s|$, pero escrito directamente sobre la tensión que el inversor puede sintetizar y evaluado con la frecuencia sincrónica *estimada*, en valor absoluto y filtrada con $\tau_f = 10$ ms, con un piso del 30 % del flujo nominal y el flujo nominal mientras la frecuencia filtrada no supera 1 rad/s.

Esta diferencia no es trivial: como el deslizamiento crece con R_r, una jaula resistiva exigía una frecuencia sincrónica mayor que la de consigna, sobremodulaba sin que el criterio llegara a detectarlo. Sustituir $p\,|\omega^*|$ por $\hat{\omega}_s = p\,\omega_m + \hat{\omega}_{slip}$ evita ese problema, pero la consecuencia es que la referencia de flujo pasa a depender de $R_{r,\mathrm{MPC}}$ a través del estimador de deslizamiento.

3.5.3. Límite de torque eléctrico impuesto por el flujo de estator

La función de costo (3.7) regula la *magnitud del flujo de estator* hacia un valor de referencia Ψ_s^*. Esta elección también condiciona el torque eléctrico máximo que el esquema puede entregar. Bajo la orientación al flujo de rotor adoptada en este capítulo, las componentes de ese flujo son (2.23)–(2.24), de modo que su magnitud resulta

$$|\boldsymbol{\psi}_s|^2 = \underbrace{(k_r\,\psi_r + L_\sigma\,i_{sd})^2}_{\text{eje d del campo}} + \underbrace{(L_\sigma\,i_{sq})^2}_{\text{torque}} \tag{3.31}$$

Si la magnitud total se mantiene constante en Ψ_s^*, todo aumento de i_{sq} debe compensarse con una *reducción* de la componente en el eje del campo y, con ella, del flujo de rotor. Despreciando el término $L_\sigma i_{sd}$, pequeño frente a $k_r \psi_r$,

$$k_r\,\psi_r \simeq \sqrt{\Psi_s^{*2} - (L_\sigma\,i_{sq})^2}, \tag{3.32}$$

y sustituyendo en la expresión del par (2.18) se obtiene el que el esquema puede entregar en función de la corriente que inyecta, a flujo de estator regulado:

$$T_e(i_{sq}) = \frac{3}{2}\, p\, i_{sq} \sqrt{\Psi_s^{*2} - (L_\sigma\, i_{sq})^2}. \tag{3.33}$$

La función (3.33) no es monótona: crece, alcanza un máximo y decrece. Anulando su derivada,

$$i_{sq}^{\text{opt}} = \frac{\Psi_s^*}{\sqrt{2}\, L_\sigma}, \qquad T_e^{\text{máx}} = \frac{3}{2}\, p\, \frac{\Psi_s^{*2}}{2\, L_\sigma}. \tag{3.34}$$

Con los valores de la plataforma ($\Psi_s^* = 0{,}988$ Wb, $L_\sigma = 13{,}12$ mH, $p = 2$) resultan $i_{sq}^{\text{opt}} = 53{,}2$ A y $T_e^{\text{máx}} = 111{,}5$ N·m, es decir $3{,}06\, T_{nom}$. (Ambas cifras arrastran la aproximación introducida en (3.32), que desprecia $L_\sigma i_{sd}$. Restituido ese término, i_{sq}^{opt} no cambia y el techo baja a 102,4 N·m.)

3.6. Sensibilidad paramétrica del control predictivo

3.6.1. Propagación del desajuste de R_r en la cadena de control

Con lo visto en este capítulo se puede determinar con precisión donde un error en la resistencia de rotor degrada el control. Sea $R_{r,\text{MPC}}$ el valor utilizado por el controlador y $R_{r,\text{MAS}}$ el valor real de la máquina. El desajuste $R_{r,\text{MPC}} \neq R_{r,\text{MAS}}$ afecta:

1. **El estimador de flujo** (3.9): una τ_r errónea distorsiona la magnitud del flujo estimado y, a través de (3.8), el ángulo de orientación γ_r. Con la orientación degradada, la descomposición *dq* pierde su significado físico: la componente i_{sd} deja de controlar exclusivamente el flujo y la i_{sq} deja de controlar exclusivamente el torque.

2. **La frecuencia de deslizamiento** (2.20): el error se propaga a la velocidad sincrónica con que se hace girar el marco de referencia, acumulándose en el ángulo.

3. **El predictor de corrientes y flujos** (3.15)–(3.19): las constantes τ_r, R_σ y τ_σ dependen de R_r, por lo que las predicciones a $k+1$ y $k+2$ se desvían de la evolución real y la función de costo (3.7) compara candidatos contra un futuro equivocado, pudiendo seleccionar vectores de conmutación subóptimos.

3.6.2. Soluciones clásicas y sus limitaciones

El problema de la variación de R_r (equivalentemente, de τ_r) es clásico en accionamientos vectoriales, y la literatura ofrece tres familias principales de soluciones [4, 10]:

- Observadores adaptativos por modelo de referencia (MRAS), que comparan dos estimadores del mismo estado con distinta sensibilidad paramétrica y adaptan el parámetro para anular la discrepancia

- Identificación paramétrica en línea, típicamente por mínimos cuadrados recursivos sobre el modelo eléctrico

- Observadores extendidos de Kalman (EKF), que aumentan el vector de estados con el parámetro incierto

Sin embargo, todas comparten tres limitaciones prácticas:

- Son *model-based* en sentido fuerte (su convergencia depende de la exactitud del resto del modelo)
- Introducen dinámicas adicionales que deben sintonizarse con cuidado
- Presentan sensibilidades cruzadas, por ejemplo, entre R_s y R_r a baja velocidad que dificultan su ajuste.

Capítulo 4

Análisis del efecto del desajuste paramétrico en el PTC

El Capítulo 3 concluyó con: la resistencia de rotor interviene simultáneamente en la orientación del marco de control y en el modelo de predicción del PTC, por lo que un valor erróneo contamina toda la cadena de estimación, predicción y selección de vectores. Este capítulo lleva ese análisis al esquema de control y caracteriza cómo se manifiesta el desajuste sobre el accionamiento según su sentido y magnitud. Las señales y estados identificados constituyen la base de los observables utilizados por la Propuesta 1 en el Capítulo 5 y de la Propuesta 2 desarrollada en el Capítulo 6.

4.1. Implementación del PTC en la plataforma PLECS

La simulación se desarrolló en PLECS y comprende el accionamiento completo: la máquina de inducción jaula de ardilla, representada mediante el bloque *Squirrel-Cage IM* parametrizado con los valores de la Tabla 4.1; el inversor fuente de tensión de dos niveles con IGBTs; el enlace de continua; y la carga mecánica. La plataforma contiene **dos bloques** `C-Script` con papeles complementarios:

- El **C-Script principal** ejecuta, una vez por período de muestreo, la totalidad del algoritmo de control descrito en el Capítulo 3, la instrumentación de diagnóstico y, la extracción de características, la inferencia y la coordinación descritas en el Capítulo 5.
- El **C-Script de carga física** genera el torque resistente aplicado al eje mediante un modelo reactivo seleccionable de bomba centrífuga o cinta transportadora. En ambos modos, la carga se opone al movimiento de la máquina, como se detalla en la Subsección 4.1.2.

La Figura 4.1 muestra el esquema tal como está implementado en la plataforma, con los dos bloques `C-Script` y la separación entre la parte de control y la de carga.

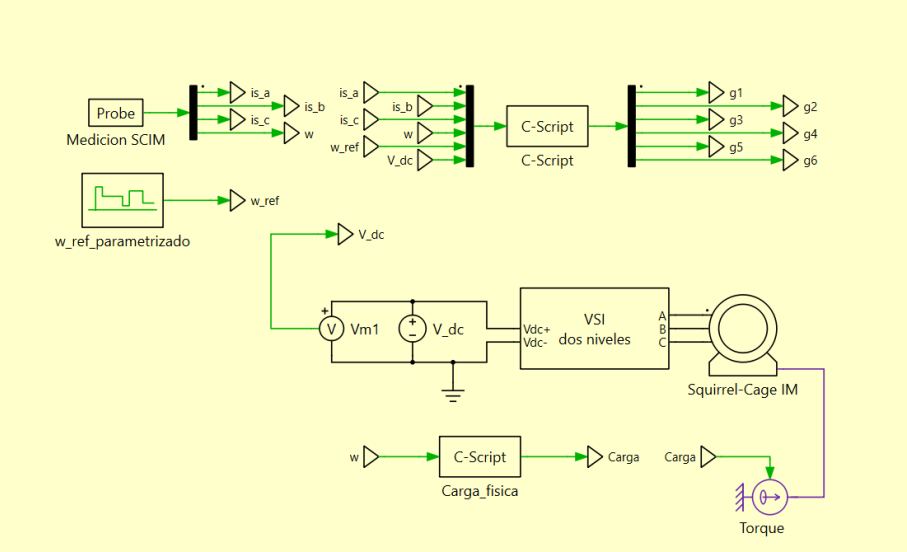


Figura 4.1: Esquema del accionamiento implementado en PLECS.

Las subsecciones que siguen detallan cada pieza de esta implementación: los parámetros con que se instancia la máquina y el controlador,

4.1.1. Parámetros de la máquina y del controlador

La Tabla 4.1 resume los parámetros de la máquina simulada y del controlador, tomados de la inicialización del modelo definitivo de la plataforma. La máquina es el modelo de jaula de ardilla de la biblioteca de PLECS con inductancias constantes: cumple las hipótesis de la Subsección 2.1.1, de modo que no satura ni tiene pérdidas en el hierro.

Tabla 4.1: Parámetros de la máquina de inducción y del controlador FCS-MPC/PTC.

Parámetro	**Símbolo**	**Valor**	**Unidad**
Resistencia de estator	R_s	1,02	Ω
Resistencia de rotor (nominal)	$R_{r,\text{nom}}$	0,83	Ω
Inductancia mutua	L_m	153,6	mH
Inductancia propia de estator	L_s	160,3	mH
Inductancia propia de rotor	L_r	160,3	mH
Momento de inercia	J	0,06	$\text{kg}\,\text{m}^2$
Pares de polos	p	2	–
Velocidad nominal	n_{nom}	1440	rpm
Torque nominal	T_{nom}	$\approx 36{,}5$	$\text{N}\,\text{m}$
Deslizamiento nominal	s_{nom}	$(1500-1440)/1500$	–
Tensión del enlace DC	V_{dc}	$380\sqrt{2}$	V
Frecuencia de muestreo del lazo	f_s	10	kHz
Referencia de flujo de estator	Ψ_s^*	$0{,}577\,V_{dc}/\omega_{red}$	Wb
Ponderación de la función de costo	λ	$2\,T_{nom}/\Psi_s^*$	–
Saturación de referencia de torque	$T^*_{\text{máx}}$	$\pm 2\,T_{nom}$	$\text{N}\,\text{m}$

4.1.2. Modelado físico de la carga mecánica

Las cargas que interesa representar (bombas y cintas transportadoras) son *reactivas*: solo pueden oponerse al movimiento y su torque se anula en reposo.

- **Bomba centrífuga**: carga cuadrática con la velocidad (ley de afinidad de las turbomáquinas [2]),

$$T_L = -K_p\,\omega_m\,|\omega_m| - B\,\omega_m, \qquad K_p = \frac{T_{nom}}{\omega_{m,\mathrm{nom}}^2}, \tag{4.1}$$

  que vale cero en reposo, crece progresivamente con la velocidad y alcanza el torque nominal a la velocidad nominal. El factor $\omega_m|\omega_m|$ garantiza el carácter reactivo (el torque siempre se opone al sentido de giro).

- **Cinta transportadora horizontal**: torque de fricción aproximadamente constante con la velocidad: la resistencia principal de una cinta es proporcional al peso de la correa y del material transportado, con regularización suave en el origen para evitar la discontinuidad de la fricción seca:

$$T_L = -T_{\mathrm{mat}}(t)\,\tanh\!\left(\frac{\omega_m}{\omega_\varepsilon}\right), \tag{4.2}$$

  donde T_{base} representa la fricción de la correa vacía, $T_{\mathrm{mat}}(t)$ el torque debido al material sobre la correa y ω_ε una velocidad de regularización pequeña (1,5 rad/s). En reposo el torque de carga es nulo: la máquina puede quedar *trancada* si su torque de arranque no vence la fricción, pero nunca es impulsada hacia atrás.

4.1.3. Separación entre la resistencia de la máquina y la del controlador

La estructura central que permite el estudio del desajuste es la parametrización **independiente** de la resistencia de rotor en dos lugares distintos de la plataforma:

- **La resistencia de la máquina simulada $R_{r,\mathrm{MAS}}$:** se entrega directamente al bloque de la máquina. Representa el valor físico real que en una máquina de laboratorio sería desconocido; su valor de referencia es el nominal, $R_{r,\mathrm{nom}} = 0{,}83\ \Omega$.
- **La resistencia inicial del controlador $R_{r,\mathrm{MPC},0}$:** se entrega al bloque `C-Script` principal como parámetro independiente. Es el valor con que el controlador construye su estimador de flujo y su predictor al comenzar la corrida.

Cuando ambos valores coinciden se habla del caso *matched* (modelo ajustado); cuando difieren, del caso *mismatch* (desajustado). En este capítulo se adopta la convención:

$$\Delta R_r = R_{r,\mathrm{MPC}} - R_{r,\mathrm{MAS}}, \qquad \varepsilon_R = \frac{R_{r,\mathrm{MPC}} - R_{r,\mathrm{MAS}}}{R_{r,\mathrm{MAS}}}, \tag{4.3}$$

de modo que $\Delta R_r < 0$ significa que el controlador *subestima* la resistencia real y $\Delta R_r > 0$ que la *sobreestima*; el error relativo ε_R acompaña al absoluto porque la gravedad de un mismo error en ohm depende del valor nominal (para esta máquina, $|\Delta R_r| = 1\ \Omega$ representa $\varepsilon_R \approx 120\,\%$).

4.2. Caracterización del comportamiento bajo desajuste: máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$)

Con la separación de la Subsección 4.1.3, la caracterización del escenario de **desajuste de modelo** consiste en barrer pares ($R_{r,\mathrm{MPC}},\ R_{r,\mathrm{MAS}}$) y observar la degradación del control. Esta sección reúne la evidencia para la configuración nominal, $R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$: cuatro barridos, tres con carga de cinta y uno con carga de bomba, con las mismas ocho combinaciones de resistencia cada uno.

El barrido emplea una **escalera fija** de ocho valores de $R_{r,\mathrm{MPC}}$, la Tabla 4.2 resume junto con el desajuste absoluto y relativo que representa cada uno.

Tabla 4.2: Escalera de $R_{r,\mathrm{MPC}}$ empleada en el barrido, con el desajuste resultante frente a la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$).

Caso	$R_{r,\mathrm{MPC}}$ [Ω]	ΔR_r [Ω]	ε_R [%]	Sentido
1	0,10	$-0,73$	-88	subestimación
2	0,30	$-0,53$	-64	subestimación
3	0,50	$-0,33$	-40	subestimación
4	0,83	$0,00$	0	**ajustado** (*matched*)
5	1,50	$+0,67$	$+81$	sobreestimación
6	3,00	$+2,17$	$+261$	sobreestimación
7	5,00	$+4,17$	$+502$	sobreestimación
8	10,00	$+9,17$	$+1105$	sobreestimación

4.2.1. Barridos con carga de cinta

4.2.1.a. Arranque en vacío

La primera maniobra es el **arranque en vacío**: partiendo del reposo, la referencia de velocidad se fija en su valor nominal (1440 rpm) y la cinta se configura sin material, de modo que el torque de carga es idénticamente nulo durante toda la corrida.

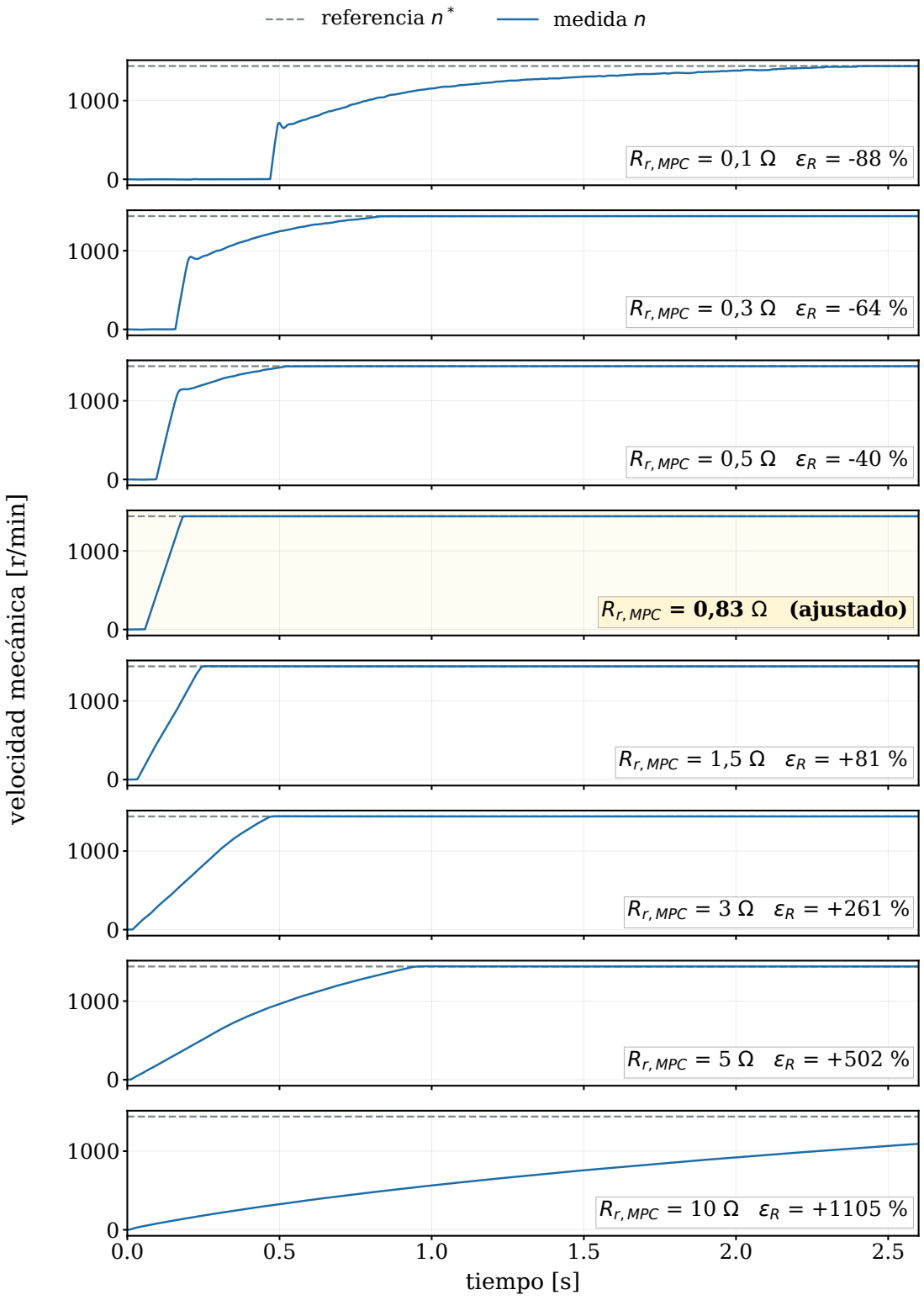


Figura 4.2: Velocidad mecánica frente a su referencia durante el arranque en vacío.

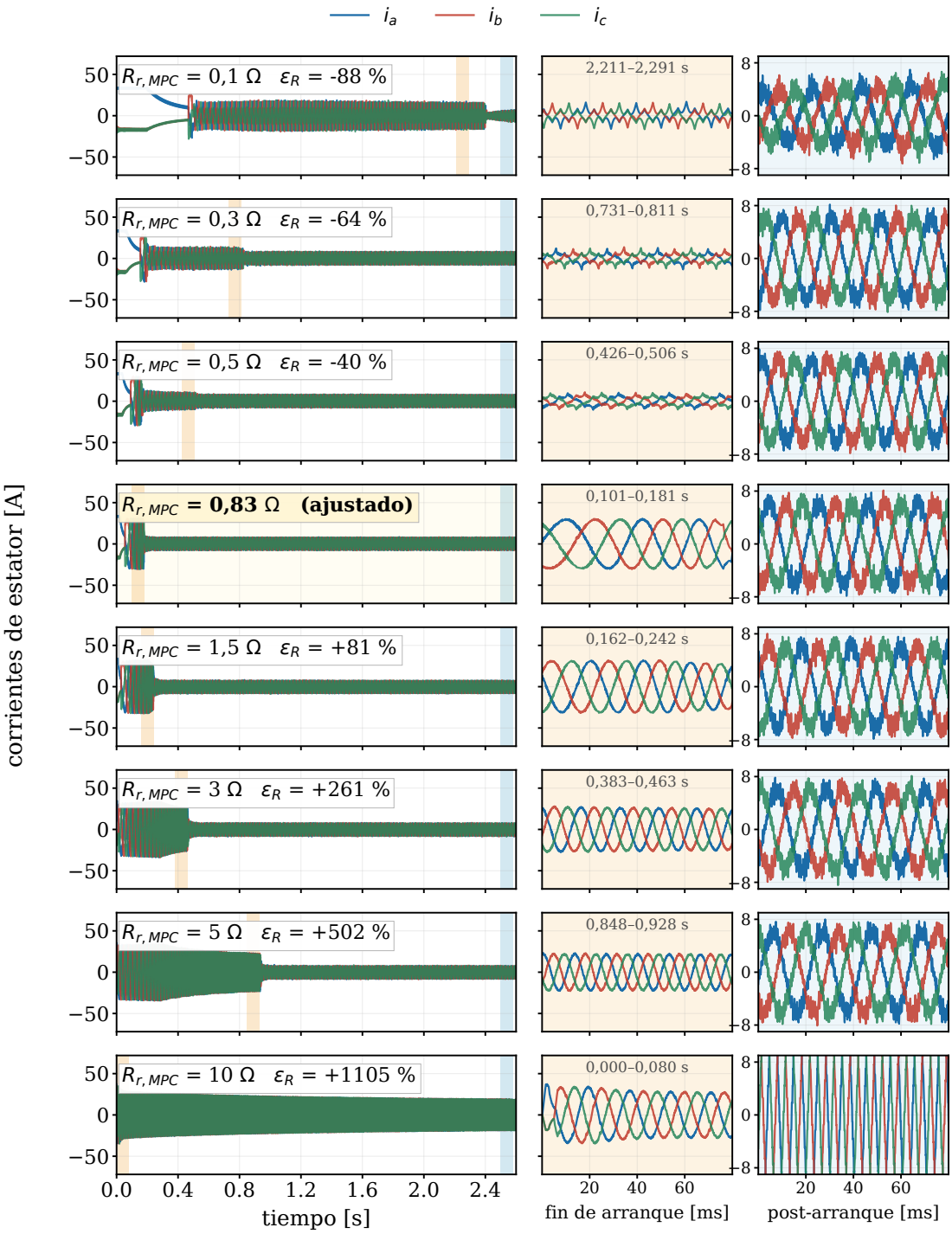


Figura 4.3: Corrientes de estator durante el arranque en vacío.

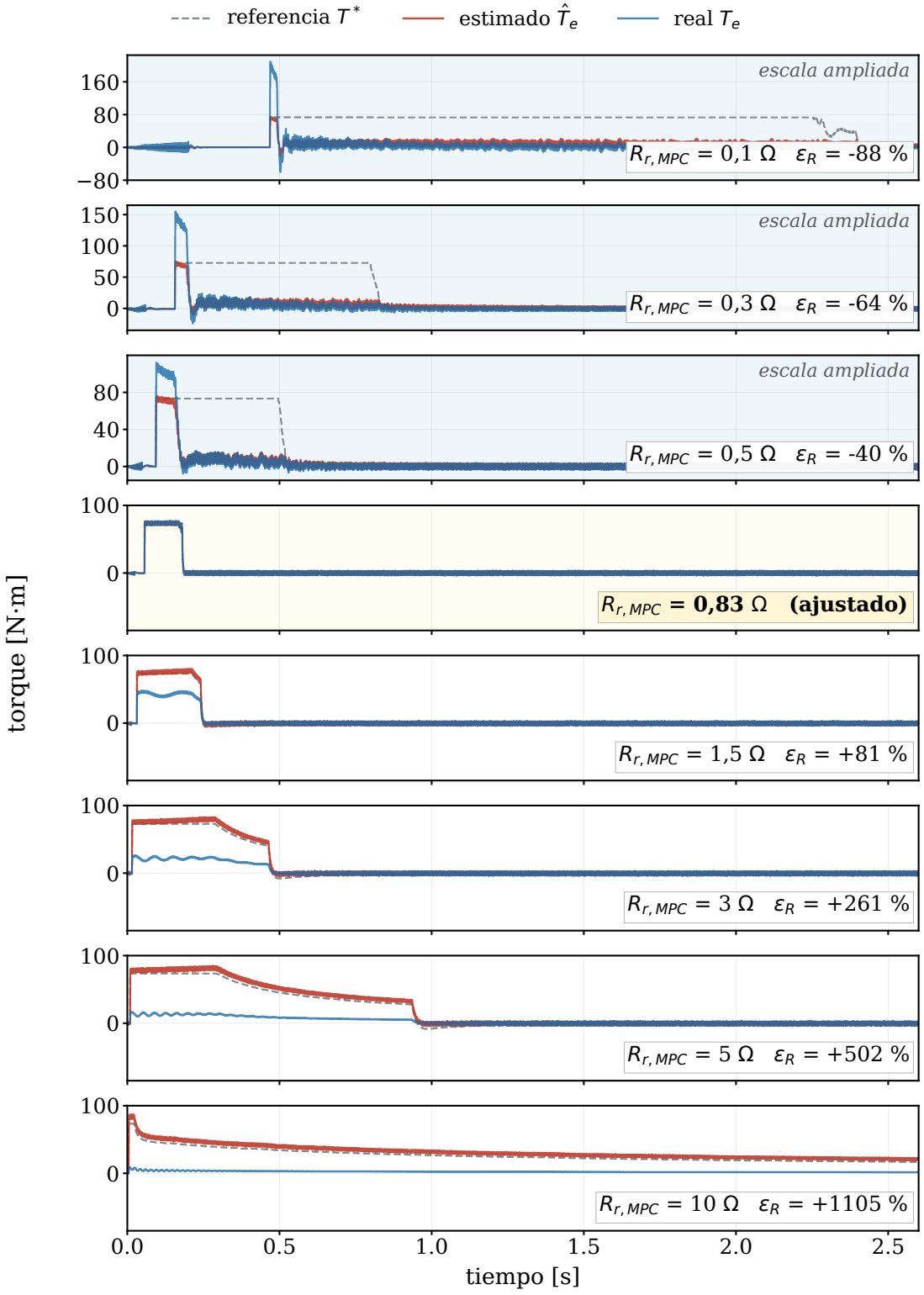


Figura 4.4: Torque de referencia T^* entregado por el PI de velocidad, torque estimado por el controlador y torque electromagnético real, durante el arranque en vacío.

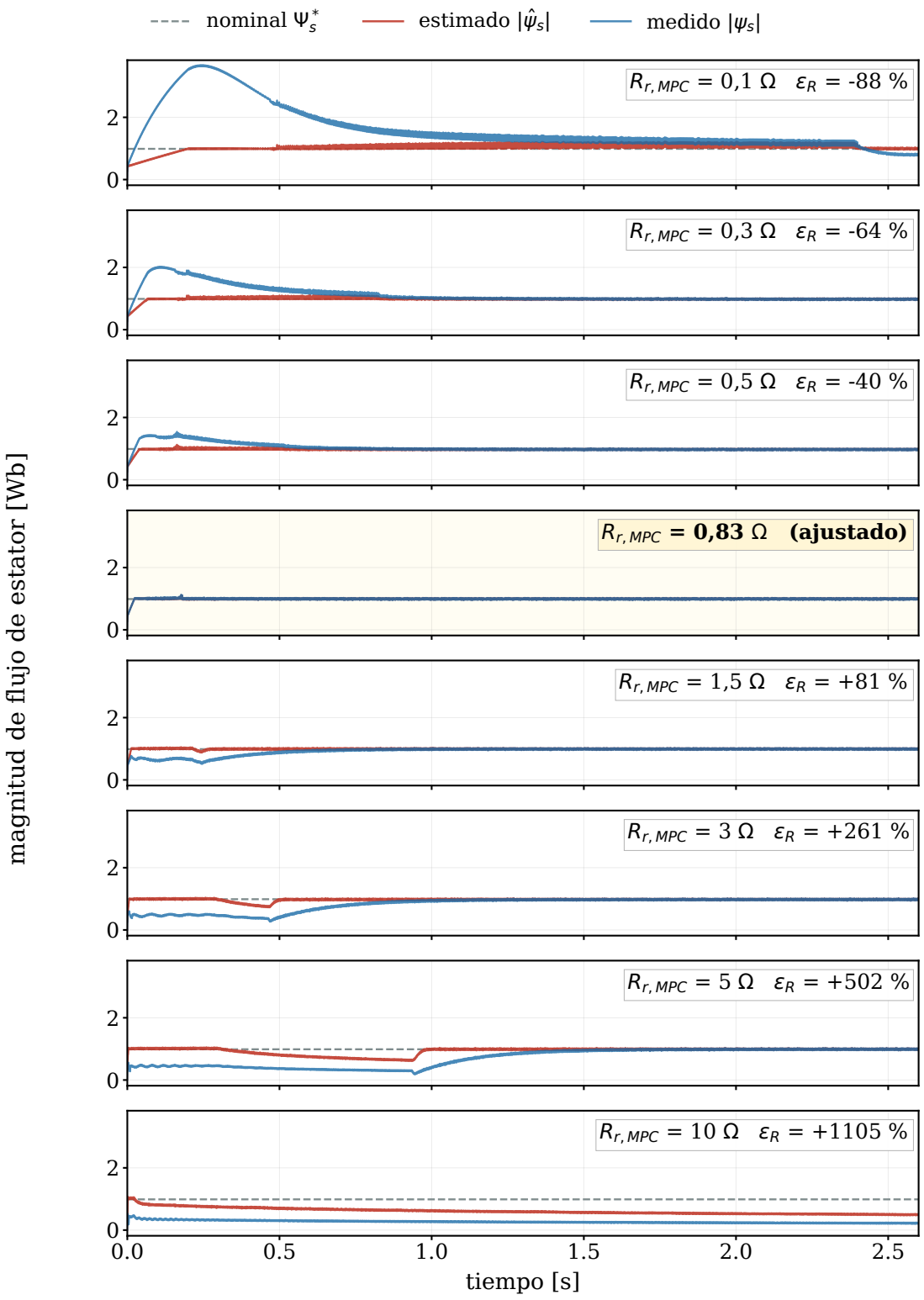


Figura 4.5: Magnitud del flujo de estator: referencia nominal, valor estimado por el controlador y valor real de la máquina, durante el arranque en vacío.

4.2.1.b. Arranque con carga de $0{,}5\,T_{nom}$

La segunda maniobra repite exactamente la escalera de la Tabla 4.2 sobre la misma máquina, cambiando una sola condición: la cinta parte cargada con material equivalente a $0{,}5\,T_{nom} = 18{,}24$ N·m, presente desde el reposo. Todo lo demás se mantiene, de modo que las diferencias respecto de la Subsección 4.2.1.a son atribuibles únicamente a la exigencia de torque.

Tabla 4.3: Arranque con carga de $0{,}5\,T_{nom}$: velocidad final (promedio de los últimos 200 ms) y error de regulación respecto de la referencia de 1440 rpm, a los ocho casos del barrido. El error del caso ajustado es la caída propia del controlador de velocidad (Sección 3.3).

$R_{r,\mathrm{MPC}}$ [Ω]	ε_R [%]	n_{fin} [rpm]	error [%]	t_{asent} [s]	Régimen
0,10	−88	424	70,6	—	degradado
0,30	−64	543	62,3	—	degradado
0,50	−40	1234	14,3	—	degradado
0,83	0	**1434**	**0,4**	**0,222**	**regula**
1,50	+81	1433	0,5	0,407	regula
3,00	+261	1091	24,2	—	degradado
5,00	+502	13	99,1	—	**rotor trancado**
10,00	+1105	4	99,7	—	**rotor trancado**

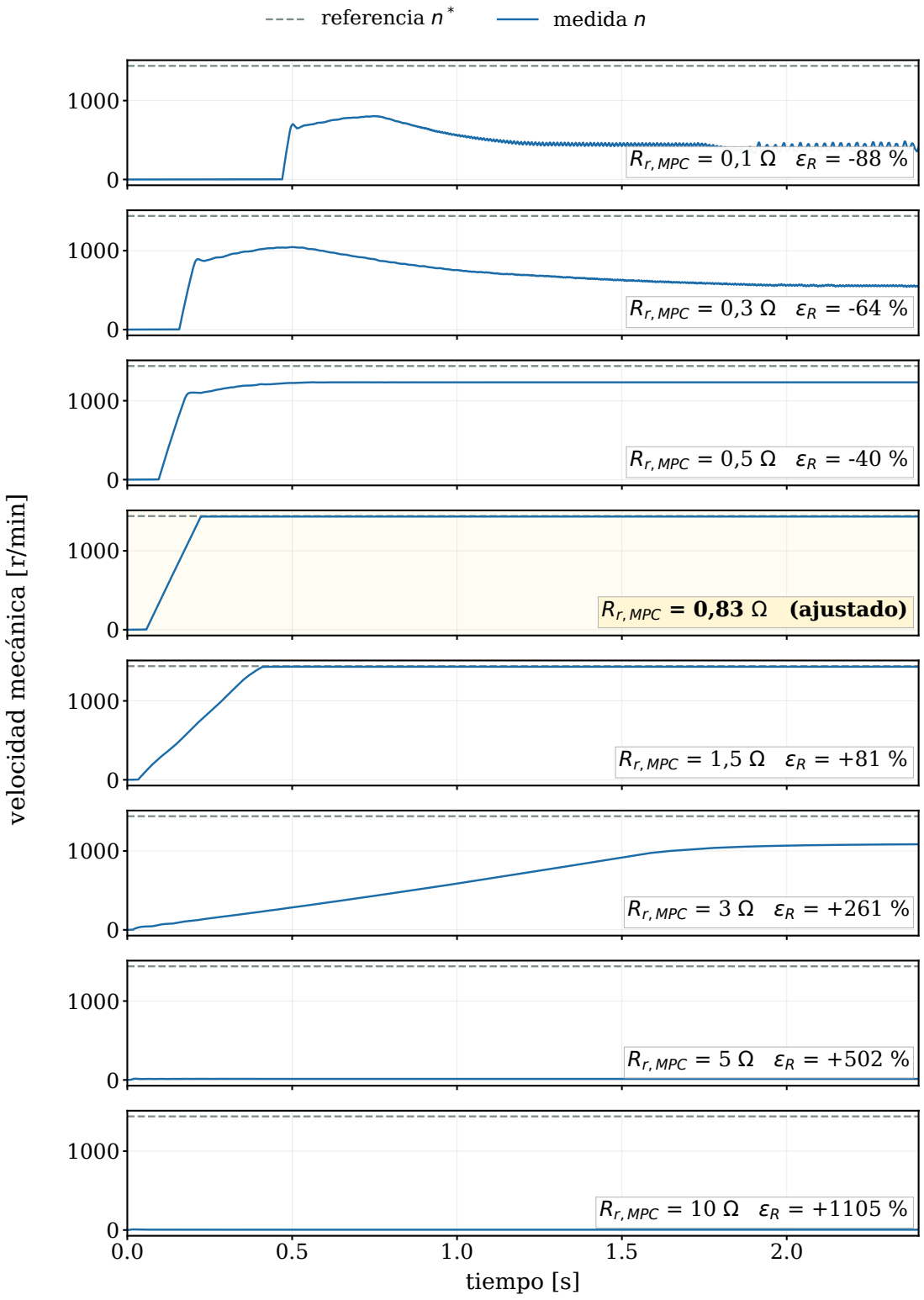


Figura 4.6: Velocidad mecánica frente a su referencia durante el arranque con $0{,}5\,T_{nom}$.

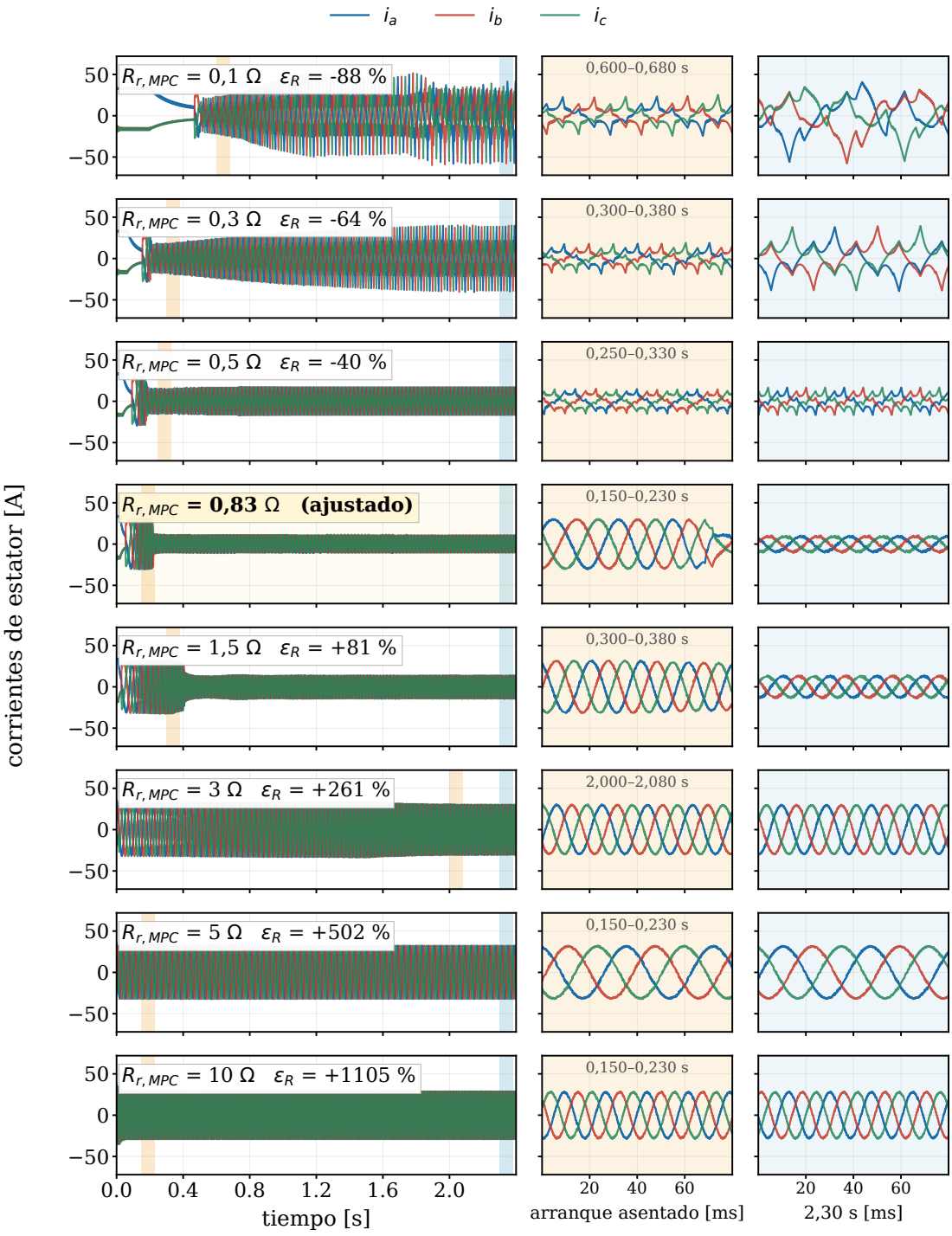


Figura 4.7: Corrientes de estator durante el arranque con $0{,}5\,T_{nom}$.

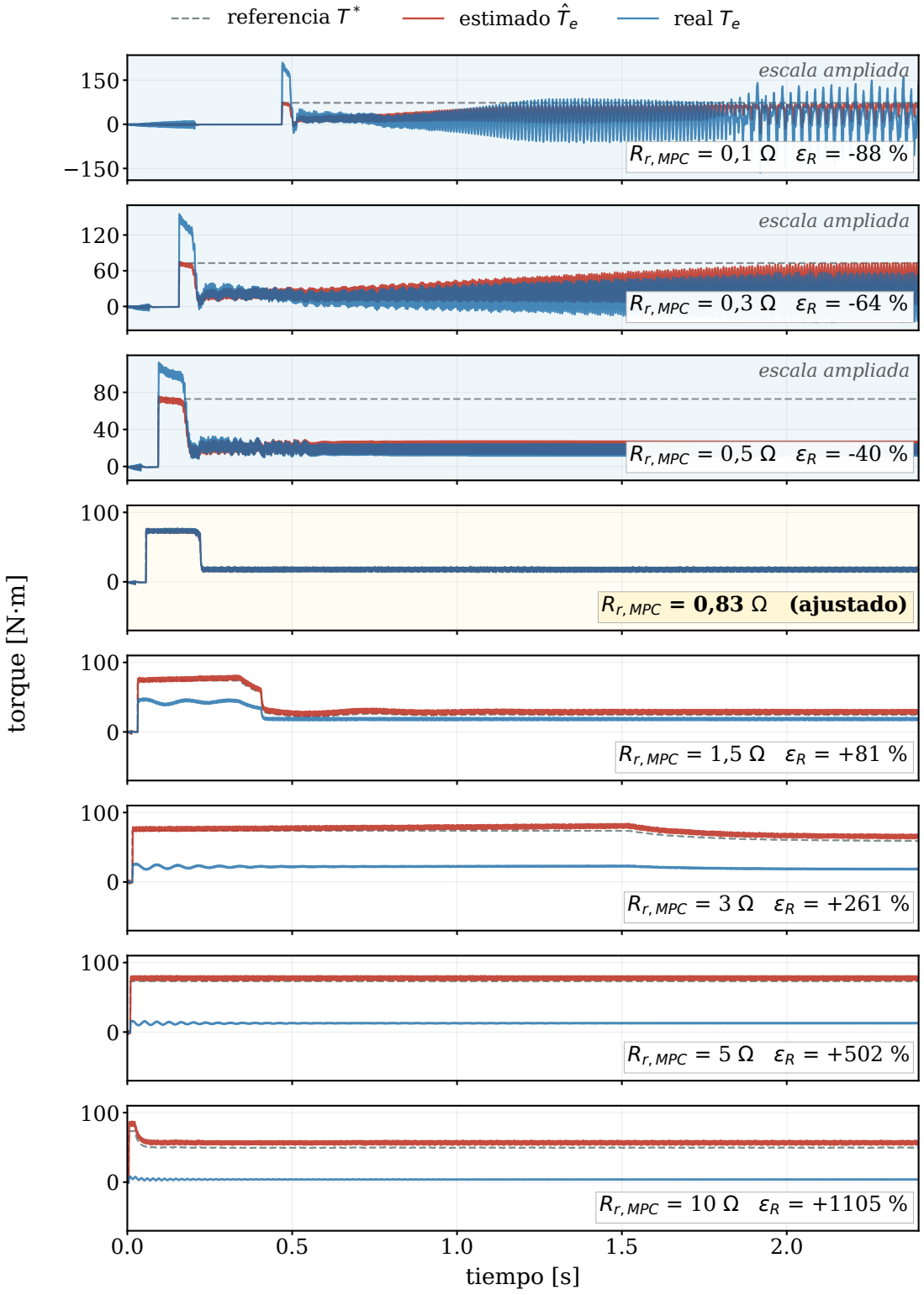


Figura 4.8: Torque de referencia, estimado y real durante el arranque con $0{,}5\,T_{nom}$. Los tres casos de subestimación llevan escala propia, señalada en el panel; los cinco restantes comparten escala.

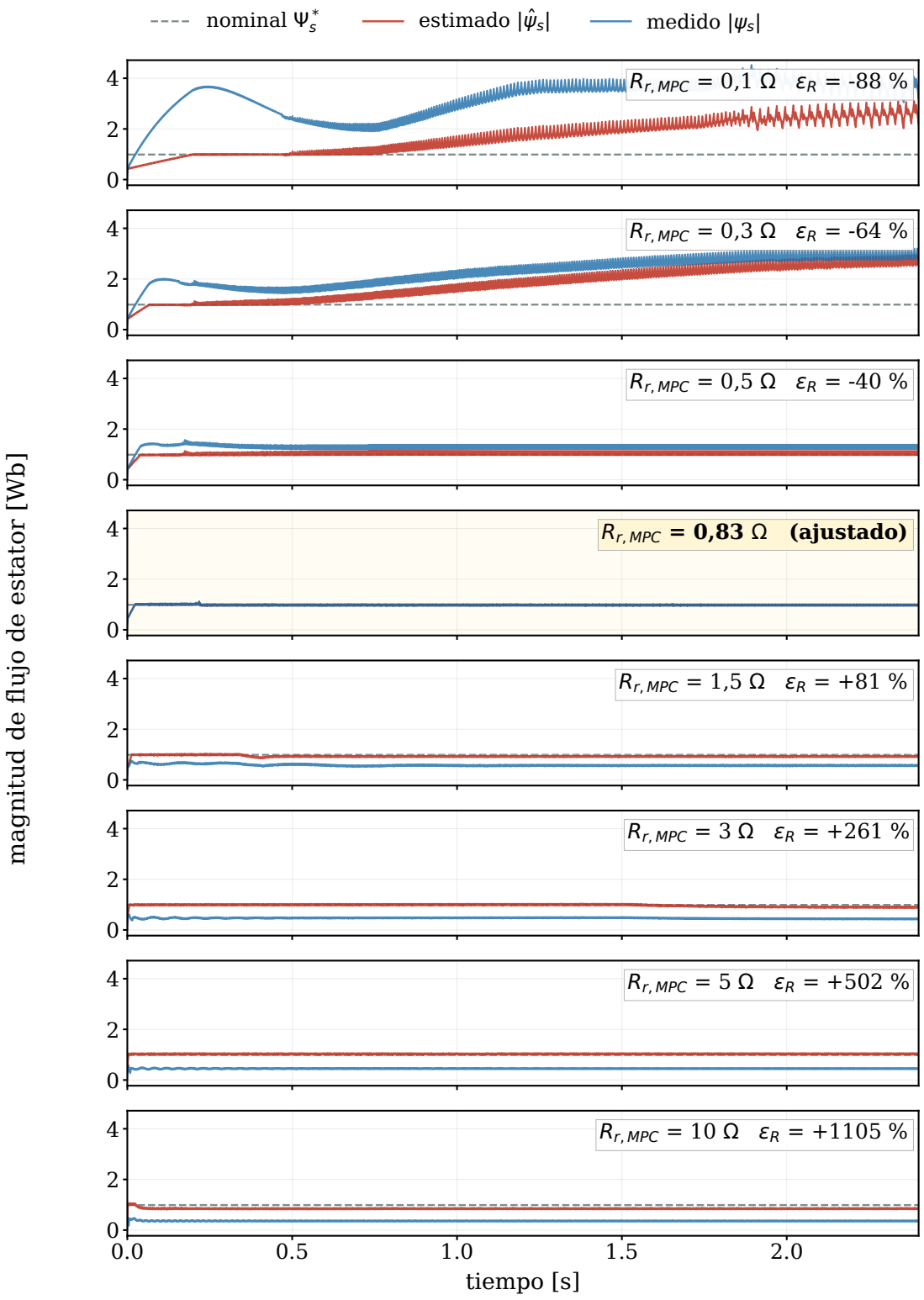


Figura 4.9: Magnitud del flujo de estator durante el arranque con $0{,}5\,T_{nom}$.

4.2.1.c. Escalón de carga sobre la máquina en marcha

Las dos maniobras anteriores solicitan torque desde el reposo o no lo solicitan en absoluto. La tercera separa ambas cosas: la máquina arranca *en vacío*, se deja asentar, y solo entonces se aplica un escalón de carga de $0{,}75\,T_{nom} = 27{,}35$ N·m. Así se distingue la capacidad de *arrancar con carga* desde el reposo o *aguantarla* ya en movimiento.

Tabla 4.4: Respuesta al escalón de $0{,}75\,T_{nom}$ aplicado a los 1,6 s sobre la máquina en estado estacionario: velocidad previa, hundimiento máximo en la ventana 1,6–2,1 s, velocidad final (promedio de los últimos 200 ms de los 4,1 s de rutina) y error de regulación resultante.

$R_{r,\mathrm{MPC}}$ [Ω]	ε_R [%]	n_{prev} [rpm]	$n_{\mathrm{mín}}$ [rpm]	caída [%]	error final [%]	Resultado
0,10	−88	1303	413	68,3	64,3	degradado[†]
0,30	−64	1440	659	54,3	71,3	degradado
0,50	−40	1440	1142	20,7	21,6	degradado
0,83	0	1440	**1432**	**0,6**	**0,6**	**recupera**
1,50	+81	1440	1359	5,6	0,9	recupera
3,00	+261	1440	805	44,1	98,9	**colapsa**
5,00	+502	1440	178	87,6	99,5	**colapsa**
10,00	+1105	757[‡]	3	99,7	99,8	**colapsa**

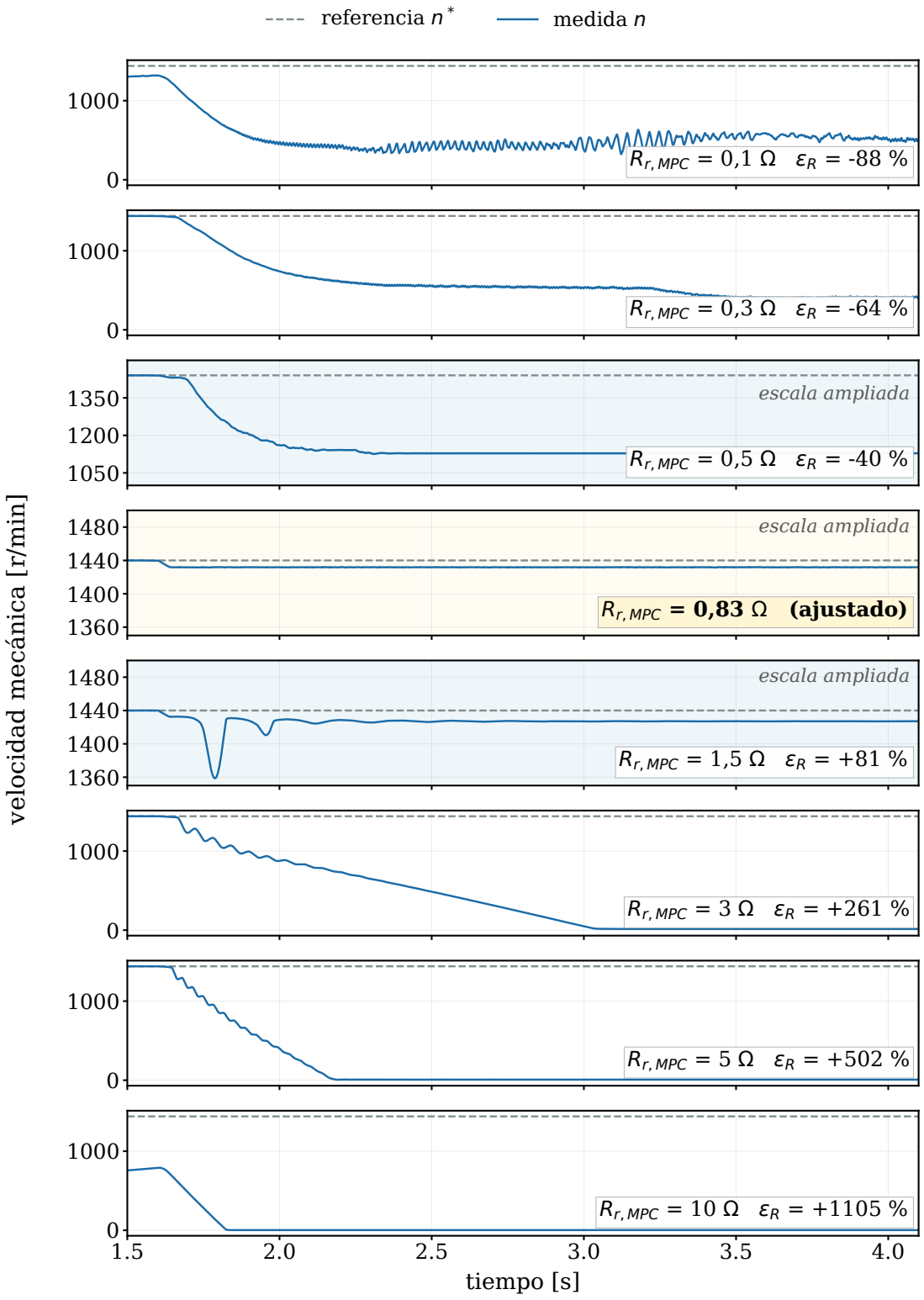


Figura 4.10: Velocidad mecánica ante el escalón de $0{,}75\,T_{nom}$ aplicado a los 1,6 s.

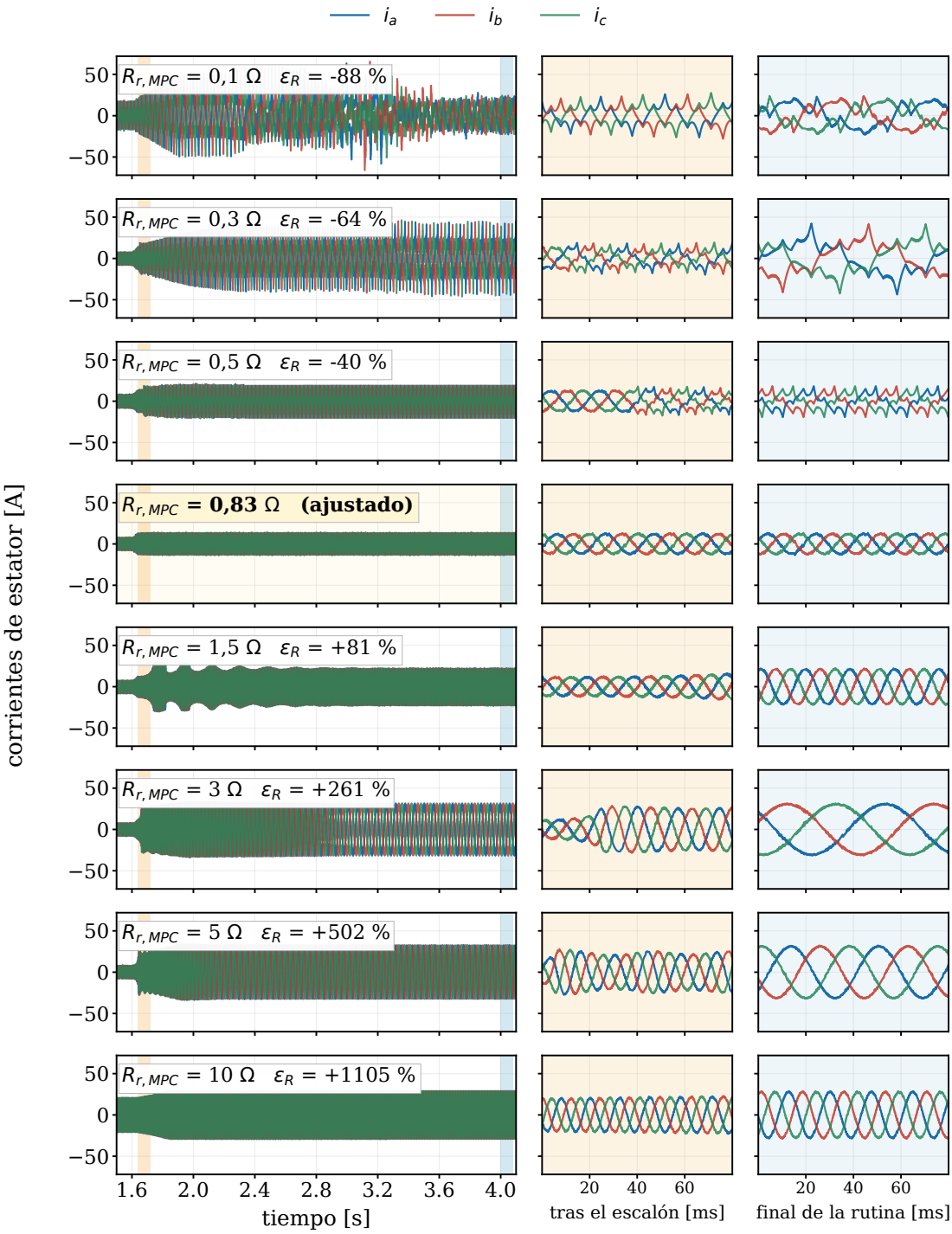


Figura 4.11: Corrientes de estator ante el escalón de $0{,}75\,T_{nom}$, desde 1,5 s.

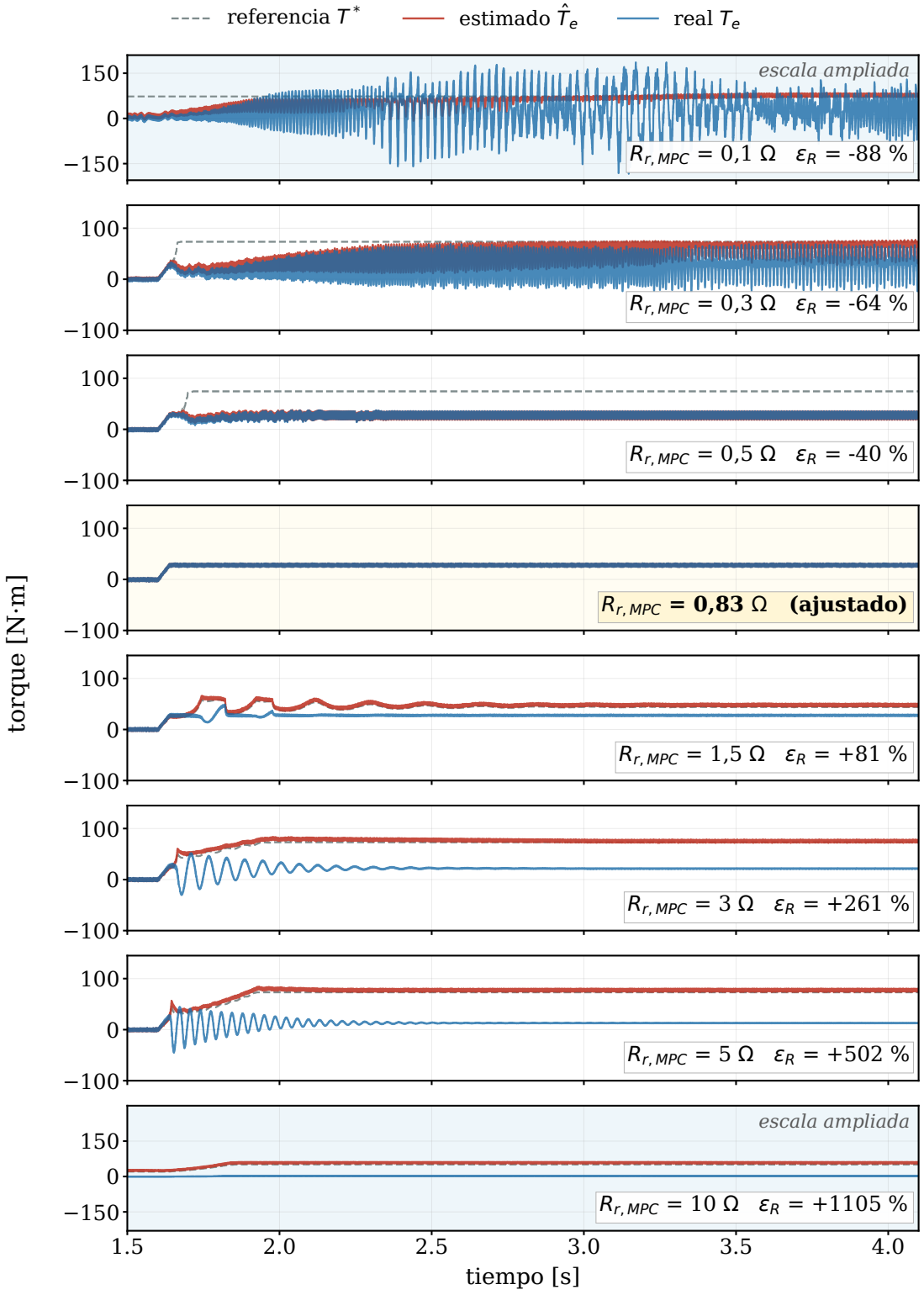


Figura 4.12: Torque de referencia, estimado y real ante el escalón de 0,75 T_{nom}, desde 1,5 s.

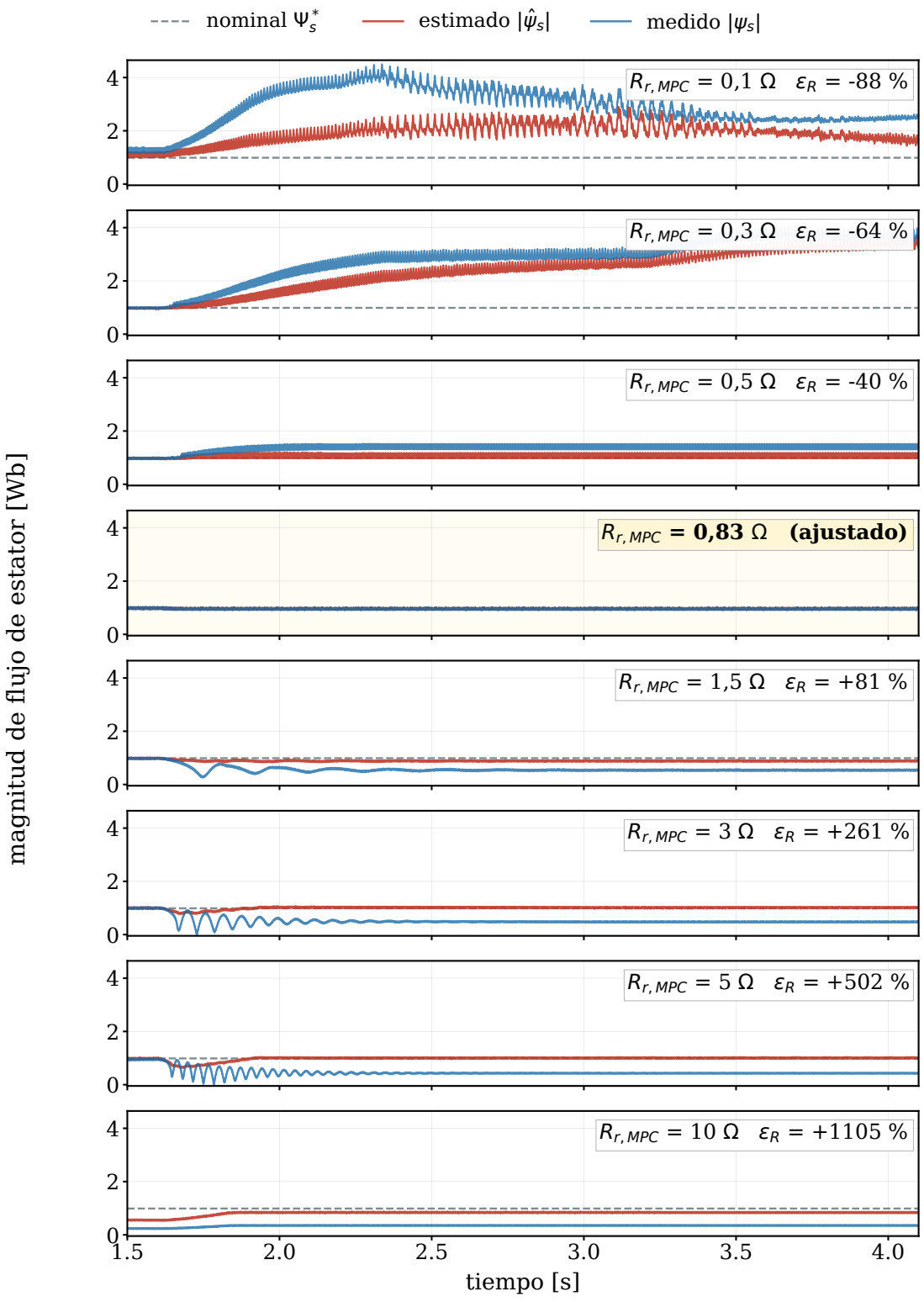


Figura 4.13: Magnitud del flujo de estator ante el escalón de 0,75 T_{nom}, desde 1,5 s.

4.2.2. Barrido con carga de bomba

Este barrido aplica la misma escalera de la Tabla 4.2 sobre el segundo tipo de carga de la plataforma, la **bomba centrífuga**, cuyo torque sigue la ley de afinidad $T_L = K\,\omega|\omega|$ con $K = T_{nom}/\omega_{nom}^2$ (Subsección 4.1.2). Se trata de una carga de torque *variable con la velocidad*, a diferencia del torque constante que opone el material sobre la cinta.

Tabla 4.5: Arranque con carga de bomba: punto de equilibrio alcanzado por cada caso del barrido. El torque de carga en el equilibrio se contrasta con el valor teórico $T_{nom}(n/n_{nom})^2$. El error del caso ajustado es la caída propia del controlador de velocidad (Sección 3.3).

$R_{r,\mathrm{MPC}}$ [Ω]	ε_R [%]	n_{eq} [rpm]	error [%]	T_L equilibrio [N·m]	Resultado
0,10	−88	536	62,8	5,06	degradado
0,30	−64	738	48,7	9,59	degradado
0,50	−40	1171	18,7	24,10	degradado
0,83	0	**1429**	**0,8**	**35,93**	**regula**
1,50	+81	1383	3,9	33,66	casi regula
3,00	+261	1047	27,3	19,28	degradado
5,00	+502	777	46,0	10,63	degradado
10,00	+1105	417	71,1	3,05	degradado

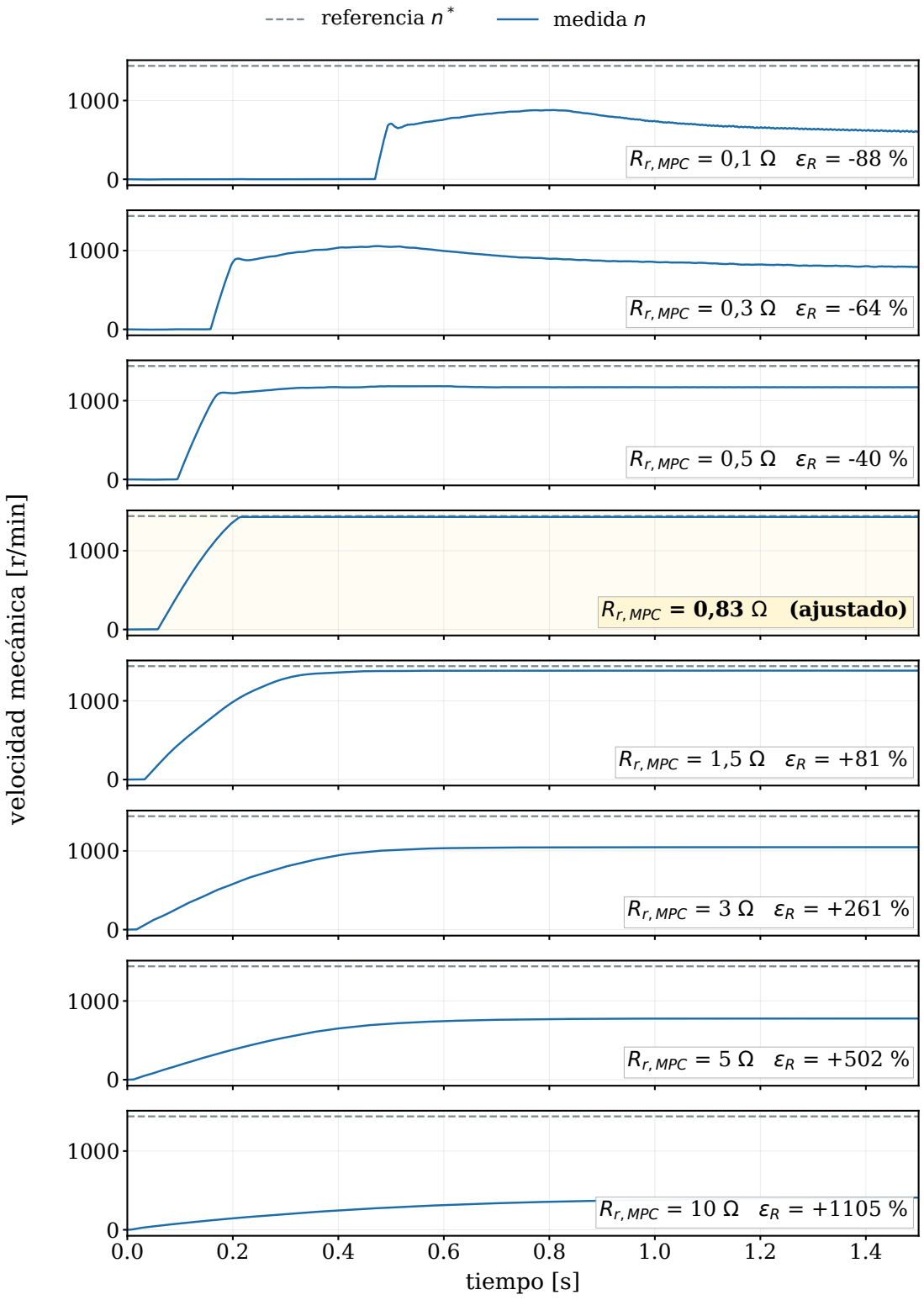


Figura 4.14: Velocidad mecánica con carga de bomba.

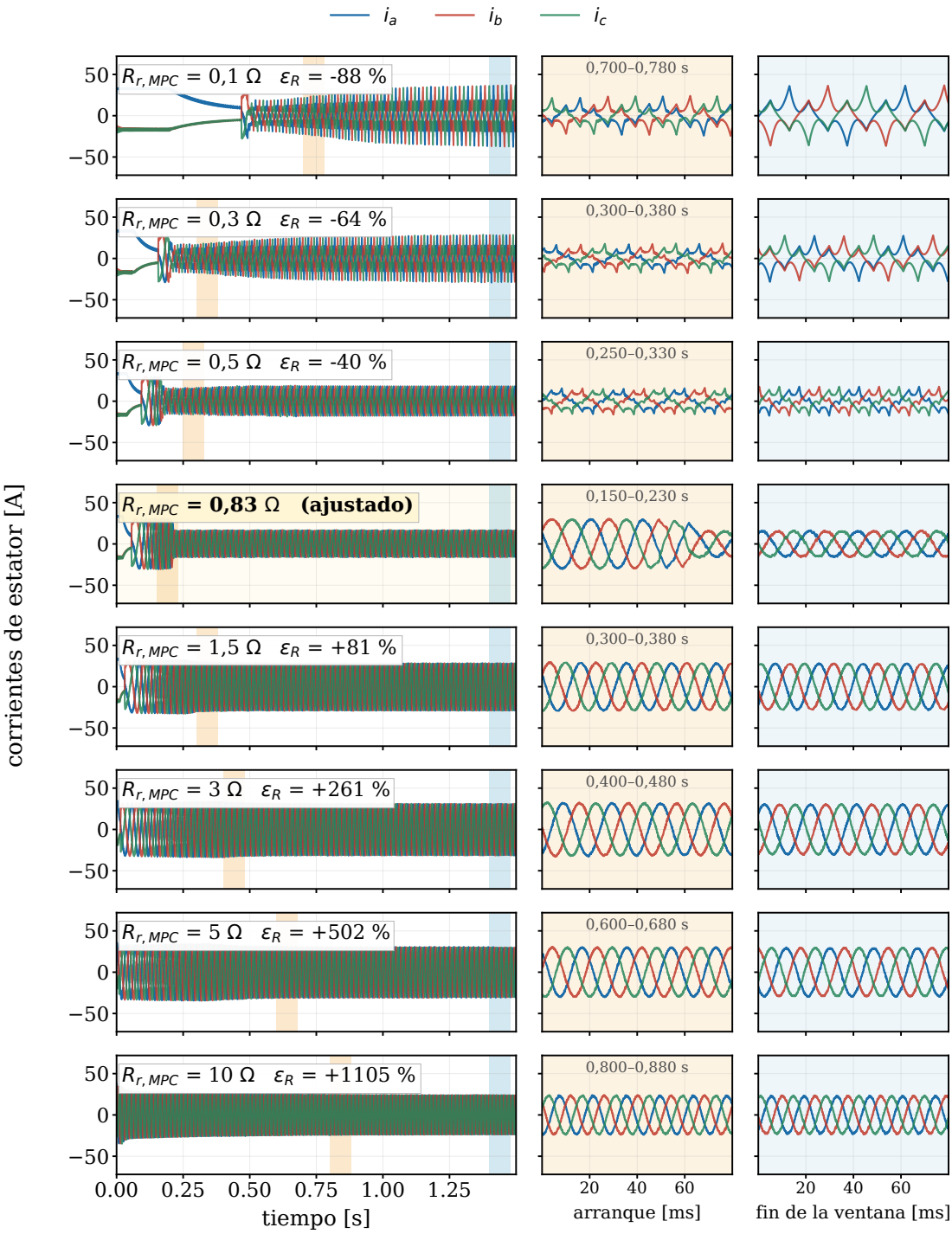


Figura 4.15: Corrientes de estator con carga de bomba.

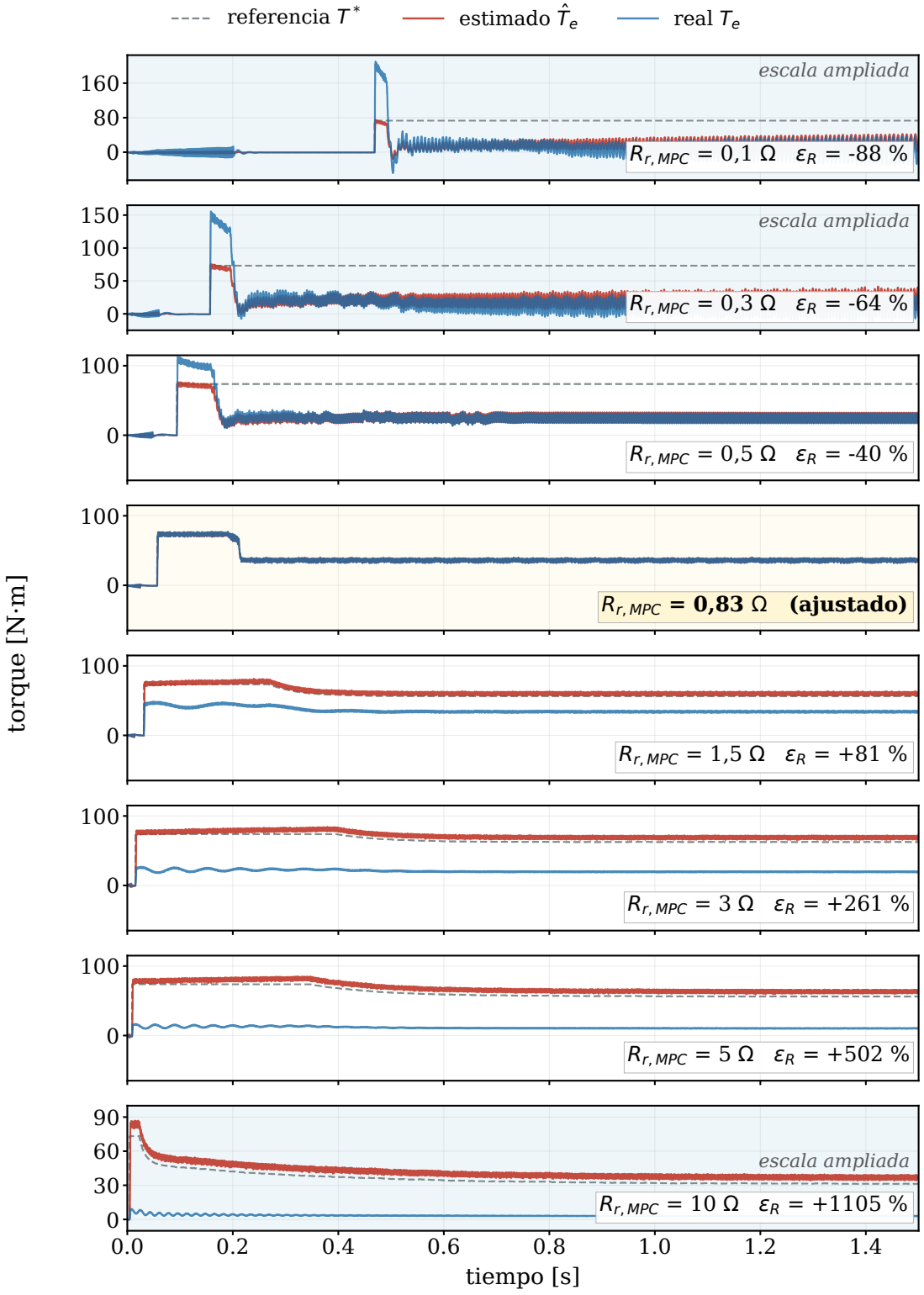


Figura 4.16: Torque de referencia, estimado y real con carga de bomba.

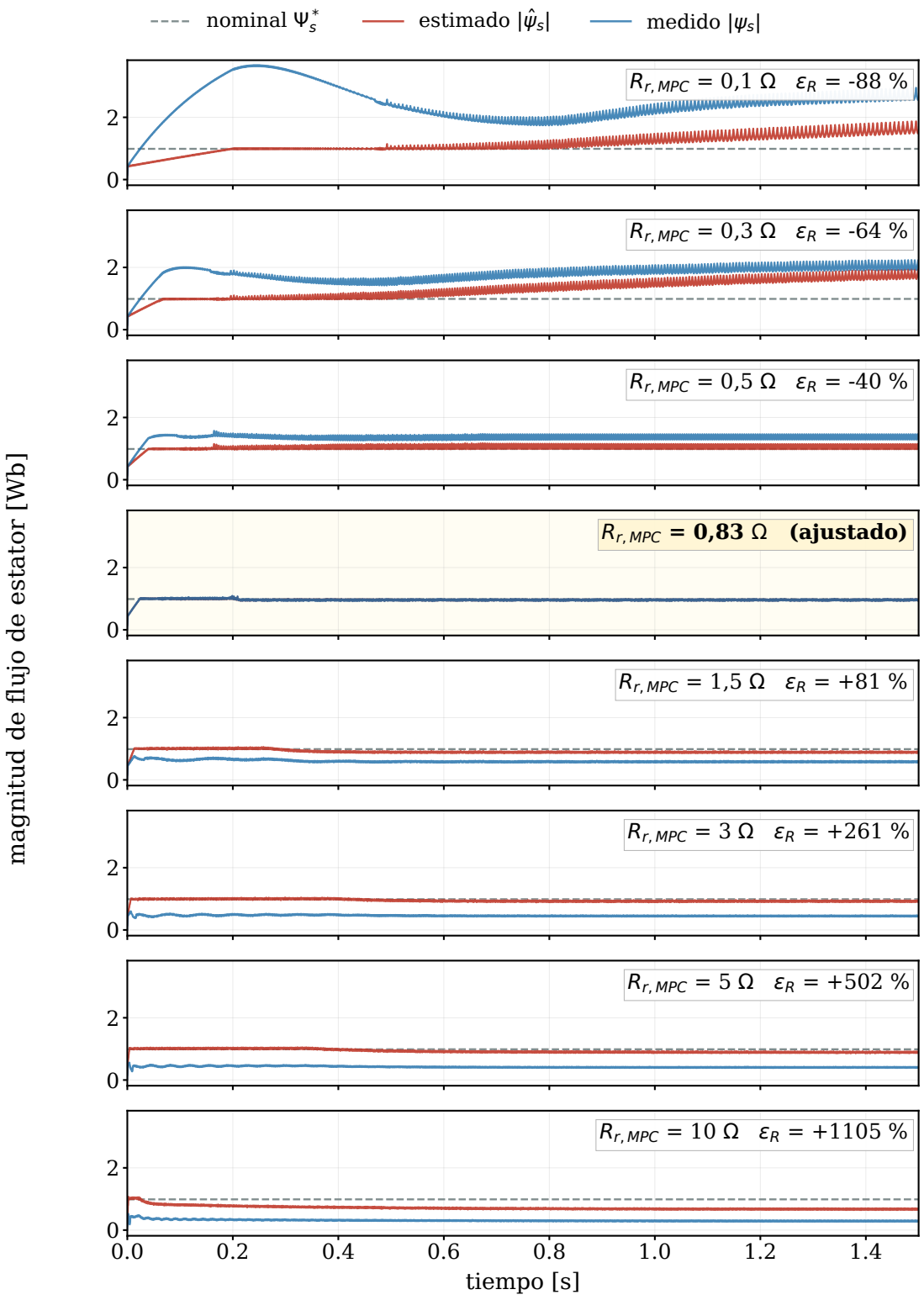


Figura 4.17: Magnitud del flujo de estator con carga de bomba.

4.2.3. Síntesis de comportamientos frente al desajuste

Esta subsección recoge lo que muestra cada uno de los cuatro barridos anteriores. Y la interpretación conjunta se desarrolla en la Sección 4.4.

4.2.3.a. El caso ajustado como patrón de comparación

Se describe con precisión el comportamiento del caso $R_{r,\mathrm{MPC}} = R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$, porque es la referencia contra la cual todo lo demás se juzga. Su rasgo distintivo es la *fidelidad*: las cuatro magnitudes se comportan como **predice** la teoría del Capítulo 3.

- **Velocidad:** En las maniobras de arranque la subida es de **aceleración prácticamente constante**. Ajustando una recta al tramo entre el 10 y el 90 % de la referencia se obtiene un coeficiente de determinación de $R^2 = 1{,}0000$ en vacío, 1,0000 con carga y 0,9958 con bomba: la velocidad es, a efectos prácticos, una rampa. Es lo esperable de un lazo que dispone del torque máximo durante todo el transitorio y lo aplica sobre una inercia constante. Frente a carga, el error entre referencia y velocidad medida es **prácticamente nulo**: 0,4 % en el arranque cargado, 0,6 % tras el escalón y 0,8 % con bomba.

- **Corrientes de fase:** Al inicio de las maniobras de arranque la magnitud es **alta**, acotada a 33,4 A mientras la máquina se magnetiza (Subsección 3.4.4), y de unos 30 A en el transitorio de aceleración que sigue, impuesta allí por la saturación de la referencia de torque. Y la forma de onda se mantiene **sinusoidal y equilibrada** durante todo el transitorio. Alcanzada la referencia de velocidad sin carga la magnitud desciende a la corriente de magnetización, unos 7,4 A, conservando la forma. Frente a carga la magnitud sube en proporción al torque demandado: 10,3 A con media carga, 13,0 A tras el escalón y 15,9 A con bomba, pero siempre con bajo contenido armónico: la distorsión medida se mantiene entre 0,10 y 0,27.

- **Torque estimado:** La estimación **coincide con el torque real**: la razón entre ambos es 1,00 en las cuatro maniobras, tanto en el transitorio como en régimen, y sus *ripple* son idénticos hasta la segunda cifra: 1,58 frente a 1,58 N·m con carga; 1,49 frente a 1,49 tras el escalón. En el arranque desde el reposo el torque **asciende como escalón**, *antes* de que la máquina empiece a girar, lo que ocurre a los 60,3 ms y permanece allí durante toda la aceleración, entre 72,85 y 73,58 N·m, un recorrido de apenas el 0,7 % en torno a la saturación de la referencia ($T^* = 72{,}95$ N·m). Es precisamente ese torque **constante** el que produce la rampa de velocidad descrita más arriba: torque constante sobre inercia constante da aceleración constante.

- **Magnitud de flujo de estator estimada:** La estimación es **casi idéntica a la real**, coinciden hasta la tercera cifra en las cuatro maniobras y ambas siguen fielmente a la referencia. Esa referencia, en cambio, ya no es exactamente la nominal: la ley de flujo dependiente del punto de operación la sitúa entre el 98,8 % del nominal en vacío y el 95,7 % con bomba, retirando el flujo justo que el deslizamiento de cada maniobra exige para que el inversor no salga de su zona lineal.

4.2.3.b. Arranque en vacío

Es la maniobra menos exigente del conjunto, porque en ella *ninguno* de los ocho casos llega a detenerse: las ocho corridas giran y siete de ellas alcanzan la referencia. Excepto en el caso de 10 Ω sobreestimado, que se estabiliza en 1171 rpm sin llegar a estacionarse cerca de la referencia; su caso se examina más abajo, porque el mecanismo que lo degrada no es el mismo que el de los demás.

Tabla 4.6: Duración de la pre-magnetización, tiempo de asentamiento de la velocidad y torque electromagnético real de *peak*, para los ocho casos del barrido con arranque en vacío.

$R_{r,\mathrm{MPC}}$ [Ω]	0,10	0,30	0,50	**0,83**	1,50	3,00	5,00	10,00
t_{premag} [ms]	469	157	94	**57**	32	16	10	5
t_{asent} [s]	2,291	0,811	0,506	**0,181**	0,242	0,463	0,928	—
$\lvert\hat{T}_e\rvert_{peak}$ [N·m]	208,3	154,3	111,4	**76,6**	48,1	26,6	16,8	9,3

- **Velocidad.** La subestimación destruye la *forma* de la rampa; la sobreestimación, su *pendiente*. Ambos extremos fallan, y por razones distintas: uno porque acelera de forma errática, el otro porque no dispone de torque con que acelerar.

  1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω la aceleración deja de ser constante: tras la espera de magnetización la máquina salta de golpe a unas 720 rpm y a partir de ahí su aceleración se desploma de 3850 a menos de 200 rpm/s. Además la **pre-magnetización** ya no es un tiempo fijo, ya que se espera a que la corriente de magnetización de rotor **estimada** alcance su valor nominal, y esa estimación evoluciona con $\tau_r = L_r/R_{r,\mathrm{MPC}}$: creer la resistencia diez veces menor equivale a creer la constante de rotor diez veces mayor, y la espera se alarga de 57 a 469 ms. Sin contar el tiempo de **pre-magnetización**, el arranque con 0,10 Ω sigue tardando 1,82 s, de modo que el retardo de arranque proviene de la sobre-magnetización que la propia subestimación provoca.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω, en cambio, la rampa sigue siendo notablemente recta pero tan tendida que ni siquiera completa el recorrido: se detiene en 1171 rpm. Hacia sobreestimación, el torque de *peak* es monótono en $R_{r,\mathrm{MPC}}$ y recorre de 208,3 a 9,3 N·m, cruzando el ajustado en 76,6: la máquina dispone de menos torque para acelerar la misma inercia, y el lazo no reacciona tarde sino que no tiene con qué.

- **Corrientes de fase.** La subestimación produce una magnitud reducida y cada vez más distorsionada; la sobreestimación, una magnitud grande, sostenida y limpia.

  1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω, una vez terminada la magnetización, la corriente cae de 26 a unos 15–17 A y en ese entorno permanece durante todo el transitorio, mientras la distorsión crece de 0,45 a 0,84.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω la corriente decae lentamente de 23 a 19 A a lo largo de toda la maniobra con una distorsión de apenas 0,06–0,08: una fundamental impecable que **no está produciendo torque útil**.

- **Torque estimado.** La subestimación oculta el *ripple*; la sobreestimación infla la magnitud.

    1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω el torque real oscila con una desviación típica de 53,0 N·m y máxima de 208,3, mientras la estimación apenas registra 18,9 N·m de *ripple*: el controlador no percibe el torque real que está aplicando.
    2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω ocurre lo contrario: la estimación alcanza 86,5 N·m cuando el torque real no supera los 9,3, casi diez veces menos, y después, el controlador sigue estimando 26,3 N·m contra 2,2 reales, lo que es coherente con que ese caso nunca alcance la referencia.

- **Magnitud de flujo de estator estimada.** La estimación permanece pegada a su valor nominal en los dos sentidos mientras el flujo real se aparta en direcciones opuestas.

    1. **Casos subestimados ($\Delta R_r < 0$).** Durante el arranque con 0,10 Ω la estimación marca 1,01 veces el nominal y el flujo real llega a 3,70: se *subestima* por un factor de casi cuatro.
    2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω la estimación marca 0,80 y el real 0,36: se *sobreestima* por más del doble. Y en estado estacionario la estimación queda en 0,57 y el flujo real en 0,26, de modo que ese caso opera permanentemente sub-magnetizado sin que el controlador lo registre.

De estas dos últimas observaciones se sigue el resultado más relevante: **las magnitudes estimadas no delatan el problema**. El controlador evalúa su función de costo sobre el torque y el flujo que él mismo estima, y en ambos el desajuste se manifiesta como un valor *razonable*: cercano a la referencia pero que no coincide con la realidad. Ya que la discrepancia solo es visible al contrastar la estimación con la medición **real**, contraste que **el controlador no puede realizar**.

4.2.3.c. Arranque con carga de $0{,}5\,T_{nom}$

La misma escalera que en vacío completaba la maniobra ocho veces de ocho deja aquí seis casos fuera de especificación: solo regulan el ajustado y el de +81 %.

- **Velocidad.** La rampa lineal se pierde en los dos sentidos, y mucho más marcadamente que en vacío.

    1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω el ajuste lineal cae a $R^2 = 0{,}540$, frente al 1,000 del ajustado: la máquina arranca, alcanza un máximo y después *decae*. El caso de 0,30 Ω ilustra ese deterioro progresivo con claridad: 759 rpm a 1 s, 566 a 2 s y 536 al término de los 3 s simulados, sin alcanzar valor estacionario alguno.
    2. **Casos sobrestimados ($\Delta R_r > 0$).** El deterioro es gradual hasta que el fallo llega de golpe. Con +261 % la máquina supera la carga y acelera, pero se estaciona muy por debajo de la referencia a 1067 rpm en 2 s y 1092 rpm a los 3 s;

y con +502 % ya no supera la carga y queda estancado con velocidad casi nula. Con la escalera empleada el rango de rotor trancado queda acotado entre 3,0 y 5,0 Ω.

- **Corrientes de fase.** El contraste de **forma** es el mismo que en vacío; pero la **magnitud** se invierte.

  1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω la corriente cae tras el *peak* inicial hasta 18,6 A y después *crece de forma sostenida* 34, 45, 46 y 52 A en los sucesivos medios segundos: es una magnitud creciente y sucia.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω la corriente se clava en 28,4 A desde el primer instante y ahí permanece los tres segundos, con una distorsión de 0,07: la máquina consume más del doble de la corriente nominal sin producir movimiento.

- **Torque estimado.** Se repite el patrón del arranque en vacío, con mayor severidad.

  1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω el torque real oscila con desviación típica de 65,6 N·m y *peak* de 208,4, mientras la estimación registra 9,0: el *ripple* real es *siete veces* el que el controlador percibe.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω la estimación se sostiene en 56,8 N·m contra un torque real de 4,9 —un factor de 11,7 invariable durante toda la corrida—, de modo que la magnitud sobreestimada no es un transitorio sino el estado permanente del lazo.

- **Magnitud de flujo de estator estimada.** Es donde esta maniobra aporta lo que el arranque en vacío no mostraba.

  1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω la estimación *deja de estar anclada* a la referencia y acompaña la deriva: en régimen marca 2,76 veces el nominal frente a un flujo real de 3,59. Sigue subestimando, pero ya no por permanecer en su valor nominal sino por quedarse corta en una divergencia que sí detecta; el lazo pierde el control de la magnitud de flujo, no solo su medida.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω, en cambio, la estimación se mantiene en 0,85 mientras el real cae a 0,37: una sobreestimación estable de más del doble, con la máquina detenida y sub-magnetizada.

De todo lo anterior se desprende que **los dos modos de falla son cualitativamente distintos y exigen respuestas distintas**. La sobreestimación produce, en su forma extrema, una falla *instantánea y estática*: la máquina nunca arranca y su estado no evoluciona, mientras que la subestimación produce una falla *lenta y dinámica*, en la que la máquina arranca, opera un tiempo apreciable y se deteriora después.

4.2.3.d. Escalón de carga sobre la máquina en marcha

Esta maniobra tiene una particularidad que la hace especialmente informativa: seis de los ocho casos llegan al escalón en el mismo estado, velocidad nominal alcanzada y flujo

real entre el 94 y el 100 % del nominal, como confirma la Figura 4.13 en el tramo previo, de modo que para ellos *todo* lo que ocurre después es atribuible a la carga y al desajuste, sin contaminación del arranque.

- **Velocidad.** Es la magnitud que separa los tres desenlaces.

  1. **Casos subestimados ($\Delta R_r < 0$).** La no linealidad se manifiesta como una consecuencia *posterior* al deterioro: con -40 % la máquina supera el escalón con una caída del 20,7 % y se estabiliza 21,6 % por debajo de la referencia, pero con -64 y -88 % el deterioro no se estabiliza dentro de la ventana simulada.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Desde $+261$ % la velocidad decae hasta el reposo con una desaceleración notablemente *recta*, lo que corresponde a un torque neto constante y negativo.

- **Corrientes de fase.** Aquí los dos sentidos se distinguen por la forma y no por la magnitud, porque en ambos extremos la corriente es grande.

  1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω sube de 30 a 46 A en los primeros 400 ms tras el escalón, alcanza 48 A con una distorsión que llega a 1,03 y solo decae a 22 A al final, cuando la máquina ya opera muy por debajo de la referencia.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω el escalón encuentra a la máquina ya degradada: la corriente sube de 25 a 29 A y ahí permanece, con una distorsión de 0,07 tras un breve transitorio de 0,21. Es prácticamente una sinusoide pura sostenida sobre una máquina que se está deteniendo.

- **Torque estimado.** Esta maniobra presenta el único caso del capítulo en que la sobreestimación además *oscila*.

  1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω el contraste es de magnitud y no de *ripple*: los dos *ripple* coinciden en 10,1 N·m, mientras la estimación llega a 50,2 frente a los 38,0 que la máquina entrega.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω, en la ventana inmediatamente posterior al escalón la estimación alcanza 54,2 N·m frente a un torque real de 4,7, y su *ripple* es de 7,8 contra apenas 0,7 del real: sobreestima a la vez la magnitud y la fluctuación, sobre una máquina que ya casi no produce torque.

- **Magnitud de flujo de estator estimada.** Muestra el efecto aislado que buscaba esta maniobra. Antes del escalón las seis estimaciones centrales coinciden con sus flujos reales en el valor nominal; la divergencia grande aparece *exactamente* al aplicarse la carga, en ambos sentidos y de forma simultánea.

  1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω la estimación se eleva hasta 2,93 veces el nominal y se asienta en 1,71, mientras el real llega a 4,53 y termina en 2,49.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω el efecto es el opuesto y ya venía de antes: ni la estimación ni el flujo real superan el nominal en ningún instante, y ambos terminan en 0,85 y 0,37. En ese caso, por tanto, no es el escalón el que rompe la regulación de flujo, viene desde antes, en el tramo en vacío.

La diferencia entre ambos sentidos es cualitativa: la subestimación conserva capacidad de torque y opera mal, de forma indefinida; la sobreestimación se detiene. En los casos que colapsan, la referencia de torque se satura al aplicarse la carga y permanece saturada durante toda la deceleración sin que el torque real llegue a igualar al de carga.

4.2.3.e. Carga de bomba

En esta rutina ningún caso se detiene: los ocho alcanzan un punto de operación estable, comprendido entre 417 y 1429 rpm. La razón está en la propia característica de la carga: ante un torque que decrece con la velocidad, una pérdida de torque produce una caída de velocidad que *reduce la propia carga* hasta que ambos se igualan. De modo que existe equilibrio para cualquier grado de desajuste. Esto convierte a la bomba en la maniobra que **permite observar los cuatro deterioros** *en estado estacionario.*

- **Velocidad.** La linealidad de la rampa se pierde en ambos sentidos y en los dos el equilibrio queda muy por debajo de la referencia. La degradación resultante es monótona y de magnitud comparable en ambos sentidos, sin discontinuidades dentro del barrido.

  1. **Casos subestimados ($\Delta R_r < 0$).** El ajuste lineal cae a $R^2 = 0{,}644$ con 0,10 Ω, con errores de regulación de velocidad en 18,7, 48,7 y 62,8 %.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** El ajuste lineal cae a $R^2 = 0{,}605$ con 10 Ω, con errores de regulación de velocidad en 27,3, 46,0 y 71,1 %. Es la única maniobra en que la sobreestimación degrada también la *forma* de la subida y no solo su pendiente.

- **Corrientes de fase.** Se repite el contraste similar a maniobras anteriores.

  1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω la magnitud cae tras el arranque inicial y vuelve a crecer hasta 37,7 A, con una distorsión alta que se estabiliza en 0,73.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω se mantiene en 23,3 A durante toda la corrida con una distorsión baja de 0,08.

- **Torque estimado.** Exhibe aquí la firma más destacable, precisamente por tratarse de un equilibrio genuino.

  1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω la estimación registra 21,3 N·m de *ripple* frente a 28,6 reales, y en el arranque el contraste es aún mayor: 17,8 estimados contra 50,4 reales.
  2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω la estimación se sostiene en 36,3 N·m frente a 3,1 N·m reales, un factor de 11,7 que *no se extingue*; con 3 y 5 Ω los factores son 3,5 y 5,9.

- **Magnitud de flujo de estator estimada.** Se aparta menos que bajo carga de torque constante, y de manera asimétrica.

1. **Casos subestimados ($\Delta R_r < 0$).** Con 0,10 Ω la estimación llega a 2,20 veces el nominal y el real a 2,94: la deriva existe, pero es menor que los 3,59 del arranque cargado, porque el torque que la máquina sostiene en su equilibrio es más pequeño.
2. **Casos sobrestimados ($\Delta R_r > 0$).** Con 10 Ω la estimación marca 0,67 frente a 0,30 del real: la sobreestimación persiste, aunque en términos absolutos el flujo estimado se aleja poco de su nominal, que es lo que hace a este caso el menos delatador de los cuatro.

El desajuste en esta rutina queda de forma permanente en la diferencia entre magnitudes estimadas y reales.

4.2.4. Resumen de características frente al desajuste

La Tabla 4.7 condensa las observaciones anteriores en una sola vista: las cuatro magnitudes graficadas frente a las cuatro maniobras, con cada casilla dividida según el sentido del desajuste.

Tabla 4.7: Resumen de la degradación de cada magnitud graficada frente al caso ajustado, por maniobra y por sentido del desajuste.

Magnitud	$\boldsymbol{\Delta R_r}$	**Arranque en vacío**	**Arranque con carga**	**Escalón de carga**	**Carga de bomba**
Velocidad	< 0	Rampa no lineal	Rampa no lineal; no alcanza la referencia	No lineal: se hunde y deriva	Rampa no lineal; equilibrio bajo
	> 0	Rampa lineal, pero más lenta	Arranque muy lento o rotor trancado	Colapsa hasta el reposo	Más lenta; equilibrio bajo
Corrientes *abc*	< 0	Magnitud reducida y THD creciente	Cae y luego crece; THD alto	Magnitud grande y THD alto	Cae y luego crece; THD alto
	> 0	Magnitud grande sostenida, sinusoidal	Magnitud grande sostenida, sinusoidal	Magnitud grande, sinusoidal	Magnitud grande sostenida, sinusoidal
Torque estimado	< 0	Subestima el *ripple* real	Subestima el *ripple* real	Sobreestima la magnitud; reproduce el *ripple* al entrar la carga, no al final	Subestima el *ripple* real
	> 0	Sobreestima la magnitud	Sobreestima la magnitud	Sobreestima la magnitud y oscila	Sobreestima la magnitud
Flujo estimado	< 0	Subestima durante el arranque	Subestima durante el arranque	Subestima al entrar la carga	Subestima
	> 0	Sobreestima en régimen	Sobreestima en régimen	Sobreestima al entrar la carga	Sobreestima, pero poco

Leída por filas, la tabla muestra que **cada magnitud se degrada de una forma característica que apenas depende de la maniobra**. El torque estimado y el flujo estimado son los casos más nítidos: su patrón se repite en las cuatro columnas, y es el escalón de carga el que introduce las dos únicas salvedades: bajo sobreestimación la estimación de torque además oscila, y bajo subestimación el desajuste se manifiesta ahí en la magnitud y no en el *ripple*, que la estimación sí reproduce. Las corrientes conservan el mismo contraste de *forma* en las cuatro maniobras, y varían únicamente en la magnitud, que depende del torque que cada rutina exige. La velocidad es la única magnitud cuyo comportamiento cambia de naturaleza entre maniobras, y lo hace porque el desenlace mismo cambia.

4.3. Caracterización del comportamiento bajo desajuste: distintos valores de $R_{r,\mathrm{MAS}}$

La sección anterior caracterizó el desajuste sobre la máquina de resistencia nominal. Esta lo extiende a otros valores de resistencia real, en cumplimiento del objetivo adicional declarado en la Subsección 1.2.3: comprobar si el comportamiento descrito es propio del punto $R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$ o se reproduce con independencia del valor nominal en torno al cual se produce el desajuste.

La distinción condiciona lo que sigue. Si los efectos caracterizados dependieran del valor nominal de R_r, un corrector ajustado en torno a $0{,}83\ \Omega$ no tendría por qué conservar su validez en otras configuraciones de resistencia; si el patrón se repite, en cambio, resulta razonable evaluar el estimador sobre un rango más amplio. Conviene recordar el alcance preciso de esta comprobación, ya enunciado en la Subsección 1.2.3: R_s, L_s, L_r y L_m permanecen fijos, por lo que se comparan cinco valores de resistencia de rotor dentro de un mismo modelo de máquina, no cinco máquinas distintas.

4.3.1. Qué cambia al mover $R_{r,\mathrm{MAS}}$: la curva de torque–deslizamiento

Antes de comparar las cuatro configuraciones adicionales de $R_{r,\mathrm{MAS}}$, se debe explicar qué cambia al modificar este parámetro, porque de lo contrario la degradación observada en los valores más resistivos, aun con el modelo perfectamente ajustado, podría interpretarse como un fallo del controlador.

La característica torque–deslizamiento clásica de la máquina de jaula [9] se escribe de forma compacta como

$$T_e(s) = \frac{2\,T_m}{\dfrac{s_m}{s} + \dfrac{s}{s_m}}, \tag{4.4}$$

donde $s = \omega_r/\omega_s$ es el deslizamiento, s_m el deslizamiento al que se produce el torque máximo y T_m ese torque máximo:

$$s_m = \frac{R_r}{\sigma X_r}, \qquad T_m = \frac{3}{2}\,p\,\frac{1-\sigma}{\sigma L_s}\cdot\frac{V_1^2}{\omega_s^2}, \qquad X_r = \omega_s L_r. \tag{4.5}$$

Las dos expresiones de (4.5) dicen cosas opuestas.

1. **El torque máximo no depende de R_r.** En T_m no aparece la resistencia de rotor: solo la razón V_1/ω_s, es decir, el flujo y las inductancias. Una jaula más resistiva *no* reduce el torque que la máquina puede producir.

2. **El deslizamiento al que se alcanza ese torque sí es proporcional a R_r.** La curva no se achata: se *estira* sobre el eje de deslizamiento.

Tabla 4.8: Estiramiento de la característica torque–deslizamiento con $R_{r,\mathrm{MAS}}$. La última columna es la velocidad que resultaría de alimentar la máquina a frecuencia nominal fija, como haría la red.

$R_{r,\mathrm{MAS}}$ [Ω]	s_m	s para T_{nom}	ω_r [rad/s]	n con $\omega_s = 2\pi 50$ [rpm]
0,50	0,121	0,022	7,0	1466
0,83	*0,201*	*0,037*	*11,7*	*1444*
1,50	0,364	0,067	21,1	1399
3,00	0,728	0,134	42,1	1299
5,00	1,213	0,223	70,2	1165

El caso de 5 Ω merece detenerse: su $s_m = 1{,}21$ cae *más allá del reposo*. Todo el rango de operación como motor, de $s = 0$ a $s = 1$, queda sobre el flanco ascendente de (4.4); esa máquina nunca llega a ver su propio torque máximo dentro de su zona útil.

Esto se explica con la velocidad mecánica:

$$\omega_m = \frac{(1 - s)\,\omega_s}{p}, \tag{4.6}$$

de modo que, *a la misma frecuencia de alimentación*, todo deslizamiento adicional se **descuenta** directamente de la velocidad.

4.3.2. Casos considerados

Se estudian cuatro valores adicionales de $R_{r,\mathrm{MAS}}$, escogidos entre los de la escalera maestra de la Tabla 4.2 para que las combinaciones puedan reutilizarse sin introducir valores nuevos. Cubren desde 0,5 hasta 5 Ω.

A diferencia del barrido nominal, que emplea ocho combinaciones por maniobra, cada caso adicional se explora con **cuatro desajustes más el caso ajustado**. La reducción es deliberada: se trata de verificar si un patrón se repite. La Tabla 4.9 recoge las combinaciones y el desajuste relativo que representa cada una.

Tabla 4.9: Combinaciones empleadas en la exploración sobre distintos valores de $R_{r,\mathrm{MAS}}$. La fila del valor nominal se incluye como referencia.

$R_{r,\mathrm{MAS}}$ [Ω]	τ_r [ms]	−80 %	−40 %	**ajustado**	+100 %	+200 %
		$R_{r,\mathrm{MPC}}$ [Ω]				
0,50	321	0,10	0,30	**0,50**	1,00	1,50
0,83	*193*	*caracterizada en la Sección 4.2 con ocho combinaciones*				
1,50	107	0,30	0,90	**1,50**	3,00	4,50
3,00	53	0,60	1,80	**3,00**	6,00	9,00
5,00	32	1,00	3,00	**5,00**	10,00	15,00

A diferencia de la sección anterior, esta explora cuatro configuraciones adicionales de $R_{r,\mathrm{MAS}}$ sobre una sola maniobra: el **arranque con carga de bomba**. El torque de la bomba depende de la velocidad, de modo que el punto de equilibrio alcanzado mide directamente la capacidad de torque disponible: si el accionamiento no consigue sostener

la referencia, se estabiliza donde puede igualar la carga, y esa velocidad de equilibrio resume la degradación en un solo número.

4.3.3. Resultados

Tabla 4.10: Velocidad de equilibrio alcanzada con carga de bomba y error de regulación resultante, para las veinte combinaciones de la Tabla 4.9. Entre paréntesis, el desajuste relativo ε_R de cada combinación.

	$R_{r,\mathrm{MAS}}$ [Ω]			
ε_R aproximado	0,50	1,50	3,00	5,00
−80 %	645	576	564	555
−40 %	1165	1156	1124	1097
ajustado	**1429**	**1429**	**1429**	**1374**
+100 %	1386	1241	1054	858
+200 %	1212	1009	777	563
error del caso ajustado	0,8 %	0,8 %	0,7 %	**4,6 %**
$\lvert T_e\rvert_{peak}$ ajustado [N·m]	76,7	76,5	76,6	76,4

La similitud del caso ajustado se conserva casi hasta el extremo del barrido. En $R_{r,\mathrm{MAS}} = 0{,}5$, 1,5 y 3 Ω la máquina ajustada alcanza la *misma* velocidad de equilibrio, 1429 rpm con un error del 0,7–0,8 %, pese a que sus constantes de tiempo de rotor difieren en un factor de seis. Solo en 5 Ω aparece un error apreciable, del 4,6 %, y ese residuo tiene la explicación que anticipa la Subsección 4.3.1 y no es un fallo del controlador.

El caso ajustado es siempre el mejor de su columna. Con la referencia dependiente del punto de operación ese artefacto desaparece: en las cuatro columnas la velocidad de equilibrio es máxima en la fila ajustada y decae monótonamente hacia ambos lados. La velocidad de equilibrio recupera así su valor como indicador de la calidad del modelo, es decir, un corrector que persiga maximizarla ya no puede ser inducido a alejarse del valor correcto.

La firma de las cuatro magnitudes se conserva. Las Figuras 4.19 a 4.22 confirman, columna por columna, lo resumido en la Tabla 4.7. Hacia subestimación la velocidad sube de forma no monótona, la corriente se distorsiona, la estimación de torque oculta el *ripple* real y el flujo real se dispara por encima del nominal mientras la estimación se queda corta. Hacia sobreestimación la rampa se conserva recta pero más lenta, la corriente permanece sinusoidal y sostenida, la estimación de torque se infla y el flujo real cae por debajo del nominal.

Consecuencia para el Capítulo 5. Este resultado es el que justifica entrenar un estimador único sobre un rango amplio de resistencias en lugar de uno por máquina: el desajuste se manifiesta de la misma forma y con severidad comparable en toda la familia explorada, y lo hace en función del error *relativo* y no del valor absoluto de $R_{r,\mathrm{MAS}}$. La salvedad del caso de 5 Ω acota ese enunciado por arriba: existe un valor de resistencia a partir del cual la máquina deja de sostener la carga nominal por sí misma, y allí la referencia contra la que se mide la degradación deja de ser el seguimiento perfecto.

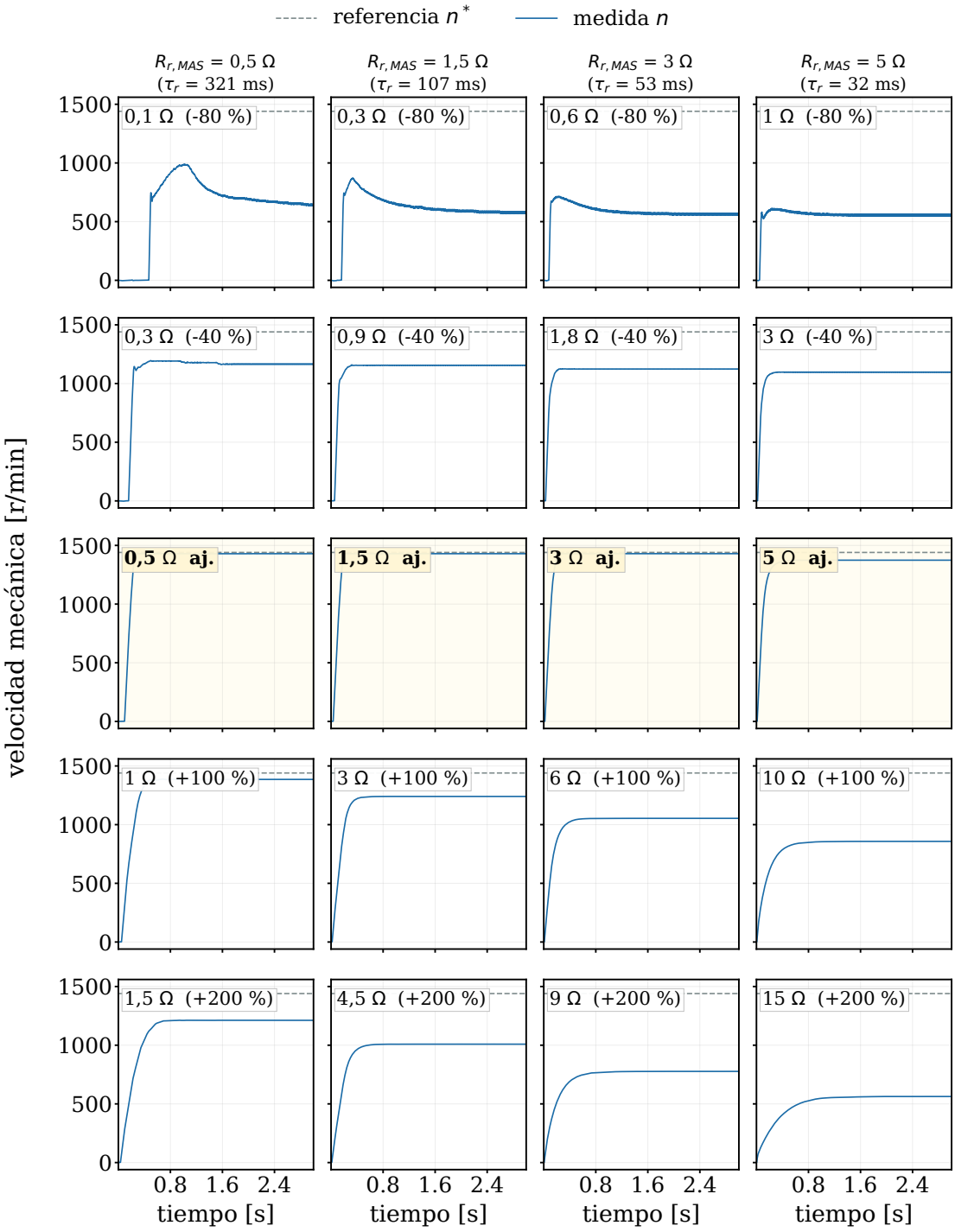
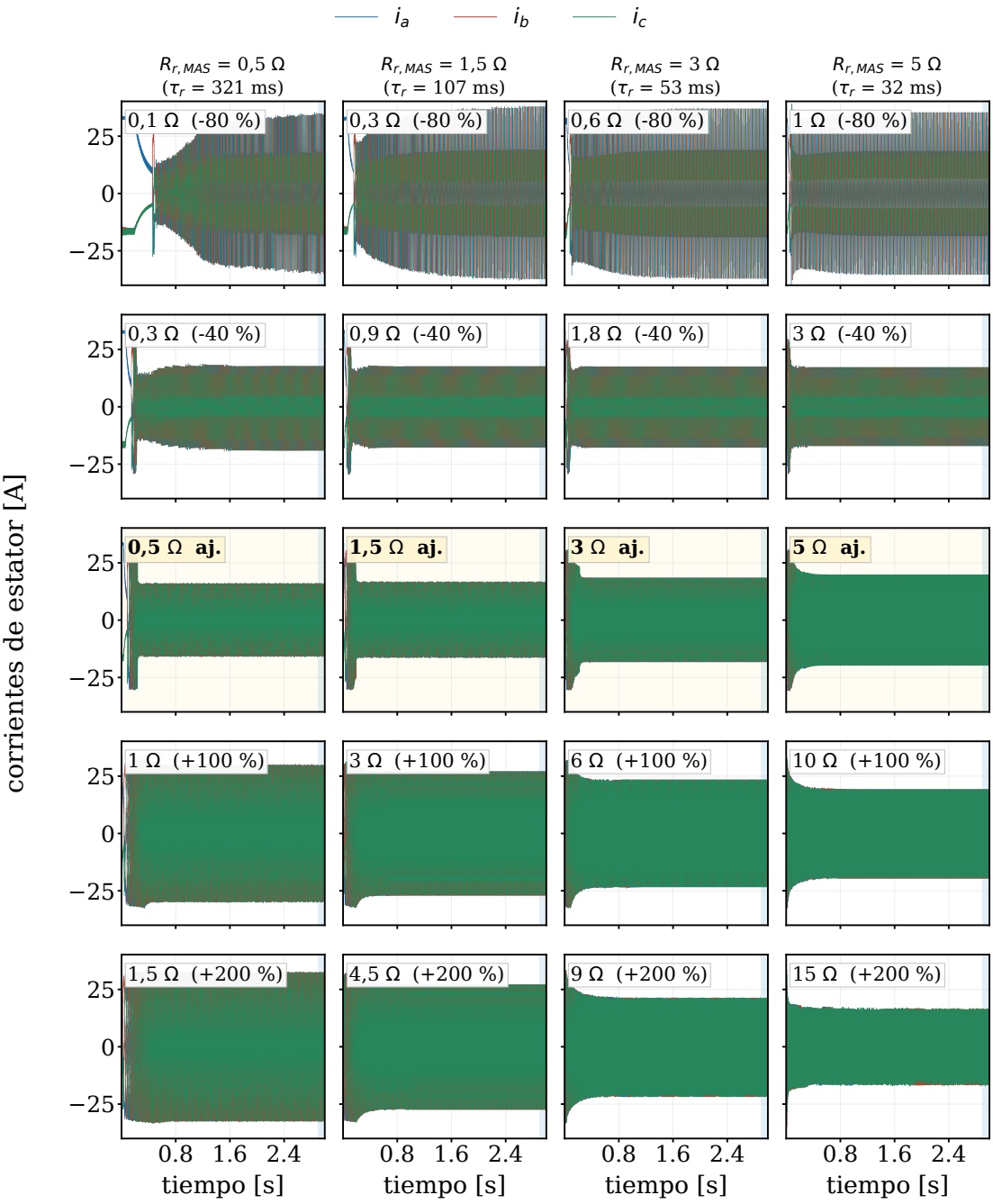
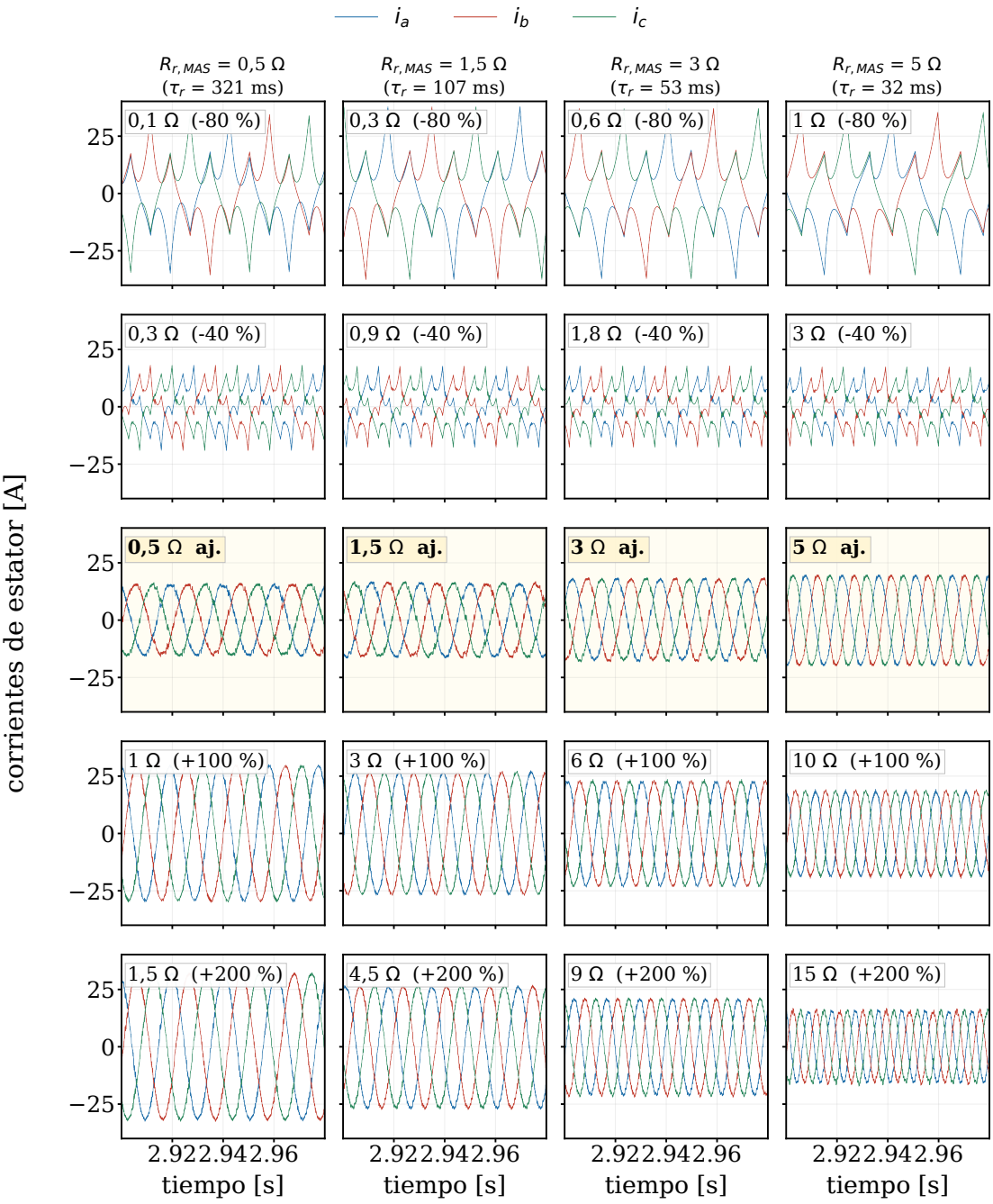


Figura 4.18: Velocidad mecánica frente a su referencia, con carga de bomba, para las veinte combinaciones de la Tabla 4.9.



La ventana sombreada de equilibrio (2,9–2,98 s) se amplía en la lámina siguiente.

Figura 4.19: Corrientes de estator durante la maniobra completa, con la misma disposición de la Figura 4.18 y escala vertical común a los veinte paneles.



Ventana de equilibrio, común a los veinte casos; misma escala vertical que la lámina anterior.

Figura 4.20: Zoom de la Figura 4.19, la amplitud y la forma de onda son comparables entre paneles.

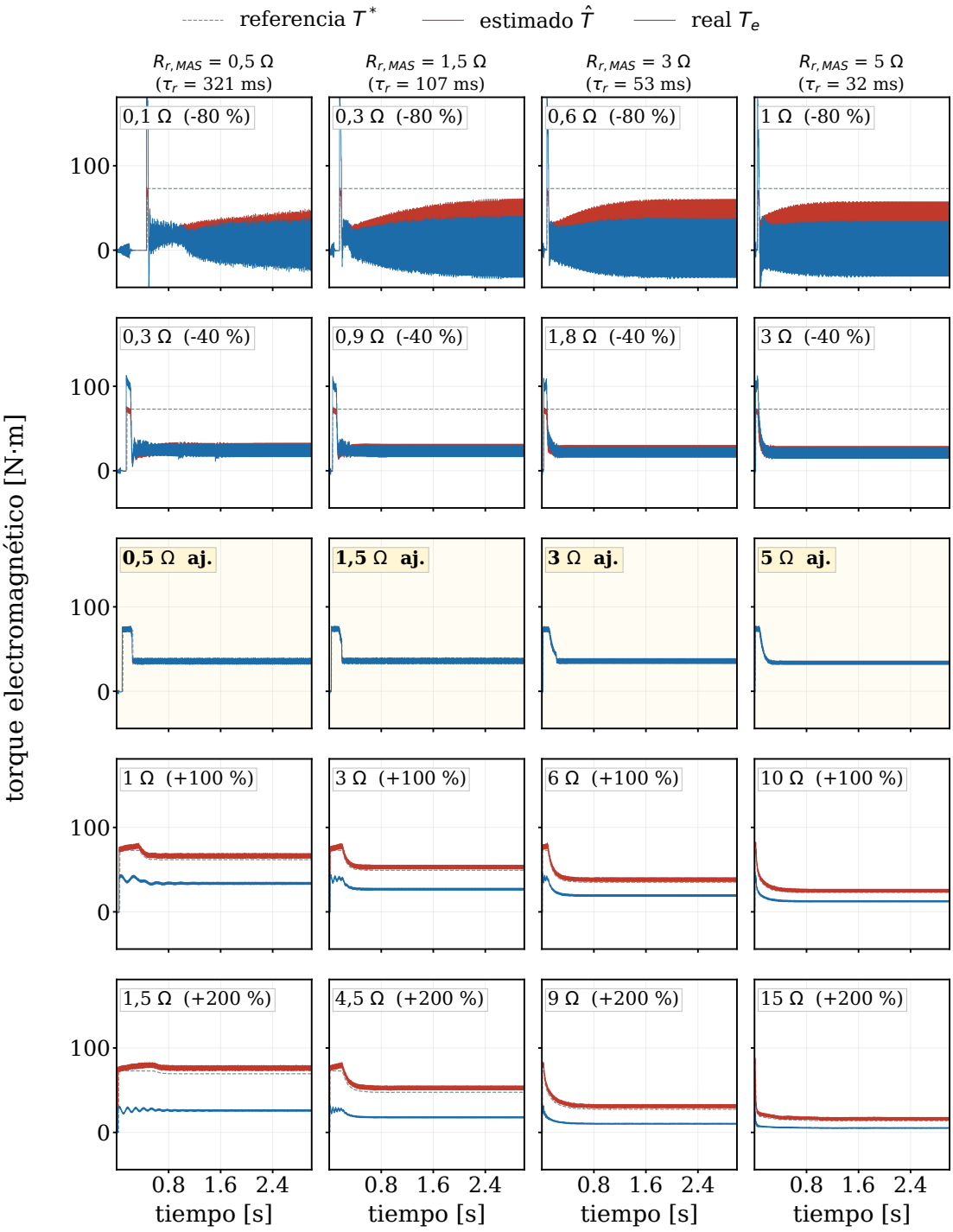


Figura 4.21: Torque de referencia, estimado y real, con la misma disposición de la Figura 4.18.

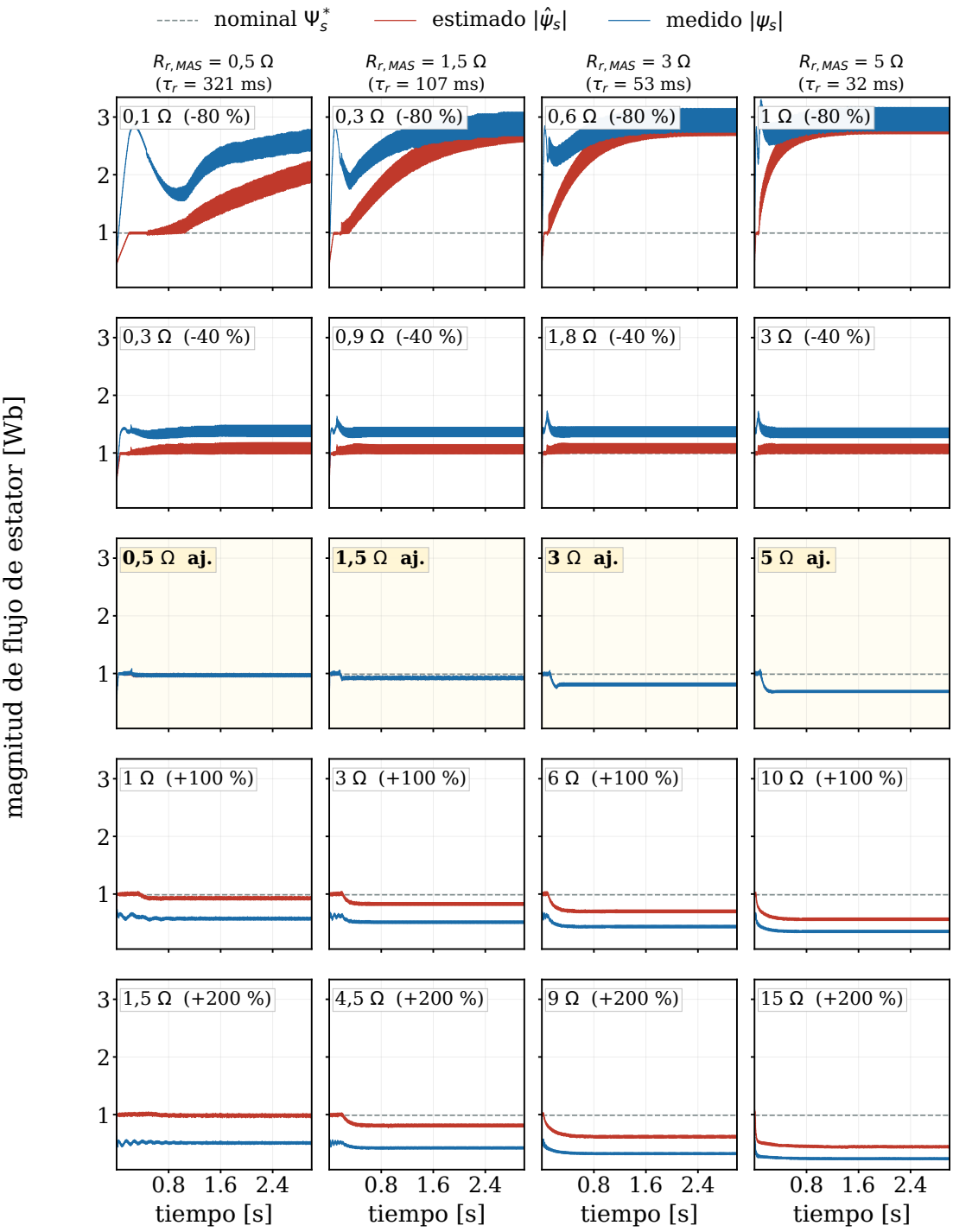


Figura 4.22: Magnitud de flujo de estator: nominal, estimada y real, con la misma disposición de la Figura 4.18.

4.4. Resumen del efecto del desajuste de $R_{r,\mathrm{MPC}}$

Las dos secciones anteriores entregaron la evidencia: el barrido detallado sobre la máquina de resistencia nominal y su extensión a otros valores de resistencia real. Esta sección la ordena y la explica. No introduce mediciones nuevas: recorre lo observado para responder *por qué* el desajuste produce esos efectos y *de qué* depende su gravedad.

4.4.1. Comportamiento con subestimación ($\Delta R_r < 0$)

Subestimar la resistencia equivale a suponer una constante de tiempo de rotor mayor que la real. El mecanismo que ordena las cuatro manifestaciones siguientes es único: $\hat{\omega}_r$ queda por debajo del deslizamiento real y con ella $\hat{\omega}_s$; como la referencia de flujo de (3.22) divide por esa frecuencia sincrónica estimada, el límite de flujo resulta sobreestimado y **el campo no llega a debilitarse**.

4.4.1.a. Velocidad

El arranque deja de ser lineal. Es la manifestación más característica de esta rama.

Bajo carga el efecto se vuelve cualitativo. Con $-88\,\%$ en el arranque cargado el ajuste lineal cae a $R^2 = 0{,}540$ y la máquina describe una trayectoria que ya no es una rampa: arranca, alcanza un máximo y después *decae*, de modo que la aceleración media del tramo resulta **negativa**: -125 rpm/s en el arranque cargado, -193 tras el escalón y -110 con bomba. No se trata de una rampa más lenta sino de una máquina que se deteriora mientras opera, y el valor final que registran las tablas debe leerse como el estado al término de la simulación y no como un punto de equilibrio.

4.4.1.b. Corrientes

A mayor desajuste, mayor distorsión. Es la única de las cuatro variables cuya degradación resulta *monótona* en el desajuste, y lo es en las tres maniobras con carga. Partiendo de una distorsión de 1,5 a 4,6 % en el caso ajustado, la subestimación la lleva a:

Maniobra	ajustado	$-40\,\%$	$-64\,\%$	$-88\,\%$
Arranque con carga	4,03 %	35,2 %	71,9 %	99,1 %
Escalón de carga	4,63 %	37,5 %	51,4 %	110,8 %
Bomba	1,46 %	31,1 %	54,1 %	72,1 %

La corriente no solo se ensucia: además *crece de forma sostenida* acompañando la consecuencia del flujo, de 18,6 a más de 52 A en el arranque cargado. Es una magnitud creciente y distorsionada, no un régimen.

4.4.1.c. Torque estimado

Conviene separar dos hechos que suelen confundirse, porque apuntan en direcciones opuestas.

El ***ripple* del torque real crece de forma dramática**: de 1,58 N·m en el caso ajustado a 4,3, 27,7 y 65,6 N·m conforme se agrava la subestimación, con excursiones instantáneas de hasta 208 N·m. La máquina sufre un castigo mecánico severo.

Pero **el estimador no es la causa**: en el caso extremo registra 9,0 N·m de rizado frente a los 65,6 reales, siete veces menos. Y su valor medio, lejos de quedarse corto, **excede al real**: el cociente T_{est}/T_e vale 1,08, 2,33 y 3,94 en el arranque cargado, y llega a 5,24 con bomba. El controlador percibe un torque medio mayor del que la máquina entrega y un *ripple* mucho menor del real.

4.4.1.d. Flujo de estator estimado

La máquina se sobre-magnetiza y el estimador se queda corto. El flujo *real* llega a 1,29, 2,98 y 3,54 veces su valor nominal en el arranque cargado, hasta 4,55 veces en los instantes de *peak*, consecuencia directa del campo que nunca se debilita. Esas magnitudes solo son posibles porque la máquina simulada tiene inductancias constantes y no satura (Subsección 4.1.1): nada limita el crecimiento del flujo. Una máquina real saturaría mucho antes, con menos flujo y bastante más corriente, de modo que los barridos con subestimación fuerte son pruebas de estrés del modelo y no predicciones del comportamiento de un accionamiento real. La *estimación*, en cambio, se sitúa por debajo de ese flujo real: el cociente $\hat{\psi}_s/\psi_s$ vale 0,81, 0,92 y 0,76 en las tres subestimaciones. El controlador ve una máquina menos magnetizada de lo que está.

El ***ripple* del flujo estimado es mayor**: de 0,0129 Wb en el caso ajustado a 0,046, 0,119 y 0,245 Wb, esto es, hasta *diecinueve veces* el valor nominal de rizado. A diferencia de lo que ocurre en la rama opuesta, aquí la estimación sí acompaña esa divergencia, registra 0,256 Wb de *ripple* frente a los 0,245 reales: el lazo pierde el control de la magnitud de flujo, no solo su medida.

4.4.2. Comportamiento con sobreestimación ($\Delta R_r > 0$)

El mecanismo es el simétrico del anterior, pero sus efectos no lo son. Al sobreestimar, $\hat{\omega}_r$ excede al deslizamiento real, $\hat{\omega}_s$ crece y la referencia de flujo de (3.22) **debilita el campo más de lo necesario**: el barrido factorial sitúa $\psi_s^*/\psi_{s,\text{nom}}$ en 0,89 frente a 0,92 del ajustado. Con menos flujo, el mismo torque exige más corriente, y en régimen con carga se emplea el 59,7 % de la capacidad de i_{sq}: seis veces lo que utiliza la rama de subestimación.

4.4.2.a. Velocidad

A mayor desajuste, menor aceleración de arranque, pero la rampa sigue siendo recta. Es exactamente lo contrario de la rama anterior: allí se destruía la *forma* de la subida, aquí se degrada su *pendiente*. La aceleración media del arranque en vacío decae de forma monótona con el desajuste,

ε_R	ajustado	+81 %	+261 %	+502 %	+1105 %
Aceleración [rpm/s]	11 631	6787	3426	1627	361
R^2 de la rampa	1,0000	1,0000	0,999	0,981	0,985

mientras la linealidad se conserva prácticamente intacta. La máquina no acelera de forma errática: acelera poco.

4.4.2.b. Corrientes

Magnitud grande y sostenida, con distorsión casi nula. La corriente se clava en 28,4 A desde el primer instante y ahí permanece, más del doble de la corriente nominal, mientras la distorsión cae a valores *inferiores* a los del caso ajustado: 1,84, 0,79, 2,73 y 1,25 % frente al 4,03 % del modelo ajustado en el arranque cargado.

Es una sinusoide impecable sostenida sobre una máquina que no está produciendo torque útil, y de ahí se concluye que: **la forma de onda, por sí sola, no informa sobre si el accionamiento funciona bien**. Una corriente limpia puede corresponder al mejor y al peor de los casos.

4.4.2.c. Torque estimado

Una sobreestimación grande y monótona del torque real, que es además el mecanismo por el que la máquina deja de acelerar. El cociente T_{est}/T_e crece sin excepción con el desajuste: 1,56, 3,51, 5,83 y 11,70 en el arranque cargado, mientras el torque real entregado cae de 18,4 a 4,9 N·m.

La manifestación más reveladora para el diagnóstico es que **el controlador cree estar entregando más torque del que se le pide**: la medición da $T_{\text{est}} = 1{,}72\,T_{\text{nom}}$ contra una referencia $T^* = 1{,}53\,T_{\text{nom}}$.

Conviene precisar el alcance de esta firma: el torque estimado excede al real *en ambos sentidos* del desajuste, describiendo una curva en V con mínimo en el caso ajustado. Lo que distingue a esta rama no es el signo del sesgo sino que sea *mayor* en el desajuste sobrestimado y que persista de forma indefinida en el punto de equilibrio.

4.4.2.d. Flujo de estator estimado

Una sobreestimación del flujo, y sin *ripple* apreciable. La estimación duplica al flujo real: el cociente $\hat{\psi}_s/\psi_s$ vale 1,63, 2,02, 2,28 y 2,27 conforme crece el desajuste, mientras el flujo real cae a 0,57, 0,45, 0,44 y 0,37 veces el nominal. La máquina opera sub-magnetizada sin que el controlador lo registre.

A diferencia de la rama opuesta, el *ripple* **no se degrada**: vale 0,0134, 0,0091, 0,0083 y 0,0082 Wb frente a los 0,0129 Wb del caso ajustado, esto es, se mantiene o incluso disminuye. La divergencia entre estimación y realidad es aquí un sesgo estable y silencioso, no una oscilación; y precisamente por ser estable resulta observable durante todo el tiempo que se desee medir.

4.4.3. Dependencia del punto de operación

Los cuatro barridos aplican la *misma* escalera de desajustes sobre la *misma* máquina y arrojan desenlaces que van desde la indiferencia total hasta la detención. Puesto que el

desajuste es idéntico en todos ellos, esa disparidad no puede atribuirse a su magnitud: la variable que cambia es la **exigencia de torque**.

La razón de fondo es que **el desajuste solo se manifiesta en las señales medidas y estimadas cuando el estimador de flujo tiene trabajo que hacer**, y eso ocurre únicamente cuando se solicita torque. En régimen permanente y sin carga se cumplen dos condiciones simultáneas que lo desactivan por completo:

1. Por una parte, el torque requerido es nulo, de modo que $i_{sq} \to 0$ y la frecuencia de deslizamiento de la condición de orientación (2.20), $\omega_r = L_m i_{sq}/(\tau_r \psi_r)$, tiende a cero aunque el valor de $\hat{\tau}_r$ sea diferente al real: el término por el cual R_r contamina la orientación está multiplicado por una corriente que es casi nula, y el ángulo γ_r deja de acumular error.

2. Por otro lado, con $d\psi_r/dt = 0$ la dinámica de flujo se reduce a $\psi_r = L_m i_{sd}$, expresión en la que R_r no interviene.

En ambas vías la propagación del error quedan anuladas en ese punto de operación, y el controlador opera sobre un modelo que, *para lo que la maniobra exige*, es efectivamente correcto. La consecuencia es doble y determina dónde tiene sentido buscar evidencia del desajuste: el PTC sostiene el seguimiento en vacío sin asistencia alguna, hasta un error de hasta +1105 % resulta indistinguible del caso ajustado. Y en vacío las señales del lazo **carecen de información** sobre R_r.

4.4.4. Propagación analítica del error

Los desenlaces recién descritos tienen un origen común, demostrable directamente desde las ecuaciones del controlador (Capítulo 3) y sin recurrir a simulación. Un error ΔR_r altera, en cadena:

1. **La constante de tiempo de rotor del modelo**, $\hat{\tau}_r = L_r/R_{r,\mathrm{MPC}}$. Con $\Delta R_r < 0$ (subestimación), $\hat{\tau}_r > \tau_r$: el controlador supone una dinámica de flujo más lenta que la real. Con $\Delta R_r > 0$, $\hat{\tau}_r < \tau_r$: la tendencia contraria.

2. **La frecuencia de deslizamiento estimada**, $\hat{\omega}_r = L_m i_{sq}/(\hat{\tau}_r \hat{\psi}_r)$ (2.20). Bajo condiciones instantáneas comparables (i_{sq}, $\hat{\psi}_r$ dadas), la subestimación de R_r produce una frecuencia de deslizamiento calculada *menor* que la que la máquina necesita, y la sobreestimación una *mayor*.

3. **El ángulo de orientación** γ_r (3.8), que integra la velocidad sincrónica construida con $\hat{\omega}_r$: el error de deslizamiento se *acumula* en el ángulo. En ambos sentidos del desajuste el marco *dq* estimado puede desviarse del flujo real; que el controlador imponga $\hat{\psi}_{rq} = 0$ en *su* marco no demuestra que ese marco esté físicamente orientado, la condición de orientación es una identidad interna del estimador, no una medición. Y con la orientación desviada, la descomposición *dq* pierde parcialmente su significado: las componentes i_{sd} e i_{sq} dejan de gobernar por separado el flujo y el torque.

4. **Los parámetros del predictor**, $R_\sigma = R_s + k_r^2 R_{r,\mathrm{MPC}}$ y $\tau_\sigma = L_\sigma/R_\sigma$: el error no queda confinado al estimador de flujo, sino que sesga también las predicciones de corriente y flujo a $k+1$ y $k+2$ (3.15)–(3.19).

5. **Las estimaciones de flujo y torque** (3.9), (3.13): magnitud y fase del flujo estimado, y con ellas el torque estimado, se apartan de las magnitudes físicas.

6. **El ordenamiento de los candidatos y la selección del vector**: la función de costo (3.7) compara los ocho candidatos contra un futuro predicho con parámetros sesgados; si el sesgo es suficiente, el orden relativo de los candidatos cambia y el vector aplicado deja de ser el óptimo físico.

La caracterización anterior identifica las variables que el desajuste afecta: la orientación γ_r, las estimaciones de flujo y torque, las predicciones de corriente, la selección de vectores, la forma de onda de las corrientes y la regulación de velocidad. Estas manifestaciones pueden transformarse en observables construidos con señales disponibles en línea. En particular, la discrepancia entre la corriente medida y la predicha proporciona el residuo físico que sustenta la Propuesta 1, mientras que las restantes variables aportan contexto para decidir si una corrección resulta admisible.

También se comprobó que la observabilidad depende de la maniobra. Los desajustes severos se manifiestan con claridad durante el arranque, mientras que los moderados requieren una condición de carga sostenida para producir evidencia suficiente. Esta relación entre régimen y observabilidad motiva la separación entre expertos de arranque y de carga que se desarrolla a continuación en el Capítulo 5; la Propuesta 2, presentada posteriormente en el Capítulo 6, aborda el mismo problema mediante una ventana temporal procesada por una única red.

Capítulo 5

Propuesta 1: Expertos PINN para la Corrección del Modelo Predictivo

Este capítulo desarrolla la primera de las dos propuestas presentadas en la Subsección 2.5.4.a: la que construye un estimador físico y emplea redes neuronales para aprender el des-sesgo que dicho estimador no puede eliminar por sí solo. La solución se compone de **dos expertos, de arranque y de carga, con dos redes direccionales cada uno**, coordinados por algoritmos de estados deterministas y complementados por vías de corrección puramente física para los regímenes en que las redes no disponen de información suficiente.

5.1. Construcción de un residuo físico universal

5.1.1. Construcción del núcleo físico: el residuo ε afín del predictor

La caracterización del Capítulo 4 mostró que el desajuste se manifiesta sobre variables medibles, pero ninguna de ellas lo cuantifica por sí sola: se necesita un observable *construido* que aísle la firma del error de R_r del resto de la dinámica. Las variables internas del controlador en coordenadas de campo son malas candidatas para ese papel, por una razón sutil: el propio ángulo γ_r con que se calculan proviene del estimador de flujo, que ya está contaminado por $R_{r,\mathrm{MPC}}$; un residuo formado sobre ellas mezclaría el error que se busca medir con el error del marco en que se mide.

Por ello, la plataforma incorpora un **predictor auxiliar de las corrientes de estator en el marco estacionario** $\alpha\beta$: el marco estacionario es fijo y no depende de ningún estimador. Además, esta predicción no interviene en la función de costo ni modifica la selección del vector de tensión; es instrumentación pura.

5.1.1.a. Predictor auxiliar de corriente en coordenadas $\alpha\beta$

1. En cada instante k, las tensiones dq actualmente aplicadas se reconstruyen en coordenadas estacionarias mediante la transformada inversa de Park:

$$v_{s\alpha}[k] = v_{sd}[k]\cos\gamma_r[k] - v_{sq}[k]\sin\gamma_r[k], \tag{5.1}$$
$$v_{s\beta}[k] = v_{sd}[k]\sin\gamma_r[k] + v_{sq}[k]\cos\gamma_r[k], \tag{5.2}$$

y, dado que el marco dq está orientado con el flujo de rotor estimado ($\hat{\psi}_{rq} = 0$), las componentes estacionarias del flujo estimado son

$$\hat{\psi}_{r\alpha}[k] = \hat{\psi}_r[k]\cos\gamma_r[k], \qquad \hat{\psi}_{r\beta}[k] = \hat{\psi}_r[k]\sin\gamma_r[k]. \tag{5.3}$$

2. A partir del modelo de corriente de la máquina y aplicando discretización *Forward Euler*, las predicciones de un paso resultan

$$\hat{i}_{s\alpha}[k+1|k] = \left(1 - \frac{T_s}{\tau_\sigma}\right) i_{s\alpha}[k] + \frac{T_s}{L_\sigma}\left[v_{s\alpha}[k] + k_r\left(\frac{\hat{\psi}_{r\alpha}[k]}{\tau_r} + \omega_{em}[k]\,\hat{\psi}_{r\beta}[k]\right)\right], \tag{5.4}$$

$$\hat{i}_{s\beta}[k+1|k] = \left(1 - \frac{T_s}{\tau_\sigma}\right) i_{s\beta}[k] + \frac{T_s}{L_\sigma}\left[v_{s\beta}[k] + k_r\left(\frac{\hat{\psi}_{r\beta}[k]}{\tau_r} - \omega_{em}[k]\,\hat{\psi}_{r\alpha}[k]\right)\right]. \tag{5.5}$$

Estas ecuaciones dependen explícitamente de la resistencia de rotor del controlador a través de

$$\tau_r = \frac{L_r}{R^{\text{eff}}_{r,\text{MPC}}}, \qquad R_\sigma = R_s + k_r^2\, R^{\text{eff}}_{r,\text{MPC}}, \qquad \tau_\sigma = \frac{L_\sigma}{R_\sigma}. \tag{5.6}$$

3. Las predicciones $\hat{i}_{s\alpha}[k+1|k]$ y $\hat{i}_{s\beta}[k+1|k]$ se almacenan en estados discretos, de modo que en el instante siguiente puedan compararse con las mediciones actuales:

$$\varepsilon_\alpha[k] = i^{\text{med}}_{s\alpha}[k] - \hat{i}_{s\alpha}[k|k-1], \qquad \varepsilon_\beta[k] = i^{\text{med}}_{s\beta}[k] - \hat{i}_{s\beta}[k|k-1]. \tag{5.7}$$

Algo importante de este predictor auxiliar: *no* entrega el valor real de R_r, porque sus predicciones usan el flujo y el ángulo estimados, que arrastran su propio sesgo. Pero su función es construir un residuo físicamente interpretable y sensible al desajuste.

5.1.1.b. El residuo de predicción $\varepsilon[k]$ como indicador del desajuste

Un residuo es la diferencia entre lo que un modelo predice y lo que realmente se mide. El predictor auxiliar de la Subsección 5.1.1.a calcula en cada ciclo la corriente que la máquina debería presentar un paso más adelante *según el modelo del controlador* (que usa $R^{\text{eff}}_{r,\text{MPC}}$); la comparación con la medición del instante siguiente, ec. (5.7), forma en notación vectorial el residuo

$$\boldsymbol{\varepsilon}[k] = \mathbf{i}^{\text{med}}_s[k] - \hat{\mathbf{i}}_s[k|k-1]. \tag{5.8}$$

Si el modelo fuese perfecto, el residuo sería nulo salvo ruido. Como el único parámetro genuinamente incierto del modelo es R_r, una parte sustancial del residuo se debe precisamente al desajuste entre $R_{r,\text{MPC}}$ y $R_{r,\text{MAS}}$: el residuo es pequeño cuando ambos valores coinciden y crece a medida que se separan.

5.1.1.c. Afinidad del predictor en R_r

La propiedad que permite relacionar el residuo con el error de resistencia es que el predictor de corriente es una **función afín** de $R^{\text{eff}}_{r,\text{MPC}}$. Una función es afín en una variable cuando depende de ella como una parte constante más una parte proporcional, $f(x) = a+b\,x$; su propiedad es que la diferencia entre dos evaluaciones de la función es *exactamente* proporcional a la diferencia de la variable, sin términos de orden superior:

$$f(x_2) - f(x_1) = b\,(x_2 - x_1), \qquad b = \frac{\partial f}{\partial x}. \tag{5.9}$$

La afinidad del predictor puede verificarse por inspección de (5.4)–(5.5): los coeficientes dependen de $R^{\text{eff}}_{r,\text{MPC}}$ de manera lineal. Es decir, vista como función de $R^{\text{eff}}_{r,\text{MPC}}$ la predicción tiene la forma de $\hat{\mathbf{i}}_s = \mathbf{A} + \boldsymbol{\varphi}\, R^{\text{eff}}_{r,\text{MPC}}$.

La importancia de esta propiedad es que, entre dos predicciones calculadas con los mismos estados y entradas, la diferencia que produce la resistencia queda expresada exactamente por una sola derivada. Que esa relación sirva además para *invertir* el residuo medido depende de hipótesis sobre la planta, que se explicitan a continuación. La afinidad se comprobó también de forma simbólica: la segunda derivada del predictor respecto de la resistencia es nula.

5.1.1.d. Sensibilidad del residuo afín

Se define la **sensibilidad** $\boldsymbol{\varphi}$ como la derivada del predictor respecto de la resistencia, evaluada con las señales del instante en que se realizó la predicción (en $k-1$). Al ser afín, esta derivada no depende del valor de $R^{\text{eff}}_{r,\text{MPC}}$ con que se evalúa el predictor en ese paso; sí depende de su historia, porque el flujo estimado $\hat{\boldsymbol{\psi}}_r$ se calcula con él. Se calcula como:

$$\varphi_\alpha[k-1] = \frac{\partial\, \hat{i}_{s\alpha}[k|k-1]}{\partial R^{\text{eff}}_{r,\text{MPC}}} = \frac{T_s}{L_\sigma}\left(-k_r^2\, i_{s\alpha}[k-1] + \frac{k_r}{L_r}\,\hat{\psi}_{r\alpha}[k-1]\right), \tag{5.10}$$

$$\varphi_\beta[k-1] = \frac{\partial\, \hat{i}_{s\beta}[k|k-1]}{\partial R^{\text{eff}}_{r,\text{MPC}}} = \frac{T_s}{L_\sigma}\left(-k_r^2\, i_{s\beta}[k-1] + \frac{k_r}{L_r}\,\hat{\psi}_{r\beta}[k-1]\right). \tag{5.11}$$

Las componentes $\hat{\psi}_{r\alpha}[k-1]$ y $\hat{\psi}_{r\beta}[k-1]$ son las del flujo girado con el ángulo $\gamma_r[k-1]$, el mismo que usó la predicción. En el código desplegado de esta propuesta, la corriente y la magnitud del flujo son las de $k-1$, pero el ángulo va un paso adelantado. La Sección G.2 mide ese desfase y muestra que su efecto sobre los resultados es pequeño.

Evaluando el predictor con los mismos estados y entradas de $k-1$ y con dos resistencias distintas, y aplicando la propiedad afín (5.9), las dos predicciones cumplen exactamente

$$\hat{\mathbf{i}}_s[k|k-1;\, R_{r,\text{MAS}}] - \hat{\mathbf{i}}_s[k|k-1;\, R^{\text{eff}}_{r,\text{MPC}}] = \boldsymbol{\varphi}[k-1]\left(R_{r,\text{MAS}} - R^{\text{eff}}_{r,\text{MPC}}\right). \tag{5.12}$$

Esta igualdad es una **identidad del predictor**, no de la planta: relaciona dos predicciones, no una predicción con una medición. Para llevarla al residuo (5.8) hay que suponer que la máquina se comporta como el predictor evaluado con su resistencia verdadera y con los mismos estados y entradas, $\mathbf{i}_s^{\text{med}}[k] = \hat{\mathbf{i}}_s[k|k-1;\, R_{r,\text{MAS}}]$. Esa hipótesis no se cumple exactamente. Si se agrupa lo que falta en un término $\boldsymbol{\eta}[k]$, el residuo medido queda

$$\boldsymbol{\varepsilon}[k] = \boldsymbol{\varphi}[k-1]\left(R_{r,\text{MAS}} - R^{\text{eff}}_{r,\text{MPC}}\right) + \boldsymbol{\eta}[k], \tag{5.13}$$

donde $\boldsymbol{\eta}[k]$ reúne cuatro fuentes de error:

- **Estado.** El predictor usa el flujo de rotor y el ángulo que estima el propio controlador, y los calcula con $R^{\text{eff}}_{r,\text{MPC}}$. Si $R^{\text{eff}}_{r,\text{MPC}} \neq R_{r,\text{MAS}}$, esos estados no son los de la máquina, y el predictor evaluado en $R_{r,\text{MAS}}$ con ellos tampoco la reproduce. Este término depende del propio desajuste.
- **Entrada y discretización.** La tensión aplicada se reconstruye en $\alpha\beta$ con el ángulo estimado, y el predictor avanza un paso de *Forward Euler*, mientras la planta evoluciona en tiempo continuo dentro del período de conmutación.
- **Modelo.** En esta plataforma los demás parámetros del controlador coinciden con los de la máquina. En un accionamiento real se agregarían, entre otros, la variación de R_s con la temperatura, la saturación de L_m y los tiempos muertos del inversor.
- **Medición.** En la simulación la corriente se muestrea sin ruido ni cuantización. En un banco real ambos entrarían en $\boldsymbol{\eta}$.

Solo si $\boldsymbol{\eta}[k] = \mathbf{0}$ el residuo medido en k es proporcional al error de resistencia, con factor de proporcionalidad igual a la sensibilidad evaluada en el instante de la predicción. Esa es la diferencia entre la *identidad del predictor* (5.12), que es exacta, y la *identificación de la planta*, que es aproximada.

5.1.2. Limitación del residuo y proyección sobre un ciclo de corriente $\hat{d}$

Aun en el caso ideal, con $\boldsymbol{\eta} = \mathbf{0}$, la relación (5.13) por sí sola no constituye todavía un observable utilizable. En cada período de muestreo entrega dos ecuaciones escalares: las componentes α y β, para una única incógnita $\delta = R_{r,\text{MAS}} - R^{\text{eff}}_{r,\text{MPC}}$, de modo que despejar el desajuste de *una sola muestra* es aritméticamente posible. Y es precisamente lo que no conviene hacer. La razón fija una decisión de diseño de la capa correctiva.

Una muestra aislada no dice en qué maniobra está la máquina.. Y la Subsección 4.4.3 estableció que el desajuste solo se manifiesta sobre las señales cuando el estimador de flujo tiene trabajo que hacer, esto es, cuando se solicita torque.

La consecuencia para el diseño es directa: antes de estimar hay que poder decidir *si la maniobra permite estimar*. Y esa decisión no se puede tomar sobre un período de conmutación.

La decisión adoptada es proyectar el residuo sobre un ciclo eléctrico completo de la corriente. La ventana deja así de ser un intervalo arbitrario de tiempo para ser una unidad física de la propia señal, y con ella se ganan cuatro cosas que una muestra no ofrece:

1. **Un valor medio robusto del desajuste.** Promediar sobre las N muestras del ciclo atenúa el *ripple* de conmutación y el ruido de medición, que sobre una muestra aislada dominan por completo la estimación.
2. **El contenido armónico.** La transformada discreta de Fourier puede calcularse sobre cualquier número de muestras; tomar exactamente un ciclo es una decisión del estimador espectral. Con la ventana igual al período de la fundamental, y mientras

la corriente se repita de un ciclo al siguiente, cada armónico cae en un solo término de la transformada, sin fuga hacia los vecinos ni ventana de ponderación, y su orden conserva el sentido de múltiplo de la fundamental. Permite medir la distorsión de la corriente y sus armónicos individuales.

3. **La evolución de la velocidad.** Dentro del ciclo se dispone de la variación de la velocidad y, entre ciclos consecutivos, de su tendencia. Es lo que separa una máquina que acelera de una que regula, y por tanto una maniobra de arranque de una de carga.

4. **La caracterización del estado.** Nivel de carga, seguimiento de la referencia de velocidad, estado de magnetización y duración del propio ciclo quedan disponibles como descriptores del punto de operación. Sobre ellos se construye la declaración de si la maniobra es informativa o no.

Un ciclo de corriente es la unidad natural de observación de este problema: la ventana más corta que contiene toda la información justa y necesaria.

Para la estimación, se usa **la proyección** del residuo. Sobre un ciclo de corriente, el sistema guarda las N muestras y se resuelve por mínimos cuadrados: se busca el δ que minimiza

$$\min_{\delta} \sum_{\text{ciclo}} \left\| \boldsymbol{\varepsilon} - \boldsymbol{\varphi}\, \delta \right\|^2, \tag{5.14}$$

cuya solución cerrada es la **proyección** del residuo sobre la sensibilidad:

$$\hat{d} = \frac{\langle \boldsymbol{\varepsilon}, \boldsymbol{\varphi} \rangle}{\langle \boldsymbol{\varphi}, \boldsymbol{\varphi} \rangle} \;\approx\; R_{r,\mathrm{MAS}} - R^{\mathrm{eff}}_{r,\mathrm{MPC}} \quad [\Omega], \tag{5.15}$$

donde $\langle \cdot, \cdot \rangle$ denota la suma de los productos muestra a muestra sobre el ciclo completo, incluidas ambas componentes. En pocas palabras: $\hat{d}$ es el error de resistencia que mejor explica el residuo observado a lo largo de un ciclo. Su unidad es el ohm y su signo indica el sentido del error, positivo si $R_{r,\mathrm{MAS}} > R^{\mathrm{eff}}_{r,\mathrm{MPC}}$ y negativo en caso contrario.

Pero sigue siendo una estimación sesgada. Se destaca esto, para no crear una expectativa que el resto del capítulo tendría que desmentir: $\hat{d}$ no entrega $R_{r,\mathrm{MAS}}$ fielmente. Sustituyendo (5.13) en (5.15),

$$\hat{d} = \left(R_{r,\mathrm{MAS}} - R^{\mathrm{eff}}_{r,\mathrm{MPC}}\right) + \frac{\langle \boldsymbol{\eta}, \boldsymbol{\varphi} \rangle}{\langle \boldsymbol{\varphi}, \boldsymbol{\varphi} \rangle} : \tag{5.16}$$

la proyección entrega el desajuste más la parte de $\boldsymbol{\eta}$ alineada con la sensibilidad. Ese segundo término es la **desviación**, y depende del **estado**, en particular del flujo de rotor estimado con la resistencia equivocada. La proyección **no puede eliminar**la por mucho que se promedie, porque $\boldsymbol{\eta}$ no es un ruido independiente de $\boldsymbol{\varphi}$. En las maniobras simuladas, donde el desajuste real se conoce, el sesgo se puede medir: aparece con el desajuste y desaparece con el modelo ajustado (Tabla 5.3 y Sección G.5). En línea, en cambio, el desajuste no se conoce: cuantificar ese sesgo y quitarlo es la tarea que se encomienda a las redes, y de ahí parte la Sección 5.2.

Y cómo se delimita el ciclo: La frontera se detecta con un criterio tipo osciloscopio (un *trigger*) sobre la corriente $i_{s\alpha}$, con **histéresis**: el detector se arma cuando la corriente desciende por debajo de $-h$ y dispara al superar $+h$, de modo que un ciclo va de un disparo al siguiente.

El umbral h se fija como una **fracción del pico del ciclo anterior** (10%), y esa referencia se elige por descarte de las dos alternativas naturales. Un tiempo mínimo entre disparos exigiría una constante temporal fija, que fallaría durante el arranque porque allí el período varía por un factor grande. Y definir la ventana a partir de la frecuencia que el propio controlador calcula sería *circular*: ω_s contiene la frecuencia de deslizamiento, que se deduce de $R^{\text{eff}}_{r,\text{MPC}}$, de modo que la ventana de análisis dependería del parámetro que se pretende estimar. Referir la histéresis a la amplitud medida no incurre en ninguno de los dos problemas y se autoescala con el punto de operación.

5.2. Consideraciones de diseño: salidas, expertos por maniobra y entradas

Establecido el núcleo físico, y sabiendo que la capa correctiva actúa sobre un único número, $R^{\text{eff}}_{r,\text{MPC}}$, esta sección construye el sistema que produce esa corrección. Las redes aprenden, fuera de línea, a transformar la estimación física sesgada y el contexto medido en una **propuesta** de corrección; los coordinadores deciden si esa propuesta es admisible dentro de cada maniobra, y el guardián de escritura decide si el candidato resultante puede copiarse al estado persistente del controlador.

5.2.1. Las salidas: corrección, confianza y dominio

La capa correctiva no reemplaza al controlador ni aprende una ley de control: conserva íntegra la estructura del PTC y corrige en línea un solo número, el parámetro efectivo de rotor con que el predictor trabaja,

$$R^{\text{eff}}_{r,\text{MPC}}[k] = R_{r,\text{MPC},0}\, k_{rr}[k], \tag{5.17}$$

donde $R_{r,\text{MPC},0}$ es el valor base ingresado por el usuario y $k_{rr}[k]$ el factor de corrección que la capa entrega.

Por tanto, el punto de partida de cómo se construye ese k_{rr} son las salidas que se le exigen a cada red, porque una red puede entrenarse para entregar prácticamente cualquier magnitud: lo que fija el diseño es decidir primero qué se le pide.

Las cuatro redes comparten una misma estructura (Subsección 5.3.1) y se distinguen por sus datos, sus etiquetas y sus pesos; las dos redes de un mismo experto comparten los datos y solo difieren en las etiquetas. Cada red posee tres salidas, que la Tabla 5.1 resume y los apartados siguientes desarrollan.

Tabla 5.1: Las tres salidas de cada red, su acotación y la etiqueta con que se entrena cada una. Sólo la primera propone un valor de resistencia; las otras dos condicionan que esa propuesta llegue a ejecutarse.

Salida	**Forma**	**Rango**	**Etiqueta de entrenamiento**	**Uso en línea**
s, corrección relativa	$S_{\text{out}} \tanh(z)$	±1,6 en arranque; ±0,7 en carga	$s^\star = \log_{10}\big(R_{r,\text{MAS}}/R^{\text{eff}}_{r,\text{MPC}}\big)$, evaluada sólo en los ciclos en que a esa red le toca actuar	Fija el candidato $\hat{R}_r = R^{\text{eff}}_{r,\text{MPC}} \cdot 10^{\,s}$
conf, confianza	$\sigma(z)$	[0; 1]	$\exp(-\lvert s - s^\star\rvert/0{,}05)$: el error que la propia red está cometiendo en ese ciclo	Condición de disparo: exige conf $\geq$ 0,30, y 0,50 en modo reforzado
dom, dominio	$\sigma(z)$	[0; 1]	Binaria: conjunción de sentido del desajuste, magnitud dentro del rango y régimen informativo	Enruta y habilita: exige dom $\geq$ 0,50, y 0,70 en modo reforzado

1. **Corrección:**

$$s = \log_{10}\left(\frac{\hat{R}_r}{R^{\text{eff}}_{r,\text{MPC}}}\right) = S_{\text{out}} \tanh(z_s), \qquad \hat{R}_r = R^{\text{eff}}_{r,\text{MPC}} \cdot 10^{\,s}, \tag{5.18}$$

   El motivo de utilizar una función logarítmica es que un mismo número expresa siempre el mismo error relativo, y **el daño que produce un desajuste depende de la razón entre la resistencia del predictor inicial y la real**. La magnitud $\hat{R}_r$ es, por tanto, un candidato reconstruido a partir de la salida neuronal; no es una medición de la resistencia de la máquina.

   El límite S_{out} de la tangente hiperbólica se determinó a partir del rango de trabajo: como tanh es asintótica, la salida *nunca alcanza* su cota, de modo que S_{out} debe fijarse con holgura sobre **el mayor desajuste** que se pretende expresar. Los valores desplegados son $S_{\text{out}} = 1{,}6$ para las redes de arranque, lo que cubre razones de hasta $40\times$, y $S_{\text{out}} = 0{,}7$ para las de carga, coherente con su dominio acotado.

2. **Dominio**: La salida dom $\in [0, 1]$ y declara que el ciclo observado pertenece a la maniobra donde esta red puede trabajar. Pero, no expresa una condición sino la conjunción de **tres** (Subsección 5.2.1.a): que el desajuste vaya en *su* sentido, que su magnitud caiga dentro del rango que la red fue entrenada a cubrir, y que el régimen sea *informativo*, esto es, que no se trate de la operación en vacío y en estado estacionario, donde según la Subsección 5.1.2 la evidencia sobre R_r no está presente en las señales medibles.

3. **Confianza**: cuánto se equivoca la red, dicho por ella misma. La salida conf $\in [0, 1]$ no mide la semejanza del ciclo presente con los del conjunto de entrenamiento. Mide algo más directo y más útil para decidir: se entrena contra el *error que la propia red está cometiendo* en ese ciclo, mediante el objetivo

$$\text{conf}^\star = \exp\left(-\frac{\lvert s - s^\star\rvert}{c}\right), \qquad c = 0{,}05, \tag{5.19}$$

   donde $s^\star$ es el desajuste verdadero (5.20). La escala c fija qué error se considera ya poco fiable: un desvío de 0,05 en log-razón (un 12 % en la resistencia) reduce

la confianza objetivo al 37 %. Es decir, la red aprende así a anticipar su propia exactitud, y no a reportar una probabilidad genérica.

La diferencia del **dominio** y **confianza** es operativa: el **dominio** delimita una región del espacio de operación, y es la misma para todos los ciclos que caen en ella; y la **confianza** varía ciclo a ciclo dentro de esa región, según lo limpio que venga el residuo, lo asentada que esté la maniobra y lo lejos que quede el punto de operación de los mejor representados en el entrenamiento. Los coordinadores utilizan la confianza como **condición de disparo**: una salida con dominio pleno pero confianza por debajo del umbral no se ejecuta, se espera al ciclo siguiente. Es lo que permite que el sistema sea selectivo, decidiendo donde la evidencia es buena y paciente donde no lo es.

5.2.1.a. Determinación del dominio: etiquetas, rangos y estados

Para que las redes aprendieran a responder si están o no en su dominio las mediciones del residuo, se crearon **etiquetas** fuera de línea a partir del desajuste verdadero en log-razón

$$s^\star = \log_{10}\left(\frac{R_{r,\mathrm{MAS}}}{R^{\mathrm{eff}}_{r,\mathrm{MPC}}}\right), \tag{5.20}$$

Sobre $s^\star$ se definen dos límites, que usan las dos redes de arranque:

1. Límite inferior $S_{\mathrm{mín}} = \log_{10}(1{,}10) = 0{,}0414$, que corresponde a un desajuste relativo del 10 % y separa el caso ajustado del desajustado.
2. Límite superior $S_{\mathrm{máx}}$ del dominio de acción, fijado en 1,30 para el arranque, una razón de 20× (exactamente, $10^{1{,}30} = 19{,}95\times$), que cubre con margen los desajustes de la campaña nominal, de hasta 11,8×. En el conjunto de arranque, sin embargo, la mayor razón con ejemplos de actuación es 13,3× (4,00 frente a 0,30 Ω); la siguiente presente, 20× (6,00 frente a 0,30 Ω), ya queda fuera de dominio.

Las dos redes de carga usan umbrales *absolutos* en lugar de relativos, coherentes con el dominio de corrección de su coordinador: el caso se considera ajustado si $|R_{r,\mathrm{MAS}} - R^{\mathrm{eff}}_{r,\mathrm{MPC}}| \le 0{,}15\ \Omega$ y fuera de dominio si esa diferencia supera los 3 Ω. Las etiquetas que siguen se definen para ellas igual, con esa diferencia en lugar de s y esos umbrales en lugar de $S_{\mathrm{mín}}$ y $S_{\mathrm{máx}}$.

Para el sentido de desajuste: $\sigma \in \{+1, -1\}$ y con $s = \sigma\, s^\star$, las cuatro etiquetas son

$$\texttt{act} = \mathbf{1}\big[S_{\mathrm{mín}} \le s \le S_{\mathrm{máx}}\big], \qquad \texttt{ood} = \mathbf{0}\big[s > S_{\mathrm{máx}}\big], \tag{5.21}$$

$$\texttt{wrong} = \mathbf{0}\big[s < -S_{\mathrm{mín}}\big], \qquad \texttt{matched} = \mathbf{0}\big[|s^\star| \le S_{\mathrm{mín}}\big]. \tag{5.22}$$

Es decir: cada red ve *todos* los ciclos del conjunto, pero cada uno etiquetado desde su propio punto de vista: sí le corresponde actuar, pertenece al sentido contrario, o el modelo ya está bien. Así aprende simultáneamente a actuar en su régimen y a no actuar fuera de él.

Y existe una quinta etiqueta **adjunta**, que **no depende del desajuste** sino de la **región ciega**: los ciclos en vacío, definidos por

$$\texttt{ciega} = \mathbf{1}\big[\text{maniobra} = \text{vacío}\big] \cdot \mathbf{1}\big[t \ge t_{\text{min,estacionario}}\big]. \tag{5.23}$$

Nótese que es una condición que exige *ambas* cosas: durante el transitorio de un arranque en vacío sí hay señal alineada con la sensibilidad, se midió $\rho^2 = 0{,}998$ en la rama sobreestimada, y por eso el transitorio no se declara ciego.

El objetivo que se entrena como salida de dominio no es `act` sino

$$\texttt{dom} = \texttt{act} \wedge \overline{\texttt{ciega}}, \tag{5.24}$$

esto es: *actúa donde te corresponde, salvo donde nadie puede saber.*

5.2.2. Expertos independientes por maniobra

La caracterización del Capítulo 4 mostró que la observabilidad del desajuste de R_r depende fuertemente de la maniobra. En consecuencia, la solución se organiza como una arquitectura de **expertos independientes por maniobra**: arranque y operación bajo carga.

1. **El arranque es un cambio de estado.** La máquina parte del reposo y debe alcanzar una referencia de velocidad, de modo que la maniobra exhibe un fenómeno que ninguna otra presenta: *el período de la corriente se acorta ciclo a ciclo* a medida que la velocidad sube. Cada ciclo se observa, por tanto, en un punto de operación distinto del anterior, y la propia secuencia de duraciones es información sobre lo que está ocurriendo. Es además el instante en que el desajuste puede ser mayor: el modelo arranca con el valor que el usuario le dio, que es justamente el que se desconoce y el punto de partida de toda rutina: si la máquina no llega a girar, no hay maniobra posterior que afinar. De ahí el criterio de diseño del experto de arranque: se entrena sobre un **rango de desajuste amplio**, aceptando a cambio menor precisión, porque su cometido es dejar la máquina andando y entregar el modelo dentro de una rango donde otro pueda trabajar.

2. **La carga es una perturbación, no un cambio de estado.** Al entrar un escalón de torque de carga la velocidad no debe cambiar: el objetivo del lazo es *mantenerla.* La máquina ya gira, el período de la corriente es estable y lo que se observa es cómo el accionamiento rechaza una perturbación y con cuánto esfuerzo lo consigue. El régimen está asentado, el residuo es de bajo ruido y aparece además un observable que durante el arranque no existe: el error de regulación de velocidad. El experto de carga se entrena, en consecuencia, sobre un **rango acotado en torno al valor vigente** y se le exige **precisión** y devolver el accionamiento a sus condiciones nominales.

De esa diferencia se sigue que los dos expertos no necesitan comparar sus salidas neuronales para decidir cuál estimación es mejor: la maniobra determina cuál posee competencia de inferencia. Esto no elimina, sin embargo, el conflicto de escritura. Cada coordinador conserva un candidato propio y ambos comparten un único estado k_{rr}; por ello, la sucesión entre escritores se resuelve más adelante mediante el guardián de la Subsección 5.2.6.

Tabla 5.2: Los dos expertos y sus cuatro redes direccionales: papel, dominio y criterio de diseño.

Redes	**Maniobra**	**Dominio**	**Papel**
`arr_subir` `arr_bajar`	Arranque	Hasta 10 Ω	**Ajuste grueso**: dejar la máquina andando y entregar el modelo dentro de una vecindad segura del valor real
`carga_subir` `carga_bajar`	Operación bajo carga	±3 Ω absolutos	**Ajuste fino**: reajustar la degradación lenta (del orden de +0,5 Ω) en torno al valor vigente

5.2.3. La asimetría de observabilidad: por qué se dividen dos redes por maniobra

El segundo nivel de partición: dos redes por maniobra, una por sentido del desajuste. Y esto proviene de una limitación de **observabilidad** del estimador, en el sentido práctico definido en la Subsección 2.5.4.a: cuánta información sobre R_r traen las señales en cada maniobra.

Un indicador de esa limitación, que la acompaña pero no la demuestra, es la alineación del residuo de predicción de corriente con su sensibilidad. Sea $\boldsymbol{\varepsilon} = \boldsymbol{i}_s - \hat{\boldsymbol{i}}_s$ el error del predictor a un paso y $\boldsymbol{\varphi} = \partial \hat{\boldsymbol{i}}_s / \partial R_{r,\mathrm{MPC}}$ su sensibilidad analítica respecto de la resistencia. Y siendo el residuo $\boldsymbol{\varepsilon} \approx \boldsymbol{\varphi}\, \Delta R_r$, el residuo acumulado sobre un ciclo de corriente es una forma **cuadrática** de la corrección δ,

$$R(\delta) = A\,\delta^2 - 2B\,\delta + C, \qquad A = \sum \boldsymbol{\varphi} \cdot \boldsymbol{\varphi}, \quad B = \sum \boldsymbol{\varepsilon} \cdot \boldsymbol{\varphi}, \quad C = \sum \boldsymbol{\varepsilon} \cdot \boldsymbol{\varepsilon}, \tag{5.25}$$

cuyo mínimo $\delta^* = B/A$ es la estimación física del error de resistencia. Y el cociente:

$$\rho^2 = \frac{B^2}{A\,C} \in [0, 1] \tag{5.26}$$

es el coeficiente de determinación de ese ajuste de un solo regresor: mide qué fracción de la energía del residuo queda en la dirección de la sensibilidad, que es la dirección en que actuaría un error de R_r. Es un indicador de **alineación**, o de calidad del ajuste, calculable en línea con las mismas sumas que ya requiere el estimador. No es una demostración de observabilidad ni de identificabilidad, y un valor alto no certifica por sí solo dos cosas:

- **Que R_r sea la única causa del residuo.** El término $\boldsymbol{\eta}$ de la ecuación (5.13) también puede estar alineado con $\boldsymbol{\varphi}$, y ρ^2 no lo separa: un residuo casi perfectamente alineado puede deberse en parte al estado y no solo a la resistencia (Sección G.5).
- **Que haya excitación suficiente.** A es la energía de la sensibilidad: mide cuánto cambiaría la corriente predicha por un error de resistencia. Cuando la maniobra excita poco la resistencia, A es pequeña, y el sesgo de la ecuación (5.16), que divide por $\langle \boldsymbol{\varphi}, \boldsymbol{\varphi} \rangle = A$, crece. ρ^2 no lo advierte, porque mide la dirección del residuo y no el tamaño de la sensibilidad: si $\boldsymbol{\varphi}$ se multiplica por un factor c, B se multiplica por c y A por c^2, y $\rho^2 = B^2/(A\,C)$ queda igual. Un ρ^2 alto puede, entonces, acompañar a una estimación muy sesgada (Sección G.5).

Tabla 5.3: Asimetría de observabilidad entre los dos sentidos del desajuste, medida sobre el barrido factorial piloto (12 848 ciclos). Las medianas de la Tabla 5.4 proceden de otro conjunto y no son directamente comparables con estas.

Indicador	**Ajustado**	**Subestimado**	**Sobreestimado**
$\rho^2 = B^2/(A\,C)$	0,004	**0,032**	**0,979**
Ganancia $\delta^\star$/gap	—	0,77	2,32
$\psi_s^\star/\psi_{nom}$ con carga	0,92	**1,0000**	0,89
$\lvert i_{sq}\rvert/i_{sq,\text{máx}}$	0,30	**0,099**	0,60
$T^\star$ del PI (p.u.)	0,99	**2,00 = tope**	1,53

El indicador ρ^2 mide qué fracción de la acumulación del residuo queda alineada con la sensibilidad; vale 0,979 sobreestimando y apenas 0,032 subestimando. La diferencia es de más de un orden de magnitud y tiene una explicación física encadenada, que es la que sostiene la asimetría; ρ^2 la refleja, pero no la prueba por sí solo, que la caracterización del Capítulo 4 desarrolla y que aquí conviene recordar en su forma breve: subestimar R_r hunde la frecuencia de deslizamiento estimada, lo que hunde ω_s, lo que eleva la cota de flujo $\psi_{lim} = v_{lim}/\omega_s$ hasta que el campo *nunca se debilita*; el flujo excedente satura la tensión disponible, la corriente de cuadratura colapsa al 9,9 % de su máximo, y como la sensibilidad

$$\boldsymbol{\varphi} = \frac{T_s}{L_\sigma}\left(-k_r^2\,\mathbf{i}_s + \frac{k_r}{L_r}\,\hat{\boldsymbol{\psi}}_r\right) \tag{5.27}$$

se **anula** cuando la corriente es puramente magnetizante, el residuo pierde precisamente la **sensibilidad** que lo hacía informativo.

El fallo por subestimación borra la evidencia de su propia existencia.

De esta constatación se siguen dos decisiones que estructuran todo el diseño. Primero, **partir por sentido y no por magnitud**: no se trata de dos escalas del mismo problema sino de dos problemas de identificación con canales de información distintos: el sobreestimado se resuelve con el residuo, cuya ganancia 2,32 es aprendible; el subestimado debe apoyarse en los *armónicos* de i_α. Segundo, **poder declarar ignorancia**, porque la propia tabla exhibe una región ($\rho^2 = 0{,}004$ en ajustado, y valores comparables en vacío) donde ninguna red debería pronunciarse.

5.2.4. El vector de entrada de los expertos

Un ciclo aislado es una fotografía instantánea; para decidir bien interesa también la *tendencia*, en particular después de una corrección. El diseño de los conjuntos de datos y de los cuatro encabezados neuronales se definió por ello sobre un vector canónico que combina el ciclo recién cerrado, el ciclo anterior, su diferencia y estadísticos de historia. La dimensión y el orden de este vector son comunes; la Subsección 5.4.2.b documenta por separado una diferencia de valores en la construcción desplegada para el experto de carga.

5.2.4.a. El vector canónico de 141 componentes

A las 27 características base (Subsección 5.2.4.b) se agregan: el valor del ciclo anterior (27), la diferencia entre el ciclo actual y el anterior (27), y agregados multi-período: media

y desviación estándar de las diez características más discriminantes sobre ventanas de los últimos 2, 5 y 10 ciclos. El vector resulta

$$\underbrace{27}_{\text{base}} + \underbrace{27}_{\text{previo}} + \underbrace{27}_{\text{diferencia}} + \underbrace{10 \times 3 \times 2}_{\text{caract.} \times \text{ventanas} \times \{\mu,\sigma\}} = 141, \tag{5.28}$$

de modo que la arquitectura de cada red recibe $\mathbf{x} \in \mathbb{R}^{141}$. Las diez características con memoria histórica son `d_hat`, `rms_eps_norm`, `thd`, `torque_resid`, `neg_seq`, `psi_ref_norm`, `err_flujo`, `isq_frac`, `wreg` y `rho2`: cinco del grupo del residuo y su firma, y cinco del contexto y la consecuencia.

Cada *grupo* cumple un papel distinto en la decisión: el ciclo actual describe el estado recién observado; el anterior aporta una referencia inmediata; y la diferencia expone la dirección del cambio, clave para distinguir una desviación persistente de un transitorio. Las medias históricas reducen la variabilidad y revelan persistencia en tres horizontes, y las desviaciones históricas miden la estabilidad temporal de la evidencia, que no debe confundirse con la salida de confianza de las redes. Todas las características son calculables en línea y se implementan en **C** con acumuladores incrementales y una memoria circular para las ventanas multi-período.

La unificación conserva una consecuencia práctica: el bloque de extracción de las 27 características y la memoria circular de los agregados multi-período son únicos y se actualizan una vez por ciclo cerrado. Los cuatro encabezados reciben vectores de 141 posiciones con el mismo orden, aunque la asignación desplegada de los bloques *previo* y *diferencia* no es idéntica entre expertos, como se explicita en la Subsección 5.4.2.b. En la generación de los datos de entrenamiento se presentó, además, un detalle de implementación, que se explica en esa misma subsección y en la Sección G.1.

5.2.4.b. Las 27 características base

Para cada ciclo válido se calculan 27 características base, comunes a los dos expertos.

El conjunto no se organiza por la naturaleza de la señal sino por el **papel** que cada característica cumple en la decisión. Se distinguen tres grupos:

1. **El residuo y su calidad** (5). Cuantifican el error de predicción del modelo y hasta qué punto ese error es atribuible a la resistencia. Es el canal que resuelve la rama sobreestimada.
2. **La firma del sentido** (12). Estado del lazo de control y forma de onda de la corriente. Determina hacia qué lado está el error.
3. **Contexto y consecuencia** (10). Nivel de carga, seguimiento de velocidad y escala temporal. Delimitan la región donde existe información y, en la rama subestimada, aportan la magnitud que el residuo pierde.

Una de las características del tercer grupo requiere definición previa, porque no es una magnitud que el controlador **ya calcule** sino un **observable construido** para este fin: el **error de regulación de velocidad**. Conviene distinguir dos magnitudes asociadas: el error físico, en radianes eléctricos por segundo,

$$e_\omega = \omega_{\text{ref}}\, p - \overline{\omega}_{el} \qquad [\text{rad/s}], \tag{5.29}$$

y su versión normalizada: la característica que consumen efectivamente las redes y los coordinadores, adimensional,

$$e_{\text{reg}} = \frac{e_\omega}{\Omega_{\text{nom},el}}, \tag{5.30}$$

donde $\Omega_{\text{nom},el}$ es la velocidad eléctrica nominal.

El error de regulación (5.30) es la traducción físicamente válida de la misma idea, mide la *consecuencia* real del desajuste sobre una variable medida.

La Tabla 5.4 las reúne. Las columnas de valor típico corresponden a las medianas medidas en régimen sobre las cuatro rutinas de caracterización del Capítulo 4 y tres resistencias nominales distintas.

Tabla 5.4: Las 27 características base. Medianas típicas sobre 80 corridas, población distinta de la de la Tabla 5.3.

Característica	**Definición (por ciclo)**	**Ajust.**	**Subest.**	**Sobreest.**
Grupo 1 — El residuo y su calidad				
`d_hat`	Proyección $\hat{d} = B/A$, ec. (5.15), en Ω	−0,022	+2,99	−4,39
`rms_eps_norm`	RMS del residuo sobre RMS de la corriente	0,0038	0,046	0,031
`rho2`	$\rho^2 = B^2/(A\,C)$: fracción del residuo alineada con la sensibilidad	0,0014	0,054	0,988
`phys_C`	$C = \sum\langle \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}\rangle$: energía cruda del residuo	—	—	—
`rr_eff`	$R^{\text{eff}}_{r,\text{MPC}}$ vigente: el punto desde el que se corrige	—	—	—
Grupo 2 — La firma del sentido de desajuste				
`psir_mean`	Magnitud media del flujo de rotor estimado [Wb]	0,852	0,963	0,817
`psi_ref_norm`	$\psi_s^\star/\psi_{s,nom}$: la consigna de flujo que el control se fija	0,920	**1,0000**	0,904
`w_slip`	Frecuencia de deslizamiento estimada [rad/s]	25,2	2,3	70,1
`isq_frac`	$\lvert i_{sq}\rvert/i_{sq,\text{máx}}$: fracción de corriente de par usada	0,275	0,083	0,569
`err_flujo`	$\psi_s^\star - \hat{\psi}_s$ [Wb]: seguimiento de la consigna de flujo	−0,002	−0,022	−0,005
`thd`	$\sqrt{\sum_{h=2}^{10} I_h^2}/I_1$ (DFT del ciclo)	0,036	**0,149**	0,012
`h3`, `h5`	Armónicos 3.º y 5.º normalizados a I_1	0,010	0,006	0,003
`h7`	Armónico 7.º normalizado a I_1	0,010	**0,146**	0,0045
`neg_seq`	Razón de secuencia negativa I_{-1}/I_1	0,008	0,008	0,003
`test_ripple`	Desviación estándar de $\hat{T}_e$ en el ciclo	0,041	0,131	0,036
`idx_switch`	Fracción de cambios del índice de vector óptimo	—	—	—
Grupo 3 — Contexto y consecuencia				
`tref_mean`	$\overline{T}^\star/T_{nom}$: par que pide el PI de velocidad	0,586	**2,00**	0,842
`test_mean`	$\overline{T}_{\text{est}}/T_{nom}$: par que el MPC cree entregar	0,593	0,426	0,951
`torque_resid`	$(\overline{T}_{\text{est}} - \frac{J}{p}\dot{\omega}_{el})/T_{nom}$	0,505	0,352	0,819
`wreg`	Error de regulación normalizado e_{reg}, ec. (5.30)	0,0074	0,099	0,0090
`rms_i`, `i_peak`	Valor eficaz y pico del vector de corriente [A]	15,1	9,5	20,5
`I1`	Amplitud de la fundamental (DFT del ciclo)	—	—	—
`cyc_nsamp`	Muestras del ciclo: duración, y peso estadístico de $\hat{d}$	—	—	—
`w_ref`	Referencia de velocidad normalizada a la nominal	—	—	—
`tpa`	$\overline{T}^\star/\text{rms}_i$: par por amperio	—	—	—

Tres comentarios sobre el conjunto.

1. Primero, todos los armónicos se normalizan respecto de la fundamental *del propio ciclo*: como durante el arranque I_1 cambia con la velocidad, esa normalización hace comparable la métrica de distorsión entre ciclos de amplitudes muy distintas.

2. Segundo, ρ^2 cumple un papel distinto del resto: no informa sobre el *valor* del desajuste sino sobre la *fiabilidad de la propia evidencia*. Es el coeficiente de determinación del ajuste por mínimos cuadrados que produce $\hat{d}$, de modo que un valor alto indica que el residuo observado está alineado con la sensibilidad, como lo estaría si lo produjera un error de resistencia, y uno bajo, que no lo está. Un valor alto es condición necesaria, no suficiente: no descarta otras causas alineadas ni garantiza excitación (Subsección 5.2.3). Entregarla como entrada permite que la red module su confianza según la calidad del canal, y el mismo indicador se reutiliza aguas abajo como veedor de los coordinadores (Subsección 5.2.5.b). Es, en sentido estricto, una característica sobre la validez de las demás.

3. Tercero, la Tabla 5.4 deja ver de un vistazo la asimetría que estructura todo el diseño. La consigna de flujo vale 1,0000 *exacto* subestimando, mientras el torque de referencia se clava en su tope de $2\,T_{nom}$: el lazo de velocidad pide todo lo que puede y no alcanza. Sobreestimando ocurre lo contrario, $\hat{T}_e > T^\star$, y ρ^2 sube a 0,988. El caso ajustado se distingue de ambos por una sola cantidad, `rms_eps_norm`, un orden de magnitud menor.

5.2.4.c. Ajuste de la entrada por sentido de giro

La máquina de inducción es simétrica respecto del sentido de giro: invertir la marcha no altera la dinámica del accionamiento, sino únicamente el signo de las magnitudes que poseen dirección y el sentido en que recorre su trayectoria el vector espacial de corriente. Un accionamiento que sigue una referencia de -1440 r/min con la misma carga se comporta igual que el que sigue $+1440$ r/min; lo que cambia es el sistema desde el cual se lo describe, no lo que ocurre en la máquina.

El diseño aprovecha esa simetría: **los expertos se definen sobre un único sentido de giro, el positivo, y las características se les presentan siempre referidas a él**. Con ello se busca que un mismo par de redes cubra los dos sentidos de marcha, sin duplicar el conjunto de entrenamiento sobre condiciones que son físicamente la misma y sin requerir un experto adicional para la marcha inversa. La referencia al sentido canónico se realiza mediante el signo de la referencia de velocidad,

$$\sigma_\omega = \operatorname{sgn}(\omega^\star) \in \{-1,\, +1\}, \tag{5.31}$$

y toma dos formas distintas según cómo se manifieste la dirección en cada característica.

Magnitudes con signo. Siete características cambian de signo con el sentido de marcha porque describen cantidades dirigidas: el torque de referencia y el estimado, el residuo de torque, el error de regulación de velocidad, la propia referencia de velocidad, el torque por unidad de corriente y el deslizamiento. Se refieren al sentido canónico multiplicándolas por σ_ω. La fracción de corriente de eje q no pertenece a este grupo: se define con el valor

absoluto $|i_{sq}|$ (Tabla 5.4), no depende del sentido de giro y se entrega tal cual, igual que en el entrenamiento.

Contenido armónico. Al invertir el sentido, el vector espacial de corriente recorre su trayectoria en el sentido opuesto, lo que equivale a conjugarlo en el plano $\alpha\beta$. Bajo esa operación las secuencias positiva y negativa **intercambian sus papeles**: la componente que en marcha directa es la fundamental pasa a ocupar el lugar de la secuencia negativa, y a la inversa. No se trata, por tanto, de un cambio de signo. Como la magnitud del armónico de orden h de una trayectoria conjugada coincide con la del orden $-h$ de la original,

$$\big|X(-h)\big| = \big|X(h)\big|, \tag{5.32}$$

basta referir el *orden* armónico al sentido canónico y evaluar en $\sigma_\omega h$. Con ello la fundamental, la distorsión armónica, los armónicos individuales y el contenido de secuencia negativa conservan su significado en ambos sentidos de marcha.

Magnitudes invariantes. Las catorce restantes no requieren transformación alguna, porque describen cantidades sin dirección: resistencias, magnitudes eficaces, contenidos de flujo y cocientes adimensionales. La Tabla 5.5 clasifica las veintisiete según su definición.

Tabla 5.5: Clasificación de las veintisiete características base según cómo se refieren al sentido de giro canónico. Las invariantes se entregan tal cual; las del grupo por signo se multiplican por σ_ω.

Clase	**Características**
Invariantes (14)	`rr_eff`, `d_hat`, `rms_eps_norm`, `phys_C`, `i_peak`, `test_ripple`, `psir_mean`, `idx_switch`, `cyc_nsamp`, `rms_i`, `psi_ref_norm`, `err_flujo`, `rho2`, `isq_frac`
Por signo (7)	`tref_mean`, `test_mean`, `torque_resid`, `wreg`, `w_ref`, `tpa`, `w_slip`
Por orden armónico (6)	`I1`, `thd`, `h3`, `h5`, `h7`, `neg_seq`

La salida no requiere transformación. La corrección que las redes entregan es el cociente logarítmico de (5.18), una razón entre resistencias, que carece de dirección; otro tanto vale para la confianza y para el dominio. Para las redes, la simetría se resuelve por completo en la entrada; para los coordinadores, que deciden cuándo se consulta a las redes, no se resolvió.

Dos precisiones de implementación cierran el mecanismo. El signo de (5.31) se determina con una histéresis del 10 % de la velocidad nominal, de modo que no conmuta en las proximidades del reposo. Y un cambio de sentido reinicializa la memoria de los agregados multi-período, porque dos de las diez características con historia (`torque_resid` y `wreg`) pertenecen al grupo con signo, y sus ventanas no deben promediar ciclos referidos a sentidos distintos.

Alcance. Como el ajuste se aplica a las entradas de las redes y no a los coordinadores, cuyas comparaciones de progreso, rearme y firma suponen el sentido positivo, esta propuesta queda validada para la marcha en ese sentido. Desde reposo con referencia negativa las redes no llegan a actuar y la vía de *breakaway* interpreta la marcha como un rotor

trancado. Tras un arranque en sentido positivo, la red de carga sigue actuando después de invertir. La Sección G.3 reúne esas comprobaciones. Toda la campaña del Capítulo 7 gira en sentido positivo, donde $\sigma_\omega = 1$ y el ajuste no modifica nada.

5.2.5. Estructura preliminar: cuatro redes y dos coordinadores

Las dos subsecciones anteriores fijan la partición: **por maniobra**, porque el desajuste se manifiesta de forma distinta al arrancar que al recibir carga; y **por sentido**, porque subestimar y sobreestimar son dos problemas de identificación con canales de información distintos. El producto de ambas particiones da la estructura definitiva: **cuatro redes**, dos por experto.

Experto	**Red**	**Rama que corrige**	**Canal dominante**
Arranque	`exp_subir`	modelo subestimado	armónicos de i_α
Arranque	`exp_bajar`	modelo sobreestimado	residuo, vía $\hat{d}$
Carga	`carga_subir`	modelo subestimado	armónicos y error de regulación
Carga	`carga_bajar`	modelo sobreestimado	residuo, vía $\hat{d}$

En cada ciclo, las dos redes de un experto publican sus salidas de manera simultánea. Esto plantea un problema que ninguna de ellas puede resolver por sí sola: **¿cuál de las dos propuestas corresponde admitir dentro de esa maniobra?**

Ese es el papel de los *coordinadores*, uno por experto: algoritmos de estados deterministas, no redes, que arbitran entre las dos ramas direccionales de su propia maniobra. La conversión posterior de los dos candidatos de coordinador en una única escritura sobre k_{rr} pertenece al guardián de escritura y no a este arbitraje de inferencia.

El reparto de responsabilidades es deliberado y se sostiene en todo el capítulo: **las redes proponen, los coordinadores admiten y el guardián autoriza la escritura**. Una red entrega su corrección, su confianza y su declaración de dominio; el coordinador decide si la propuesta se acepta, con qué magnitud y en qué momento, y el coordinador de carga puede además revertir su propio candidato. Ninguna red ni coordinador escribe por sí solo directamente sobre $R^{\text{eff}}_{r,\text{MPC}}$.

Sobre esa base, cada coordinador resuelve tres cuestiones:

1. **Seguridad**: cuál de sus dos redes es competente en el ciclo observado, que se resuelve por la declaración de dominio y no por el signo de la corrección propuesta.
2. **Admisión**: si la propuesta reúne las condiciones para ejecutarse: confianza suficiente, persistencia a lo largo de varios ciclos y ausencia de veto por pérdida de regulación.
3. **Construcción del candidato**: qué fracción de la propuesta se aplica, con qué límites y qué estado interno conserva el coordinador mientras progresa la maniobra.

La **sucesión entre expertos** se resuelve en otra capa: el guardián de escritura de la Subsección 5.2.6. Mantener separados ambos niveles evita llamar "arbitraje" tanto a la selección de una red como a la exclusión mutua de dos escritores, que son decisiones distintas.

5.2.5.a. Diferencias: maniobra de carga vs el arranque

La maniobra de carga impone más requisitos que explican diseño posterior:

- **Repetición.** El arranque es un episodio único; los escalones de carga pueden repetirse en una misma corrida. El experto debe poder *rearmarse* tras cada evento y acumular correcciones sucesivas.
- **Detección indirecta del inicio.** No existe una medición en línea del torque de carga real; el inicio de la maniobra se infiere de variables sustitutas: el aumento del valor eficaz de la corriente o del torque de referencia que exige el lazo de velocidad.
- **Régimen asentado en vez de transitorio.** La carga se observa en régimen estacionario: el residuo es de bajo ruido, pero su sesgo depende del punto de operación y del nivel de carga, no del estado transitorio.

El núcleo físico, en cambio, es idéntico: el mismo residuo (5.8), la misma sensibilidad analítica (5.10)–(5.11) y la misma proyección (5.15), calculados por el mismo código validado.

5.2.5.b. Coordinador de arranque y algoritmo físico: breakaway

El coordinador de arranque se ejecuta **una vez por ciclo de corriente cerrado** y su tarea es convertir dos propuestas simultáneas en, a lo sumo, una acción sobre el parámetro.

Qué recibe. Por un lado, las **seis salidas** de sus dos redes: la tripleta (s, conf, dom) de la rama subestimada y la de la sobreestimada. Por otro, el **contexto físico** de la maniobra, que no proviene de red alguna: el progreso de velocidad $\omega_m/\omega_m^\star$ y la etapa derivada de él (el código usa máx($\omega_m^\star$, 10^{-6}) como denominador, de modo que el progreso solo está definido para referencias positivas; Sección G.3), las banderas de magnetización y de arranque activo, el torque de referencia normalizado, la duración real del ciclo, y las medianas móviles de cinco ciclos de $\hat{d}$, del residuo normalizado $\|\boldsymbol{\varepsilon}\|/\|\mathbf{i}\|$ y de ρ^2.

Qué entrega. Un único candidato de $R_{r,\mathrm{MPC}}^{\mathrm{eff}}$ y un veredicto que indica de dónde salió: sin acción, aplicación de red, paso físico de *breakaway*, abstención declarada o modo seguro. Nada más: el coordinador no escribe sobre el controlador, propone.

Cómo relaciona lo uno con lo otro. La cadena tiene tres filtros en serie, y el orden importa porque cada uno puede detener a los siguientes.

1. **Primero la física, y sin consultar a las redes.** Se forman dos juicios con las medianas móviles. El primero decide *si hay desajuste*: si el residuo normalizado no supera 0,0054 el caso se declara ajustado y no se toca nada. El segundo decide *si el residuo respalda*: según ρ^2 supere o no 0,50, el sistema exige a las redes límites normales o reforzados. La razón de este orden es la condición de rechazo: ninguna red puede actuar sobre un modelo que la física declara ya ajustado, por segura que se muestre.
2. **Después, propuesta de las redes.** Una red participa si su dominio y su confianza superan los umbrales vigentes y su propuesta apunta en su propio sentido. Si participan dos redes, dirige la **firma**, que deduce el sentido del desajuste de la física

medida: $\psi_s^\star/\psi_{s,nom} \geq 0{,}999$ indica subestimación y $\hat{T}_e > T^\star$, sobreestimación. Si la firma contradice a la red enrutada, el contador de confirmaciones se reinicia y no se actúa.

3. **Por último la confirmación.** La propuesta debe sostenerse durante los ciclos exigidos (límites) antes de que la corrección se aplique. Es lo que impide que un ciclo atípico mueva el modelo.

Los dos tipos de límites son la traducción directa de esa lógica: cuando el canal del residuo no respalda la estimación, la red debe mostrarse más conservadora durante más tiempo.

Tabla 5.6: Parámetros del coordinador de arranque. La columna reforzada rige cuando $\rho^2 \leq 0{,}50$, esto es, cuando el residuo no respalda la estimación de la red.

Condición	**Normal**	**Reforzada**
Dominio mínimo	0,50	0,70
Confianza mínima	0,30	0,50
Ciclos de confirmación	2	5
Fracción del paso aplicado	1,0	0,5
Comunes		
Residuo bajo el cual se declara ajustado	$\tau_\varepsilon = 0{,}0054$	
Torque bajo el cual se declara ciego	$\lvert T^\star\rvert/T_{nom} < 0{,}12$	
Paso máximo por aplicación	$\lvert s\rvert \leq 0{,}15$ (factor 1,41)	
Aplicaciones máximas por maniobra	10	
Saturación del parámetro	[0,1; 10] Ω	

Confirmada la firma, la corrección se aplica en log-razón sobre el valor vigente,

$$s^{\text{apl}} = \text{sat}_{\pm 0{,}15}\big[\,0{,}85\;\phi\;s\,\big], \qquad R^{\text{eff}}_{r,\text{MPC}} \leftarrow \text{sat}_{[0{,}1;\,10]}\big(R^{\text{eff}}_{r,\text{MPC}} \cdot 10^{\,s^{\text{apl}}}\big), \tag{5.33}$$

con $\phi = 1$ si el residuo respalda y $\phi = 0{,}5$ si no, y con un factor de amortiguación 0,85 que evita aplicar la propuesta completa de una vez. La corrección se descarta si cae bajo la banda muerta

$$\Delta_{\text{dead}} = \text{máx}\big(0{,}05\,R^{\text{eff}}_{r,\text{MPC}};\;0{,}05\;\Omega\big), \tag{5.34}$$

esto es, el mayor entre un 5 % del valor vigente y un piso absoluto. El criterio es *relativo* porque la resolución del estimador lo es: medida como tres desviaciones estándar de la estimación sobre el caso ajustado, vale un 1,5 % constante del valor vigente; el término absoluto no es un criterio de diseño sino un piso de ruido.

Además, si la referencia de velocidad cae bajo 5 rad/s mecánicos y vuelve a subir, el coordinador **se rearma**, porque eso constituye otra maniobra de arranque. Se reinicia la máquina de decisión y se *conserva* $R^{\text{eff}}_{r,\text{MPC}}$, de modo que lo aprendido antes no se pierde.

Algoritmo físico: breakaway. Toda la cadena anterior presupone que la máquina *progresa* durante la maniobra. Hay una situación en que eso no ocurre: el **arranque rotor trancado**, con la máquina magnetizada y girando a velocidad casi nula. La cual produce un bloqueo mutuo: la máquina no arranca sin corrección, y la corrección no llega porque la máquina no arranca. Y la abstención de las redes es *correcta*, por lo cual, esperar a que las redes respondan sería esperar indefinidamente.

La solución está en que el residuo no pierde toda su información: pierde la **magnitud**, pero conserva el **signo**, medido acertado en el 99 % de los ciclos de esa banda. Sobre ese único canal se construye el *breakaway*, que no consulta ninguna salida neuronal.

Cómo se implementó. Cuatro reglas, todas sobre el signo de la mediana de $\hat{d}$:

- **Dos puertas de entrada.** Desde el estado magnetizado, si la máquina lleva cinco ciclos sin declarar arranque activo, o si acumula más de 150 ms en ese estado. Y desde el estado de análisis, si con carga presente el *avance* de la velocidad en cinco ciclos es inferior al 1 % de la referencia. Ambas exigen además $|\mathrm{med}\,\hat{d}| \geq 0{,}10\ \Omega$. Con referencia negativa la primera puerta se abre aunque la máquina gire, porque el arranque activo no se declara nunca (Sección G.3).
- **Acción.** Pasos de 0,5 Ω cada tres ciclos sobre el valor efectivo, en la dirección del signo.
- **Bisección al cambiar de sentido.** Un cambio de signo significa que el valor ya quedó encerrado, de modo que el paso se parte por la mitad, con mínimo de 0,05 Ω.
- **Cesión a las redes.** Tras tres cambios de sentido la vía devuelve el control al análisis normal. El criterio de salida no es haber acertado el valor, sino **haber entrado en el dominio donde las redes pueden hablar**, que es exactamente su objetivo.

5.2.5.c. Coordinador dual de carga y vía de emergencia

El coordinador de carga resuelve el mismo problema que el de arranque: dos propuestas por ciclo, una sola acción; pero sobre una maniobra que se repite, lo que cambia tanto lo que recibe como cuándo se le permite actuar.

Qué recibe. Las seis salidas de sus dos redes y el contexto de la maniobra: valor eficaz de corriente, torque de referencia y velocidad normalizados, aceleración, el error de regulación e_{reg} (5.30) (que en el arranque no existe), la mediana de $\hat{d}$ y el $R^{\mathrm{eff}}_{r,\mathrm{MPC}}$ vigente.

Qué entrega. Igual que el de arranque: un candidato de $R^{\mathrm{eff}}_{r,\mathrm{MPC}}$ y un veredicto. La diferencia está en que su decisión es *por evento* y no por maniobra, porque los escalones de carga pueden repetirse dentro de una misma corrida.

Cómo relaciona lo uno con lo otro. Sobre la máquina se definen estados de trabajo: ARRANQUE → fin de arranque → asentamiento → espera de escalón → transitorio → análisis → aplicación. El paso desde el primer estado exige una velocidad media positiva, de modo que con referencia negativa desde reposo el coordinador no sale de él (Sección G.3); y durante el análisis rigen seis reglas:

1. **Análisis frente a torque.** Si $|T^{\star}|/T_{nom} < 0{,}30$ el coordinador no acumula evidencia. La compuerta se evalúa *antes* de poblar el buffer de análisis, lo que impone de hecho cinco ciclos consecutivos con carga sin necesidad de un contador aparte. El umbral separa las dos poblaciones con holgura: en vacío el torque de referencia normalizado ocupa el intervalo 0,003–0,009 y con carga el 0,975–2,000.
2. **Selección de las dos propuestas.** Sobre medianas de cinco ciclos, una red participa si su dominio $\geq 0{,}50$ y su confianza $\geq 0{,}30$, umbrales que suben a 0,70 y 0,50

cuando $\rho^2 \leq 0{,}50$. Si piden actuar las dos, decide el signo de la mediana de $\hat{d}$ con magnitud mínima de 0,30 Ω; si ninguna, no se actúa. Se exige además coherencia con la firma de rama definida en la Subsección 5.2.5.b, que aquí solo se consulta por encima del 75 % de la velocidad nominal: por debajo de esa banda un campo sin debilitar no distingue la subestimación de una máquina que sencillamente gira despacio, de modo que la firma se abstiene en lugar de vetar sin fundamento.

3. **Análisis continuo, con rearme.** Bastan quince ciclos asentados y con carga para abrir ventana, sin que el torque tenga que cambiar. La detección de escalón —un aumento superior al 15 % de la corriente eficaz o del torque de referencia sobre una línea base de tres ciclos— se conserva, pero su papel es iniciar un evento fresco. Tras cada corrección se dejan pasar veinte ciclos de transitorio antes de volver a juzgar.

4. **Aplicación amortiguada y acotada.** La corrección se aplica en log-razón,

$$s^{\mathrm{apl}} = \mathrm{sat}_{\pm 0{,}10}\big[\,0{,}85\,s\,\big], \qquad R^{\mathrm{eff}}_{r,\mathrm{MPC}} \leftarrow \mathrm{sat}_{[0{,}1;\,10]}\big(R^{\mathrm{eff}}_{r,\mathrm{MPC}} \cdot 10^{\,s^{\mathrm{apl}}}\big), \tag{5.35}$$

   limitando cada aplicación a un factor de 1,26 frente al 1,41 del arranque, coherente con su papel de precisión, con la misma banda muerta (5.34), a lo sumo tres compromisos por evento y veinticinco ciclos de enfriamiento tras cada uno. Su dominio de acción es $|\delta R_r| \leq 3\ \Omega$ absolutos.

5. **Rectificación sobre la tendencia del residuo.** Tras cada corrección se compara el residuo normalizado posterior con el previo, y el veredicto solo vale si la carga no cambió entre ambos (tolerancia del 10 % sobre la corriente eficaz). Si el residuo empeora más de un 10 %, el coordinador restaura su candidato anterior y cierra el evento. La restauración se entrega como una decisión activa, de modo que el controlador vuelve a trabajar con el candidato anterior (Sección G.4).

6. **Convergencia.** Dentro de una maniobra de carga, no se admiten cambios de dirección: un cambio de signo se interpreta como convergencia y lo cierra.

Algoritmo físico: vía emergencia. El coordinador analiza en régimen asentado, condición necesaria para que las redes operen dentro de su distribución de entrenamiento. Esa prudencia tiene un punto ciego simétrico al del arranque: si el desajuste combinado con una carga muy alta, esto impide que la velocidad se asiente, la ventana de análisis nunca se llena, las redes jamás son consultadas y la degradación progresa sin respuesta. La **vía de emergencia** cubre ese hueco y, como el *breakaway*, es una regla física que no consulta a las redes.

Cómo se implementó. La estructura es la del *breakaway* con dos añadidos que la maniobra de carga exige: el *pacing* y la rectificación, porque aquí la corrección surte efecto con la **constante de tiempo de la inercia** y no de inmediato:

- **Disparo por pérdida sostenida de regulación.** Se arma cuando $e_{\mathrm{reg}} > 0{,}08$ durante seis ciclos consecutivos, con carga presente y con el estimador físico confirmando desajuste ($|\mathrm{med}\,\hat{d}| \geq 0{,}30\ \Omega$).

- **Paso relativo y presupuesto acotado.** Cada tres ciclos aplica un paso del 25 % del *menor* entre el valor vigente y el valor base de la resistencia, con piso de 0,05 Ω, en la dirección del signo de la mediana de $\hat{d}$. El total corregido por evento no puede superar 2 Ω, deliberadamente por debajo de los 3 Ω del dominio del experto: una vía que corrige a ciegas no debe poder recorrer todo el rango que la red sí puede juzgar.
- ***Pacing* por recuperación.** Antes de cada paso se comprueba si la regulación ya está mejorando en más de un 5 % respecto del ciclo previo; si mejora, el paso se pospone. Sin esto la vía sobrepasaría, porque seguiría corrigiendo mientras la inercia todavía responde a la corrección anterior.
- **Rectificación.** Si tras corregir el signo de la mediana de $\hat{d}$ se invierte de forma sostenida con la regulación aún degradada, la vía re-engancha en sentido contrario con presupuesto renovado, hasta dos inversiones por episodio.
- **Reinicio ante cambios de referencia.** Un salto de la referencia de velocidad superior a 0,02 normalizado por ciclo reinicia el contador y pospone el paso: un transitorio de referencia no es un evento de carga.
- **Sucesión.** Recuperada la regulación, la vía cede el control como un evento fresco del coordinador normal, cuyas redes refinan ya en régimen asentado.

Solo el ciclo en que se produce un paso efectivo de emergencia entrega una decisión activa hacia la escritura, por la misma línea custodiada que la ruta neuronal normal (Subsección 5.2.6.a).

Que este reparto se cumple en operación no es una suposición del diseño. Las vías físicas son reglas deterministas impuestas por diseño, y cabe preguntarse si terminan sustituyendo a las redes en lugar de apoyarlas. La medición de quién mueve efectivamente el parámetro en cada escenario —pasos de *breakaway*, aplicaciones de la red de arranque y compromisos de la de carga— se presenta como evidencia en la Subsección 7.1.6 del capítulo de resultados.

5.2.6. La escritura del parámetro: guardián

Los dos coordinadores resuelven, cada uno en su maniobra, qué candidato puede proponer. Queda una decisión diferente, que conecta toda la capa con el controlador: **cuál coordinador posee autorización de escritura y cómo se actualiza $R^{\text{eff}}_{r,\text{MPC}}$ dentro del algoritmo del PTC**.

5.2.6.a. Algoritmo de escritura: guardián.

Cada coordinador mantiene **su propio** valor de resistencia como estado interno y lo actualiza con sus reglas: el de arranque conserva un candidato y el de carga otro. El PTC no lee directamente ninguno de ellos; lee **una sola variable**, el factor de corrección persistente k_{rr}, del que se reconstruye el valor efectivo:

$$R^{\text{eff}}_{r,\text{MPC}} = R_{r,\text{MPC},0} \cdot \text{sat}\big(k_{rr}\big). \tag{5.36}$$

El guardián de escritura es, entonces, **la regla que decide cuál de los dos coordinadores puede copiar su valor interno sobre esa variable compartida en cada ciclo**. No selecciona redes ni estima el parámetro. En concreto:

- **No arbitra la inferencia.** La elección entre las dos redes direccionales ocurre dentro de cada coordinador mediante dominio, confianza y contexto físico. El guardián de escritura no vuelve a comparar esas salidas.
- **No verifica si la corrección fue buena.** Esa comprobación existe, pero está *dentro* del coordinador de carga: es la rectificación sobre el residuo de la Subsección 5.2.5.c, que compara el residuo antes y después de corregir y restaura el candidato previo si empeora. El guardián de escritura solo controla la propiedad de la variable compartida. Cuando la rectificación restaura el candidato previo, lo entrega por la misma puerta que sus correcciones.
- **No es un bloque ni una red.** Son dos condiciones de escritura implementadas en el propio `C-Script`, una por experto, evaluadas al final de la ejecución en que se cierra el ciclo de corriente.

El problema que resuelve se ve mejor planteado al revés. Si ambos expertos escribieran cada ciclo se pisarían: el de arranque alcanza su valor y lo *mantiene*, de modo que seguiría republicándolo indefinidamente y borraría el refinamiento posterior del de carga. La regla que lo impide es la **política de escritura por ventana**: cada experto escribe solo mientras está activo en su propia ventana de maniobra, y como el factor es un estado discreto persistente, al dejar de escribir conserva el último valor sin que nadie deba reafirmarlo.

5.2.6.b. Las dos puertas de escritura

La **puerta de arranque** exige que su coordinador no esté en modo seguro y que el experto de carga no haya comprometido todavía ninguna corrección:

```
if (PINN_V2_APPLY > 0.5 && dec2 != AC2_SAFE_D
    && g_k2.committed == 0)   /* carga comiteo -> arranque cede */
    k_rr_arr_state = g_ac2.rr_effect / Rr_MPC_base;
```

Listing 5.1: Puerta de escritura del experto de arranque.

Nótese lo que *no* exige: que se haya seleccionado una red.

La **puerta de carga** es más restrictiva: exige su propia bandera, la de aplicación, y una **decisión activa en ese mismo ciclo**.

```
if (PINN_carga > 0.5 && PINN_V2_APPLY > 0.5 && kdec == K2_ACTIVE)
    k_rr_arr_state = g_k2.rr_effect / Rr_MPC_base;
```

Listing 5.2: Puerta de escritura del experto de carga: exige decisión activa.

La asimetría entre ambas responde a cómo trabaja cada experto: el de arranque publica su estado de forma continua mientras dura su ventana: que en la mayoría de los ciclos eso equivale a reescribir el mismo número; mientras que el de carga actúa por eventos discretos y solo tiene algo nuevo que decir en el ciclo en que compromete una corrección. Por eso mismo, la rama que restaura el candidato previo entrega también una decisión activa, para que la puerta de carga deje pasar la reversión.

Dos consecuencias completan el mecanismo:

1. **La sucesión es permanente y de un solo sentido.** La marca de compromiso del coordinador de carga cierra la puerta de arranque, y no al revés. Se activa tanto cuando la red de carga compromete una corrección como cuando lo hace su vía de emergencia. La justificación es de secuencia: cuando el experto de carga tiene algo que decir, el de arranque ya cumplió su papel, de modo que cederle la pluma no le cuesta nada. Sin esta condición, un arranque cargado sobre una máquina degradada seguiría escribiendo sobre las correcciones de carga.

2. **La ausencia de escritor no restablece nada.** Si ninguna puerta se abre, el valor permanece:

$$k_{rr} \leftarrow \begin{cases} R_{r,\text{coord,carga}}/R_{r,\text{MPC},0}, & \text{si la puerta de carga está abierta,} \\ R_{r,\text{coord,arr}}/R_{r,\text{MPC},0}, & \text{si la puerta de arranque está abierta y carga no ha comprometido,} \\ k_{rr}^{\text{anterior}}, & \text{en cualquier otro caso.} \end{cases} \tag{5.37}$$

5.2.6.c. Cierre del lazo sobre el controlador

La escritura de k_{rr} ocurre hacia el final de la ejecución en que se cierra el ciclo de corriente. Al comienzo de la ejecución *siguiente*, el factor de k_{rr} es aplicado sobre el valor efectivo (5.17) y las constantes derivadas del modelo (5.6).

5.2.7. Estructura final: comparativa de los dos expertos

La Tabla 5.7 resume la comparación entre ambos expertos. El núcleo físico, las 27 características, la dimensión y el orden del vector canónico, la arquitectura, la forma de la función de pérdida y el criterio de partición son comunes; el peso de uno de los términos de silencio, $\mathcal{L}_{\text{dom}0}$, y los umbrales de las etiquetas, no. Las maniobras difieren en el dominio, el conjunto de entrenamiento, los mecanismos de seguridad de sus coordinadores y, en el despliegue evaluado, en el contenido efectivo de los bloques temporales *previo* y *diferencia*.

Tabla 5.7: Comparación entre el experto de arranque y el experto de carga: mismo núcleo físico, distinta maniobra.

Aspecto	**Experto de arranque**	**Experto de carga**
Señal / residuo	Residuo de corriente $\alpha\beta$; proyección $\hat{d}$	El mismo residuo y $\hat{d}$
Sensibilidad $\boldsymbol{\varphi}$	Analítica, ecs. (5.10)–(5.11)	La misma
Entrada desplegada	27 base; $\mathbf{x} \in \mathbb{R}^{141}$ con ciclo previo y diferencia efectivos	Misma dimensión; previo igual al actual y diferencia nula; historia móvil efectiva
Arquitectura y pérdida	$141-64-64$; ec. (5.39)	Las mismas, salvo el peso de $\mathcal{L}_{\text{dom0}}$: 24 en lugar de 48
Criterio de las dos redes	Por **sentido** (subir/bajar)	Por **sentido** (subir/bajar)
Maniobra	Transitorio de partida; se rearma por maniobra	Carga sostenida; análisis continuo
Papel	**Excepción**: dejar la máquina andando	**Precisión**: refinar lo que el arranque dejó
Dominio de corrección	Hasta 10 Ω (razón 20×)	±3 Ω absolutos
Cota de salida S_{out}	1,6	0,7
Datos de entrenamiento	Cinco campañas, 767 corridas, con vacío	Cuatro campañas, 483 corridas, **sin** vacío
Resistencia real en los datos	12 valores, de 0,30 a 10,00 Ω	19 valores, de 0,83 a 6,33 Ω
Compuerta de carga	$\lvert T^{\star}\rvert/T_{nom} \geq 0{,}12$	$\lvert T^{\star}\rvert/T_{nom} \geq 0{,}30$
Modo grueso	*Breakaway* físico por signo del residuo	Vía de emergencia por pérdida de regulación
Limitación conocida	Rama subestimada sin magnitud fiable	Resolución relativa de $\sim 1{,}5\,\%$

5.3. Redes MLP: arquitectura y entrenamiento de las redes PINN

Las cuatro redes del sistema comparten arquitectura, salidas y la forma de la función de pérdida: lo propio de cada experto son los escenarios con que se generó su conjunto de datos, los umbrales de sus etiquetas, el peso de $\mathcal{L}_{\text{dom0}}$, la capacidad discriminante de sus características y sus limitaciones conocidas. Esta sección describe primero una sola vez lo común y entra después en lo particular de cada uno.

5.3.1. Arquitectura y salidas de las cuatro redes

Las cuatro redes del sistema comparten la misma arquitectura, un MLP deliberadamente compacto y se entrenan por separado con sus propias etiquetas sobre el conjunto de ejemplos sobre su maniobra:

$$141 \longrightarrow 64 \xrightarrow{\tanh} 64 \xrightarrow{\tanh} \{\, s,\ \text{conf},\ \text{dom}\,\}, \tag{5.38}$$

con 13 443 parámetros entrenables por red, inicialización de Glorot, sin *weight decay* (que degradó la exactitud en los ensayos) y con recorte de gradiente. El cuerpo es común a las tres cabezas, que son capas lineales independientes sobre la última representación oculta: la de log-razón cierra con $S_{\text{out}}\tanh(\cdot)$ según (5.18), y las de confianza y dominio con función logística.

El entrenamiento supervisa cada salida para que, una vez congelados los pesos, la red entregue por ciclo la propuesta con el significado de la Subsección 5.2.1. La cabeza de log-razón aprende $s^\star$ (5.20) *solo donde le corresponde actuar*; la de dominio aprende la etiqueta `dom` (5.24), que es la que identifica la maniobra; y la de confianza se ajusta al objetivo autocalibrado (5.19).

5.3.2. La función de pérdida

El entrenamiento es **supervisado** en el sentido de la Subsección 2.4.3: sobre los escenarios simulados se calculan las etiquetas de la Subsección 5.2.1.a, y el descenso de gradiente por minilotes ajusta los pesos y sesgos de cada red hasta que sus tres salidas reproducen esas etiquetas. El procedimiento es el estándar; lo que distingue a estas redes es *qué* se coloca dentro de la función de costo.

Cada red minimiza una pérdida de siete términos, agrupables en tres papeles: ajustar donde corresponde, callar donde no, y respetar la física:

$$\mathcal{L} = \underbrace{w_d\,\mathcal{L}_{\text{data}} + w_{\text{dom}}\,\mathcal{L}_{\text{dom}} + w_c\,\mathcal{L}_{\text{conf}}}_{\text{supervisado}} + \underbrace{w_q\,\mathcal{L}_{\text{quiet}} + w_{\text{d0}}\,\mathcal{L}_{\text{dom0}} + w_{\text{c0}}\,\mathcal{L}_{\text{conf0}}}_{\text{silencio}} + \underbrace{w_{\text{dir}}\,\mathcal{L}_{\text{dir}}}_{\text{física}}. \tag{5.39}$$

Los términos de datos, confianza y silencio son errores cuadráticos; la cabeza de dominio, al resolver una decisión binaria, se ajusta por entropía cruzada; y $\mathcal{L}_{\text{dir}}$ es una penalización física. También cambian el subconjunto de ciclos sobre el que se ajustan y el peso con que ingresan. La Tabla 5.8 los reúne.

Tabla 5.8: Los siete términos de la pérdida: qué compara cada uno, sobre qué ciclos se ajustan y con qué peso entra, y el origen de ese peso. El conjunto `fuera` es la unión del caso ajustado, el sentido contrario, el fuera-de-dominio por magnitud y la región ciega, que solo existe en el conjunto de arranque. Cuando el peso difiere entre expertos se indica como arranque/carga.

Término	**Qué compara**	**Ciclos**	**Peso**	**Origen del peso**
Supervisado: ajustar donde corresponde				
$\mathcal{L}_{\text{data}}$	s contra el desajuste verdadero $s^\star$	actuación	1,0	Término principal; fijado al diseñar la red
$\mathcal{L}_{\text{dom}}$	dom contra la etiqueta `dom`	todos	0,5	Fijado al diseñar la red
$\mathcal{L}_{\text{conf}}$	conf contra conf*	actuación	0,1	Fijado al diseñar la red
Silencio: callar donde no				
$\mathcal{L}_{\text{quiet}}$	s contra cero	ajustados	1,5	Fijado al diseñar la red
$\mathcal{L}_{\text{dom0}}$	dom contra cero	`fuera`	**48/24**	Ensayo y error: de 3 a 48 en arranque y de 3 a 36 en carga, una semilla por valor
$\mathcal{L}_{\text{conf0}}$	conf contra cero	`fuera`	0,5	Fijado al diseñar la red
Física: respetar el signo del residuo				
$\mathcal{L}_{\text{dir}}$	$\max(0, -s\ \text{sign}(s^\star))$, ponderado por ρ^2	actuación	0,2	Fijado al diseñar la red

Dos lecturas de la tabla ordenan el diseño. La primera es que **los términos de silencio son la mitad de la función objetivo**: no se le pide a la red que acierte, solo que se

calle. La segunda es el peso de $\mathcal{L}_{\text{dom0}}$, **casi cincuenta veces** el del término de datos en las redes de arranque y la mitad de eso en las de carga: callar fuera de dominio pesa mucho más que acertar dentro. Es el único peso que se ajustó, por ensayo y error y con una semilla por valor, sin una búsqueda sistemática: se conservó en cada experto un valor que no dejaba falsos positivos sobre el modelo ajustado de la partición de prueba. La red de arranque lleva el doble porque, además de los tramos con carga, recorre el vacío y la región ciega, donde abundan los ciclos en que debe callar; la de carga solo recibe ciclos de régimen de carga. Con estos pesos, sobre la partición de prueba, ninguna red propone una corrección sobre el modelo ajustado (0 de 2 026 ciclos en cada red de arranque, 603 de ellos en vacío), a costa de un recall menor, y en la región ciega queda un 2,2 % (43 de 1 928 ciclos). Son cifras fuera de línea, con el vector de entrenamiento (Sección G.1), y sobre la misma partición con que se eligió el peso; la medida independiente es la campaña en lazo cerrado del Capítulo 7.

El término físico y por qué no es el residuo minimizado. La versión anterior incorporaba como término físico la propia forma cuadrática del residuo, $A\,\delta^2 - 2B\,\delta + C$, que se minimiza exactamente en $\delta = B/A = \hat{d}$; empujar la salida de la red hacia ese mínimo parecía la manera natural de informarla con física. Ese término se **eliminó**, y la razón es la constatación más importante de este trabajo respecto del paradigma PINN canónico: *el residuo no se minimiza con el modelo ajustado.* La razón entre el δ que minimiza el residuo y el desajuste real vale 2,32 en la rama sobreestimada y 0,77 en la subestimada (Tabla 5.3). Minimizar el residuo, por tanto, no lleva a $R_{r,\text{MAS}}$. Un término que premiara esa minimización estaría empujando a la red exactamente hacia otro punto de error.

Y el término $\mathcal{L}_{\text{dir}}$ actual que conserva del residuo $\hat{d}$, es ρ^2, pero la dirección correcta empleada durante el entrenamiento proviene de la etiqueta supervisada $s^\star$. Su expresión es

$$\mathcal{L}_{\text{dir}} = \frac{1}{N_{\text{act}}} \sum_{\text{act}} \rho^2 \ \text{máx}\big(0,\ -s \cdot \text{sign}(s^\star)\big), \tag{5.40}$$

que penaliza las propuestas cuyo signo contradice el **signo** verdadero del desajuste, solo sobre ciclos de actuación, y pondera esa penalización por ρ^2. Donde el residuo explica bien lo observado ($\rho^2 \to 1$), la restricción direccional pesa con toda su fuerza; donde no informa ($\rho^2 \to 0$, como ocurre en parte de la rama subestimada y en la región ciega), la restricción se atenúa y la red puede apoyarse en los demás descriptores.

Por lo cual, se trata de un **estimador neuronal híbrido informado por física**, no de una PINN canónica que imponga el residuo de una ecuación diferencial sobre la salida. La supervisión aporta $s^\star$ y su signo; la física aporta el predictor auxiliar, $\hat{d}$ como entrada y ρ^2 como ponderación de la restricción direccional.

5.3.3. Entrenamiento de las redes PINN

Esta sección describe, para cada experto, los escenarios con que se generó su conjunto de datos, cómo se reparte ese conjunto entre entrenamiento, validación y prueba, y qué filas usa cada una de sus dos redes.

5.3.3.a. Entrenamiento del experto de arranque

El conjunto se construye sobre un rango amplio de resistencia real y de resistencia del modelo: la resistencia *real* de la máquina recorre doce valores entre 0,30 y 10,00 Ω, y la resistencia con que se *inicializa* el modelo del controlador, $R_{r,\mathrm{MPC},0}$, toma veintiséis valores distintos en ese mismo rango. Lo que permanece fijo en todo el conjunto son los demás parámetros de la máquina, de modo que el dominio es amplio en el *parámetro* R_r y único en los demás parámetros de *máquina*.

Conviene subrayar que las dos resistencias se barren de forma **independiente**: no se trata de un desajuste fijo aplicado a distintas máquinas, sino del producto de las dos rejillas. De ahí que la razón $R_{r,\mathrm{MPC},0}/R_{r,\mathrm{MAS}}$ cubra desde 0,03 hasta 33 veces, con los dos casos: subestimación y sobreestimación; y el caso ajustado representados a la vez en ocho de los doce niveles de resistencia: 0,30 Ω no tiene subestimados, 10,00 Ω no tiene sobreestimados, y 1,33 y 2,33 Ω solo aparecen ajustados.

El conjunto se construyó en **cinco campañas complementarias** (Tabla 5.9).

Tabla 5.9: Las cinco campañas que componen el conjunto de datos del experto de arranque. Bajo cada total de corridas, entre paréntesis, las de entrenamiento, validación y prueba.

Campaña	**Corr.**	**Ciclos**	$\mathbf{t_{sim}}$	$\mathbf{R_{r,MAS}}$ [Ω]	$\mathbf{R_{r,MPC,0}}$	**Carga y referencia**
Factorial de rango ancho	256 (178/39/39)	30 858	3,0 s	0,30; 0,83; 1,50; 2,50; 4,00; 6,00; 8,00; 10,00	los mismos ocho valores, cruzados con $R_{r,\mathrm{MAS}}$ (8×8)	vacío, cinta al 50 y al 100 %, bomba; referencia nominal
Saltos de $R^{\mathrm{eff}}_{r,\mathrm{MPC}}$ con carga	282 (198/41/43)	48 424	4,0 s	0,83; 1,83; 2,83	0,40; 0,70; 1,50 y 2,50 veces $R_{r,\mathrm{MAS}}$, más un salto a 0,60; 0,85; 1,20 o 1,80 veces el valor vigente, en $t = 2{,}0$ o 2,9 s	cinta al 50 y al 100 %, bomba; referencia nominal
Saltos en vacío	94 (61/18/15)	17 375	4,0 s	0,83; 1,83; 2,83	la misma rejilla de la campaña anterior	vacío; referencia nominal
Referencia de velocidad variable	90 (61/16/13)	11 666	3,0 s	0,83; 2,83	0,36; 1,00 y 2,50 veces $R_{r,\mathrm{MAS}}$	vacío, cinta al 50 %, bomba; referencia de 0,40; 0,70; 1,00; 1,20 y 1,40 de la nominal
Ajustados a referencia variable	45 (38/1/6)	6 795	3,0 s	1,33; 1,83; 2,33	$= R_{r,\mathrm{MAS}}$ (modelo ajustado)	las mismas tres cargas y las mismas cinco referencias
Total	**767** (536/115/116)	**115 118**				

De las 767 corridas resultan 115 118 ciclos, que se reparten en 58 943 de la rama sobreestimada, 39 563 de la subestimada y 16 612 con el modelo ajustado, que son los que enseñan a no actuar. La partición se hace por corrida completa y no por ciclo, para que la red no pueda memorizar una corrida: con semilla fija y proporciones de 70, 15 y 15 %, quedan 536 corridas de entrenamiento (80 237 ciclos), 115 de validación (17 907) y 116

de prueba (16 974). Solo las de entrenamiento ajustan los pesos y la normalización de las entradas; el entrenamiento dura 400 épocas fijas, y la prueba produce las métricas fuera de línea. Las dos redes recorren las mismas filas y se distinguen por sus etiquetas: `arr_subir` recibe corrección supervisada en 29 107 filas y `arr_bajar` en 37 318, y se les enseña a no reclamar el caso en 55 720 y 46 593, respectivamente; las dos cifras se solapan en la región ciega.

La **fase de saltos** es la que resolvió el problema más severo del entrenamiento. Con R_r fijo por corrida, todo ciclo del conjunto describe una máquina cuyo modelo nunca ha sido corregido; pero en operación la red observa, la mayor parte del tiempo, estados que *siguen* a una corrección previa del coordinador.

La **tercera campaña** corrige un efecto colateral de la segunda, y es un ejemplo de por qué conviene medir la composición del conjunto y no solo su tamaño: como la fase de saltos no contenía ninguna corrida en vacío, al integrarla la proporción de ciclos ciegos cayó del 9,7 al 3,8 %, y el entrenamiento resultante mejoraba con carga y empeoraba en vacío. Repitiendo la campaña con perfil en vacío, la región ciega volvió al 12,3 % y la regresión desapareció. Las dos campañas de referencia variable no alteran esa proporción: en el conjunto final la región ciega sigue siendo el 12,3 % de los ciclos.

Las **dos últimas campañas** emplean ciclos de **debilitamiento de campo**, la red aprende a distinguir un flujo reducido *legítimo* de uno reducido por error de modelo. La cuarta campaña barre la referencia entre 0,40 y 1,40 de la nominal, ya que por encima de 1,05 la velocidad síncrona supera la nominal y el control entra en campo debilitado.

La **quinta campaña** es, a su vez, la corrección de la cuarta, y repite el mecanismo de la tercera. Al incorporar la referencia variable, esos puntos de operación nuevos quedaron representados por treinta corridas ajustadas contra sesenta desajustadas, y la red aprendió que *esa región* era anómala: los falsos positivos sobre el modelo ajustado, que eran nulos, subieron al 0,653 %, y todos ellos provenían de la referencia de 0,70 con cinta al 50 %. La respuesta fue **añadir ejemplos ajustados en los puntos que faltaban**: tres resistencias reales adicionales, con el modelo ajustado, sobre las mismas cinco referencias y las mismas tres cargas.

5.3.3.b. Entrenamiento del experto de carga

La generación de escenarios de carga combina dos familias:

- **Escenarios de R_r fijo** en estado de carga constante, con combinaciones de R_r y nivel de carga, y casos desajustados por dirección, incluyendo casos ajustados en todos los niveles de carga y casos fuera de dominio (desajustes mayores que 3 Ω) para calibrar el bloqueo.
- **Escenarios de estado degradado**, pensados en origen para la red `carga_subir`, la de la dirección más exigente, aunque, como todo el conjunto, los recorren las dos redes: rutinas de *múltiples escalones* de torque con arranques en vacío y arranques ya cargados, y escenarios de *salto post-corrección* en que, con las redes desactivadas, se fuerza a $R_{r,\mathrm{MPC}}^{\mathrm{eff}}$ a saltar de un valor a otro en medio de una carga sostenida. Los ciclos posteriores al salto son, por construcción, los estados "post-corrección" que la red observará en lazo cerrado real cuando el coordinador ya haya aplicado

una corrección previa; sin estos escenarios, la red solo vería estados limpios que no representan su operación real.

El conjunto resultante se detalla en la Tabla 5.10. Comparte con el experto de arranque los saltos de $R^{\text{eff}}_{r,\text{MPC}}$ y las dos campañas de referencia variable, estas sin sus corridas en vacío. Su campaña propia son las **rutinas de escalones de carga**, que barren **desviaciones** absolutas entre el modelo y la máquina en lugar de razones, y esa es la diferencia que separa a los dos expertos. El conjunto no contiene corridas en vacío: solo los ciclos que el filtro temporal clasifica como régimen de carga, incluidos 168 que tocan la parada final de las rutinas.

Tabla 5.10: Las cuatro campañas que componen el conjunto de datos del experto de carga. Cuando $R_{r,\text{MPC},0}$ se expresa como factor, este multiplica a la resistencia real de cada corrida; en las rutinas, el modelo y la máquina comparten un valor base y uno de los dos se desplaza en la desviación indicada. Bajo cada total de corridas, entre paréntesis, las de entrenamiento, validación y prueba.

Campaña	**Corr.**	**Ciclos**	$\mathbf{t}_{\text{sim}}$	$\mathbf{R}_{\mathbf{r},\text{MAS}}$ [Ω]	$\mathbf{R}_{\mathbf{r},\text{MPC},\mathbf{0}}$	**Carga y referencia**
Rutinas de escalones de carga	111 (78/15/18)	14 817	4,9 s	18 valores entre 0,83 y 6,33: las bases 0,83; 1,83; 2,83 y 3,83; esas bases más 0,30; 0,80; 1,50 o 2,50; y 4,83, solo ajustado	la base cuando el modelo está subestimado, y la base más 0,30; 0,80; 1,50 o 2,50 Ω cuando está sobreestimado; más cinco casos ajustados	tres perfiles de carga, descritos en la Tabla 5.11; referencia de velocidad nominal; en los últimos 0,2 s, parada con la cinta descargada
Saltos de $R^{\text{eff}}_{r,\text{MPC}}$ con carga	282 (197/40/45)	32 743	4,0 s	0,83; 1,83; 2,83	x0,40; x0,70; x1,50 y x2,50 veces $R_{r,\text{MAS}}$, más un salto a x0,60; x0,85; x1,20 o x1,80 veces el valor vigente, en $t = 2{,}0$ o 2,9 s	cinta al 50 y al 100 %, bomba; referencia nominal
Referencia de velocidad variable	60 (41/10/9)	5 193	3,0 s	0,83; 2,83	x0,36; x1,00 y x2,50 veces $R_{r,\text{MAS}}$	cinta al 50 %, bomba; referencia de 0,40; 0,70; 1,00; 1,20 y 1,40 de la velocidad nominal
Ajustados a referencia variable	30 (22/7/1)	3 246	3,0 s	1,33; 1,83; 2,33	$= R_{r,\text{MAS}}$ (modelo ajustado)	las mismas dos cargas y las mismas cinco referencias
Total	**483** (338/72/73)	**55 999**				

La partición sigue el mismo criterio, sorteada por separado: 338 corridas de entrenamiento (39 280 ciclos), 72 de validación (8 403) y 73 de prueba (8 316), de modo que una corrida compartida puede caer en particiones distintas para cada experto. `carga_subir` recibe corrección supervisada en 13 529 filas y `carga_bajar` en 15 666.

Las maniobras de la primera campaña son las que dan su nombre al experto, y conviene detallarlas porque de ellas depende qué eventos aprende a distinguir. Se ensayaron tres perfiles (Tabla 5.11).

Tabla 5.11: Los tres curvas de escalones de carga con que se generó la campaña. Cada corrida de la campaña de resistencias se simuló con los tres.

Perfil	**Carga en cada tramo** $[\times T_{\text{nom}}]$	**Qué aporta**
Creciente desde vacío	$0 \to 0{,}45 \to 0{,}75 \to 0{,}95 \to 1{,}00$	Cuatro escalones ascendentes tras un arranque en vacío: el caso de referencia del experto
Alternante desde vacío	$0 \to 0{,}60 \to 0{,}90 \to 0{,}60 \to 0{,}90$	La carga también *baja*. Es el evento que el coordinador debe saber distinguir de una deriva del parámetro, porque los dos reducen la corriente
Creciente arrancando cargado	$0{,}45 \to 0{,}45 \to 0{,}75 \to 0{,}95 \to 1{,}00$	Sin tramo en vacío: el arranque ocurre ya bajo carga, y el experto recibe la máquina desde una condición que no vio nacer

5.4. Implementación en línea en la plataforma PLECS

Esta sección explica el flujo de trabajo real del sistema dentro del bloque `C-Script` de la plataforma.

5.4.1. Ubicación en el lazo del PTC y despliegue en C

Las redes neuronales están implementadas dentro de la propia simulación del MPC, pero **al final del ciclo de trabajo del control predictivo**. En cada período de muestreo, el PTC ejecuta primero su secuencia completa, y solo entonces, con la decisión de control ya tomada, la capa correctiva de expertos procesa las mediciones en línea de ese período.

5.4.2. Flujo 1: mediciones de un ciclo hasta el primer vector de entrada

El primer flujo de trabajo recolecta las mediciones dentro de cada ciclo de corriente y, sobre la base de ciclos válidos consecutivos, ensambla el vector de entrada que ambos expertos comparten. La Figura 5.1 recorre el camino completo: de la memoria del ciclo al vector, del vector a los dos expertos y de sus árbitros al guardián de escritura.

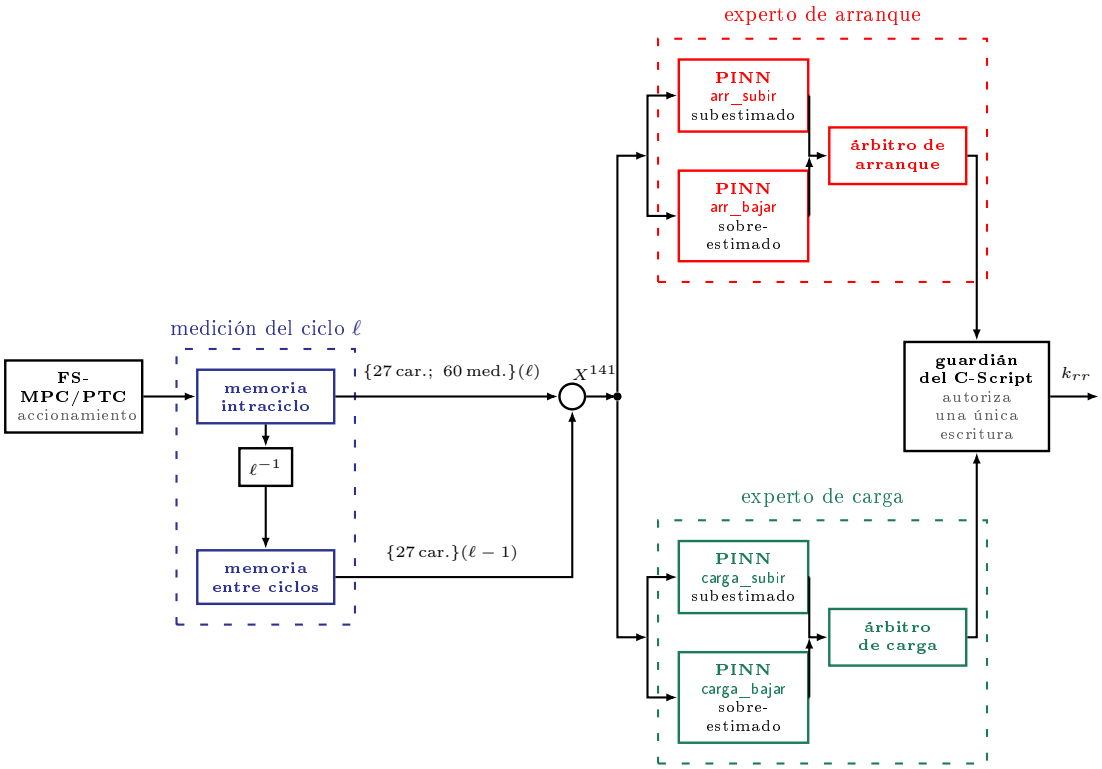


Figura 5.1: Flujo 1. De la medición de un ciclo válido al vector de entrada común, y de éste a los dos expertos duales y al guardián de escritura del `C-Script`.

5.4.2.a. Memoria de un ciclo válido

1. **Periodo de conmutación:** En cada ejecución del `C-Script` (una por período $T_s = 100\ \mu$s), la ruta dispone de las mediciones físicas (i_a, i_b, i_c, ω_m), de la referencia de velocidad y de variables internas del controlador: la tensión aplicada en $\alpha\beta$, el flujo de rotor estimado, el torque de referencia que entrega el PI de velocidad, el torque estimado, el índice del vector óptimo, y los valores $R^{\text{eff}}_{r,\text{MPC}}$ y $R_{r,\text{MPC},0}$, etc.

   Con esas señales se evalúa el predictor auxiliar (5.4)–(5.5) usando el $R^{\text{eff}}_{r,\text{MPC}}$ vigente; la predicción se retiene una muestra y, al llegar la medición siguiente, se forman el residuo $\boldsymbol{\varepsilon}[k]$ (5.7) y la sensibilidad $\boldsymbol{\varphi}[k-1]$ (5.10)–(5.11), con la alineación temporal ya discutida en la Subsección 5.1.1, salvo el ángulo del flujo de la sensibilidad, que va un paso adelantado (Sección G.2).

2. **Acumulación intraciclo:** Durante el ciclo no se guardan todas las señales: casi todas se resumen a medida que llegan las muestras, y solo la corriente se guarda completa.

   - **Lo que se resume muestra a muestra.** De cada señal se lleva una suma, un máximo o un contador, y al cerrar el ciclo esas cuentas dan los promedios, los valores RMS, los picos, las conmutaciones y la duración del ciclo. Del mismo modo se acumulan tres sumas del residuo y la sensibilidad, $A = \langle \boldsymbol{\varphi}, \boldsymbol{\varphi} \rangle$, $B =$

$\langle \boldsymbol{\varepsilon}, \boldsymbol{\varphi} \rangle$ y $C = \langle \boldsymbol{\varepsilon}, \boldsymbol{\varepsilon} \rangle$, con las que al cierre se calculan la proyección $\hat{d} = B/A$ (5.15) y el veedor $\rho^2 = B^2/(A\,C)$.

- **Lo que se guarda completo.** Las dos componentes de la corriente, $i_{s\alpha}$ e $i_{s\beta}$, se guardan muestra a muestra en un *buffer* de hasta 2 304 muestras. Al cerrar el ciclo, una transformada discreta de Fourier, sin ventana, del vector espacial $i_{s\alpha} + j\, i_{s\beta}$ entrega la fundamental y los armónicos de la Tabla 5.4: los de orden $+2$ a $+10$, que giran en el mismo sentido que la fundamental, y la secuencia negativa, que es el de orden -1.

Si un ciclo es más largo que el *buffer*. Las 2 304 muestras equivalen a 0,23 s, es decir, a un ciclo de corriente de 4,34 Hz. Un ciclo más largo llena el *buffer*, y sus muestras siguientes ya no se guardan: los armónicos salen de las primeras 2 304, que no son un ciclo completo, mientras todo lo que se resume muestra a muestra sigue usando el ciclo entero. En los 141 222 ciclos de los conjuntos de datos de los expertos no ocurre nunca: el más largo tiene 2 154 muestras. La campaña no registró la duración de sus ciclos.

3. **Cierre de un ciclo válido:** La ruta permanece activa solo si la máquina está magnetizada, ha transcurrido el tiempo inicial de asentamiento del flujo ($t > 0{,}10$ s) y la amplitud del vector de corriente supera 2 A; si alguna condición se pierde, el ciclo parcial se descarta.

   La frontera entre ciclos es el cruce ascendente por cero de $i_{s\alpha}$ resuelto con el **disparador de Schmitt** de la Subsección 5.1.2. El ciclo que se cierra se acepta solo si contiene al menos 25 muestras. Entonces se calculan sus características y se ejecuta la secuencia de actualización temporal descrita en la Subsección 5.4.2.b; después se reinician los acumuladores y la muestra que detectó el cruce se incorpora como primera muestra del ciclo nuevo.

Al cierre se evalúa el observable físico del desajuste en su forma acumulada, proyección $\hat{d}$ (5.15), y se forman las 27 características base de la Tabla 5.4, comunes a los cuatro expertos.

5.4.2.b. Memorias entre ciclos y construcción de los vectores de entrada

Se denota con ℓ el índice del ciclo válido recién cerrado (el índice k queda reservado a instantes).

La implementación mantiene dos recursos compartidos: un vector `fv_prev` destinado a conservar las 27 características del ciclo anterior y una memoria circular de diez ciclos para las diez características clave enumeradas en la Subsección 5.2.4.a. No se almacena un registro completo de todos los ciclos.

La composición canónica de la ecuación (5.28) contiene $\boldsymbol{f}[\ell]$, $\boldsymbol{f}[\ell - 1]$, su diferencia $\Delta\boldsymbol{f}[\ell]$ y la media y desviación estándar de cada característica clave sobre ventanas de 2, 5 y 10 ciclos. El ciclo recién cerrado ingresa a la memoria circular *antes* de calcular los agregados, de modo que estos incluyen al ciclo actual; mientras no se acumulan ciclos suficientes, las ventanas usan los m ciclos disponibles, sin relleno artificial.

En el despliegue definitivo, sin embargo, el orden de instrucciones produce dos contenidos temporales distintos (Listado B.4). Primero se inserta $\boldsymbol{f}[\ell]$ en la memoria circular. Si

ya existe un ciclo previo, el vector de arranque se construye *antes* de sobrescribir `fv_prev`; por tanto, sus bloques contienen efectivamente $\boldsymbol{f}[\ell]$, $\boldsymbol{f}[\ell-1]$ y $\boldsymbol{f}[\ell]-\boldsymbol{f}[\ell-1]$. A continuación, el programa copia $\boldsymbol{f}[\ell]$ sobre `fv_prev` y recién entonces construye el vector de carga. En consecuencia, para las dos redes de carga desplegadas,

$$\boldsymbol{f}_{\text{prev,carga}}[\ell] = \boldsymbol{f}[\ell], \qquad \Delta\boldsymbol{f}_{\text{carga}}[\ell] = \mathbf{0},$$

aunque sus 60 agregados multi-período sí contienen historia efectiva y se calculan sobre la misma memoria circular que los del arranque.

Esta secuencia también cambia la condición de partida: las redes de arranque se evalúan desde el **segundo ciclo válido**, cuando existe un ciclo anterior real; las redes de carga se evalúan desde el **primer ciclo válido**, porque la bandera de memoria se activa antes de construir su vector. La dimensión sigue siendo 141 en las cuatro redes, pero no es correcto afirmar que las cuatro reciben la misma historia temporal.

Relación con el vector de entrenamiento. Los datos de entrenamiento se generaron con un extractor en Python que forma el bloque previo y la diferencia con el ciclo anterior real y lee las variables internas del controlador con un período de atraso. Por eso el vector que reciben las redes en línea difiere del de entrenamiento en algunas componentes (Sección G.1). Esa diferencia no altera los resultados del Capítulo 7, que se midieron con el vector tal como lo construye el `C-Script`; lo que acota es el alcance de las métricas fuera de línea de las redes, calculadas con el vector de entrenamiento.

5.4.3. Flujo 2: trabajo de las redes neuronales

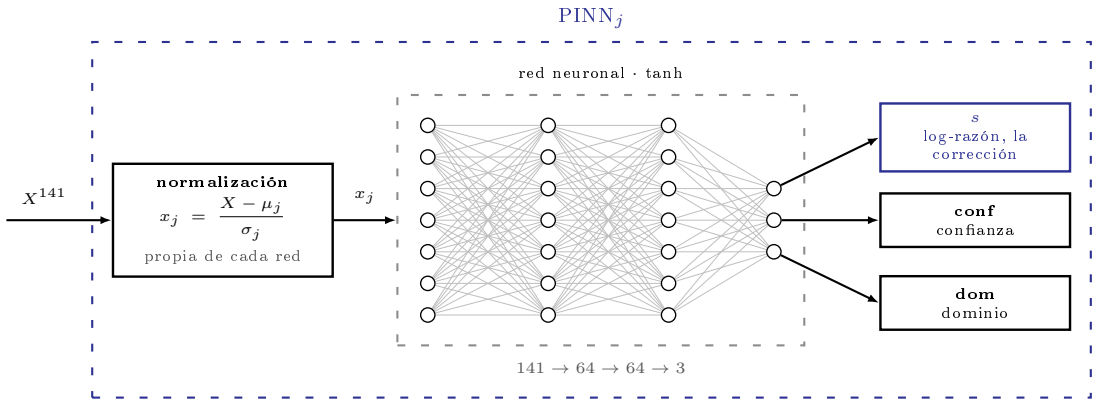


Figura 5.2: Flujo 2. Esquema de trabajo de una MLP común: inferencia sobre el vector de 141 componentes, y las tres salidas (s, conf, dom) como propuesta hacia los coordinadores.

El segundo flujo recibe los vectores construidos por el flujo anterior y ejecuta la inferencia (Figura 5.2).

1. Cada encabezado neuronal aplica primero su normalización $x_{n,j} = (x_j - \mu_j)/\sigma_j$, donde $\boldsymbol{\mu}$ y $\boldsymbol{\sigma}$ son parámetros fijados en el entrenamiento. Las dos redes de arranque comparten un normalizador y las dos de carga comparten otro, porque sus datos se normalizaron en común dentro de cada experto.

2. Sigue el paso hacia adelante de (5.38): dos capas ocultas de 64 neuronas con activación tanh y tres cabezas lineales de salida (Subsección 5.2.1), que se cierran con

$$s = S_{\text{out}} \tanh(z_s), \qquad \text{conf} = \sigma(z_c), \qquad \text{dom} = \sigma(z_d), \tag{5.41}$$

El candidato en ohm, $\hat{R}_r = R^{\text{eff}}_{r,\text{MPC}} \cdot 10^{\,s}$, y la corrección implícita $\delta R_r = \hat{R}_r - R^{\text{eff}}_{r,\text{MPC}}$ se calculan fuera de la red. Con la salida en s, el candidato se construye siempre como un múltiplo del valor vigente, y el limite de corrección en $[0{,}1; 10]\ \Omega$ queda como responsabilidad de los coordinadores.

5.4.4. Flujo 3: capa correctiva online

El tercer flujo somete las cuatro propuestas neuronales a los dos coordinadores deterministas, cuyo diseño se desarrolló en las Subsecciones 5.2.5.b y 5.2.5.c, y a las guardas de escritura de la Subsección 5.2.6.

5.4.5. Cierre del lazo sobre el controlador

Tras el tercer flujo, cada coordinador entrega su salida a las guardas de escritura definidas en la Subsección 5.2.6.

5.4.5.a. Sintesis: secuencia completa de la operación en línea

La secuencia completa puede resumirse así:

1. medir las corrientes y la velocidad, y guardar el contexto del control interno del PTC;
2. predecir la corriente del paso siguiente con el predictor auxiliar y retener la predicción una muestra;
3. al llegar la medición, formar el residuo $\boldsymbol{\varepsilon}$ y la sensibilidad $\boldsymbol{\varphi}$, y acumular la evidencia del ciclo en curso;
4. en cada disparo del *trigger* sobre $i_{s\alpha}$, cerrar el ciclo valido, y calcular sus características;
5. incorporar el ciclo actual a la memoria circular, construir el vector de arranque con el ciclo previo efectivo cuando esté disponible, actualizar `fv_prev` y construir después el vector de carga con previo igual al actual y diferencia nula;
6. ejecutar las redes habilitadas para ese ciclo e interpretar sus tres salidas (s, conf, dom) como propuestas;
7. enviar las propuestas a los coordinadores y, cuando corresponde, utilizar las vías físicas de respaldo;
8. evaluar las dos guardas finales de escritura y conservar o actualizar el factor de corrección k_{rr};

9. en la ejecución siguiente, reconstruir $R^{\text{eff}}_{r,\text{MPC}}$, recalcular las constantes del modelo y cerrar así el lazo de corrección sobre el PTC.

La Figura 5.3 sitúa esa secuencia dentro del accionamiento: en negro el control predictivo tal como se describió en el Capítulo 3, en rojo la capa correctiva y las mediciones que consume, y en azul el único camino por el que la capa toca al controlador, la resistencia efectiva del modelo interno.

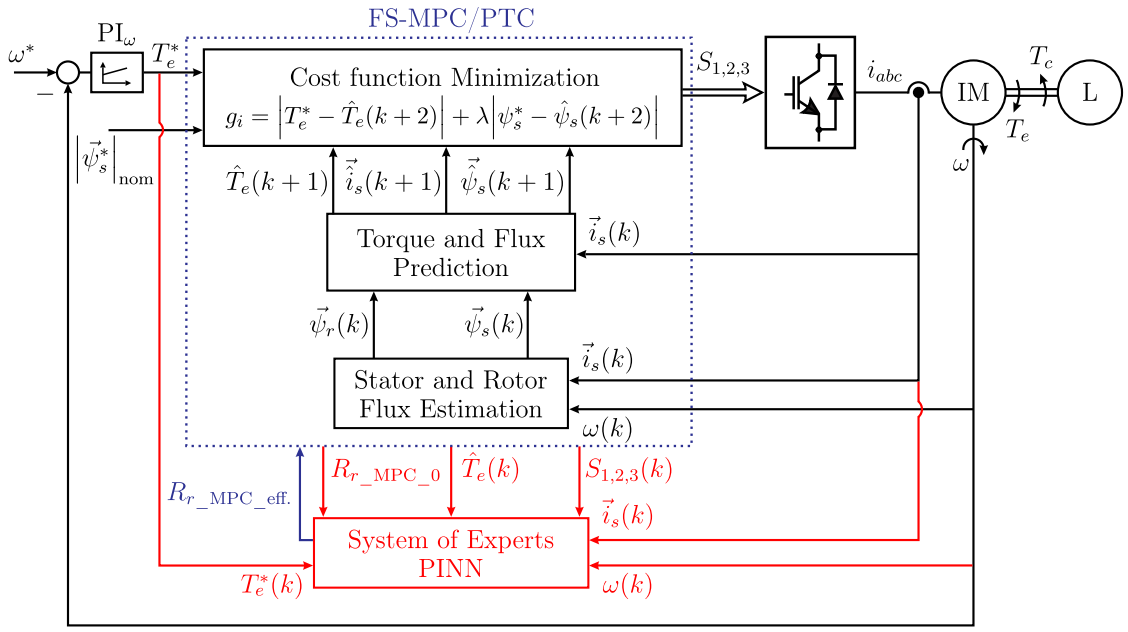


Figura 5.3: Esquema de control completo con la propuesta 1.

En síntesis, las cuatro redes no miden la resistencia real de la máquina: interpretan evidencia física condensada y proponen una resistencia objetivo. Los coordinadores, las vías físicas de respaldo y el guardián de escritura deciden después si el modelo interno del controlador puede cambiar. Con ello queda cerrada la primera propuesta.

5.4.6. Configuración final de entrenamiento y ejecución

La Tabla 5.12 reúne en un solo lugar la configuración con que se entrenaron las cuatro redes desplegadas y que usan las corridas del Capítulo 7. Los valores se leyeron de los programas de entrenamiento y de la configuración guardada junto con los pesos; las redes de carga se reentrenaron con esa configuración y resultaron idénticas bit a bit a las desplegadas.

Tabla 5.12: Configuración final de las cuatro redes de la Propuesta 1.

Aspecto	**Experto de arranque**	**Experto de carga**
Redes	`exp_subir`, `exp_bajar`	`carga_subir`, `carga_bajar`
Arquitectura	Perceptrón $141 \to 64 \to 64$ con tangente hiperbólica y tres cabezas: $s = s_{\text{máx}} \tanh(\cdot)$, conf y dom con sigmoide; 13 443 parámetros; inicialización de Glorot	
Techo de la salida	$s_{\text{máx}} = 1{,}6$	$s_{\text{máx}} = 0{,}7$
Orden de entradas	Las 27 características de la Tabla 5.4 en ese orden, las mismas del ciclo anterior, su diferencia, y media y desviación de diez de ellas sobre 2, 5 y 10 ciclos (5.28); el orden queda guardado con los pesos	
Normalización	$(x - \mu)/\sigma$ por componente, con μ y σ calculados sobre la partición de entrenamiento de cada experto, y copiados a la cabecera en C	
Etiquetas	$s^\star = \log_{10}(R_{r,\text{MAS}}/R^{\text{eff}}_{r,\text{MPC}})$; dom y conf* según la Subsección 5.2.1.a	
Pesos de la pérdida	$\mathcal{L}_{\text{dom0}} = 48$; el resto, Tabla 5.8	$\mathcal{L}_{\text{dom0}} = 24$; el resto, Tabla 5.8
Optimizador	Adam, tasa inicial 10^{-3} con decaimiento exponencial de 0,995 por época, recorte del gradiente a norma 1, sin *weight decay*	
Lote y épocas	Lotes de 2048 ciclos; 400 épocas fijas, sin parada temprana	
Semilla	0	
Modelo elegido	El de la última época; la validación y la prueba no eligen la época	
Partición	Por corrida completa (Tabla 5.9)	Por corrida completa (Tabla 5.10)
Umbrales	Los de los coordinadores y el guardián se fijaron durante el desarrollo, con análisis de los conjuntos de entrenamiento y rutinas de verificación en lazo cerrado, sin reservar un conjunto propio; la quinta campaña del arranque y el peso de $\mathcal{L}_{\text{dom0}}$ se definieron a partir de resultados de la prueba (Subsección 5.3.3.a)	

Capítulo 6

Propuesta 2: Una Sola Red como Estimador Inverso Amortizado

Este capítulo desarrolla la segunda de las propuestas de capa correctiva presentadas en la Subsección 2.5.4.b: una **única red** que recibe una ventana de mediciones y propone el desajuste del modelo del controlador.

6.1. Construcción de una red neuronal como estimador de desajuste

La caracterización del Capítulo 4 estableció que el desajuste de la resistencia de rotor se manifiesta sobre variables que el accionamiento ya mide, y de manera especialmente nítida sobre la forma de las corrientes de estator. Esta propuesta lleva esa constatación a su formulación más directa: si la información está en la corriente, se entrena una red que la lea y entregue el desajuste.

6.1.1. Mismo principio de corrección

Esta propuesta al igual que la vista en el Capítulo 5, **no toca** la función de costo del MPC, ni el horizonte de predicción, ni las señales de conmutación del inversor. Lo único que modifica es el valor de resistencia de rotor con que el control predictivo trabaja,

$$R_{r,\mathrm{MPC}}^{\mathrm{eff}} \;\leftarrow\; R_{r,\mathrm{MPC}}^{\mathrm{eff}} \cdot 10^{\,s}, \tag{6.1}$$

donde s es la corrección que la capa entrega. Y con el valor actualizado se recalculan las constantes derivadas del modelo, de modo que la corrección se propaga a la vez al estimador de flujo, a la orientación de campo y al predictor de corriente, sin tocar la ley de control.

6.1.2. Estimación directa de la red desde la forma de onda

La red resuelve el problema inverso: deducir el parámetro a partir de su efecto sobre las señales no se resuelve en línea cada vez que hace falta, sino **una sola vez, fuera de**

línea y sobre cientos de trayectorias y los más de mil ejemplos que se extraen de ellas. Su solución queda *amortizada* en los pesos de una red que en operación solo debe evaluarse, con un costo por decisión que es el de una pasada hacia adelante.

6.2. Diseño de las entradas y salidas

En esta propuesta, la red lee forma de onda y entrega un factor de corrección. Para ello se definen qué señales entran y sobre qué ventana, qué información adicional necesita la red para interpretarla, y qué produce exactamente.

6.2.1. Las salidas de la capa y el techo de la corrección

La red entrena tres salidas, pero la capa correctora **utiliza sólo dos**, con funciones distintas: una fija el candidato de resistencia y la otra condiciona que ese candidato llegue a escribirse. La tercera, una medida de observabilidad del parámetro, que se entrenó junto con las anteriores, pero que **no se utilizó en la capa correctiva**; se documenta al presentar la función de pérdida (Subsección 6.3.2) y no vuelve a aparecer. La Tabla 6.1 recoge las dos que sí gobiernan la capa, con su forma, su rango y la etiqueta contra la que cada una se entrena.

Tabla 6.1: Las dos salidas que gobiernan la capa, su acotación y la etiqueta con que se entrena cada una. Sólo la primera actúa sobre el modelo del controlador.

Salida	Forma	Rango	Etiqueta de entrenamiento	Uso en línea
s, corrección relativa	$s_{\text{máx}} \tanh(z)$	$[-0{,}7;\ 0{,}7]$	$s^\star = \log_{10}\big(R_{r,\text{MAS}}/R^{\text{eff}}_{r,\text{MPC}}\big)$, usada solo en ventanas con $\lvert s^\star\rvert \le 0{,}7$; las que exceden el techo se entrenan como dominio 0	Fija el candidato $R^{\text{eff}}_{r,\text{MPC}} \cdot 10^{\,\alpha s}$
dom, dominio	$\sigma(z)$	$[0;\ 1]$	Binaria: 1 en ventana de maniobra con $\lvert s^\star\rvert \le 0{,}7$, 0 en ventana de baja excitación o fuera del techo	Veta la escritura cuando cae por debajo de 10^{-3}

1. La corrección. $s = \log_{10}\Big(R_{r,\text{MAS}}/R^{\text{eff}}_{r,\text{MPC}}\Big)$. En operación entrega el **factor** por el que propone multiplicar el valor vigente del controlador $R^{\text{eff}}_{r,\text{MPC}}$, y por eso necesita saber desde dónde corrige, con el vector de contexto de la Subsección 6.2.3.

   La salida de corrección se acota mediante

   $$s = s_{\text{máx}}\ \tanh(z), \qquad s_{\text{máx}} = 0{,}7, \tag{6.2}$$

   donde z es la salida lineal de la última capa. El límite $s_{\text{máx}}$ es una **decisión de diseño**: acota la corrección a un factor $10^{0{,}7} = 5{,}01$ por inferencia, lo cual es una propiedad de seguridad, por muy fuera de dominio que se encuentre, la red no puede solicitar un disparate. Y fue la base del conjunto de datos: ninguna etiqueta de regresión lo excede, porque en la primera campaña las razones se eligieron dentro de ese intervalo y en la segunda las ventanas que lo exceden se entrenaron como dominio 0 en lugar de recortarse.

Además, este límite impide *proponer* un desajuste grande de una sola vez, ya que la red debe proponer niveles de corrección de forma iterativa.

2. El dominio. Es una **clasificación binaria** y se entrena como tal, su etiqueta no admite matices: vale 1 en las ventanas que el generador marcó como de maniobra observable y 0 en las que marcó como de baja excitación: máquina detenida, seguimiento en vacío o con carga insuficiente para excitar las señales que informan sobre R_r. Vale también 0 en las ventanas con carga cuyo desajuste excede el límite superior. Esta salida es por tanto la probabilidad estimada de que la ventana observada sea viable, y es la que habilita o veta la escritura.

6.2.1.a. Zona muerta

La zona muerta no es una propiedad de la red, que entrega su solicitud sin umbral alguno: es una de las condiciones del guardián de escritura (Subsección 6.2.4), y se describe aquí porque actúa sobre la salida s. La escritura del parámetro está restringida para trabajar con un desajuste superior a una zona muerta. Expresando la solicitud de la red como un porcentaje, $10^{|s|} - 1$, sólo se escribe si esa magnitud supera el 7,5 % de desajuste. Una vez que la capa entra en la zona muerta, volver a corregir exige superar de nuevo ese rango durante **dos decisiones consecutivas del mismo signo**, condición que evita que una ventana aislada reactive la corrección sobre un modelo **ya ajustado.**

6.2.2. Entrada: ventana de tiempo con seis canales

La entrada principal de la red es una ventana de 0,3 s de medición reciente, presentada como una matriz de seis canales por 256 muestras. Los canales y su normalización se recogen en la Tabla 6.2.

Tabla 6.2: Canales de entrada de la red y su base de normalización.

Canal	Señal	Base
0	$i_{s\alpha}$ — corriente de estator	16,70 A
1	$i_{s\beta}$	16,70 A
2	$v_{s\alpha}$ — tensión aplicada por el inversor	537,40 V
3	$v_{s\beta}$	537,40 V
4	ω_e — velocidad eléctrica medida	314,16 rad/s
5	ω_e^* — referencia de velocidad	314,16 rad/s

La elección de estos seis canales, y no de otros, tiene una justificación por pares. Las **corrientes** porque su nivel de corriente, forma y distorsión, presentan la información de desajuste frente a exigencia de torque. Las **tensiones** porque son los vectores de tensión aplicados para cumplir las referencias de control. Y la **velocidad medida junto a su referencia** es lo que permite distinguir maniobras: una referencia que sube con corriente que sube corresponde a un arranque o a un seguimiento de velocidad, mientras que una referencia constante con velocidad que cae ligeramente y corriente que sube corresponde a un escalón de carga, porque el lazo de velocidad pidió torque al detectar la caída de velocidad. Su diferencia es además el error de seguimiento.

Construcción de la ventana en línea. Antes de llegar a la red, el `C-Script` prepara la ventana en tres pasos fijos, sin pesos aprendidos; la red convolucional de la subsección siguiente trabaja recién sobre el resultado. Las muestras crudas llegan al bloque de control cada $T_s = 100$ μs. Pero no se conservan todas las muestras dentro de 0,3 s (3000 muestras por canal), porque presentarlas todas a la red sería costoso. En su lugar (1) se promedian de a cinco muestras, produciendo muestras reducidas de 500 μs que (2) se almacenan en un buffer circular de 600 posiciones. En el instante de decidir, (3) ese buffer se **remuestrea por interpolación lineal a** 256 **puntos** uniformemente repartidos sobre los últimos 0,3 s, y cada canal se divide por su base de la Tabla 6.2. La Figura 6.1 recoge la cadena completa.

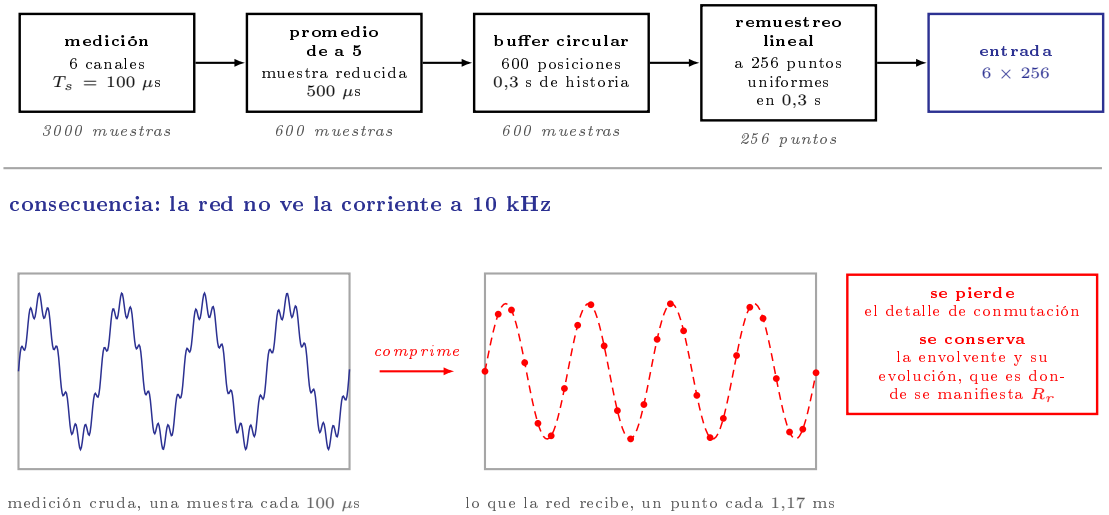


Figura 6.1: Construcción de la ventana de entrada. La compresión resultante, un punto cada 1,17 ms.

Es decir, **la red convolucional no ve la corriente a** 10 **kHz**. Ve 0,3 s de historia comprimidos a 256 puntos, es decir, un punto cada 1,17 ms. Los detalles de conmutación se pierden por construcción; pero lo que se conserva es la forma de la corriente y su evolución, que es donde el desajuste de R_r se manifiesta.

Qué deja pasar esta preparación. Promediar de a cinco muestras funciona como un filtro pasabajos: la fundamental y los armónicos bajos pasan casi intactos, con el 97,5 % de su amplitud a 250 Hz, y el contenido de frecuencia alta se atenúa. Después, con 256 puntos en 0,3 s, la ventana solo puede representar frecuencias de hasta 425 Hz, es decir, hasta el octavo armónico de 50 Hz. Lo que queda por encima de ese límite, sobre todo el rizado de conmutación, no se elimina con otro filtro: se superpone a las frecuencias bajas como un ruido (*aliasing*) que la red recibe junto con la señal. El entrenamiento usó exactamente la misma preparación sobre las trayectorias simuladas, de modo que la red aprendió con ventanas construidas igual que las que recibe en línea.

6.2.2.a. De la ventana al descriptor: una red convolucional

La ventana no entra directamente a la parte de la red que estima la corrección. Antes pasa por una **red convolucional**, que es una red especializada en procesar datos

estructurados en cuadrículas, como imágenes.

Esta red es necesaria porque la ventana son $6 \times 256 = 1536$ números, y una capa densa que los llevase a 32 salidas necesitaría 49 184 pesos y, sobre todo, aprendería **una plantilla distinta para cada instante** de la ventana: lo que reconociese en la muestra 40 no lo reconocería en la 180. Y eso es justamente lo contrario de lo que hace falta, porque la firma del desajuste aparece en la forma de la corriente, pero no siempre en el mismo lugar de los 0,3 s.

Una convolución resuelve las dos cosas a la vez. Su unidad de cálculo es un **filtro**: en la primera capa, una plantilla de 9 puntos consecutivos por los 6 canales, esto es, 54 pesos. El filtro se apoya sobre un tramo de la ventana, multiplica sus pesos por los valores que tiene debajo, suma, añade un sesgo y aplica la no linealidad; el número que resulta indica cuánto se parece ese tramo de señal a lo que el filtro busca. Después **se desliza** y repite la operación, avanzando de dos en dos muestras. La misma plantilla se emplea en todas las posiciones, de modo que lo aprendido sirve en cualquier instante de la ventana y el costo baja de 49 184 pesos a 1 760.

Dos escalas conviene retener, porque fijan qué puede ver cada capa. Los 9 puntos del primer filtro son unos 10,6 ms, es decir, **media onda** a la frecuencia nominal; y encadenando las tres capas, cada salida de la última depende de 37 muestras de la entrada, unos 42 ms, **algo más de dos períodos eléctricos**. La escala de los detectores está por tanto ajustada al ciclo de corriente, que es donde el desajuste se manifiesta.

Lo que el codificador **recibe** es entonces la matriz de 6×256, y lo que **entrega** es otra matriz, de 64×32: sesenta y cuatro descriptores aprendidos, evaluados en treinta y dos instantes. El intercambio que hacen las tres capas es siempre el mismo: **el tiempo se comprime y la descripción se enriquece**. Cada convolución avanza de a dos muestras y rellena con ceros cada extremo con la mitad entera de su núcleo, 4, 3 y 2 muestras, de modo que la longitud se reduce exactamente a la mitad en cada capa: de 256 a 128, 64 y 32.

Aún es muy pesado: Treinta y dos instantes siguen siendo **demasiados** y por eso se toma el **valor medio** de cada uno de los 64 descriptores a lo largo de los 32 instantes, y también su **valor máximo**. El resultado son $64 + 64 = 128$ números.

Esos 128 números son el descriptor de tamaño fijo con que la ventana queda representada, y son los que se suman con el vector de contexto de la subsección siguiente.

6.2.3. Adicional a la ventana: vector de contexto

La forma de onda no contiene toda la información que la red necesita. Falta, de manera esencial, **dónde está parado el controlador**: la misma ventana de señal admite interpretaciones distintas según cuál sea el valor de resistencia con que el MPC está prediciendo. Por eso la red recibe, además de los valores de ventana, un pequeño vector de contexto cuya primera componente es

$$c_0 = \log_{10}\left(\frac{R^{\text{eff}}_{r,\text{MPC}}}{R_{r,\text{nom}}}\right), \tag{6.3}$$

donde $R_{r,\mathrm{nom}} = 0{,}83\ \Omega$ es la resistencia nominal de placa de la máquina con que se entrenó esta red. Y la componente (6.3) dice cuánto $R^{\mathrm{eff}}_{r,\mathrm{MPC}}$ se aparta del valor nominal de placa.

La segunda componente del contexto es la **frecuencia de la corriente en la ventana**. Se obtiene contando ciclos con un disparador de Schmitt sobre $i_{s\alpha}$, con histéresis del 10 % de la amplitud y dos guardas: una amplitud mínima de 2,0 A, por debajo de la cual la entrada vale cero, y una separación mínima entre disparos expresada en *tiempo*, que impide que el *ripple* de conmutación se cuente como un ciclo. Esta magnitud arrastra implícitamente la frecuencia síncrona y con ella el régimen de operación, incluido si la máquina trabaja en debilitamiento de campo, situación en que el flujo disponible y con él la observabilidad de R_r cambia.

En la red desplegada, esta segunda componente es el número de ciclos contados dividido por la duración de la ventana y por 50 Hz. Las tres componentes restantes son una codificación *one-hot* de la maniobra actual, cuyo uso no se implementó en el flujo de trabajo: en línea quedan fijas en escalón de carga.

6.2.4. Estructura final: red más algoritmo de escritura

Fijadas las salidas y las entradas, la propuesta queda definida por dos piezas de naturaleza distinta que conviene no confundir.

1. Una **red neuronal inversa amortizada**, que lee la ventana de corriente, tensión y velocidades, junto con el contexto del controlador, y entrega por decisión la propuesta de la Subsección 6.2.1: corrección y dominio. Es la única pieza que aprende, y su comportamiento queda fijado al congelar los pesos.
2. Un **guardián de escritura** conservador, que recibe esa propuesta junto con el estado de operación y decide si el parámetro del controlador llega a modificarse.

Por lo cual, la red aporta la capacidad de leer una forma de onda; y el algoritmo de escritura permite la aplicación. Una salida de la red **nunca llega directamente al controlador**.

El guardián evalúa **cuatro condiciones** antes de iniciar una corrección con rampa: la habilitación de la maniobra, comprobación de desajuste, trabajo fuera de la zona muerta y confirmación de evidencia.

6.2.4.a. Bloqueo en marcha invertida

La Propuesta 1 cubre la marcha inversa cambiando el signo de sus características con dirección (Subsección 5.2.4.c). En la red única no basta con cambiar el signo de la velocidad, porque recibe las señales crudas: al invertir el giro, también la componente β de la corriente y de la tensión cambia de signo, y habría que corregir cuatro de sus seis canales. Ese ajuste no se implementó. La Tabla 6.3 muestra, sobre la red desplegada, por qué hace falta. Con la ventana inversa tal como llega, la red propone una corrección mucho menor que en marcha directa y deja de vetar las ventanas en vacío. Cambiando solo la velocidad, la mediana se acerca a la necesaria, pero ventana a ventana la propuesta oscila mucho más y tampoco veta el vacío. Solo con los cuatro canales cambiados reproduce la marcha directa.

Tabla 6.3: Red desplegada ante las ventanas de la marcha inversa, comparadas con las de la misma rutina en sentido positivo (Sección 7.3), con la capa desactivada y el modelo fijo en 3,83 Ω sobre la máquina de 0,83 Ω: 46 ventanas, una cada 0,1 s. Corrección necesaria: $s^\star = -0{,}664$. Vetos distintos: ventanas en que la decisión de dominio difiere de la que toma la red con la ventana directa.

Ventana entregada a la red	**s propuesta, mediana (rango)**	**Diferencia con la directa**	**Vetos distintos**
Directa (referencia)	−0,588 (−0,671 a −0,496)	—	—
Inversa, tal como llega (la desplegada)	−0,342 (−0,556 a −0,069)	0,254 (máx. 0,511)	5 de 46
Inversa, solo las velocidades con signo cambiado	−0,642 (−0,682 a −0,217)	0,114 (máx. 0,421)	5 de 46
Inversa con $i_{s\beta}$, $v_{s\beta}$, ω_e y ω_e^* con signo cambiado	−0,586 (−0,668 a −0,504)	0,017 (máx. 0,050)	1 de 46

Por eso el guardián **veta la escritura** cuando la media de la velocidad o de su referencia en la ventana de la decisión es menor que −0,02 en por unidad, unos 30 r/min. En marcha inversa la capa conserva la corrección que traía, pero no identifica la resistencia de rotor.

6.3. Arquitectura y entrenamiento

Con las entradas y las salidas fijadas, esta sección desarrolla cómo se obtiene la red que las conecta.

6.3.1. Arquitectura de la red

La red queda formada por dos tramos encadenados. El primero es la **red convolucional** de la Subsección 6.2.2.a, que lleva la ventana de 6×256 a un descriptor de 128 números. El segundo es una **red MLP** que recibe esos 128 números, con las 5 componentes de contexto (133 entradas) y produce las salidas de la red. La Tabla 6.4 contabiliza las tres neuronas de esa última capa, que son las que el modelo desplegado tiene.

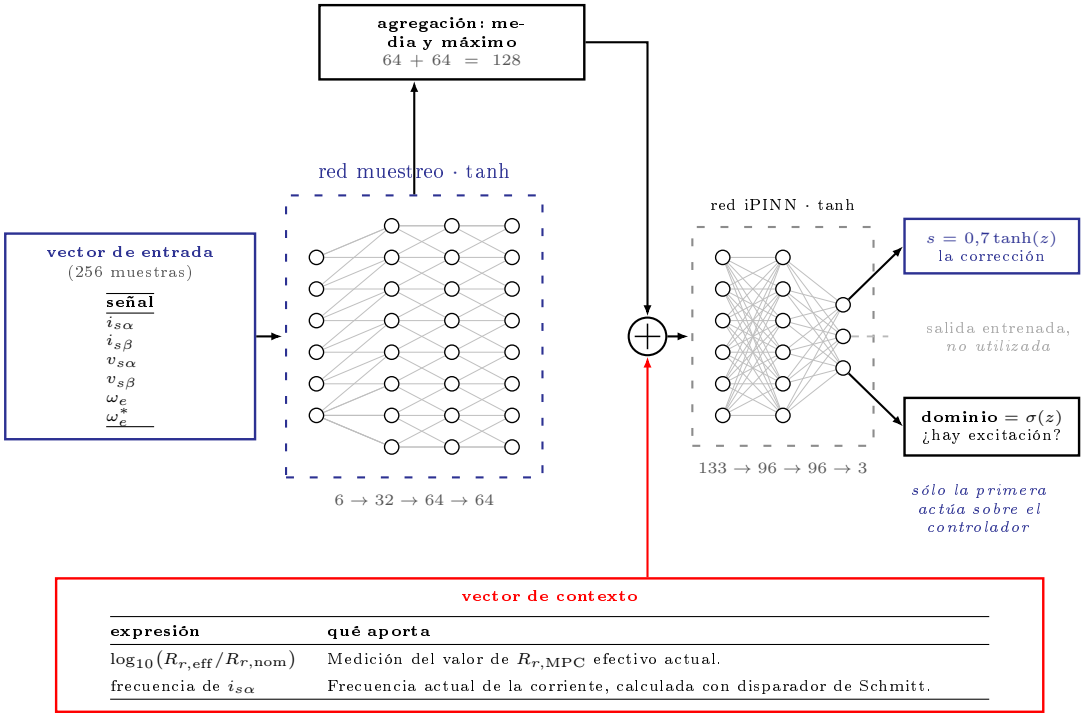


Figura 6.2: Arquitectura de la red. La red convolucional con la agregación por valores medios y máximos produce 128 componentes que se suman con el vector de contexto antes de la red amortizada inversa (iPINN). La última capa tiene tres neuronas, pero sólo dos alimentan a la capa correctora.

Tabla 6.4: Arquitectura de la red. Todas las capas ocultas emplean tangente hiperbólica y la inicialización es Xavier normal con sesgos nulos. El cómputo se realiza en doble precisión.

Capa	Salida	Núcleo / paso	Parámetros
Entrada (ventana)	6×256	—	—
Conv1D $6 \to 32$	32×128	9 / 2	1 760
Conv1D $32 \to 64$	64×64	7 / 2	14 400
Conv1D $64 \to 64$	64×32	5 / 2	20 544
Agregación media ‖ máximo	128	—	—
Agregación con contexto	133	—	—
Capa $133 \to 96$	96	—	12 864
Capa $96 \to 96$	96	—	9 312
Capa $96 \to 3$	3	—	291
		Total	**59 171**

Tres decisiones merecen justificación.

Los **núcleos decrecientes** 9, 7 y 5. El primero es el más largo porque trabaja sobre la señal sin resumir y necesita abarcar media onda; los siguientes operan sobre descriptores que ya condensan un tramo de tiempo, de modo que un núcleo más corto les basta para

cubrir el mismo horizonte físico.

El **paso 2 en las tres convoluciones** reduce la longitud de 256 a 32 sin necesidad de capas de submuestreo aparte, y es lo que hace que el campo receptivo crezca hasta los dos períodos eléctricos citados en la Subsección 6.2.2.a.

La **activación es tangente hiperbólica** en todas las capas ocultas: una no linealidad suave y centrada en cero, coherente con entradas normalizadas que toman signo.

Conviene notar cómo se reparten los 59 171 parámetros: 36 704 están en las tres convoluciones y 22 467 en la cabeza densa. Es decir, **casi dos tercios del modelo se dedican a leer la forma de onda**, y sólo un tercio a convertir esa lectura en una corrección.

6.3.2. La función de pérdida

El entrenamiento es supervisado en el sentido de la Subsección 2.4.3. Y la función de pérdida combina un término supervisado con tres auxiliares,

$$\mathcal{L} = 10\,\mathcal{L}_{\text{ratio}} + 0{,}5\,\mathcal{L}_{\text{dom}} + 0{,}2\,\mathcal{L}_{\text{fis}} + 0{,}2\,\mathcal{L}_{\text{obs}}, \tag{6.4}$$

donde $\mathcal{L}_{\text{ratio}}$ es el error cuadrático entre la corrección estimada y la verdadera, evaluado sólo sobre las ventanas de dominio 1, esto es, de maniobra y con $|s^\star| \le 0{,}7$; $\mathcal{L}_{\text{dom}}$ es la entropía cruzada del dominio contra su etiqueta binaria; y $\mathcal{L}_{\text{fis}}$ es el término que incorpora la física.

El término restante, $\mathcal{L}_{\text{obs}}$, ajusta la **tercera salida** de la red contra una medida de observabilidad del parámetro derivada del mismo perfil físico. Se registra aquí porque formó parte del entrenamiento y sus pesos son los que se desplegaron, pero **esa salida no interviene en ninguna decisión de la capa**: fue un intento de complementación que **no llegó a incorporarse** al flujo de trabajo.

El término físico. La física se incorpora mediante un **perfil de residuo precalculado**.

Para cada ventana i del conjunto se construye, *antes* del entrenamiento, la función

$$E_i(s) \;=\; \frac{1}{2K\,I_b^2}\sum_{k=1}^{K}\Big\|\,\mathbf{i}_s^{\text{med}}[k] - \hat{\mathbf{i}}_s\big[k \mid R_{r,\text{MPC}}^{\text{eff}}\cdot 10^{\,s}\big]\,\Big\|^2, \tag{6.5}$$

esto es: se toma un valor candidato de resistencia, se integra con él el modelo de la máquina para predecir la corriente un paso adelante en cada muestra de la ventana, y se acumula el error cuadrático contra la corriente efectivamente medida en el paso siguiente. K es el número de pasos de la ventana, I_b la corriente base de la Tabla 6.2, y el factor $1/2$ promedia las componentes α y β del error. El cálculo es costoso, pero ocurre una sola vez por ventana y **fuera de línea.**

Durante el entrenamiento, el término físico es el valor de ese perfil *interpolado* en la corrección que la red propone,

$$\mathcal{L}_{\text{fis}} \;=\; \frac{1}{N_m}\sum_{i\in\mathcal{M}}\ln\!\Big(1+\frac{E_i(s_i)}{E_{\text{esc}}}\Big), \tag{6.6}$$

donde s_i es la corrección que la red propone para la ventana i, y el promedio recorre $\mathcal{M}$, las ventanas de maniobra con $|s^\star| \le 0{,}7$ del lote (N_m de sus N ventanas), las mismas

sobre las que se evalúa $\mathcal{L}_{\text{ratio}}$. El perfil se interpola linealmente entre sus nodos, lo que lo hace derivable respecto de s y, por tanto, utilizable por el optimizador sin diferenciación automática sobre la ecuación de la máquina. El término empuja la estimación hacia el mínimo de $E(s)$, que es el valor de resistencia que mejor explica la corriente observada según el modelo del predictor.

El perfil se evalúa en 61 valores de s entre $-0{,}7$ y $0{,}7$. Para cada uno, el flujo de rotor se integra por el método de Heun con la resistencia candidata, a partir de la corriente y la velocidad medidas y desde flujo nulo al comienzo de la trayectoria; la corriente del paso siguiente se predice por Euler, y el error se normaliza por la corriente base de la Tabla 6.2, de modo que $E(s)$ es adimensional. A diferencia del residuo de la Propuesta 1, que conserva el flujo estimado con la resistencia equivocada y por eso se minimiza lejos del valor real (Tabla 5.3), aquí cada candidato integra su propio flujo, de modo que con la resistencia verdadera el modelo reproduce la máquina.

Así, mientras $\mathcal{L}_{\text{ratio}}$ compara la corrección propuesta con la etiqueta, $\mathcal{L}_{\text{fis}}$ la compara con la física: crece cuanto peor reproduce el modelo, con la resistencia propuesta, la corriente medida, y su mínimo queda, en mediana, a menos de 0,01 en s de la corrección real. Como E no llega a cero ni con la resistencia correcta, porque la predicción de un paso nunca es exacta, se divide por su valor típico en ese punto, $E_{\text{esc}} = 4{,}48 \cdot 10^{-3}$, que es la mediana sobre las 1 285 ventanas de entrenamiento; así el término queda del orden de uno, comparable con los demás. El logaritmo no mueve el mínimo, pero impide que unas pocas ventanas mal reproducidas, con E de hasta 197 veces el valor típico, dominen el término: una ventana diez veces peor que lo típico aporta $\ln 11 \approx 2{,}4$ y no 10. El 1 dentro del logaritmo hace, además, que el término valga cero con error nulo y nunca sea negativo.

El peso 0,2 es el único que se eligió con un barrido (Tabla 6.5). La estimación empeora a medida que crece, porque el mínimo de $E(s)$, aunque centrado en la corrección real, se dispersa de una ventana a otra; por eso el término físico acompaña al supervisado y no lo reemplaza. Los otros tres pesos se fijaron al diseñar la red, sin una búsqueda sistemática.

Tabla 6.5: Los cuatro pesos de la función de pérdida (6.4) y su origen. Barrido: error porcentual medio de validación de la estimación, promedio de dos semillas, con la misma arquitectura y los mismos datos.

Término	**Peso**	**Qué compara**	**Origen del peso**
$\mathcal{L}_{\text{ratio}}$	10	La corrección propuesta con la etiqueta	Término principal; fijado al diseñar la red
$\mathcal{L}_{\text{dom}}$	0,5	El dominio con su etiqueta	Fijado al diseñar la red
$\mathcal{L}_{\text{fis}}$	0,2	La corrección propuesta con el perfil físico	Barrido de 0,2 a 30: el error sube de 6,1 a 10,3 %; sin logaritmo, 8,2 % con 0,2
$\mathcal{L}_{\text{obs}}$	0,2	La observabilidad, que no se usa en línea	Fijado al diseñar la red

Con esta función, se incorpora la información física durante el entrenamiento por medio de un perfil producido por el modelo de la máquina; y durante la inferencia no se resuelve nuevamente ese perfil. La red entrega una propuesta amortizada, mientras que la autoridad para escribir permanece en reglas deterministas.

6.3.3. Los escenarios del conjunto de datos

Los escenarios se generan sobre tres variantes del modelo de la plataforma del Capítulo 4, con los mismos parámetros de máquina y las mismas bases de normalización, y de sus trayectorias se recortan ventanas, casi todas de 0,3 s. **Cada ventana es un ejemplo de entrenamiento independiente**: la red no aprende la rutina, aprende a leer un tramo de forma de onda y decir cuánto está desajustado el modelo. El conjunto se construyó en **dos campañas sucesivas**, y la segunda existe porque la primera resultó insuficiente por una razón que la medición dejó clara (Tabla 6.6).

Tabla 6.6: Las dos campañas que componen el conjunto de datos de la red desplegada, por familia. Trayectorias: total y, entre paréntesis, de entrenamiento, validación y prueba. Ejemplos: ventanas de entrenamiento de regresión y de dominio 0, las únicas que ajustan los pesos. 1 266 de los 1 463 ejemplos miden 0,3 s; el resto, de la familia de legado y del aumento de 0,10 Ω, mide entre 0,04 y 1,0 s. El sumando +72 son las trayectorias de los tres conjuntos de evaluación heredados, que no aportan ejemplos de entrenamiento. Toda ventana se remuestrea a 256 puntos.

Familia	**Tray.**	**Ejemplos** **regr./dom. 0**	$\mathbf{R}_{\mathbf{r},\mathrm{MAS}}$ [Ω]	$\mathbf{R}^{\mathrm{eff}}_{\mathbf{r},\mathrm{MPC}}$ inicial [Ω]	**Maniobra, carga, referencia y duración**
Desajuste constante: lazo abierto, $R^{\mathrm{eff}}_{r,\mathrm{MPC}}$ fijo en toda la trayectoria					
Matriz velocidad–carga	135 (75/30/30)	100/50	0,70; 0,85; 1,00; 1,20 y 1,40 (en entrenamiento, 0,70; 1,00 y 1,40)	0,5; 1,0 y 2,0	Arranque desde reposo y escalón de cinta a los 0,35 s, sin escalón si la carga es nula; carga de 0, 50 y 100 %; referencia de 0,25; 0,50; 0,75; 1,00 y 1,10 pu; 1,0 s (4,5 s los ajustados)
Legado de tres maniobras	129 (57/0/0+72)	152/57	15 valores de 0,25 a 10,00 (24 más, de 0,33 a 7,60, solo en las heredadas)	0,5; 1; 2 y 5 (0,83; 1,5 y 3,0 solo en las heredadas)	43 de cada maniobra, a 1,00 pu: arranque en vacío (0,2 s), escalón de cinta 0 → 50 → 100 % (1,05 s) e inversión sin carga (1,05 s)
Aumento de 0,10 Ω	20 (8/4/8)	42/0	0,10	0,1; 0,2; 0,4 y 0,5	Arranque y escalón de cinta; referencia de 0,50 y 1,10 pu en entrenamiento; 2,5 s (4,5 s los ajustados)
Lazo cerrado: la capa actuando, ventanas tomadas en los instantes de decisión					
Arranque con carga constante	93 (65/12/16)	620/30	0,50; 0,83; 1,00; 3,00 y 5,00	$R_{r,\mathrm{MAS}}$ dividido por 0,1 a 10, dentro de [0,05; 10]; después, lo que escribe la capa	Arranque desde reposo con cinta constante, sin escalón, en tres puntos: 0,50 pu con 100 %, 1,00 pu con 50 % y 0,25 pu con 100 %; 5,0 s, diez ventanas por corrida
Cuatro rutinas de perfil variable	52 (36/10/6)	371/41	Los mismos cinco	$R_{r,\mathrm{MAS}}$ dividido por 0,2; 1 o 5; después, lo que escribe la capa	13 corridas de cada rutina a 1,00 pu, de 4,9 a 5,5 s: escalera de carga (0 → 45 → 75 → 95 → 100 %), parada y rearranque con cinta al 50 %, vacío seguido de carga plena, e inversión de giro con cinta al 50 %
Total	**429** (241/56/60+72)	**1 285/178**	19 valores de 0,10 a 10,00 en entrenamiento (42 contando evaluación)		

La diferencia entre los valores iniciales de $R^{\text{eff}}_{r,\text{MPC}}$ de las dos campañas es la que las distingue y merece leerse con cuidado. En la primera, la resistencia del modelo es un **valor prescrito** que se fija al comenzar la corrida y no se mueve: diez valores en total entre sus tres familias. En la segunda solo se prescribe el valor inicial, que resulta de dividir $R_{r,\text{MAS}}$ por razones de 0,1 a 10. A partir de ahí son **los valores que la propia capa fue escribiendo** a lo largo de la trayectoria; de ahí que aparezcan 670 valores distintos y que la razón de desajuste cubra dos órdenes de magnitud en vez de un factor diez. Solo entran como ejemplos de regresión las ventanas cuyo desajuste cabe bajo el techo de la salida, entre 0,2 y 5 veces; las demás se entrenan como dominio 0.

La partición asigna cada trayectoria completa a un solo conjunto (Tabla 6.6). Solo los 1 463 ejemplos de entrenamiento ajustan los pesos; la validación decide qué época se conserva y cuándo se detiene el entrenamiento, y la prueba y los conjuntos heredados solo se informan.

Conviene subrayar la cifra de resistencia real, porque de ella depende cómo han de leerse los resultados: la red *no* se entrena alrededor de la resistencia de placa sino sobre dos órdenes de magnitud, de 0,10 a 10,00 Ω, aunque el 85,5 % de los ejemplos de regresión está en los cinco valores del lazo cerrado. El rango es comparable al del experto de arranque, de 0,30 a 10,00 Ω (Subsección 5.3.3.a). La resistencia de la campaña de evaluación, 0,83 Ω, está dentro de ese dominio, aunque sus puntos de partida de 5,83 y 9,83 Ω implican desajustes de 7,0 y 11,8 veces, mayores que el techo de los ejemplos de regresión. Y lo que permanece fijo son los restantes parámetros de la máquina y las bases con que se normalizan las entradas: el dominio es amplio en el *parámetro* y único en la *máquina*.

Por qué hizo falta la segunda campaña: el desplazamiento de covariables. La primera versión del conjunto se generó íntegramente con desajuste *constante*, es decir, sin que la capa correctora actuase. Esa elección introduce un problema conocido en aprendizaje por imitación: la red se entrena en un régimen y opera en otro. Tras la primera corrección el punto de operación cambia por completo: el modelo predice mejor, mejora la magnetización, sube el torque, sube la velocidad. De modo que los instantes en que la red realmente trabaja apenas estaban representados en su entrenamiento.

La evidencia de que el problema era real y no teórico se acumuló de forma característica: cada vez que mejoraban las métricas fuera de línea, el desempeño en lazo cerrado empeoraba. Un modelo con un 35 % menos de error de prueba resultó un 47 % peor sobre la batería de escenarios. La solución adoptada es la estándar para ese problema: **recoger los datos con la propia política y reentrenar**. Se ejecutaron las trayectorias con la corrección encendida, usando la red desplegada en ese momento, anterior a la definitiva, y se extrajeron las ventanas en los instantes de decisión, etiquetándolas con el $R_{r,\text{MAS}}$ verdadero y el $R^{\text{eff}}_{r,\text{MPC}}$ vigente en cada una. Esas ventanas se *agregan* al conjunto anterior en lugar de reemplazarlo, y la red definitiva se entrenó sobre el conjunto agregado: se trata de una sola iteración de agregación. La segunda campaña quintuplica a la primera en casos (1 528 ventanas frente a 284 trayectorias), aunque proviene de solo 145 corridas, y aporta 1 062 de los 1 463 ejemplos de entrenamiento.

Una precaución resulta imprescindible al generarlas: **no muestrear durante la rampa de corrección**. Si el valor de resistencia cambia dentro de la ventana, la etiqueta es ambigua. La geometría temporal de la capa lo garantiza por construcción, pero al generar

los datos se impone además de forma explícita, descartando toda ventana en que $R^{\text{eff}}_{r,\text{MPC}}$ no sea constante.

6.4. La capa correctora en línea

Esta sección describe cómo se aplican las correcciones y qué condiciones deben cumplirse.

La Figura 6.3 sitúa la capa dentro del accionamiento, con el mismo código de color que la Figura 5.3 de la propuesta anterior: en negro el control predictivo tal como se describió en el Capítulo 3, en rojo la red y las mediciones que consume, y en azul el único camino por el que la capa toca al controlador, la resistencia efectiva del modelo interno. La comparación entre ambas figuras hace visible la diferencia entre las dos propuestas: donde la primera coloca un sistema de expertos con sus árbitros y su guardián, la segunda coloca una sola red.

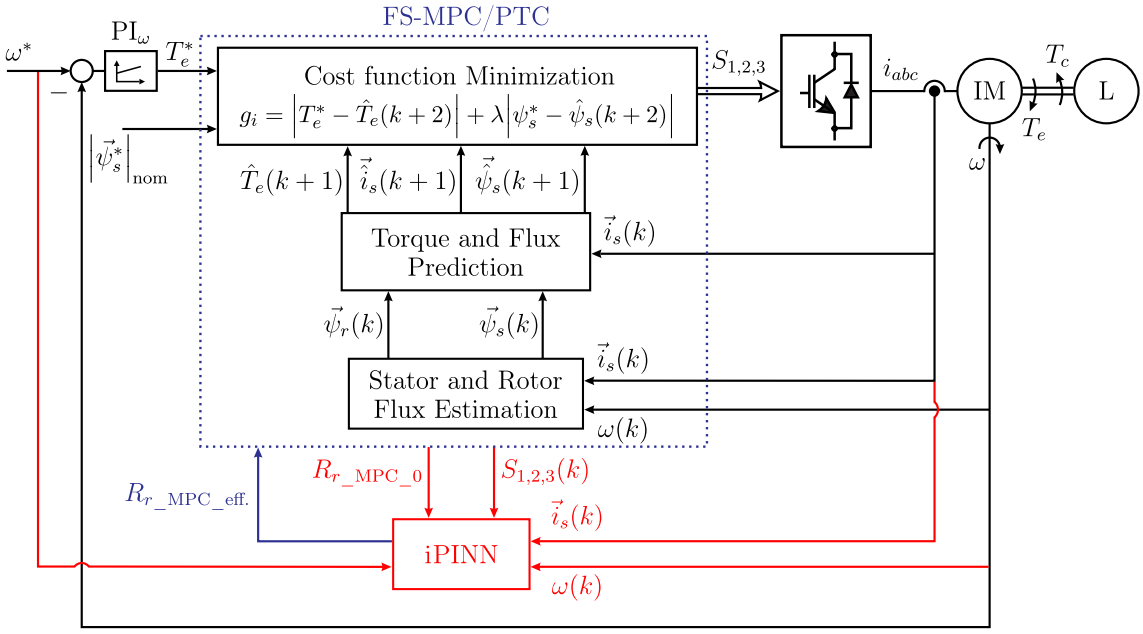


Figura 6.3: Esquema de control completo con la propuesta 2.

6.4.1. Cadencia y aplicación de la corrección

La lógica en línea se reduce a una secuencia corta:

1. Cada 0,4 s, a partir del primer segundo de simulación, se arma la ventana y se ejecuta la pasada hacia adelante.

2. Antes de escribir, el guardián comprueba en este orden: que la capa esté habilitada; que ni la velocidad medida media ni su referencia media indiquen marcha invertida (Subsección 6.2.4.a); que la salida de dominio no active el veto, salvo cuando $|s| \geq 0{,}95 s_{\text{máx}}$; y que se cumpla el estado de zona muerta

3. Y finalmente, cuando se obtienen dos decisiones consecutivas del mismo signo por encima del 7,5 %. Se permiten actuar se define el candidato mediante:

$$R^{\text{eff}}_{r,\text{MPC}} \leftarrow R^{\text{eff}}_{r,\text{MPC}} \cdot 10^{\alpha s}, \qquad \alpha = 0{,}5, \tag{6.7}$$

que se aplica mediante una rampa de 0,1 s en lugar de un escalón. El factor α significa que **solo se aplica la mitad de la corrección solicitada**, lo que amortigua las oscilaciones y hace que la convergencia sea gradual.

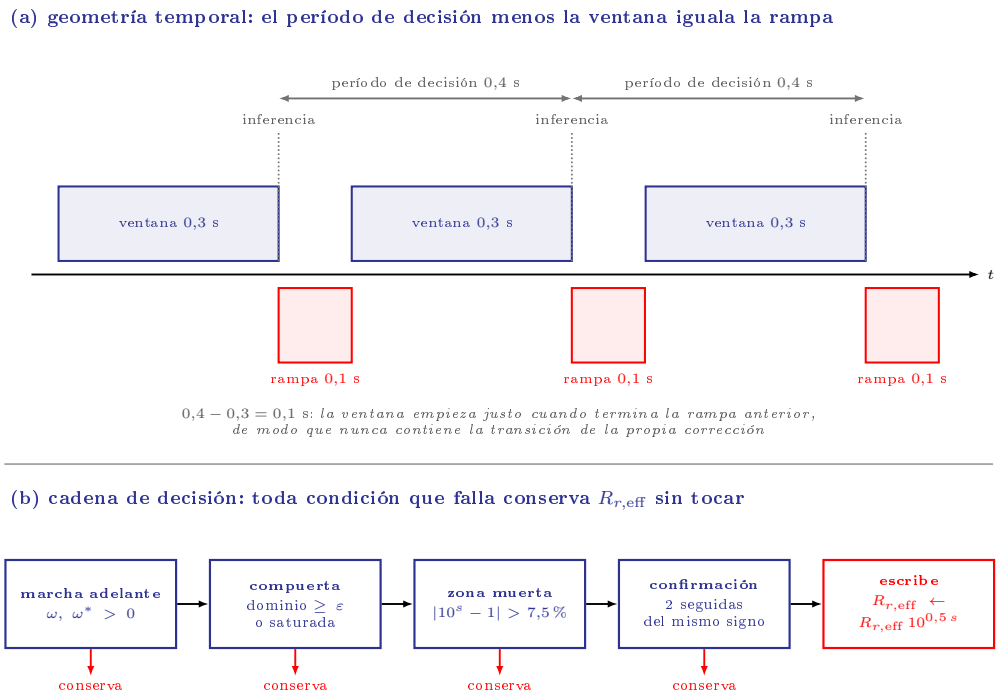


Figura 6.4: Capa correctora. (a) Geometría temporal: la ventana de observación comienza donde termina la rampa anterior, de modo que ninguna ventana contiene la transición de la propia corrección. (b) Cadena de decisión: la escritura exige que se cumplan las cuatro condiciones, y el fallo de cualquiera de ellas conserva el valor vigente en lugar de modificarlo.

6.4.2. Configuración final de entrenamiento y ejecución

La Tabla 6.7 reúne la configuración de la red desplegada, leída del archivo de configuración y del registro de entrenamiento guardado junto con los pesos, y verificada contra el conjunto reconstruido: la escala del término físico recalculada coincide con la registrada.

Tabla 6.7: Configuración final de la red de la Propuesta 2.

Aspecto	**Valor**
Arquitectura	Tres convoluciones 1D de 32, 64 y 64 canales, núcleos de 9, 7 y 5, paso 2 y relleno de 4, 3 y 2 muestras; agregación por media y máximo; tronco $133 \to 96 \to 96$; tres salidas, con $s = 0{,}7\tanh(\cdot)$; tangente hiperbólica; 59 171 parámetros (Tabla 6.4)
Ventana	0,3 s de la trayectoria promediada por bloques de cinco muestras y remuestreada por interpolación lineal a 256 puntos; seis canales en el orden de la Tabla 6.2
Contexto	Cinco componentes: $\log_{10}(R^{\text{eff}}_{r,\text{MPC}}/0{,}83)$, frecuencia de ciclos sobre 50 Hz y la codificación de maniobra (arranque, escalón de carga, inversión), fija en escalón de carga en línea
Normalización	Bases físicas fijas de la Tabla 6.2, no estimadas de los datos; E_{esc} del término físico, sobre las ventanas de entrenamiento
Etiquetas	$s^\star = \log_{10}(R_{r,\text{MAS}}/R^{\text{eff}}_{r,\text{MPC}})$ recortada a $\pm 0{,}7$; dominio 1 en las ventanas excitadas y 0 en las de baja excitación; confianza a partir de la curvatura relativa de $E(s)$ en $s^\star$
Pesos de la pérdida	10, 0,5, 0,2 y 0,2 (6.4); término físico logarítmico (6.6)
Optimizador	Adam, tasa inicial 10^{-3} con decaimiento exponencial hasta el 5 % en 1600 épocas; recorte del gradiente a norma 1; doble precisión
Lote y épocas	Lotes de 32 ventanas; en cada época se sortean, con reposición, tantas ventanas como ejemplos de entrenamiento, 1 463 (46 lotes), con peso inversamente proporcional al número de ejemplos de su maniobra; rotación aleatoria del plano $\alpha\beta$ en cada lote; hasta 1600 épocas, con parada temprana tras 350 sin mejorar la validación
Semillas	1234 y 2026; se despliega la 2026, por convención del proyecto
Modelo elegido	La época con menor error porcentual medio de validación: la 667 de 1017 con la semilla 2026. La arquitectura, entre varias variantes y semillas, y los umbrales del guardián se eligieron con una batería de once escenarios en lazo cerrado, distinta de la partición de prueba
Partición	Por trayectoria completa: 241 de entrenamiento, 56 de validación y 60 de prueba, más 72 trayectorias heredadas de evaluación; de ellas se extraen 1 463 ejemplos de entrenamiento (Subsección 6.3.3)

Capítulo 7

Resultados y Análisis

Este capítulo presenta la evaluación de las dos propuestas desarrolladas en los Capítulos 5 y 6, bajo distintos grados de desajuste paramétrico (Sección 1.2).

El capítulo se organiza en cuatro secciones. Las Secciones 7.1 y 7.2 evalúan cada propuesta por separado. La Sección 7.3 sigue una rutina específica en el dominio del tiempo para observar *cómo* corrige cada propuesta. La Sección 7.4 las contrasta.

7.1. Resultados de la Propuesta 1: el sistema de expertos

7.1.1. Organización de la evaluación

La evaluación de esta primera propuesta comprende 2418 corridas: 806 condiciones de operación, cada una simulada tres veces y variando únicamente los aportes de la capa correctiva.

Los tres modos de operación:

1. *Solo MPC*: la capa desactivada, que es la línea base y reproduce el comportamiento caracterizado en el Capítulo 4
2. *MPC + PINN arranque*: sólo el experto de arranque de la Sección 5.2.5 y su coordinador
3. *MPC + PINN arr.+carga*: la capa completa, con los dos expertos y los dos coordinadores de la Sección 5.4.

Esto permite atribuir cada diferencia a uno de los dos expertos, cada uno con su coordinador y sus rutas físicas, y no al azar del escenario. No separa, en cambio, las redes de las reglas físicas que las acompañan: ningún modo apaga el *breakaway*, la vía de emergencia o las bandas muertas manteniendo las redes, ni las redes manteniendo esas reglas.

Cómo leer las figuras. Todas las figuras de esta sección comparten la misma organización, que conviene describir una vez. El **eje horizontal** recorre la resistencia de rotor con que se inicializa en el modelo del control predictivo, $R_{r,\mathrm{MPC},0}$, en escala logarítmica. La línea vertical discontinua marca el valor ideal del control (caso ajustado), de modo que a su izquierda el modelo subestima y a su derecha sobreestima. Las tres curvas corresponden

a los tres modos de operación. En las rutinas de cinta, las **columnas** de paneles recorren la carga presente durante el arranque (T_{ini}) y las **filas** el nivel al que se lleva la carga tras el escalón (T_{fin}), ambos en fracciones del torque nominal; por encima de la diagonal el escalón aumenta la carga, sobre ella la mantiene y por debajo la reduce. En todos los indicadores, menos es mejor.

Ventanas. Los indicadores se calculan sobre los 0,5 s finales de cada rutina, ya en régimen tras el escalón. Las señales procedentes del bloque de control se adelantan una muestra antes de operar, para compensar el retardo de un período de muestreo que introduce el puerto de salida.

Tabla 7.1: Indicadores de desempeño empleados en la evaluación, en el orden en que se presentan.

Indicador	**Definición**	**Unidad**
Residuo de R_r	$\lvert R_{r,\mathrm{MAS}} - \mathrm{mediana}(R_{r,\mathrm{MPC}}^{\mathrm{eff}})\rvert$	Ω
Error de velocidad	$\mathrm{RMSE}(\omega^* - \omega_m)$	r/min
THD de corriente	$100\sqrt{(\mathrm{THD}_a^2 + \mathrm{THD}_b^2 + \mathrm{THD}_c^2)/3}$	%
Error de estim. de torque	$\mathrm{RMSE}(T_e - \hat{T}_e)$	N·m
Ripple de torque	$\mathrm{RMS}(T_e - \overline{T_e})$	N·m
Error de estim. de flujo	$\mathrm{RMS}\sqrt{(\psi_{sd} - \hat{\psi}_{sd})^2 + (\psi_{sq} - \hat{\psi}_{sq})^2}$	Wb
Ripple de flujo	$\mathrm{RMS}(\lvert\boldsymbol{\psi}_s\rvert - \overline{\lvert\boldsymbol{\psi}_s\rvert})$	Wb

Los dos indicadores de flujo son distintos. El **error de estimación** es *vectorial*: compara el flujo de estator de la máquina con el que estima el controlador, cada uno en los ejes dq de su propio flujo de rotor. El ***ripple*** se refiere solo a la *magnitud* $\lvert\boldsymbol{\psi}_s\rvert$ de la máquina.

El piso del error de velocidad. El controlador de velocidad empleado no integra en la escala de las corridas y deja un error de régimen proporcional al torque de carga, $T_L/(k_w\, p)$ (Sección 3.3). Por eso el modelo ajustado nunca llega a error cero: su error de velocidad crece con la carga final, y en las tablas de este capítulo vale, por ejemplo, 5,5 r/min con la cinta a medio par y 10,8 r/min con la bomba a velocidad nominal. Ninguna corrección de R_r puede bajar de ese piso, y un error de velocidad igual al del modelo ajustado indica que la capa correctiva recuperó por completo el punto de operación.

Qué cuenta como éxito y como fallo. Los criterios de este capítulo se evalúan sobre la misma ventana de los 0,5 s finales:

- **Alcanza su consigna**: la velocidad media de la ventana difiere de su referencia en menos del 2 %. Exigir en su lugar que el error RMS de velocidad quede bajo el 2 % de la referencia da los mismos recuentos en todas las columnas de la Tabla 7.21. Con una tolerancia del 5 % solo cambia una condición: el sistema de expertos pasa del 93,7 al 95,2 % en la cinta.
- **Colapso**: el error RMS de velocidad supera la mitad de la consigna; la máquina queda lejos de su referencia, con el rotor trancado o girando muy por debajo de ella. En el subconjunto nominal común colapsan sin corrección el 44,4 % de las condiciones desajustadas de cinta y el 14,3 % de las de bomba; con el sistema de expertos, el 1,6 y el 0 %, y con la red única, ninguna. De las 28 condiciones de cinta que colapsan sin corrección, 21 quedan con el rotor trancado, entre 2 y 26 r/min, y 7 giran entre

445 y 673 r/min; las 4 de bomba giran entre 424 y 695 r/min, y la única del sistema de expertos, a 158 r/min.

- **Protege el caso ajustado**: en una condición que parte con el modelo ajustado, la mediana de la resistencia aplicada en la ventana queda a menos de 0,05 Ω de 0,83 Ω. Aunque se evalúa al final, el resultado vale para toda la corrida: ninguna de las trece condiciones ajustadas sale de esa banda y vuelve. El sistema de expertos no mueve el parámetro en ninguna; la red única lo mueve en tres, dos dentro de la banda y una fuera, que se aleja desde 1,05 s y no regresa. Esa es la definición de **falso positivo** (Subsección 7.2.2).
- **Corrida fallida**: ninguna de las 2418 corridas de la Propuesta 1 ni de las 208 de la Propuesta 2 terminó con un error de simulación; todas llegan a su tiempo final y entregan todos sus indicadores. En este capítulo, fallar significa no alcanzar la consigna.

Cómo se calcula la distorsión armónica. Para cada fase, por ejemplo i_a, se toma la corriente de la ventana final, 5000 muestras, se le resta su media y se calcula su espectro con una ventana de Hann, con una resolución de 2 Hz. En ese espectro se lee la amplitud de la fundamental, $I_{a,1}$, que es el pico más alto, y la de los armónicos $h = 2, \ldots, 10$, cada una como el máximo del espectro a menos de 4 Hz de h veces la fundamental. La distorsión de la fase es

$$\mathrm{THD}_a = \frac{\sqrt{\sum_{h=2}^{10} I_{a,h}^2}}{I_{a,1}},$$

y el indicador de la Tabla 7.1 combina las tres fases en media cuadrática. Como solo suma hasta el décimo armónico, 500 Hz a velocidad nominal, mide la distorsión de bajo orden: el rizado de conmutación, de frecuencia mayor, queda fuera. Las dos campañas lo calculan con el mismo programa.

No es el descriptor que ve la red. La característica `thd` de la Propuesta 1 (Tabla 5.4) se calcula de otro modo: sobre un solo ciclo, con una DFT rectangular del vector espacial y solo con las componentes que giran en el sentido de la fundamental (Subsección 5.4.2.a). No son intercambiables: en el tramo con carga de la rutina de la Sección 7.3, con el modelo ajustado, el descriptor vale 4,31 % en mediana por ciclo y el indicador 2,84 % sobre la máquina nominal, y 5,04 y 1,09 % sobre la de 3,00 Ω. La Propuesta 2 no usa un descriptor de distorsión: su red recibe la ventana de corriente.

Estructura de la sección. La Subsección 7.1.2 presenta la rutina completa a velocidad nominal sobre la máquina de resistencia nominal, y la Subsección 7.1.3 la repite sobre las otras cuatro resistencias reales evaluadas. La Subsección 7.1.4 presenta la carga de bomba, cuya ley de afinidad plantea una exigencia distinta, y la Subsección 7.1.5 repite la vista principal a velocidad reducida.

7.1.2. Rutina cinta a velocidad nominal: resistencia de rotor nominal 0.83 Ω

Vista principal de la campaña sobre la máquina de resistencia nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$): la máquina arranca con la carga inicial que indica la columna y a mitad de rutina recibe un escalón que la lleva al nivel que indica la fila. Son 63 condiciones desajustadas por modo de operación, a 1440 r/min.

La Tabla 7.2 resume el bloque antes de presentarlo: promedia las nueve facetas de carga de cada escalón, de modo que cada columna condensa un punto de la escalera y cada bloque de filas, un modo de operación. Leída de izquierda a derecha reproduce el eje horizontal de las figuras que siguen.

Tabla 7.2: Síntesis de las figuras del bloque: máquina de $R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$, carga de cinta (variable) y velocidad nominal.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	608,4	5,5	7,5	250,6	568,3	877,4	954,9	1021,6
THD de corriente [%]	32,62	1,98	1,53	1,07	0,96	1,04	1,33	1,51
Error estim. de torque [N·m]	26,73	0,01	9,09	19,90	31,65	39,79	44,59	40,18
Ripple de torque [N·m]	23,70	1,56	1,13	0,95	0,78	0,73	0,66	0,27
Error estim. de flujo [Wb]	0,2155	0,0004	0,1971	0,3161	0,3990	0,4746	0,5209	0,5598
MPC + PINN arranque								
Residuo de R_r [Ω]	0,211	0,000	0,241	0,348	0,291	0,513	0,361	0,367
Error de velocidad [r/min]	34,9	5,5	6,3	6,8	12,1	147,9	15,8	17,2
THD de corriente [%]	4,53	1,98	1,20	1,11	0,95	0,93	0,98	1,06
Error estim. de torque [N·m]	2,84	0,01	4,05	6,06	5,03	9,33	6,38	6,51
Ripple de torque [N·m]	1,87	1,56	1,31	1,24	1,28	1,18	1,22	1,22
Error estim. de flujo [Wb]	0,1068	0,0004	0,1039	0,1450	0,1219	0,1780	0,1476	0,1491
MPC + PINN arr.+carga								
Residuo de R_r [Ω]	0,077	0,000	0,083	0,072	0,091	0,294	0,127	0,126
Error de velocidad [r/min]	15,0	5,5	5,5	5,5	5,6	147,0	13,6	10,9
THD de corriente [%]	2,57	1,98	1,83	2,05	2,06	1,24	2,75	2,24
Error estim. de torque [N·m]	0,80	0,01	0,37	0,31	0,49	5,05	0,64	0,60
Ripple de torque [N·m]	1,70	1,56	1,53	1,50	1,48	1,37	1,74	1,63
Error estim. de flujo [Wb]	0,0370	0,0004	0,0166	0,0129	0,0170	0,0839	0,0352	0,0329

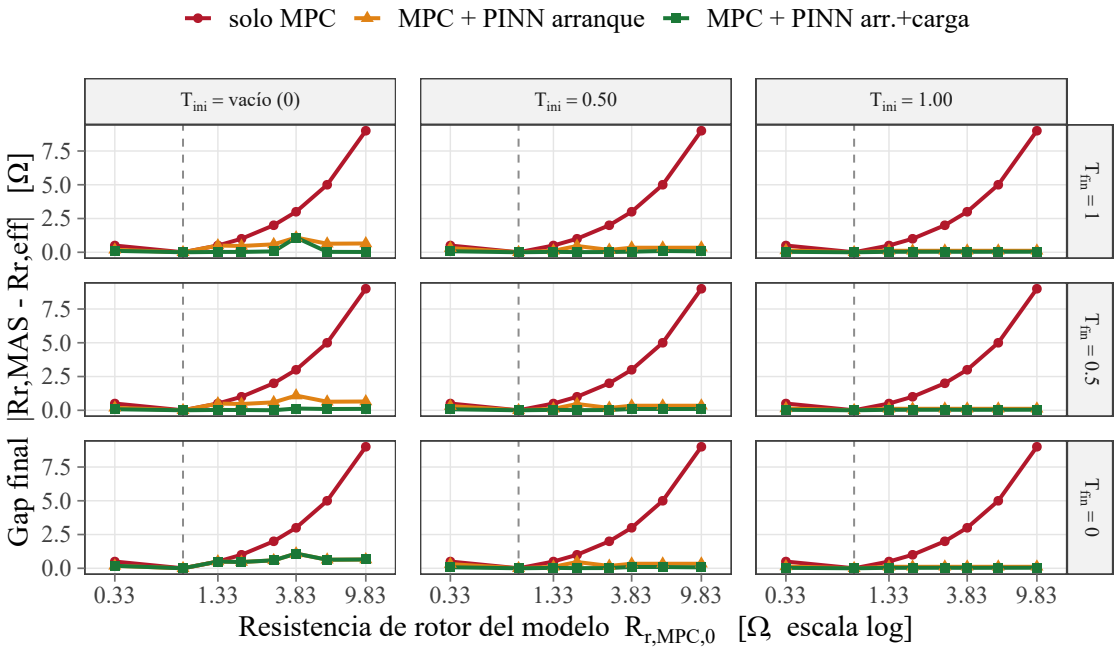


Figura 7.1: Residuo de la resistencia de rotor: distancia entre el modelo del controlador y el valor real de la máquina al terminar la rutina. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

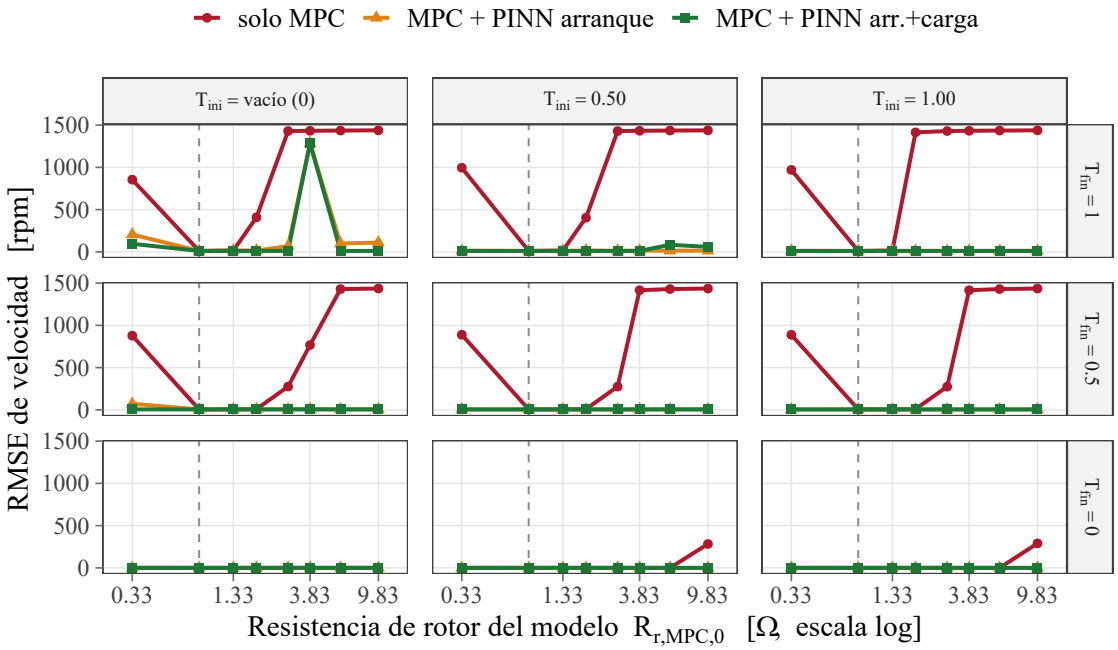


Figura 7.2: Error de seguimiento de velocidad respecto de su referencia al terminar la rutina. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

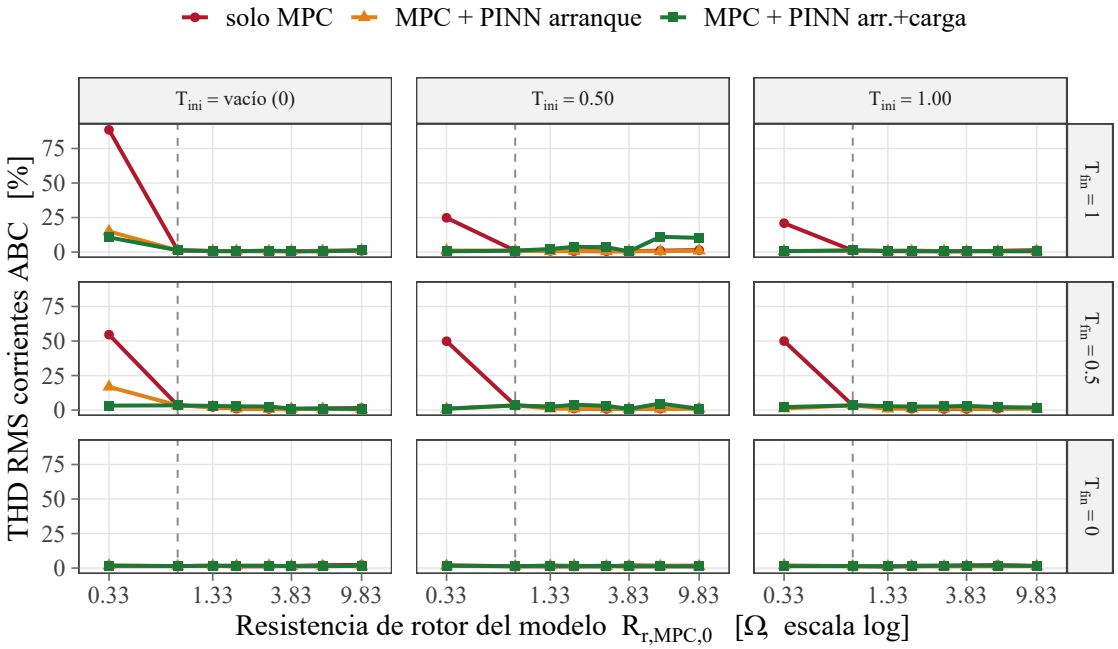


Figura 7.3: Distorsión armónica de las corrientes de estator. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

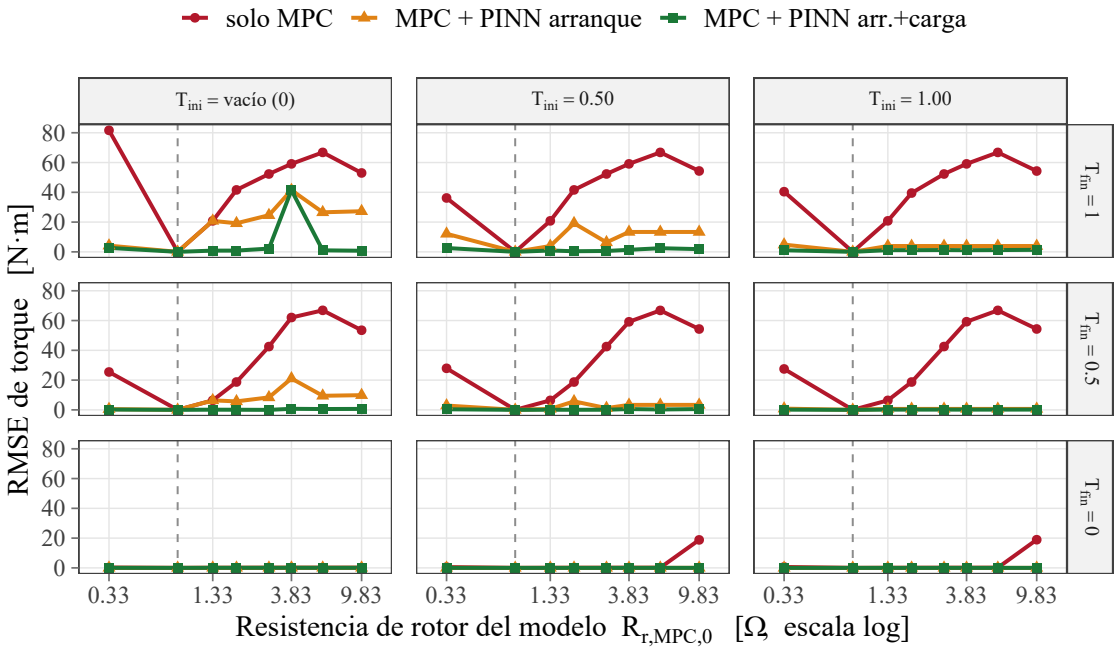


Figura 7.4: Error de estimación de torque: discrepancia entre el torque real y el que estima el observador del controlador. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

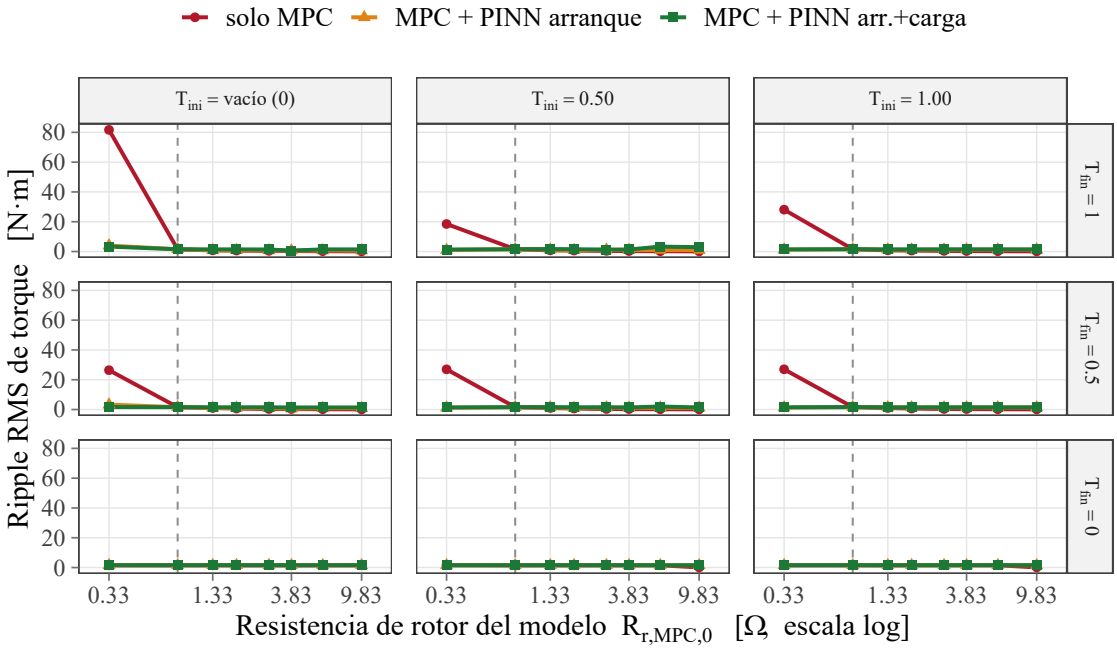


Figura 7.5: *Ripple* RMS del torque entregado por la máquina. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

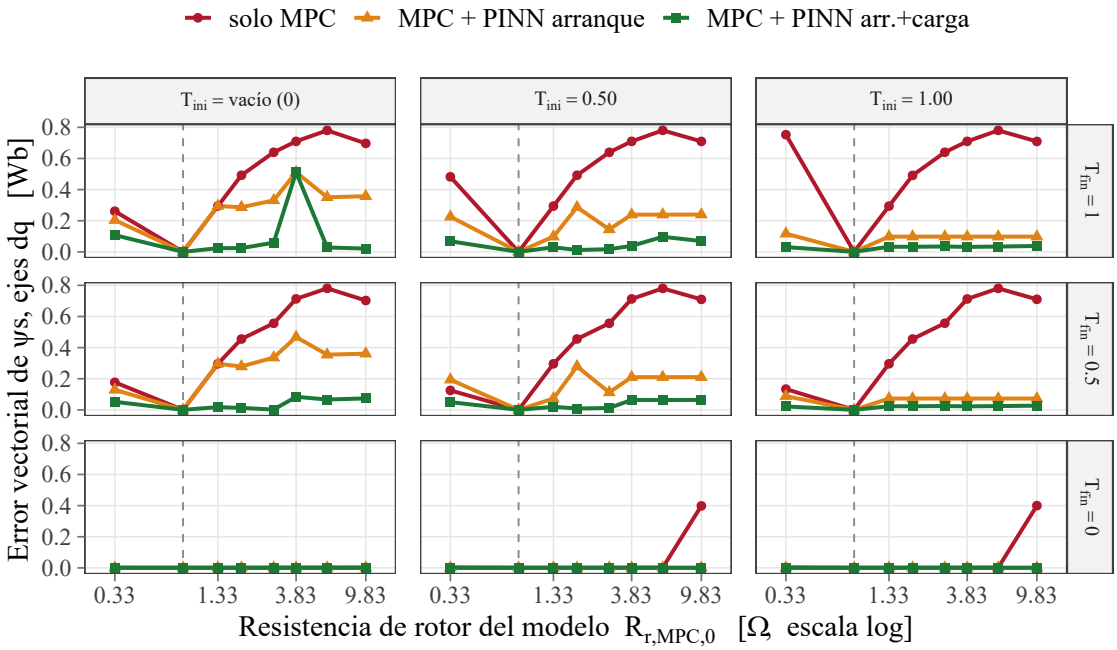


Figura 7.6: Error vectorial de estimación del flujo de estator, en ejes *dq* (Tabla 7.1). Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

Conviene observar que el escalón $R_{r,\mathrm{MPC},0} = 3{,}83\ \Omega$ rompe la monotonía y deja un residuo mayor que los de 5,83 y 9,83 Ω, que son desajustes más severos; su origen no está en el alcance de las redes sino en la interacción entre el límite de tasa de corrección y la duración de la ventana de arranque, y se documenta en el Anexo E.

7.1.3. Rutina completa a velocidad nominal, para cada resistencia de rotor real

Esta subsección repite la rutina y la velocidad para otras cuatro configuraciones de resistencia real del rotor. Lo medido aquí es la **capacidad frente a distintos valores de $R_{r,\mathrm{MAS}}$, la mayoría dentro del dominio de entrenamiento**, no una transferencia a otra máquina. La campaña usa cinco valores de $R_{r,\mathrm{MAS}}$ con las demás magnitudes de la máquina invariantes. Las excepciones son de extrapolación y conviene tenerlas presentes al leer esta subsección: 0,10 Ω queda por debajo de las menores resistencias reales con que se entrenaron los dos expertos, 0,30 Ω el de arranque y 0,83 Ω el de carga; 0,50 Ω queda por debajo del mínimo del experto de carga; y los puntos de partida de 12 y 14 Ω de las configuraciones de 3,00 y 5,00 Ω superan el mayor valor inicial del modelo visto en entrenamiento, 10 Ω. Los valores de 0,50 Ω (para el experto de arranque), 3,00 y 5,00 Ω no están exactamente en el entrenamiento, pero quedan entre valores entrenados. El punto de partida de 9,50 Ω de la configuración de 0,50 Ω, diecinueve veces el valor real, cae además en la banda de razones entre 13,3 y 20 en que el conjunto de arranque no tiene ejemplos.

Recordar que cada configuración tiene su propia escalera de $R_{r,\mathrm{MPC},0}$, centrada en su valor real, de modo que el eje horizontal recorre en todos los casos el desajuste *relativo* y

la línea vertical discontinua se sitúa en una resistencia distinta en cada figura.

Y de los siete indicadores se grafican aquí sólo los dos que responden directamente a esa pregunta: el **residuo de la resistencia de rotor**, que dice cuánto se acerca el modelo del controlador al valor real, y el **error de seguimiento de velocidad**, que dice si el accionamiento entrega el punto de operación pedido. Los cinco restantes: distorsión armónica de corriente, errores de estimación de torque y flujo, y sus respectivos indicadores de *ripple*, se presentan para cada configuración en la Sección D.1 del Anexo D.

Sus valores figuran en las Tablas 7.3 a 7.6, una por máquina, construidas igual que la del bloque anterior: un bloque de filas por modo de operación y una columna por escalón de la escalera propia de esa máquina, con el caso ajustado sombreado.

Tabla 7.3: Síntesis de las figuras del bloque: máquina de $R_{r,\mathrm{MAS}} = 0{,}10\ \Omega$, carga de cinta (variable) y velocidad nominal.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]						
	0,10	0,60	1,10	2,10	3,10	5,10	9,10
Solo MPC							
Residuo de R_r [Ω]	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	5,5	949,3	957,1	959,6	1036,2	1149,4	1316,0
THD de corriente [%]	2,02	74,02	29,80	1,79	0,94	1,08	1,48
Error estim. de torque [N·m]	0,01	40,60	44,91	50,73	62,89	70,55	61,18
Ripple de torque [N·m]	1,57	2,44	2,27	1,73	0,20	0,04	0,02
Error estim. de flujo [Wb]	0,0004	0,5555	0,6417	0,7712	0,7794	0,8450	0,7765
MPC + PINN arranque							
Residuo de R_r [Ω]	0,000	0,666	0,505	0,991	0,931	1,071	0,643
Error de velocidad [r/min]	5,5	809,0	551,8	619,6	599,1	742,6	709,8
THD de corriente [%]	2,02	62,83	51,55	24,09	13,41	22,93	79,05
Error estim. de torque [N·m]	0,01	41,49	27,90	35,96	29,56	38,29	46,86
Ripple de torque [N·m]	1,57	9,77	12,31	11,30	5,06	8,54	23,10
Error estim. de flujo [Wb]	0,0004	0,5350	0,3685	0,5073	0,3984	0,4834	0,4590
MPC + PINN arr.+carga							
Residuo de R_r [Ω]	0,000	0,465	0,470	0,895	0,835	0,939	0,643
Error de velocidad [r/min]	5,5	635,1	398,1	619,4	598,9	722,4	709,8
THD de corriente [%]	2,02	49,07	46,26	26,30	15,60	23,25	79,05
Error estim. de torque [N·m]	0,01	38,04	22,25	35,54	29,25	38,04	46,88
Ripple de torque [N·m]	1,57	8,48	11,51	11,26	5,13	8,92	23,11
Error estim. de flujo [Wb]	0,0004	0,4615	0,3312	0,5012	0,3948	0,4778	0,4597

Tabla 7.4: Síntesis de las figuras del bloque: máquina de $R_{r,\mathrm{MAS}} = 0{,}50\ \Omega$, carga de cinta (variable) y velocidad nominal.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]						
	0,50	1,00	1,50	2,50	3,50	5,50	9,50
Solo MPC							
Residuo de R_r [Ω]	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	5,5	40,9	480,6	921,8	955,1	957,0	1084,4
THD de corriente [%]	2,11	1,03	1,20	0,99	0,96	1,27	1,22
Error estim. de torque [N·m]	0,01	16,18	27,74	39,88	43,29	47,36	45,61
Ripple de torque [N·m]	1,55	1,00	0,85	0,71	0,65	0,58	0,20
Error estim. de flujo [Wb]	0,0004	0,2868	0,3804	0,4844	0,5201	0,5668	0,6307
MPC + PINN arranque							
Residuo de R_r [Ω]	0,000	0,333	0,271	0,335	0,331	0,378	0,332
Error de velocidad [r/min]	5,5	29,1	23,6	163,5	163,4	164,1	163,5
THD de corriente [%]	2,11	1,32	1,39	1,60	12,12	1,53	12,20
Error estim. de torque [N·m]	0,01	10,79	8,52	9,87	9,90	11,03	9,90
Ripple de torque [N·m]	1,55	1,18	1,24	1,22	1,22	1,25	1,22
Error estim. de flujo [Wb]	0,0004	0,1913	0,1637	0,1766	0,1770	0,1826	0,1769
MPC + PINN arr.+carga							
Residuo de R_r [Ω]	0,000	0,069	0,110	0,150	0,150	0,198	0,164
Error de velocidad [r/min]	5,5	5,6	5,7	135,6	141,6	161,4	132,1
THD de corriente [%]	2,11	1,53	1,72	1,73	1,72	9,36	1,89
Error estim. de torque [N·m]	0,01	0,93	1,64	4,30	4,14	4,72	4,21
Ripple de torque [N·m]	1,55	1,50	1,49	1,48	1,52	1,42	1,47
Error estim. de flujo [Wb]	0,0004	0,0258	0,0532	0,0671	0,0618	0,0795	0,0646

Tabla 7.5: Síntesis de las figuras del bloque: máquina de $R_{r,\mathrm{MAS}} = 3{,}00\ \Omega$, carga de cinta (variable) y velocidad nominal.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]									
	1,00	2,00	2,50	**3,00**	3,50	4,00	5,00	6,00	8,00	12,00
Solo MPC										
Residuo de R_r [Ω]	2,000	1,000	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	629,0	199,4	110,2	5,4	33,3	97,1	211,5	414,9	781,4	955,7
THD de corriente [%]	23,77	17,79	9,23	1,68	1,07	1,00	1,33	1,24	1,57	1,89
Error estim. de torque [N·m]	27,23	2,25	1,25	0,01	2,36	5,06	11,33	17,47	25,85	19,36
Ripple de torque [N·m]	19,73	3,32	2,58	1,38	1,25	1,16	0,99	0,89	0,82	0,69
Error estim. de flujo [Wb]	0,2594	0,1743	0,0813	0,0004	0,0628	0,1179	0,1942	0,2441	0,3347	0,3120
MPC + PINN arranque										
Residuo de R_r [Ω]	0,705	0,591	0,232	0,000	0,361	0,593	0,664	0,255	0,465	0,271
Error de velocidad [r/min]	122,4	88,6	35,4	5,4	24,0	67,7	115,5	31,3	44,7	26,0
THD de corriente [%]	10,01	7,91	4,01	1,68	1,10	3,59	8,78	2,40	2,76	1,70
Error estim. de torque [N·m]	1,94	1,83	0,68	0,01	1,69	2,55	1,98	1,07	2,15	1,32
Ripple de torque [N·m]	2,53	2,19	1,85	1,38	1,30	1,66	2,37	1,47	1,40	1,32
Error estim. de flujo [Wb]	0,1148	0,0934	0,0371	0,0004	0,0459	0,0780	0,1045	0,0349	0,0593	0,0336
MPC + PINN arr.+carga										
Residuo de R_r [Ω]	0,193	0,222	0,139	0,000	0,150	0,201	0,142	0,091	0,202	0,052
Error de velocidad [r/min]	5,4	5,5	5,5	5,4	5,5	5,5	5,5	5,5	5,5	5,4
THD de corriente [%]	3,08	2,50	1,64	1,68	2,07	2,34	2,13	1,89	1,95	1,84
Error estim. de torque [N·m]	0,34	0,30	0,33	0,01	0,46	0,46	0,38	0,22	0,47	0,23
Ripple de torque [N·m]	1,38	1,38	1,47	1,38	1,46	1,41	1,40	1,45	1,38	1,35
Error estim. de flujo [Wb]	0,0133	0,0148	0,0136	0,0004	0,0161	0,0157	0,0142	0,0074	0,0154	0,0082

Tabla 7.6: Síntesis de las figuras del bloque: máquina de $R_{r,\mathrm{MAS}} = 5{,}00\ \Omega$, carga de cinta (variable) y velocidad nominal.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]										
	2,00	3,00	4,00	4,50	**5,00**	5,50	6,00	7,00	8,00	10,00	14,00
Solo MPC											
Residuo de R_r [Ω]	3,000	2,000	1,000	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	647,8	266,9	180,6	102,6	59,8	119,0	172,7	300,2	494,5	800,7	955,5
THD de corriente [%]	23,72	22,17	10,31	5,83	2,11	1,26	1,60	1,68	1,42	1,34	2,15
Error estim. de torque [N·m]	22,86	2,77	1,45	0,92	0,01	1,47	3,09	6,93	10,42	15,32	12,19
Ripple de torque [N·m]	17,56	3,49	2,66	1,88	1,25	1,26	1,20	1,02	0,97	0,89	0,73
Error estim. de flujo [Wb]	0,3071	0,2163	0,0984	0,0487	0,0005	0,0390	0,0739	0,1174	0,1643	0,2302	0,2294
MPC + PINN arranque											
Residuo de R_r [Ω]	0,463	0,791	0,446	0,340	0,000	0,408	0,689	0,122	0,882	0,028	0,449
Error de velocidad [r/min]	98,8	124,3	93,3	87,0	59,8	90,8	138,9	73,5	164,8	61,7	89,7
THD de corriente [%]	4,55	9,07	5,04	5,02	2,11	1,68	1,63	2,49	7,24	1,26	1,63
Error estim. de torque [N·m]	0,80	1,22	0,77	0,63	0,01	1,17	2,08	0,35	1,62	0,12	1,37
Ripple de torque [N·m]	1,87	2,10	1,82	1,68	1,25	1,26	1,24	1,27	2,18	1,33	1,38
Error estim. de flujo [Wb]	0,0462	0,0837	0,0429	0,0330	0,0005	0,0323	0,0521	0,0100	0,0819	0,0033	0,0351
MPC + PINN arr.+carga											
Residuo de R_r [Ω]	0,228	0,291	0,228	0,206	0,000	0,195	0,134	0,105	0,209	0,028	0,146
Error de velocidad [r/min]	58,8	59,4	62,4	60,3	59,8	54,3	54,5	68,0	78,6	61,7	76,3
THD de corriente [%]	1,70	1,39	2,41	2,13	2,11	1,77	2,26	2,58	2,41	1,26	1,85
Error estim. de torque [N·m]	0,28	0,16	0,40	0,27	0,01	0,33	0,25	0,23	0,57	0,12	0,41
Ripple de torque [N·m]	1,37	1,24	1,35	1,25	1,25	1,25	1,26	1,26	1,83	1,33	1,34
Error estim. de flujo [Wb]	0,0101	0,0067	0,0127	0,0126	0,0005	0,0121	0,0097	0,0080	0,0189	0,0033	0,0123

El resultado es que **la corrección no queda restringida al valor nominal de placa**. Entre 0,50 y 5,00 Ω, el residuo permanece entre 0,124 y 0,177 Ω, y los indicadores que dependen del modelo interno *mejoran* al crecer la resistencia real: el error de estimación de torque cae de 3,32 a 0,30 N·m y el de flujo de 0,059 a 0,011 Wb entre las configuraciones de 0,50 y 5,00 Ω. La configuración de 0,10 Ω, en cambio, queda fuera del alcance útil: el residuo promedio de 0,708 Ω equivale a siete veces su propia resistencia, y allí la capa no mejora el *ripple* de torque y la distorsión respecto de no actuar, porque el controlador sigue operando con un modelo todavía gravemente equivocado. Es además la única configuración que queda por debajo de la menor resistencia real con que se entrenaron los dos expertos, y cuatro de sus seis puntos de partida desajustados superan las 20 veces el valor real, razón a partir de la cual las redes de arranque se entrenaron para abstenerse.

Dos observaciones sobre la fila de referencia. En la máquina de 5 Ω el modelo ajustado ya arrastra 59,8 r/min de error de seguimiento, frente a los 5,5 r/min de las demás, que son la caída propia del controlador de velocidad: a esa resistencia y con la cinta cargada al máximo, la máquina no alcanza su consigna ni con el modelo correcto, de modo que los 63,4 r/min de la capa completa están sobre el límite físico y no sobre el ideal.

7.1.3.a. Máquina de 0,10 Ω

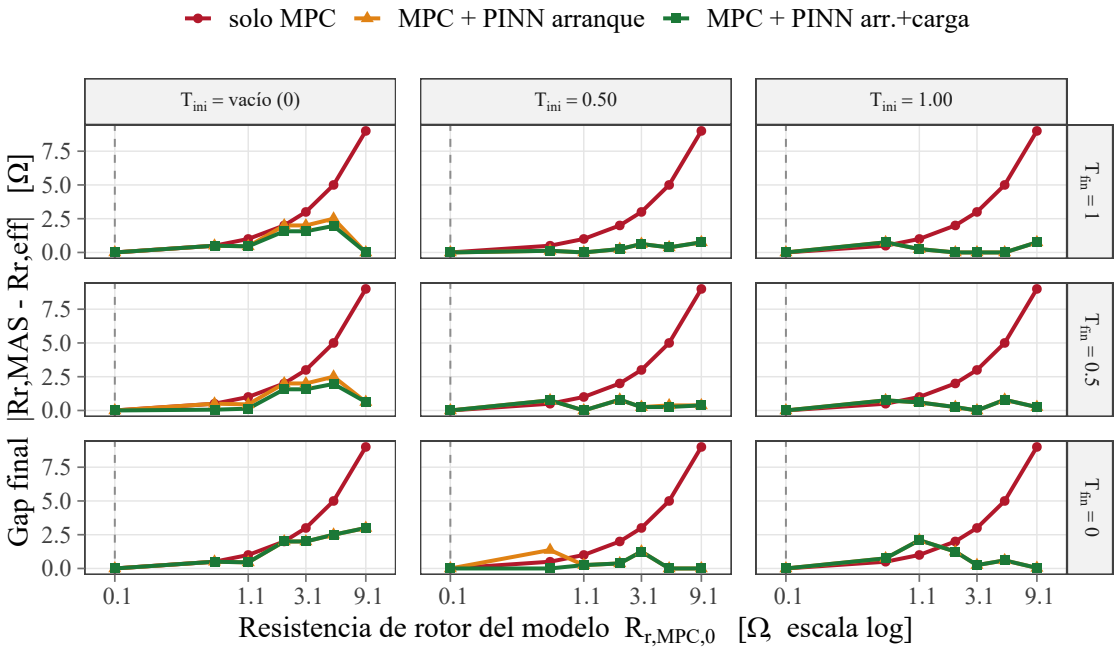


Figura 7.7: Residuo de la resistencia de rotor: distancia entre el modelo del controlador y el valor real de la máquina al terminar la rutina. Máquina de 0,10 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

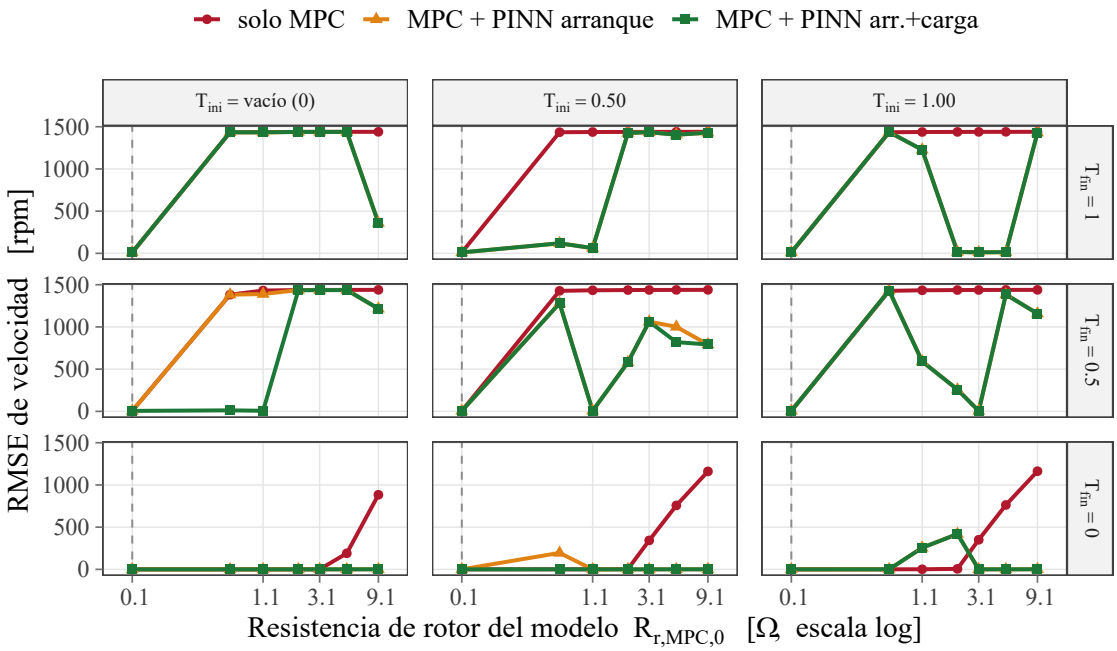


Figura 7.8: Error de seguimiento de velocidad respecto de su referencia al terminar la rutina. Máquina de 0,10 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

7.1.3.b. Máquina de 0,50 Ω

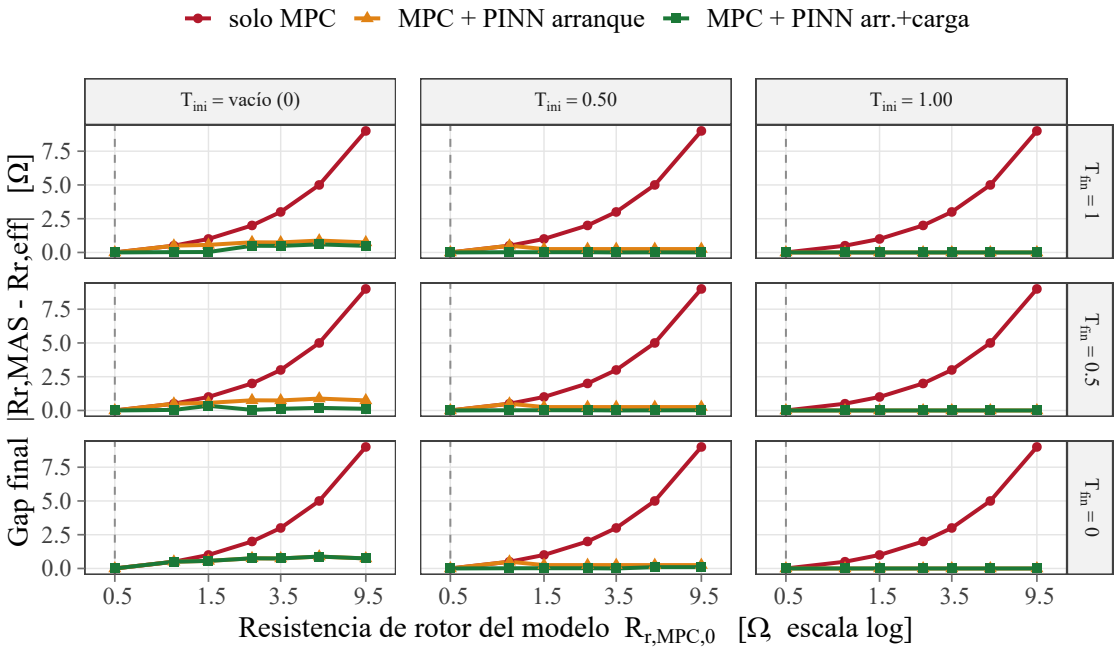


Figura 7.9: Residuo de la resistencia de rotor: distancia entre el modelo del controlador y el valor real de la máquina al terminar la rutina. Máquina de 0,50 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

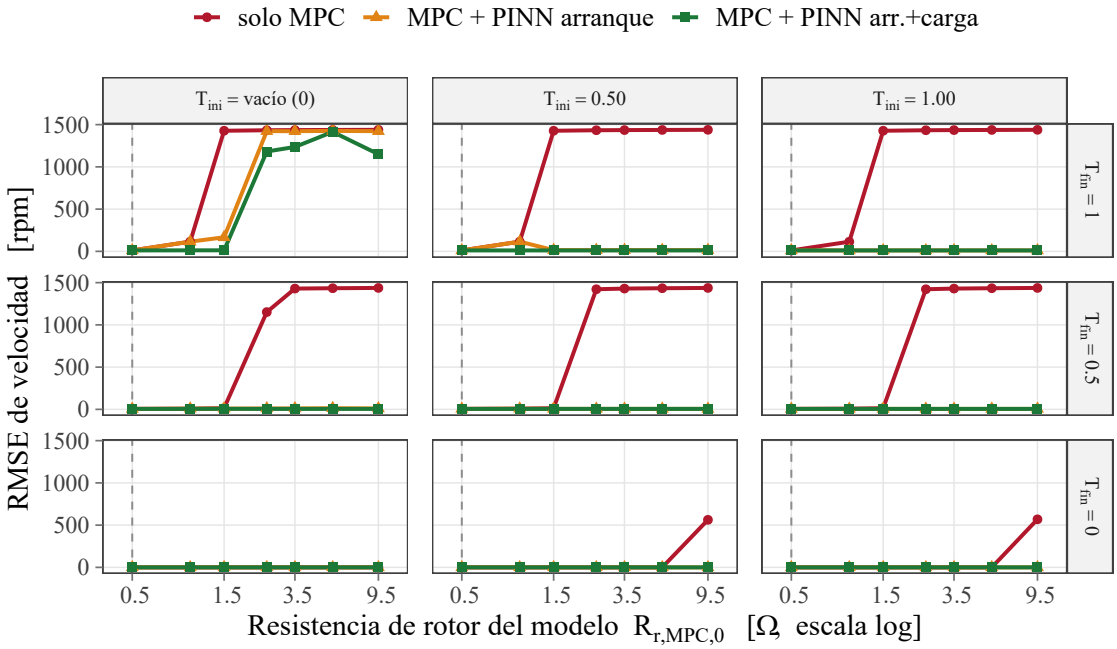


Figura 7.10: Error de seguimiento de velocidad respecto de su referencia al terminar la rutina. Máquina de 0,50 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

7.1.3.c. Máquina de 3,00 Ω

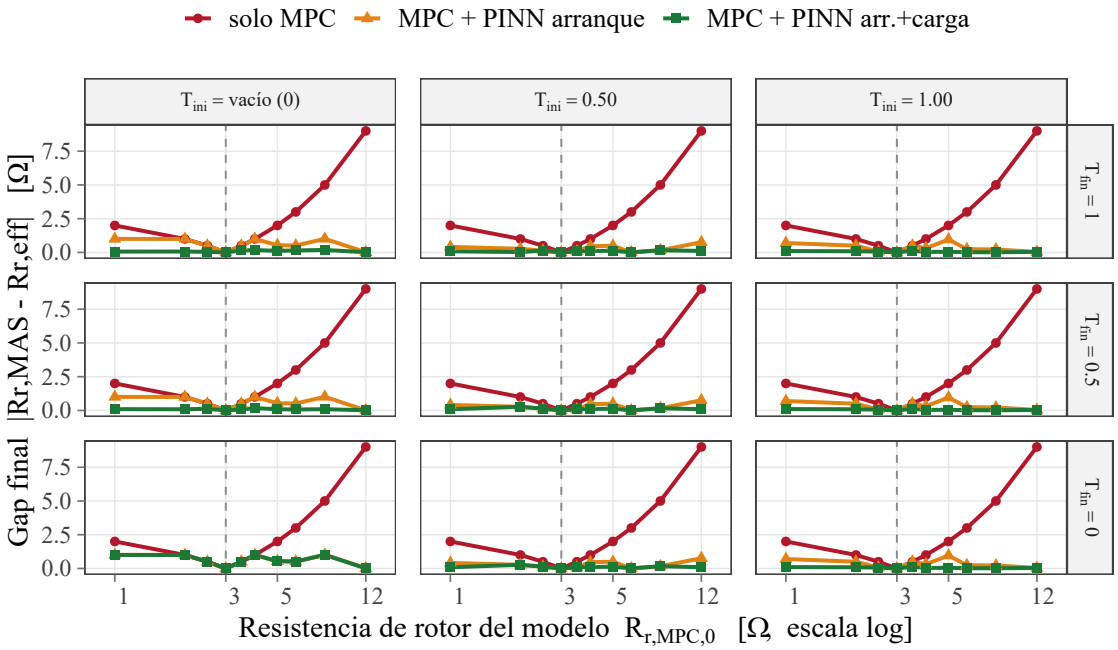


Figura 7.11: Residuo de la resistencia de rotor: distancia entre el modelo del controlador y el valor real de la máquina al terminar la rutina. Máquina de 3,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

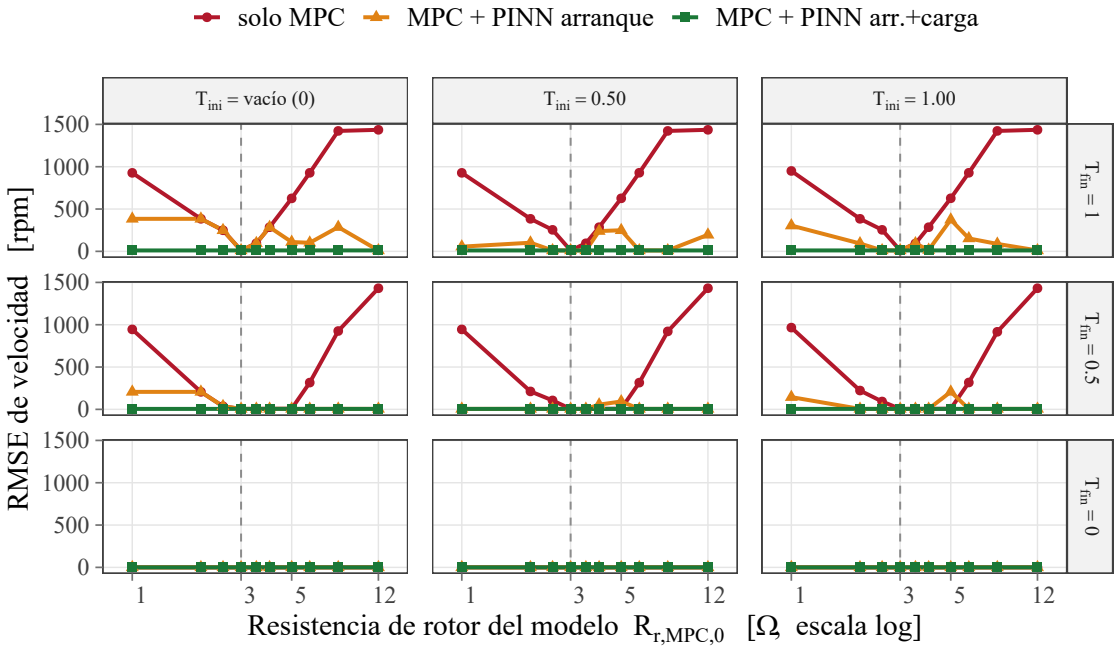


Figura 7.12: Error de seguimiento de velocidad respecto de su referencia al terminar la rutina. Máquina de 3,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

7.1.3.d. Máquina de 5,00 Ω

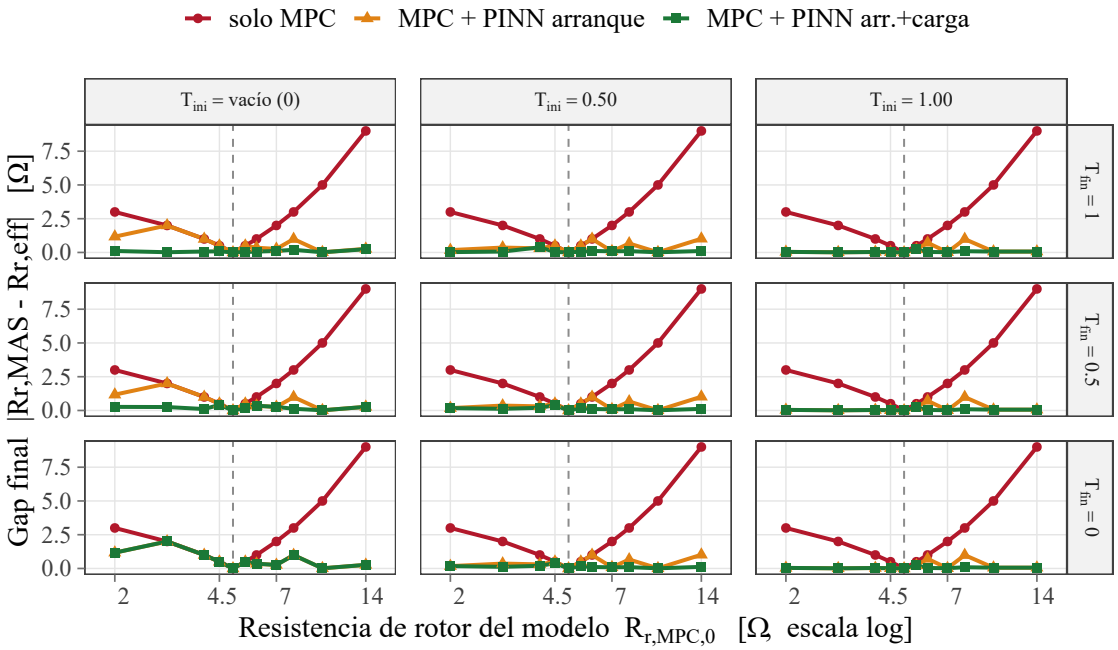


Figura 7.13: Residuo de la resistencia de rotor: distancia entre el modelo del controlador y el valor real de la máquina al terminar la rutina. Máquina de 5,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

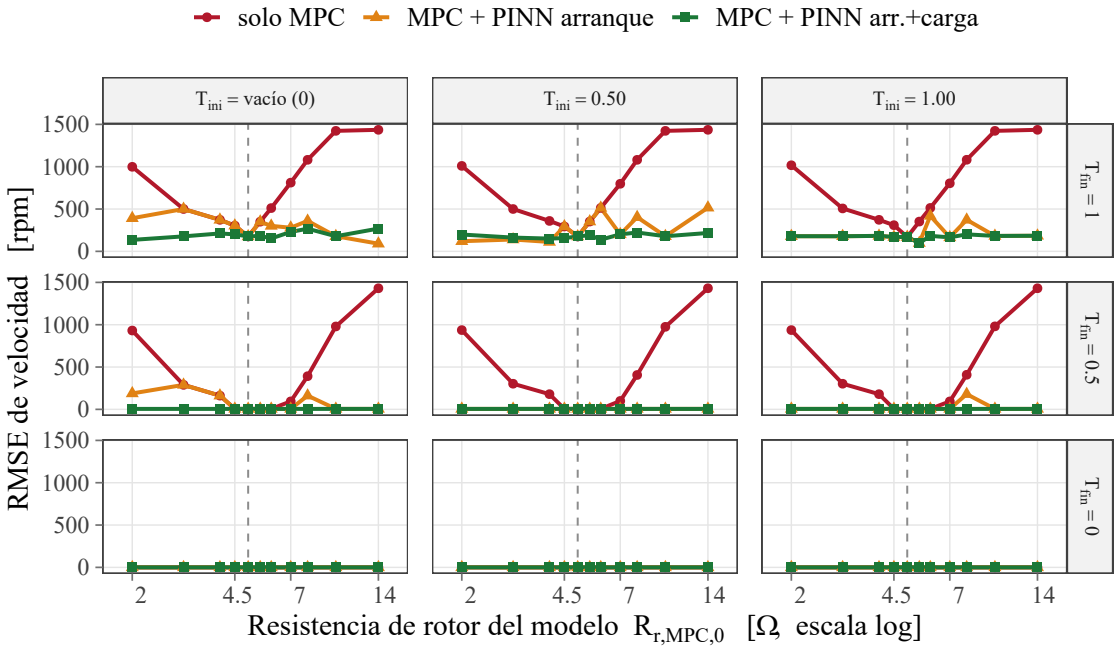


Figura 7.14: Error de seguimiento de velocidad respecto de su referencia al terminar la rutina. Máquina de 5,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

7.1.4. Carga de bomba

La carga de bomba descrita en la Sección 4.1.2 depende cuadráticamente de la velocidad, de modo que no se aplica como un escalón de torque independiente. Las facetas recorren aquí las cuatro referencias de velocidad, de 750 a 1440 r/min. Son 28 condiciones desajustadas por modo sobre la configuración nominal.

Este resultado es diferente respecto de la carga tipo cinta, porque hay dos indicadores que apuntan en direcciones opuestas: con carga de bomba la identificación es **peor**, pero la condición de operación es **mejor**. La carga de bomba disminuye al caer la velocidad, de modo que basta una corrección parcial para que la máquina *escape* del punto degradado y se estabilice. La cinta, en cambio, impone un torque que no cede y exige una corrección más precisa para superarlo; cuando la consigue, la señal disponible es más identificable.

Las Tablas 7.7 a 7.10 resumen el bloque que sigue, una por referencia de velocidad. A diferencia de las de cinta, cada celda corresponde aquí a una *única* corrida: con esta carga la referencia de velocidad fija también el torque solicitado, de modo que no existen niveles de carga que promediar.

Tabla 7.7: Carga de bomba a 750 r/min sobre la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$). Las columnas recorren la resistencia con que se inicializa el modelo del controlador; la sombreada es el **caso ajustado**, en el que la capa no debe actuar.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	4,9	2,9	2,1	1,7	5,6	9,2	55,2	325,9
THD de corriente [%]	1,28	2,07	1,07	1,43	1,44	4,12	1,58	1,37
Error estim. de torque [N·m]	5,77	0,00	1,96	2,63	13,54	27,83	49,89	33,78
Ripple de torque [N·m]	2,33	1,52	1,38	1,31	0,59	1,60	0,19	0,08
Error estim. de flujo [Wb]	0,2689	0,0002	0,1214	0,2199	0,6351	0,6903	0,6575	0,5487
MPC + PINN arranque								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	3,297	2,658
Error de velocidad [r/min]	4,9	2,9	2,1	1,7	5,6	9,2	10,6	8,4
THD de corriente [%]	1,28	2,07	1,07	1,43	1,44	4,12	0,90	1,65
Error estim. de torque [N·m]	5,77	0,00	1,96	2,63	13,54	27,83	32,66	24,47
Ripple de torque [N·m]	2,33	1,52	1,38	1,31	0,59	1,60	0,79	0,96
Error estim. de flujo [Wb]	0,2689	0,0002	0,1214	0,2199	0,6351	0,6903	0,7584	0,6838
MPC + PINN arr.+carga								
Residuo de R_r [Ω]	0,307	0,000	0,500	1,000	1,418	3,000	3,297	2,658
Error de velocidad [r/min]	3,8	2,9	2,1	1,7	1,9	9,2	10,6	8,4
THD de corriente [%]	2,16	2,07	1,07	1,43	2,11	4,12	0,90	1,65
Error estim. de torque [N·m]	2,58	0,00	1,96	2,63	1,33	27,83	32,66	24,47
Ripple de torque [N·m]	1,85	1,52	1,38	1,31	1,15	1,60	0,79	0,96
Error estim. de flujo [Wb]	0,1254	0,0002	0,1214	0,2199	0,3610	0,6903	0,7584	0,6838

Tabla 7.8: Carga de bomba a 1000 r/min sobre la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$). Las columnas recorren la resistencia con que se inicializa el modelo del controlador; la sombreada es el **caso ajustado**, en el que la capa no debe actuar.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	7,3	5,2	5,7	8,2	14,2	80,0	305,2	575,9
THD de corriente [%]	3,95	1,00	0,97	1,01	0,66	0,88	1,58	1,37
Error estim. de torque [N·m]	4,63	0,01	4,40	14,73	37,40	52,19	49,89	33,78
Ripple de torque [N·m]	3,04	1,52	1,11	0,79	0,51	0,30	0,19	0,08
Error estim. de flujo [Wb]	0,4312	0,0002	0,2870	0,4919	0,6641	0,6478	0,6575	0,5487
MPC + PINN arranque								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	1,881	2,092	1,514
Error de velocidad [r/min]	7,3	5,2	5,7	8,2	14,2	13,5	14,7	11,3
THD de corriente [%]	3,95	1,00	0,97	1,01	0,66	0,65	0,65	0,65
Error estim. de torque [N·m]	4,63	0,01	4,40	14,73	37,40	34,76	39,34	26,49
Ripple de torque [N·m]	3,04	1,52	1,11	0,79	0,51	0,49	0,69	0,58
Error estim. de flujo [Wb]	0,4312	0,0002	0,2870	0,4919	0,6641	0,6542	0,6704	0,6053
MPC + PINN arr.+carga								
Residuo de R_r [Ω]	0,415	0,000	0,009	1,000	0,588	0,529	2,092	1,514
Error de velocidad [r/min]	6,5	5,2	5,2	8,2	6,1	5,9	14,7	11,3
THD de corriente [%]	1,10	1,00	0,69	1,01	0,99	1,59	0,65	0,65
Error estim. de torque [N·m]	2,77	0,01	0,03	14,73	5,96	4,89	39,34	26,49
Ripple de torque [N·m]	2,43	1,52	1,54	0,79	1,04	1,10	0,69	0,58
Error estim. de flujo [Wb]	0,3252	0,0002	0,0059	0,4919	0,3315	0,3019	0,6704	0,6053

Tabla 7.9: Carga de bomba a 1250 r/min sobre la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$). Las columnas recorren la resistencia con que se inicializa el modelo del controlador; la sombreada es el **caso ajustado**, en el que la capa no debe actuar.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	484,2	8,1	10,7	14,9	174,2	330,0	555,2	825,9
THD de corriente [%]	54,55	0,77	0,52	0,61	0,72	0,88	1,58	1,37
Error estim. de torque [N·m]	15,65	0,01	12,63	29,23	47,39	52,19	49,89	33,78
Ripple de torque [N·m]	13,44	1,53	0,98	0,68	0,39	0,30	0,19	0,08
Error estim. de flujo [Wb]	0,1037	0,0002	0,3362	0,5041	0,5883	0,6478	0,6575	0,5487
MPC + PINN arranque								
Residuo de R_r [Ω]	0,484	0,000	0,500	0,466	0,588	0,529	0,634	0,651
Error de velocidad [r/min]	10,6	8,1	10,7	10,5	11,5	11,0	11,8	11,9
THD de corriente [%]	0,49	0,77	0,52	0,78	1,23	0,45	0,82	0,53
Error estim. de torque [N·m]	12,14	0,01	12,63	11,57	15,49	13,56	16,96	17,51
Ripple de torque [N·m]	0,98	1,53	0,98	1,01	0,87	0,94	0,90	0,88
Error estim. de flujo [Wb]	0,3281	0,0002	0,3362	0,3191	0,3755	0,3497	0,3949	0,4016
MPC + PINN arr.+carga								
Residuo de R_r [Ω]	0,484	0,000	0,500	0,466	0,588	0,529	0,634	0,651
Error de velocidad [r/min]	10,6	8,1	10,7	10,5	11,5	11,0	11,8	11,9
THD de corriente [%]	0,49	0,77	0,52	0,78	1,23	0,45	0,82	0,53
Error estim. de torque [N·m]	12,14	0,01	12,63	11,57	15,49	13,56	16,96	17,51
Ripple de torque [N·m]	0,98	1,53	0,98	1,01	0,87	0,94	0,90	0,88
Error estim. de flujo [Wb]	0,3281	0,0002	0,3362	0,3191	0,3755	0,3497	0,3949	0,4016

Tabla 7.10: Carga de bomba a 1440 r/min sobre la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$). Las columnas recorren la resistencia con que se inicializa el modelo del controlador; la sombreada es el **caso ajustado**, en el que la capa no debe actuar.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	674,1	10,8	15,7	149,5	364,2	520,0	745,2	1015,9
THD de corriente [%]	54,55	1,02	0,86	0,42	0,72	0,88	1,58	1,37
Error estim. de torque [N·m]	15,65	0,01	20,25	33,84	47,39	52,19	49,89	33,78
Ripple de torque [N·m]	13,44	1,53	0,81	0,58	0,39	0,30	0,19	0,08
Error estim. de flujo [Wb]	0,1037	0,0001	0,2981	0,4415	0,5883	0,6478	0,6575	0,5487
MPC + PINN arranque								
Residuo de R_r [Ω]	0,484	0,000	0,112	0,087	0,174	0,132	0,207	0,156
Error de velocidad [r/min]	15,5	10,8	11,5	11,3	12,0	11,7	12,3	11,9
THD de corriente [%]	0,79	1,02	0,55	0,62	0,58	0,61	0,61	1,12
Error estim. de torque [N·m]	19,52	0,01	3,82	2,94	6,22	4,59	7,52	5,50
Ripple de torque [N·m]	0,83	1,53	1,28	1,33	1,20	1,25	1,15	1,22
Error estim. de flujo [Wb]	0,2935	0,0001	0,0977	0,0779	0,1444	0,1136	0,1665	0,1312
MPC + PINN arr.+carga								
Residuo de R_r [Ω]	0,030	0,000	0,031	0,087	0,027	0,036	0,029	0,029
Error de velocidad [r/min]	10,9	10,8	10,9	11,3	10,9	11,0	10,9	10,9
THD de corriente [%]	0,62	1,02	0,64	0,62	1,02	0,58	0,80	0,80
Error estim. de torque [N·m]	0,99	0,01	1,00	2,94	0,87	1,17	0,93	0,94
Ripple de torque [N·m]	1,42	1,53	1,42	1,33	1,43	1,44	1,43	1,42
Error estim. de flujo [Wb]	0,0286	0,0001	0,0289	0,0779	0,0254	0,0338	0,0269	0,0274

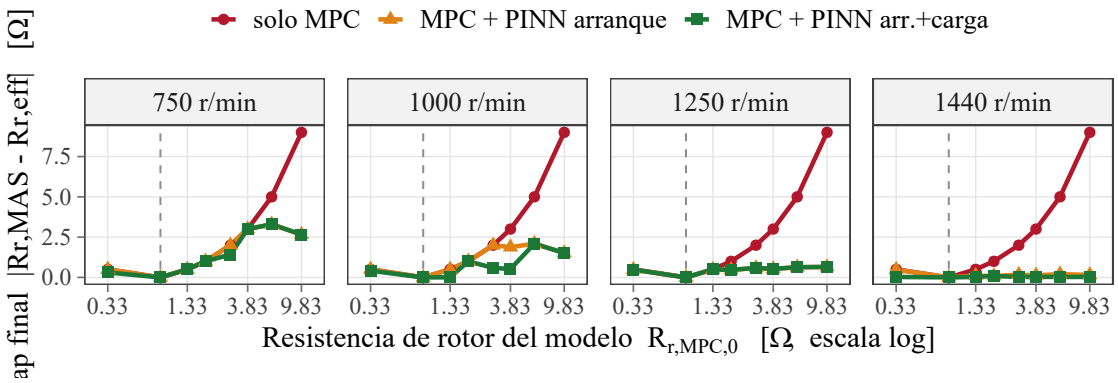


Figura 7.15: Residuo de la resistencia de rotor: distancia entre el modelo del controlador y el valor real de la máquina al terminar la rutina.

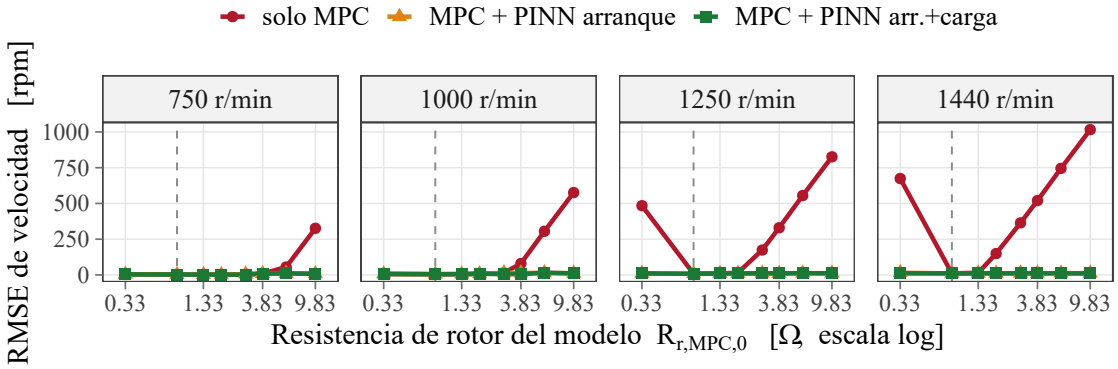


Figura 7.16: Error de seguimiento de velocidad respecto de su referencia al terminar la rutina.

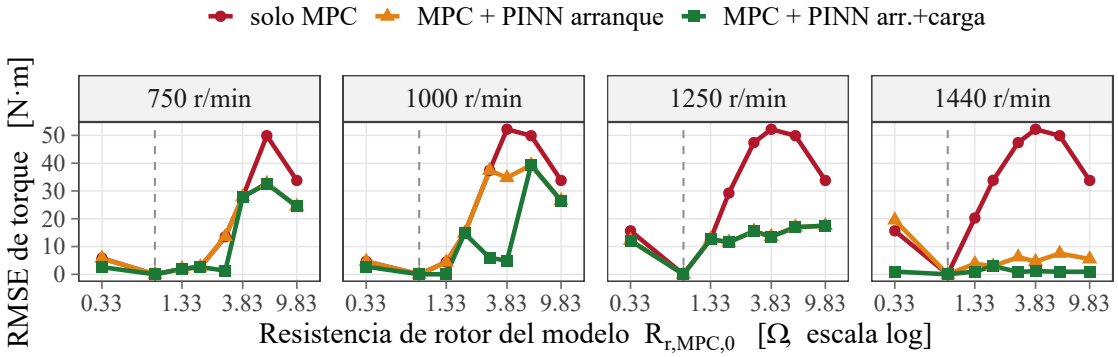


Figura 7.17: Error de estimación de torque: discrepancia entre el torque real y el que estima el observador del controlador.

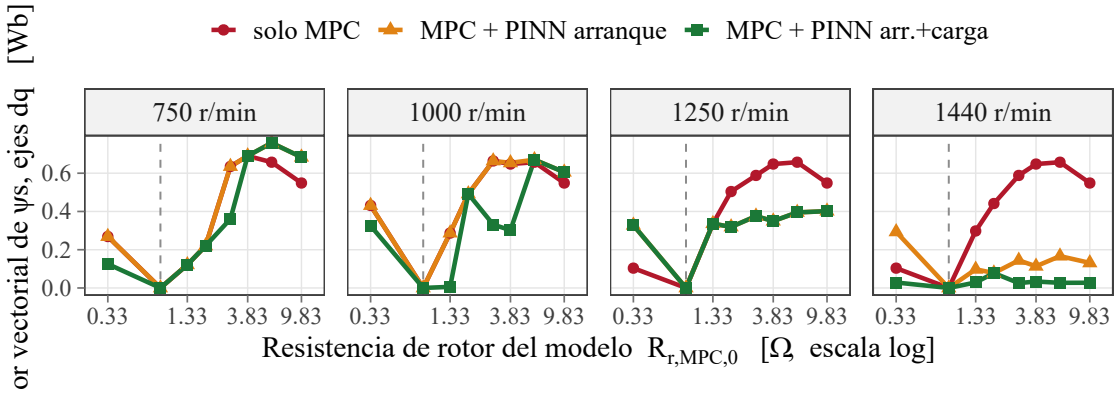


Figura 7.18: Error vectorial de estimación del flujo de estator, en ejes *dq* (Tabla 7.1).

De los siete indicadores se grafican los cuatro que discriminan: el residuo de R_r, el error de velocidad y los dos errores de estimación. Los dos de *ripple* y la distorsión se presentan en la Sección D.2 del Anexo D: con esta carga sólo se apartan de su valor nominal en el extremo de subestimación. Sus valores son 1,20 N·m de rizado de torque contra 1,53 del

modelo ajustado, 0,0179 Wb de rizado de flujo contra 0,0136, y 1,07 % de distorsión contra 1,21 %.

Los indicadores que dependen del modelo interno conservan el rastro de la identificación incompleta; los que describen la máquina real quedan restituidos. Es la manifestación más clara de un hecho que recorre todo el capítulo: **operar bien y estimar bien no son la misma cosa**, y el desajuste puede quedar oculto en las variables de la máquina mientras persiste en el modelo del controlador.

7.1.5. Modo cinta: diferente velocidad de operación

Las subsecciones anteriores de carga cinta corresponden a velocidad nominal. Y esta repite las mismas rutinas de arranque y escalón de carga, sobre la máquina nominal a 1000 r/min, es decir, al 69 % de la velocidad nominal.

Como en la subsección anterior, se grafican sólo el residuo de R_r y el error de velocidad, que son los dos indicadores en los que la comparación con la velocidad nominal es directa; los cinco restantes están en la Sección D.3 del Anexo D.

Las Tablas 7.11 y 7.12 presentan ambas velocidades una sobre otra, con idéntica máquina, idéntica carga e idéntica escalera, de modo que la comparación se lee columna a columna y lo único que las separa es la referencia de velocidad.

Tabla 7.11: Velocidad de referencia **nominal**, 1440 r/min: máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$), carga de cinta. Reproduce los datos de la Tabla 7.2 y se repite aquí para comparar columna a columna con la tabla siguiente.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	608,4	5,5	7,5	250,6	568,3	877,4	954,9	1021,6
THD de corriente [%]	32,62	1,98	1,53	1,07	0,96	1,04	1,33	1,51
Error estim. de torque [N·m]	26,73	0,01	9,09	19,90	31,65	39,79	44,59	40,18
Ripple de torque [N·m]	23,70	1,56	1,13	0,95	0,78	0,73	0,66	0,27
Error estim. de flujo [Wb]	0,2155	0,0004	0,1971	0,3161	0,3990	0,4746	0,5209	0,5598
MPC + PINN arranque								
Residuo de R_r [Ω]	0,211	0,000	0,241	0,348	0,291	0,513	0,361	0,367
Error de velocidad [r/min]	34,9	5,5	6,3	6,8	12,1	147,9	15,8	17,2
THD de corriente [%]	4,53	1,98	1,20	1,11	0,95	0,93	0,98	1,06
Error estim. de torque [N·m]	2,84	0,01	4,05	6,06	5,03	9,33	6,38	6,51
Ripple de torque [N·m]	1,87	1,56	1,31	1,24	1,28	1,18	1,22	1,22
Error estim. de flujo [Wb]	0,1068	0,0004	0,1039	0,1450	0,1219	0,1780	0,1476	0,1491
MPC + PINN arr.+carga								
Residuo de R_r [Ω]	0,077	0,000	0,083	0,072	0,091	0,294	0,127	0,126
Error de velocidad [r/min]	15,0	5,5	5,5	5,5	5,6	147,0	13,6	10,9
THD de corriente [%]	2,57	1,98	1,83	2,05	2,06	1,24	2,75	2,24
Error estim. de torque [N·m]	0,80	0,01	0,37	0,31	0,49	5,05	0,64	0,60
Ripple de torque [N·m]	1,70	1,56	1,53	1,50	1,48	1,37	1,74	1,63
Error estim. de flujo [Wb]	0,0370	0,0004	0,0166	0,0129	0,0170	0,0839	0,0352	0,0329

Tabla 7.12: Velocidad de referencia **reducida**, 1000 r/min, esto es, el 69 % de la nominal: máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$), carga de cinta. Lo único que cambia respecto de la tabla anterior es la referencia de velocidad.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	202,8	5,5	7,3	199,0	335,0	587,5	661,6	664,6
THD de corriente [%]	24,76	1,19	1,29	1,20	1,06	1,12	1,27	1,47
Error estim. de torque [N·m]	26,38	0,01	8,26	18,90	31,01	39,76	44,58	36,10
Ripple de torque [N·m]	24,90	1,53	1,19	1,00	0,91	0,73	0,66	0,57
Error estim. de flujo [Wb]	0,2712	0,0002	0,2133	0,3294	0,4336	0,4744	0,5206	0,4886
MPC + PINN arranque								
Residuo de R_r [Ω]	0,417	0,000	0,500	0,833	1,500	1,294	1,364	1,493
Error de velocidad [r/min]	95,8	5,5	7,3	92,2	225,8	224,9	225,2	227,6
THD de corriente [%]	17,16	1,19	1,29	1,24	1,21	1,19	1,20	1,22
Error estim. de torque [N·m]	19,49	0,01	8,26	15,44	23,44	21,21	21,90	23,25
Ripple de torque [N·m]	19,21	1,53	1,19	1,06	1,03	1,03	1,06	0,94
Error estim. de flujo [Wb]	0,2247	0,0002	0,2133	0,2907	0,3598	0,3446	0,3480	0,3495
MPC + PINN arr.+carga								
Residuo de R_r [Ω]	0,371	0,000	0,064	0,184	1,026	0,564	0,737	0,859
Error de velocidad [r/min]	105,8	5,5	5,5	5,6	222,1	114,1	115,0	175,3
THD de corriente [%]	10,56	1,19	1,34	1,26	1,26	1,22	1,40	1,17
Error estim. de torque [N·m]	14,71	0,01	0,11	1,29	16,96	7,72	10,39	11,93
Ripple de torque [N·m]	13,64	1,53	1,52	1,47	1,19	1,28	1,27	1,21
Error estim. de flujo [Wb]	0,1713	0,0002	0,0052	0,0504	0,2569	0,1534	0,1840	0,1969

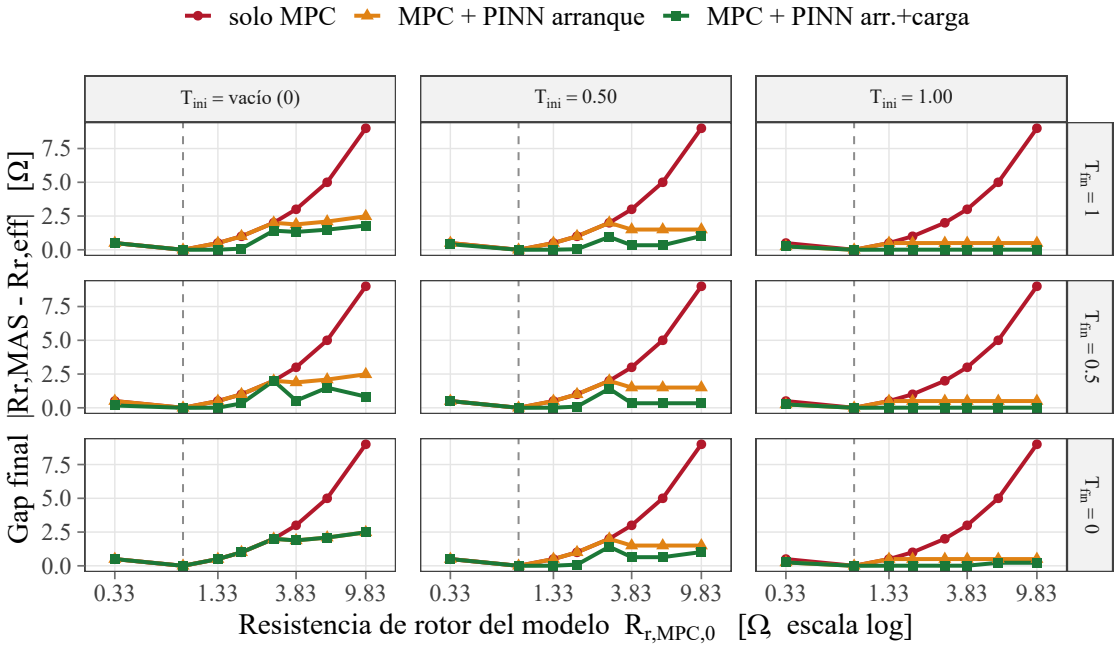


Figura 7.19: Residuo de la resistencia de rotor: distancia entre el modelo del controlador y el valor real de la máquina al terminar la rutina. Velocidad de referencia 1000 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

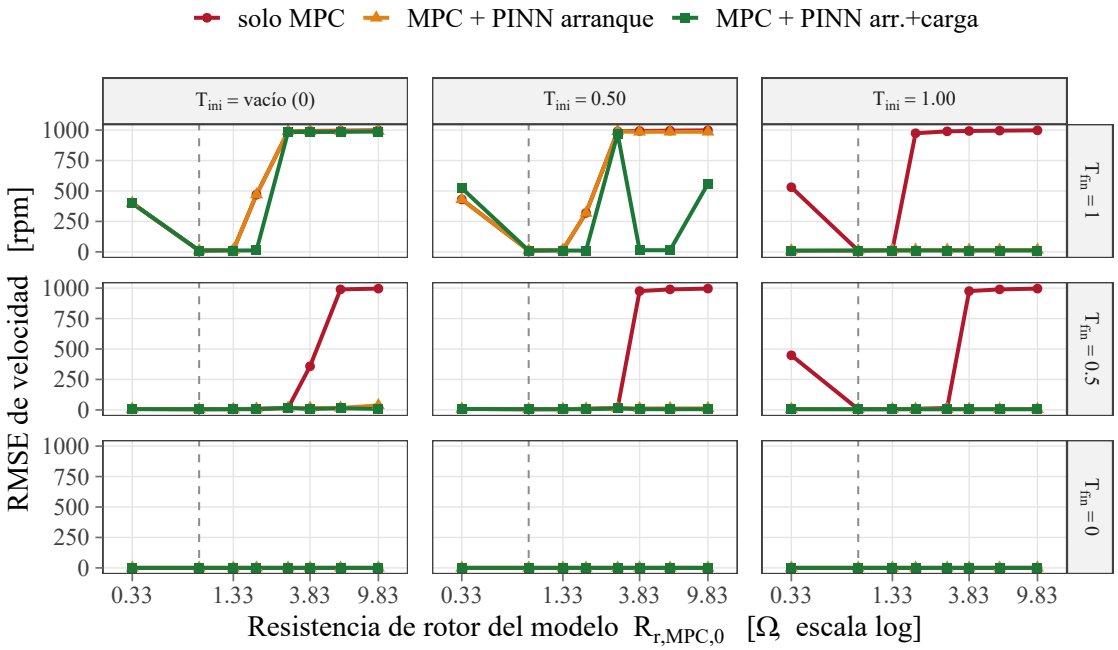


Figura 7.20: Error de seguimiento de velocidad respecto de su referencia al terminar la rutina. Velocidad de referencia 1000 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

7.1.6. Análisis de sucesión: quién corrige el parámetro

Los resultados anteriores miden *cuánto* recupera la capa correctiva. Pero, la evaluación no responde: **quién, dentro de la capa, mueve el parámetro**. El diseño del Capítulo 5 afirma un reparto en dos escalas: vías físicas sobre el signo del residuo cuando las redes no proponen corrección, y redes sobre su magnitud cuando el régimen se lo permite. Para verificarlo en lugar de suponerlo, se implementaron 8 escenarios construidos a mano, cada uno simulado con los expertos operando sobre el mismo modelo que produjo los resultados anteriores. Los tres actores son identificables con el `C-Script`, con un contador: pasos del *breakaway* y vía de emergencia, aplicaciones de la red de arranque y aplicaciones de la red de carga.

Tabla 7.13: Quién mueve $R^{\text{eff}}_{r,\text{MPC}}$ en cada rutina: pasos del *breakaway* y de la vía de emergencia, aplicaciones de la red de arranque y compromisos de la red de carga

(a) Escrituras de cada actor, y valor de $R^{\text{eff}}_{r,\text{MPC}}$ al comienzo y al final de la corrida, en Ω. Carga escalonada: el perfil creciente desde vacío de la Tabla 5.11. *En la máquina de 5 Ω el valor no se asienta y oscila en ese intervalo hasta el final.

Caso	Inicial	Objetivo	Final	Vía física	Red arr.	Red carga
Arranque sobreestimado, cinta 50 %	3,00	0,83	0,823	1	2	2
Ajustado, cinta 50 %	0,83	0,83	0,830	0	0	0
Rotor trancado, cinta nominal	3,00	0,83	0,861	4	0	1
Deriva +0,5 Ω, carga escalonada	0,83	1,33	1,263	0	1	1
Sobreestimado en 1,5 Ω, carga escalonada	2,33	0,83	0,856	0	1	3
Subestimado 10×, cinta nominal	0,10	1,00	1,072	7	0	1
Subestimado 10×, cinta 50 %	0,10	1,00	1,036	3	1	2
Subestimado 6×, máquina de 5 Ω, cinta nominal	0,83	5,00	4,83–5,08*	42	0	0

(b) Traza de dos casos: cada escritura sobre $R^{\text{eff}}_{r,\text{MPC}}$, con su actor, el valor que propone y el que se aplica, en Ω.

t [s]	Actor	Antes	Propuesta	Aplicada
Arranque sobreestimado, cinta 50 % *(objetivo* 0,83*)*				
0,439	Vía física (*breakaway*)	3,000	—	2,500
1,065	Red de arranque	2,500	0,830	1,770
1,128	Red de arranque	1,770	0,760	1,253
2,072	Red de carga	1,253	0,828	0,995
2,663	Red de carga	0,995	0,796	0,823
Sobreestimado en 1,5 Ω*, carga escalonada (objetivo* 0,83*)*				
0,285	Red de arranque	2,330	0,937	1,650
1,408	Red de carga	1,650	0,814	1,310
2,213	Red de carga	1,310	0,827	1,041
3,225	Red de carga	1,041	0,827	0,856

Tres puntos.

1. **Primero, los tres actores trabajan.** La vía física interviene en cinco casos, la red de arranque en cuatro y la de carga en seis. En los tres casos en que el *breakaway* no llega a activarse, son las redes las únicas que actúan, y en el ajustado no actúa nadie, que es lo que se le pide. La vía física no desplazó a las redes, salvo en la máquina de 5 Ω, que se analiza en el tercer punto.

2. **Segundo: la red de arranque también es gruesa.** Sus cinco propuestas se alejan del valor real entre un 0 y un 35 %, contra un máximo de 4,1 % en las diez de la red de carga. La traza de la Tabla 7.13b lo muestra en el caso sobreestimado con carga escalonada: la red de arranque propone 0,937 para un objetivo de 0,83 y da un solo paso, de 2,33 a 1,650, y es la red de carga la que cierra, en tres escrituras, hasta 0,856. En la deriva ocurre lo mismo en sentido contrario: la red de arranque lleva $R^{\text{eff}}_{r,\text{MPC}}$ de 0,83 a 1,071, y la de carga, a 1,263. La distinción entre la vía física y la red de arranque no es, por tanto, de precisión sino de *qué leen*: la vía usa el signo del residuo, la red usa su magnitud, y ambas cumplen el mismo papel de acercamiento

rápido. La precisión la aporta el experto de carga, que hace la última escritura en seis de los siete casos con desajuste.

3. **Tercero, en la máquina de 5 Ω la vía física da 42 pasos y ninguna red toma el relevo.** La vía llega al entorno del valor real en ocho pasos, hacia 1,7 s; se pasa una vez, hasta 5,33 Ω, y desde 1,8 s oscila entre 4,83 y 5,08 Ω hasta el final de la corrida: 32 de los 42 pasos son ese vaivén. La causa es un límite de la máquina y no un error del modelo. Con 5 Ω y par nominal, la velocidad no supera el 88 % de la referencia ni siquiera con el modelo ajustado y sin expertos, y el coordinador de arranque trata como estancado todo arranque cargado que, por debajo del 90 % de la referencia, avance menos de un 1 % en cinco ciclos (Subsección 5.2.5.b). Por eso la vía física vuelve a entrar cada 0,25 s, aproximadamente, y cada escritura reinicia el análisis del coordinador de carga, que no sale de la espera de asentamiento. Las redes, por tanto, no se abstienen: nunca llegan a evaluar. El vaivén queda dentro del 3,4 % del valor real, y la velocidad alcanza el 98 % de la que logra el modelo ajustado.

De ahí se mantiene el orden de la arquitectura **sobre la partición por maniobra y precisión, con escala de corrección**. La corrección gruesa y rápida corresponde al experto de arranque, que la ejerce por dos mecanismos según disponga o no de magnitud fiable (la red cuando ρ^2 lo permite, la vía física cuando no); la fina corresponde al experto de carga, que refina en régimen asentado hasta la resolución del estimador; y los coordinadores deciden en cada instante cuál capa tiene la palabra.

El rescate físico no reemplaza a las redes: las habilita.

7.2. Resultados de la Propuesta 2: estimador inverso de una sola red

La red única del Capítulo 6 se evaluó sobre el **subconjunto nominal común** de la campaña del sistema de expertos: la rutina completa de cinta a velocidad nominal y la carga de bomba a cuatro velocidades, ambas con $R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$. Este subconjunto contiene 91 condiciones desajustadas y 13 ajustadas; al ejecutar cada una en los modos *Solo MPC* y *MPC + iPINN*, la campaña de la segunda propuesta comprende 208 corridas. Por tanto, la comparación entre propuestas es directa dentro de estas 104 condiciones, pero no debe extenderse a las restantes configuraciones de $R_{r,\mathrm{MAS}}$ de la campaña de expertos.

Que la comparación sea legítima descansa en dos comprobaciones previas. La primera es que **el accionamiento es el mismo**: con la capa correctiva desactivada, los dos modelos entregan las mismas señales sobre un escenario común, sin diferencia apreciable en velocidad, torque ni resistencia efectiva. La segunda es que los indicadores se calculan con el mismo código, sobre la misma ventana de los 0,5 s finales y con el mismo adelanto de una muestra en las señales del controlador.

7.2.1. Rutina cinta a velocidad nominal

Vista principal sobre la máquina de resistencia nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$): la máquina arranca con la carga inicial que indica la columna y a mitad de rutina recibe un escalón que la lleva al nivel que indica la fila. Son 63 condiciones desajustadas por modo, a 1440 r/min.

La Tabla 7.14 resume el bloque con el mismo formato empleado para el sistema de expertos: una columna por escalón de la escalera, cada celda promediando las nueve facetas de carga, y la columna del caso ajustado sombreada. Al disponer esta propuesta de un solo modo de corrección, la tabla contrasta dos brazos en lugar de tres. Y cada celda promedia las nueve facetas de carga de ese escalón.

Tabla 7.14: Síntesis de las figuras del bloque: estimador inverso de una sola red sobre la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$), carga de cinta y velocidad nominal. Las columnas recorren la resistencia con que se inicializa el modelo del controlador; la sombreada es el **caso ajustado**

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	608,4	5,5	7,5	250,6	568,3	877,4	954,9	1021,5
THD de corriente [%]	32,62	1,98	1,53	1,07	0,96	1,04	1,33	1,51
Error estim. de torque [N·m]	26,73	0,01	9,09	19,90	31,65	39,79	44,59	40,18
Ripple de torque [N·m]	23,70	1,56	1,13	0,95	0,78	0,73	0,66	0,27
Error estim. de flujo [Wb]	0,2155	0,0004	0,1971	0,3161	0,3990	0,4746	0,5209	0,5598
MPC + iPINN								
Residuo de R_r [Ω]	0,118	0,003	0,076	0,138	0,250	0,363	0,073	0,049
Error de velocidad [r/min]	6,7	5,5	5,5	5,6	5,6	5,6	5,5	5,6
THD de corriente [%]	3,77	2,07	1,73	1,98	1,62	1,73	2,02	1,74
Error estim. de torque [N·m]	0,71	0,11	0,32	0,49	0,52	0,51	0,34	0,44
Ripple de torque [N·m]	1,99	1,57	1,49	1,48	1,47	1,48	1,51	1,49
Error estim. de flujo [Wb]	0,0383	0,0037	0,0117	0,0178	0,0189	0,0175	0,0134	0,0165

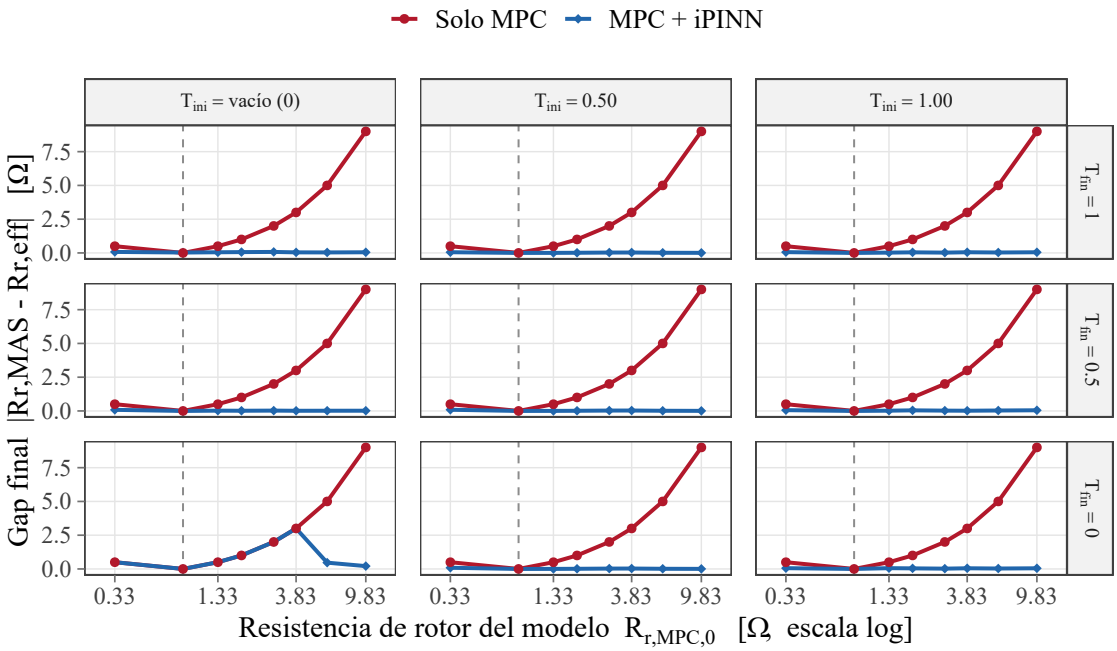


Figura 7.21: Residuo de la resistencia de rotor: distancia entre el modelo del controlador y el valor real de la máquina al terminar la rutina. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

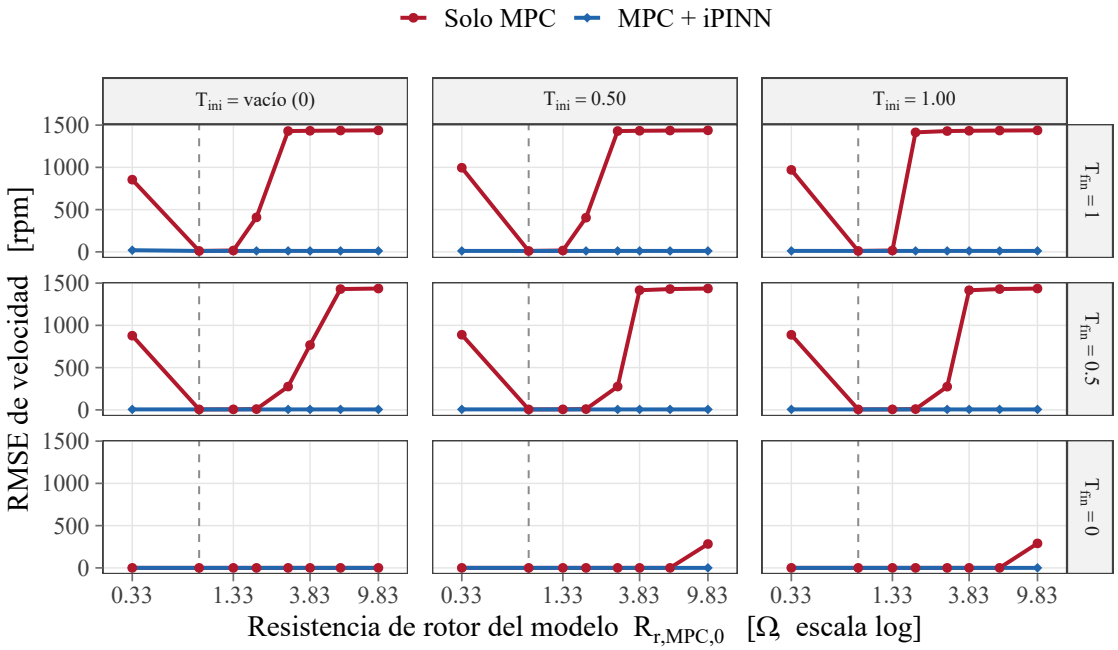


Figura 7.22: Error de seguimiento de velocidad respecto de su referencia al terminar la rutina. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

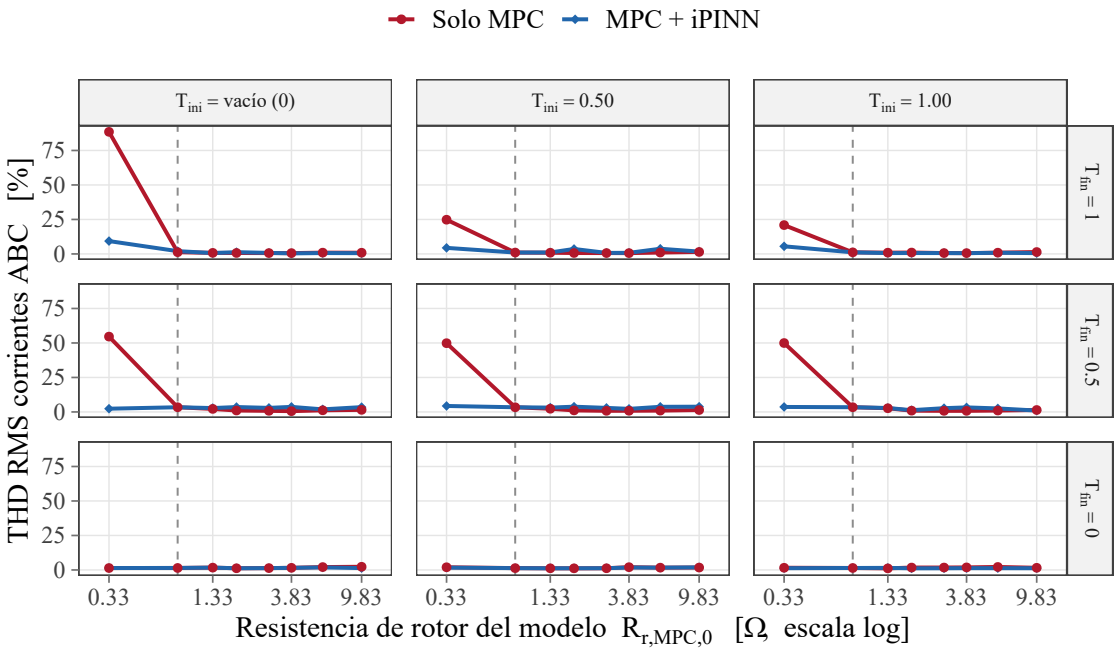


Figura 7.23: Distorsión armónica de las corrientes de estator. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

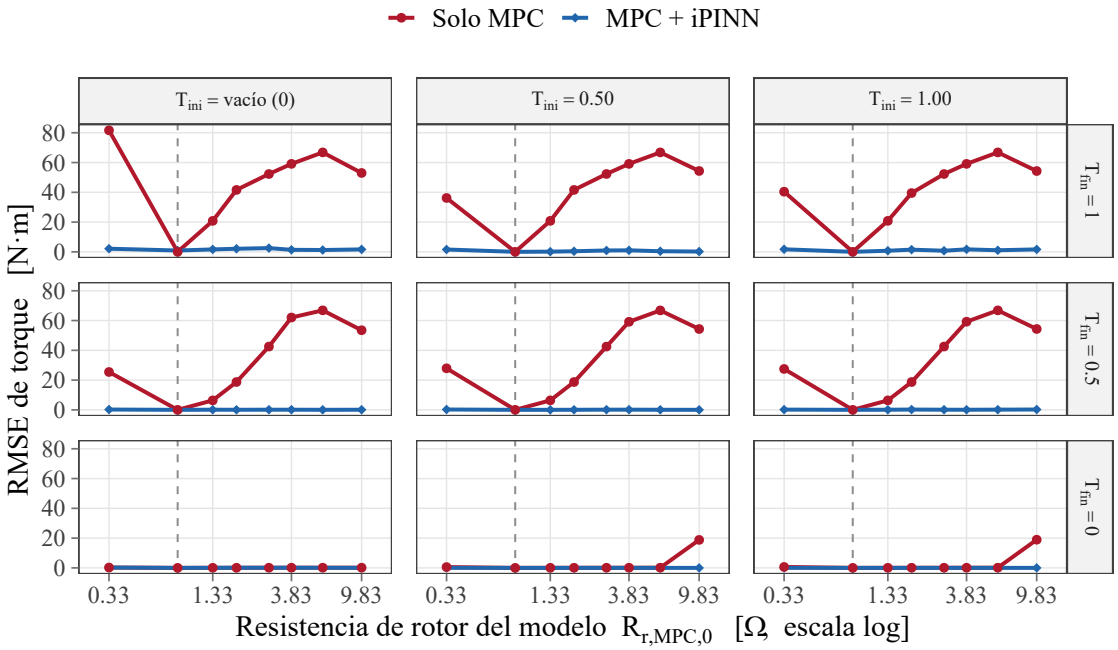


Figura 7.24: Error de estimación de torque: discrepancia entre el torque real y el que estima el observador del controlador. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

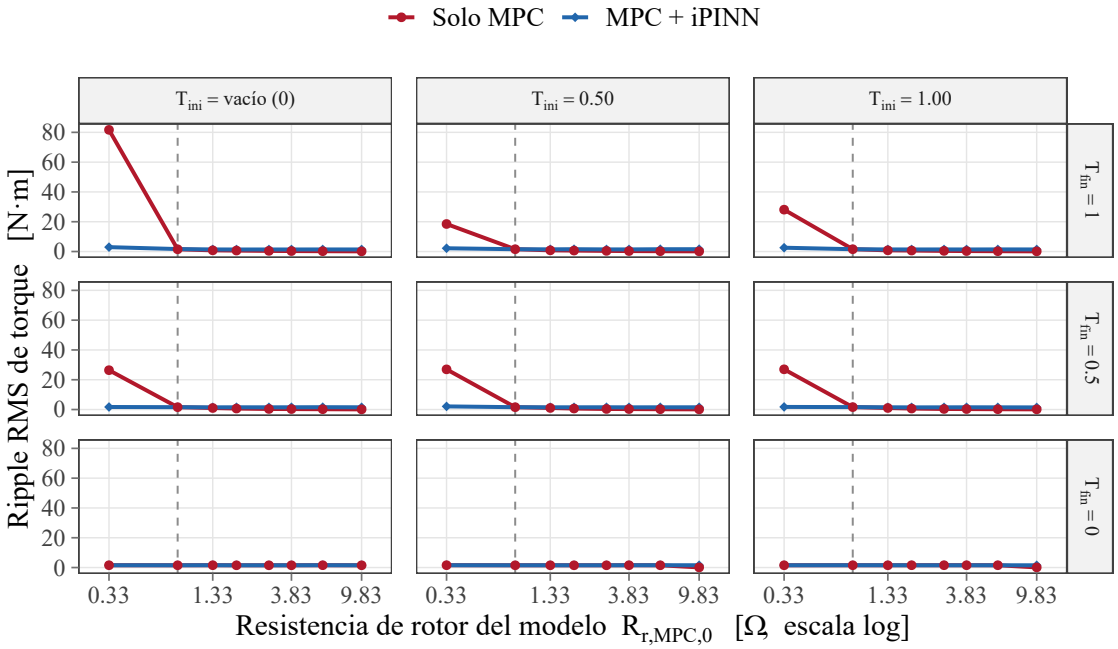


Figura 7.25: *Ripple* del torque electromagnético entregado. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

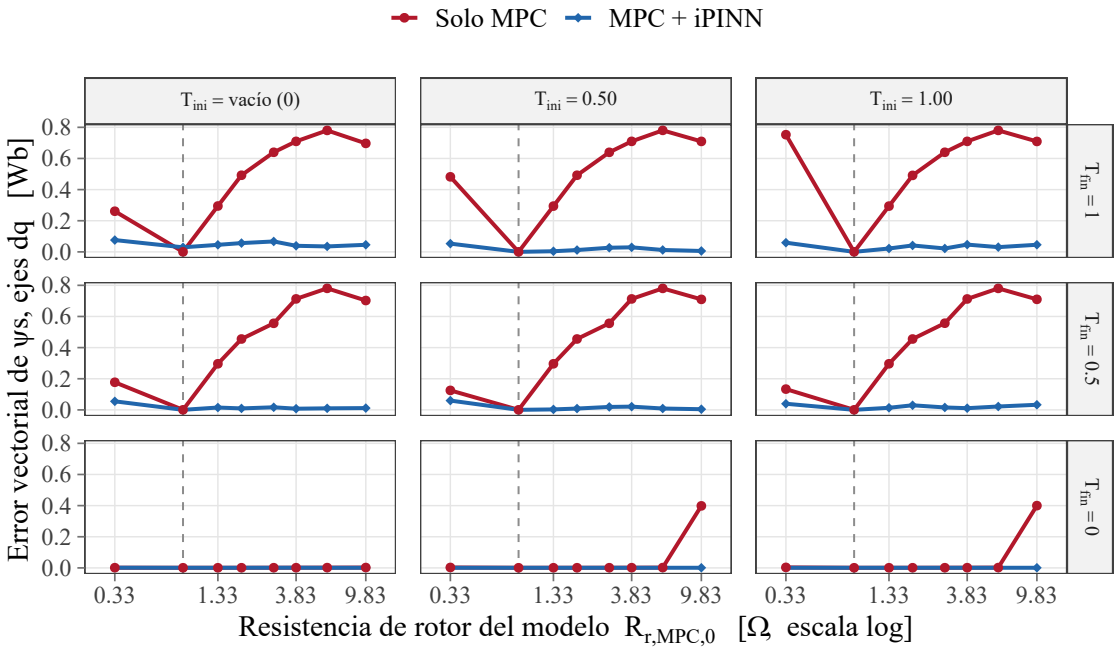


Figura 7.26: Error vectorial de estimación del flujo de estator, en ejes *dq* (Tabla 7.1): discrepancia entre el flujo real y el estimado por el controlador. Velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

El resultado principal está en los dos primeros indicadores, el residuo de resistencia y el error de velocidad. El residuo de resistencia cae de 3,000 a **0,153 Ω** de media y el error de velocidad de 612,7 a **5,7** r/min, con lo que la fracción de condiciones en que el accionamiento alcanza su consigna pasa del 44,4 al **100** % y desaparecen los colapsos. Los indicadores de forma de onda acompañan: la distorsión armónica baja de 5,72 a 2,09 % y el error de estimación de torque de 30,27 a 0,48 N·m.

Conviene detenerse en la **faceta inferior izquierda**, que es la única donde la capa no aplana el residuo. Corresponde a arrancar en vacío y permanecer en vacío: la corrección sigue a la línea base hasta desajustes de unos 3 Ω y solo actúa por encima. En las ocho facetas restantes el residuo medio queda entre 0,021 y 0,055 Ω; en esta vale 1,097 Ω. No es un fallo del estimador sino la **región ciega** anunciada en la Subsección 4.4.3: sin torque solicitado el desajuste no se manifiesta en las señales, y una red que no ve no debe corregir. Que la corrección aparezca solo en los desajustes grandes es coherente con ello, porque allí el transitorio de arranque sí deja firma aunque el estado final no la tenga.

7.2.2. Carga de bomba

La carga de bomba obedece a la ley de afinidad descrita en la Sección 4.1.2, de modo que la carga *es* la velocidad y no se aplica en escalón. Las facetas recorren las cuatro referencias de velocidad, de 750 a 1440 r/min. Son 28 condiciones desajustadas por modo.

Las Tablas 7.15 a 7.18 resumen el bloque que sigue, una por referencia de velocidad. Como en el bloque equivalente del sistema de expertos, cada celda corresponde a una *única* corrida: con esta carga la referencia de velocidad fija también el torque solicitado.

Las columnas recorren la resistencia con que se inicializa el modelo del controlador; la sombreada es el **caso ajustado**, en el que la capa no debe actuar.

Tabla 7.15: Carga de bomba a 750 r/min sobre la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$). Cada celda corresponde a una única corrida: con esta carga la velocidad de referencia fija también el torque solicitado, de modo que no hay facetas que promediar.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	4,9	2,9	2,1	1,7	5,6	9,2	55,2	325,9
THD de corriente [%]	1,28	2,07	1,07	1,43	1,44	4,12	1,58	1,37
Error estim. de torque [N·m]	5,77	0,00	1,96	2,63	13,54	27,83	49,89	33,78
Ripple de torque [N·m]	2,33	1,52	1,38	1,31	0,59	1,60	0,19	0,08
Error estim. de flujo [Wb]	0,2689	0,0002	0,1214	0,2199	0,6351	0,6903	0,6575	0,5487
MPC + iPINN								
Residuo de R_r [Ω]	0,332	0,260	0,313	0,246	0,257	0,225	0,287	0,231
Error de velocidad [r/min]	2,3	2,5	2,4	2,5	2,5	2,5	2,4	2,5
THD de corriente [%]	1,48	1,33	1,07	1,10	1,53	1,32	1,26	1,25
Error estim. de torque [N·m]	1,51	1,23	1,42	1,17	1,22	1,09	1,33	1,12
Ripple de torque [N·m]	1,43	1,44	1,43	1,44	1,45	1,45	1,43	1,45
Error estim. de flujo [Wb]	0,0817	0,0697	0,0819	0,0664	0,0688	0,0614	0,0761	0,0629

Tabla 7.16: Carga de bomba a 1000 r/min sobre la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$). Cada celda corresponde a una única corrida: con esta carga la velocidad de referencia fija también el torque solicitado, de modo que no hay facetas que promediar.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	7,3	5,2	5,7	8,2	14,2	80,0	305,2	575,9
THD de corriente [%]	3,95	1,00	0,97	1,01	0,66	0,88	1,58	1,37
Error estim. de torque [N·m]	4,63	0,01	4,40	14,73	37,40	52,19	49,89	33,78
Ripple de torque [N·m]	3,04	1,52	1,11	0,79	0,51	0,30	0,19	0,08
Error estim. de flujo [Wb]	0,4312	0,0002	0,2870	0,4919	0,6641	0,6478	0,6575	0,5487
MPC + iPINN								
Residuo de R_r [Ω]	0,048	0,041	0,091	0,057	0,052	0,049	0,049	0,078
Error de velocidad [r/min]	5,1	5,1	5,1	5,1	5,1	5,1	5,1	5,1
THD de corriente [%]	0,77	1,13	0,91	0,67	0,93	0,88	0,85	1,46
Error estim. de torque [N·m]	0,13	0,11	0,28	0,16	0,14	0,13	0,13	0,23
Ripple de torque [N·m]	1,50	1,50	1,44	1,50	1,48	1,49	1,49	1,48
Error estim. de flujo [Wb]	0,0300	0,0255	0,0557	0,0356	0,0325	0,0302	0,0304	0,0477

Tabla 7.17: Carga de bomba a 1250 r/min sobre la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$). Cada celda corresponde a una única corrida: con esta carga la velocidad de referencia fija también el torque solicitado, de modo que no hay facetas que promediar.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	484,2	8,1	10,7	14,9	174,2	330,0	555,2	825,9
THD de corriente [%]	54,55	0,77	0,52	0,61	0,72	0,88	1,58	1,37
Error estim. de torque [N·m]	15,65	0,01	12,63	29,23	47,39	52,19	49,89	33,78
Ripple de torque [N·m]	13,44	1,53	0,98	0,68	0,39	0,30	0,19	0,08
Error estim. de flujo [Wb]	0,1037	0,0002	0,3362	0,5041	0,5883	0,6478	0,6575	0,5487
MPC + iPINN								
Residuo de R_r [Ω]	0,045	0,000	0,027	0,033	0,033	0,026	0,018	0,029
Error de velocidad [r/min]	8,0	8,1	8,1	8,2	8,1	8,1	8,2	8,1
THD de corriente [%]	0,96	0,77	0,80	0,85	0,92	0,72	0,90	0,78
Error estim. de torque [N·m]	0,62	0,01	0,45	0,54	0,55	0,44	0,51	0,48
Ripple de torque [N·m]	1,65	1,53	1,50	1,50	1,50	1,50	1,52	1,50
Error estim. de flujo [Wb]	0,0392	0,0002	0,0236	0,0280	0,0287	0,0227	0,0188	0,0252

Tabla 7.18: Carga de bomba a 1440 r/min sobre la máquina nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$). Cada celda corresponde a una única corrida: con esta carga la velocidad de referencia fija también el torque solicitado, de modo que no hay facetas que promediar.

Indicador	$R_{r,\mathrm{MPC},0}$ [Ω]							
	0,33	**0,83**	1,33	1,83	2,83	3,83	5,83	9,83
Solo MPC								
Residuo de R_r [Ω]	0,500	0,000	0,500	1,000	2,000	3,000	5,000	9,000
Error de velocidad [r/min]	674,1	10,8	15,7	149,5	364,2	520,0	745,2	1015,9
THD de corriente [%]	54,55	1,02	0,86	0,42	0,72	0,88	1,58	1,37
Error estim. de torque [N·m]	15,65	0,01	20,25	33,84	47,39	52,19	49,89	33,78
Ripple de torque [N·m]	13,44	1,53	0,81	0,58	0,39	0,30	0,19	0,08
Error estim. de flujo [Wb]	0,1037	0,0001	0,2981	0,4415	0,5883	0,6478	0,6575	0,5487
MPC + iPINN								
Residuo de R_r [Ω]	0,048	0,000	0,021	0,059	0,057	0,067	0,055	0,048
Error de velocidad [r/min]	10,8	10,8	10,9	11,1	11,1	11,2	11,1	11,0
THD de corriente [%]	3,09	1,02	3,16	0,62	0,52	1,08	0,83	0,57
Error estim. de torque [N·m]	1,39	0,01	0,69	1,94	1,89	2,24	1,80	1,58
Ripple de torque [N·m]	1,97	1,53	1,41	1,36	1,36	1,37	1,36	1,40
Error estim. de flujo [Wb]	0,0472	0,0001	0,0203	0,0540	0,0525	0,0613	0,0502	0,0446

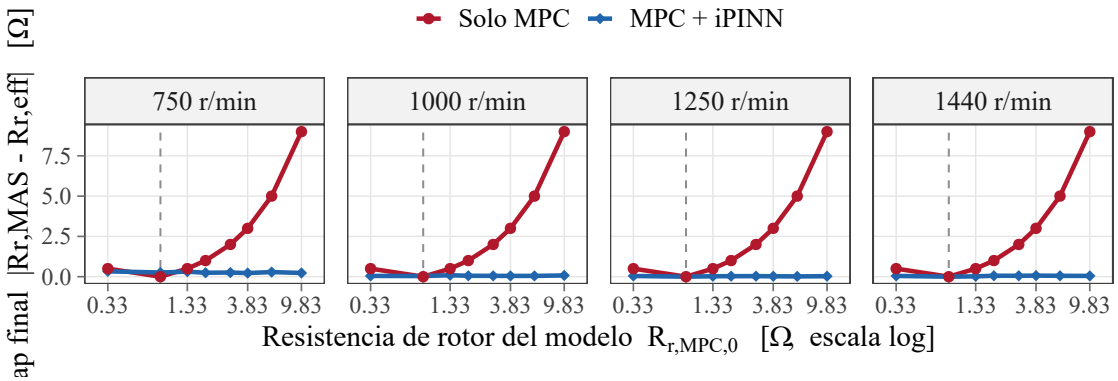


Figura 7.27: Residuo de la resistencia de rotor al terminar la rutina. Carga de bomba; las facetas recorren la referencia de velocidad, que en esta carga fija también el torque solicitado. Menor es mejor.

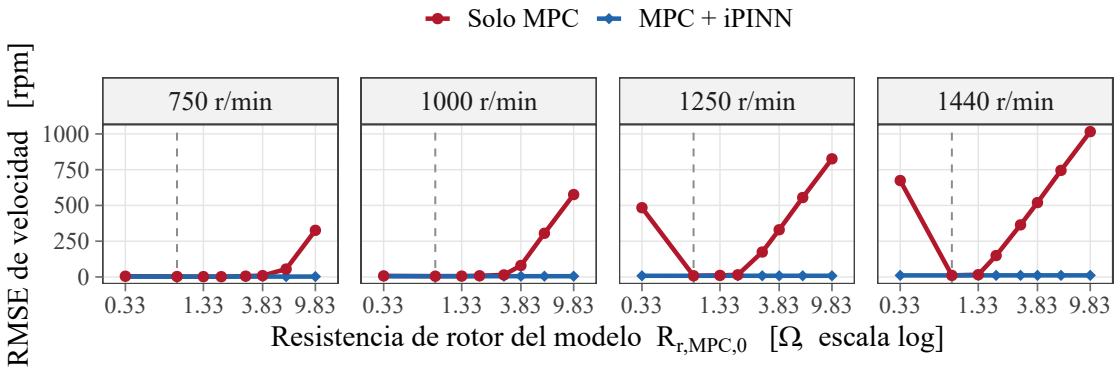


Figura 7.28: Error de seguimiento de velocidad respecto de su referencia al terminar la rutina. Carga de bomba; las facetas recorren la referencia de velocidad, que en esta carga fija también el torque solicitado. Menor es mejor.

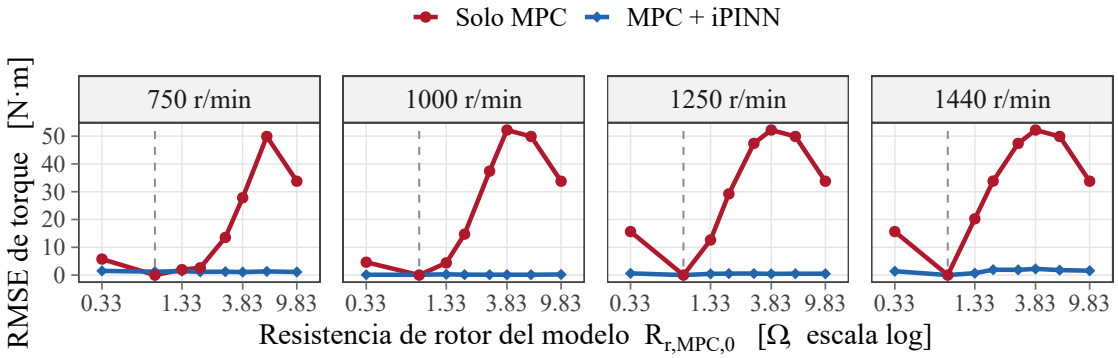


Figura 7.29: Error de estimación de torque: discrepancia entre el torque real y el estimado. Carga de bomba; las facetas recorren la referencia de velocidad, que en esta carga fija también el torque solicitado. Menor es mejor.

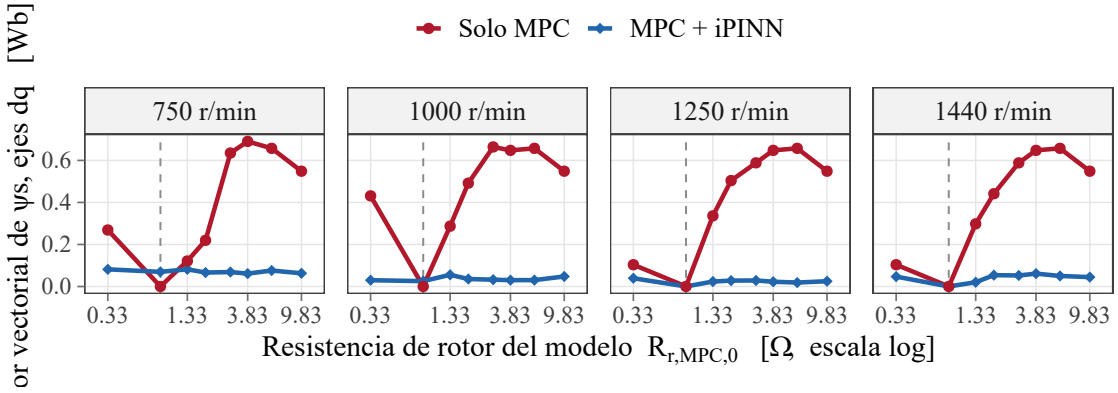


Figura 7.30: Error vectorial de estimación del flujo de estator, en ejes *dq* (Tabla 7.1). Carga de bomba; las facetas recorren la referencia de velocidad, que en esta carga fija también el torque solicitado. Menor es mejor.

El comportamiento reproduce el de la cinta y en un aspecto lo mejora: el residuo cae de 3,000 a **0,103 Ω** y el error de velocidad de 260,0 a **6,7** r/min, con el **100** % de las condiciones alcanzando su consigna frente al 42,9 % de la línea base. El error de estimación de torque baja de 29,51 a 0,90 N·m.

La condición de rechazo. Sobre las trece condiciones en que el modelo ya parte ajustado, la capa se abstiene en **doce**: en diez conserva el parámetro sin alteración alguna y en las dos restantes lo desplaza menos de 0,05 Ω, una diferencia que no altera el comportamiento del controlador. La excepción es la bomba a 750 r/min, donde desplaza el modelo de 0,830 a 1,090 Ω y con ello degrada la estimación de torque. Es el único falso positivo de la campaña y delimita el alcance de la propuesta: a carga cuadrática a aproximadamente la mitad de la velocidad nominal, queda fuera de esa cobertura. La carga de bomba no aparece en ningún ejemplo de entrenamiento de esta propuesta, a ninguna velocidad; aun así, a 1000 r/min la capa solo desplazó el modelo ajustado en 0,041 Ω, y a 1250 y 1440 r/min no lo movió.

7.3. Comparación: evaluación sobre una rutina completa

Las dos secciones anteriores resumen miles de corridas en indicadores escalares medidos sobre una ventana final. Esa vista responde a *cuánto* recupera cada propuesta, pero no muestra *cómo* lo hace: en qué instante interviene cada capa, de qué forma escribe la corrección y qué está haciendo la máquina mientras tanto. Esta sección cierra esa brecha presentando las señales en el dominio del tiempo de una única rutina, ejecutada sobre las tres configuraciones y sobre dos máquinas distintas.

La rutina se compone de cuatro tramos encadenados en 5 s, descritos en la Tabla 7.19. Su interés está en que combina las tres condiciones que el trabajo distingue: arranque en vacío, entrada de carga e inversión de giro, y las recorre dos veces en sentidos opuestos.

Tabla 7.19: Rutina empleada en esta sección. La carga es de tipo cinta, de modo que el torque resistente no cede con la velocidad. La referencia de velocidad corresponde a la nominal de la máquina, 1440 r/min.

Tramo	Intervalo	Referencia de velocidad	Carga
1	0,0 – 1,0 s	+1440 r/min	vacío
2	1,0 – 2,5 s	+1440 r/min	cinta, $0{,}5\,T_{nom}$
3	2,5 – 3,5 s	−1440 r/min	vacío
4	3,5 – 5,0 s	−1440 r/min	cinta, $0{,}5\,T_{nom}$

Análisis frente a distintas resistencias $R_{r,\mathrm{MAS}}$. La rutina se repite sobre la máquina de resistencia nominal ($R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$) con el modelo del controlador inicializado en $R_{r,\mathrm{MPC},0} = 3{,}83\ \Omega$, y sobre una máquina de $R_{r,\mathrm{MAS}} = 3{,}00\ \Omega$ con $R_{r,\mathrm{MPC},0} = 7{,}00\ \Omega$. En ambos casos el modelo *sobreestima* la resistencia; la primera lo hace en un factor 4,6 y la segunda en 2,3.

Las cinco magnitudes. Cada lámina apila cinco paneles que comparten el eje de tiempo: corrientes de estator, velocidad medida frente a su referencia, torque de referencia frente al estimado y al real, magnitud del flujo de estator en las mismas tres versiones, y por último la resistencia de rotor que el controlador aplica frente al valor real de la máquina. Es importante señalar que **las mediciones reales de las variables, no son**

observables en ninguna propuesta, y ambas propuestas solo funcionan con los valores estimados.

7.3.1. Rutinas con resistencia nominal

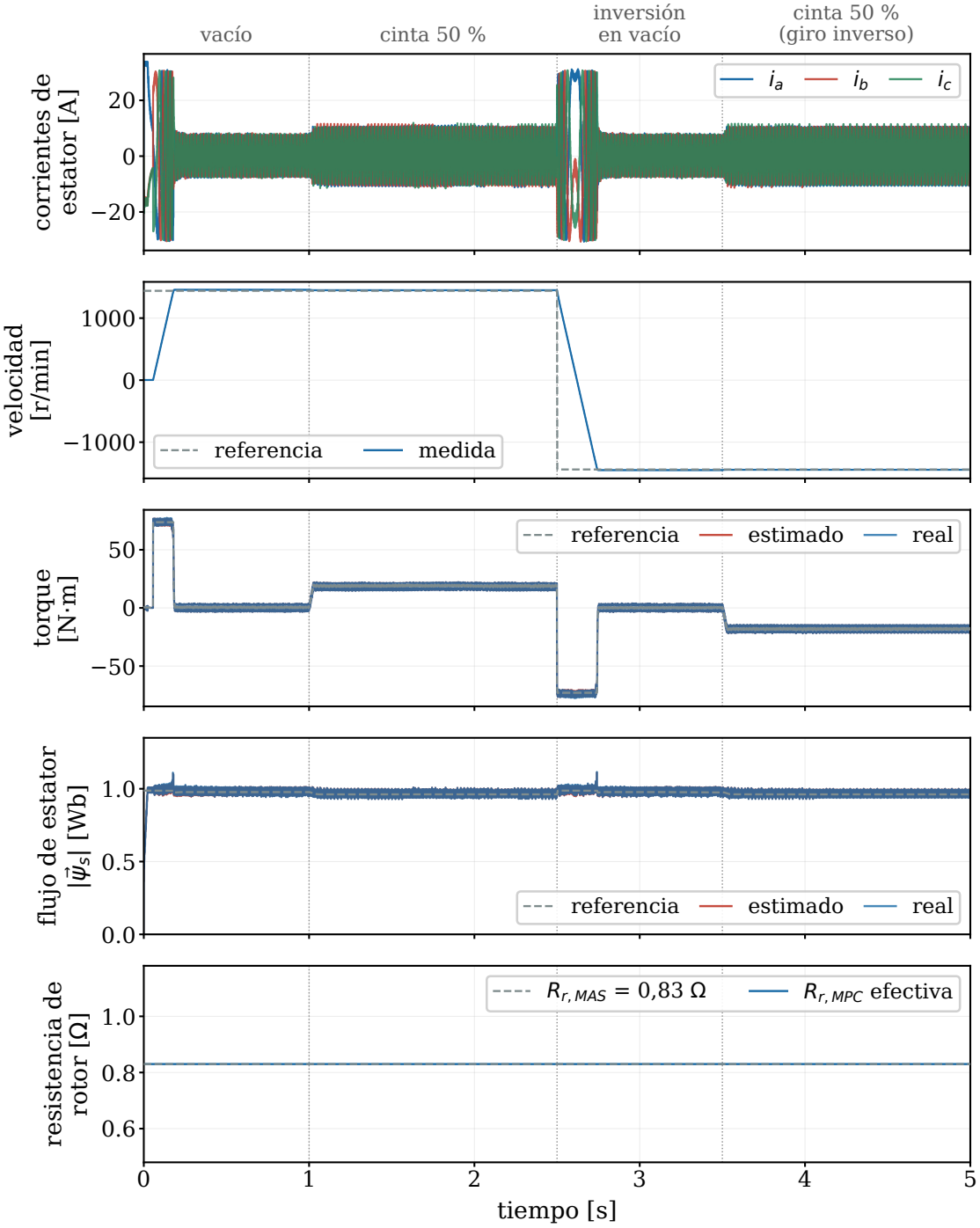


Figura 7.31: Rutina completa sobre la máquina nominal con el modelo del controlador ajustado a su valor real ($R_{r,\mathrm{MPC}} = R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$) y sin capa correctiva.

7.3.1.a. Sin corrección

$R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$ con el modelo del controlador inicializado en 3,83 Ω.

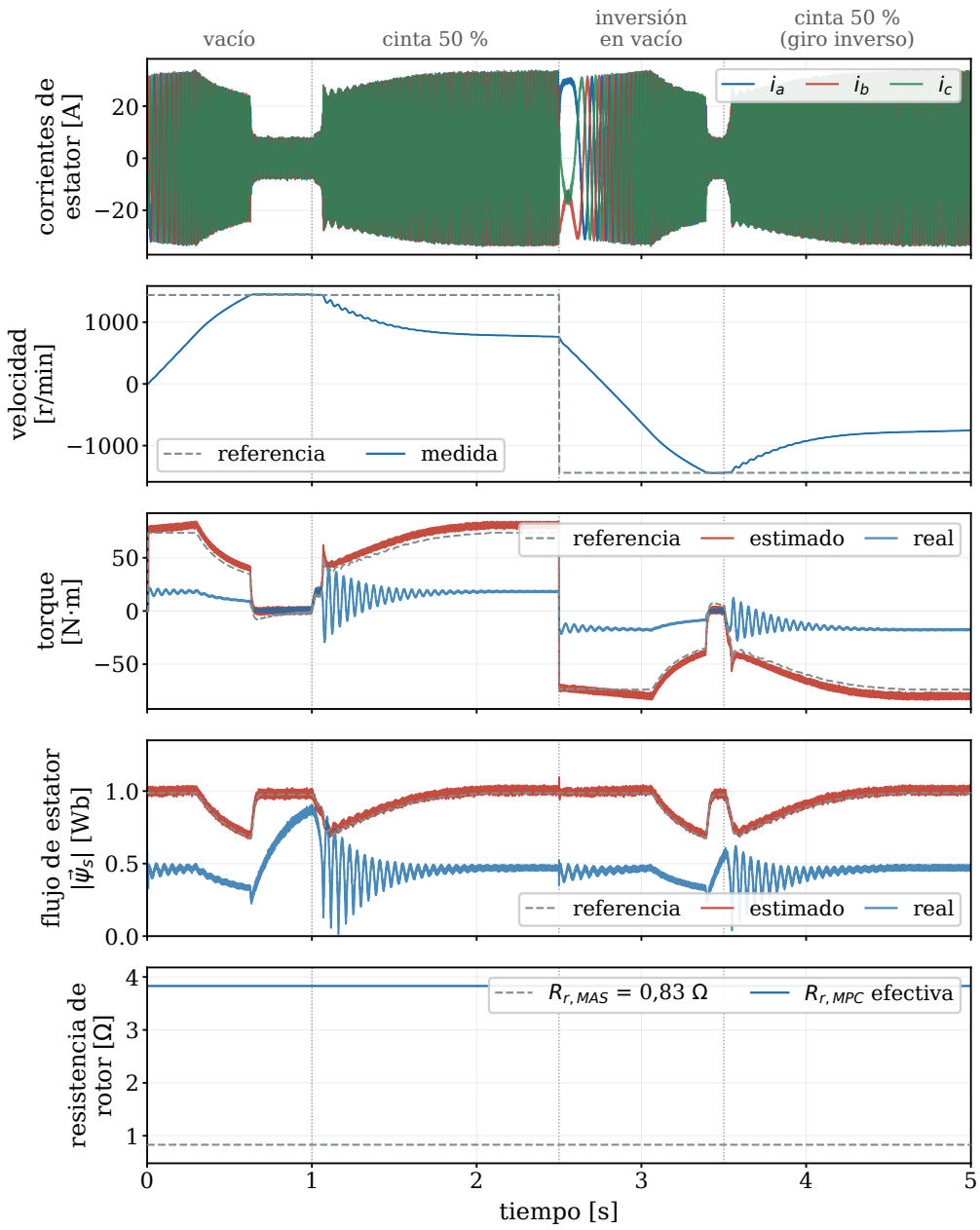


Figura 7.32: Rutina completa sobre la máquina nominal con el modelo sobreestimado en un factor 4,6 y sin capa correctiva. La resistencia aplicada permanece en su valor inicial durante toda la corrida.

La Figura 7.32 reproduce en el tiempo el fenómeno que el Capítulo 4 caracterizó, y lo hace visible en una sola corrida:

1. **En vacío el desajuste no se manifiesta en la velocidad.** El primer tramo alcanza 1440,8 r/min frente a una referencia de 1440, con 0,8 r/min de error. Un observador que midiera únicamente la velocidad concluiría que el accionamiento está sano.

2. **Pero sí se manifiesta en el flujo.** En ese mismo tramo el flujo real de la máquina vale 0,739 Wb frente a los 0,977 Wb de la referencia, mientras el controlador estima 0,977 Wb: cree tener el flujo que ha pedido. La máquina opera sub-magnetizada, con un control subóptimo.

3. **Con carga el accionamiento no entrega el punto de operación.** El segundo tramo se estabiliza en 767 r/min, con 673 r/min de error, y el cuarto reproduce la cifra en sentido inverso. El flujo real cae a 0,471 Wb y el torque estimado alcanza 80,29 N·m contra 17,93 N·m reales: un error de estimación de 62,35 N·m.

4. **Más corriente.** El *peak* de corriente de fase bajo carga es de 33,2 A, frente a los 11,9 y 14,4 A que las dos propuestas necesitan para entregar el mismo torque. Operar con el modelo equivocado no solo degrada el resultado: encarece obtenerlo.

7.3.1.b. Propuesta 1

$R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$ con el modelo del controlador inicializado en 3,83 Ω.

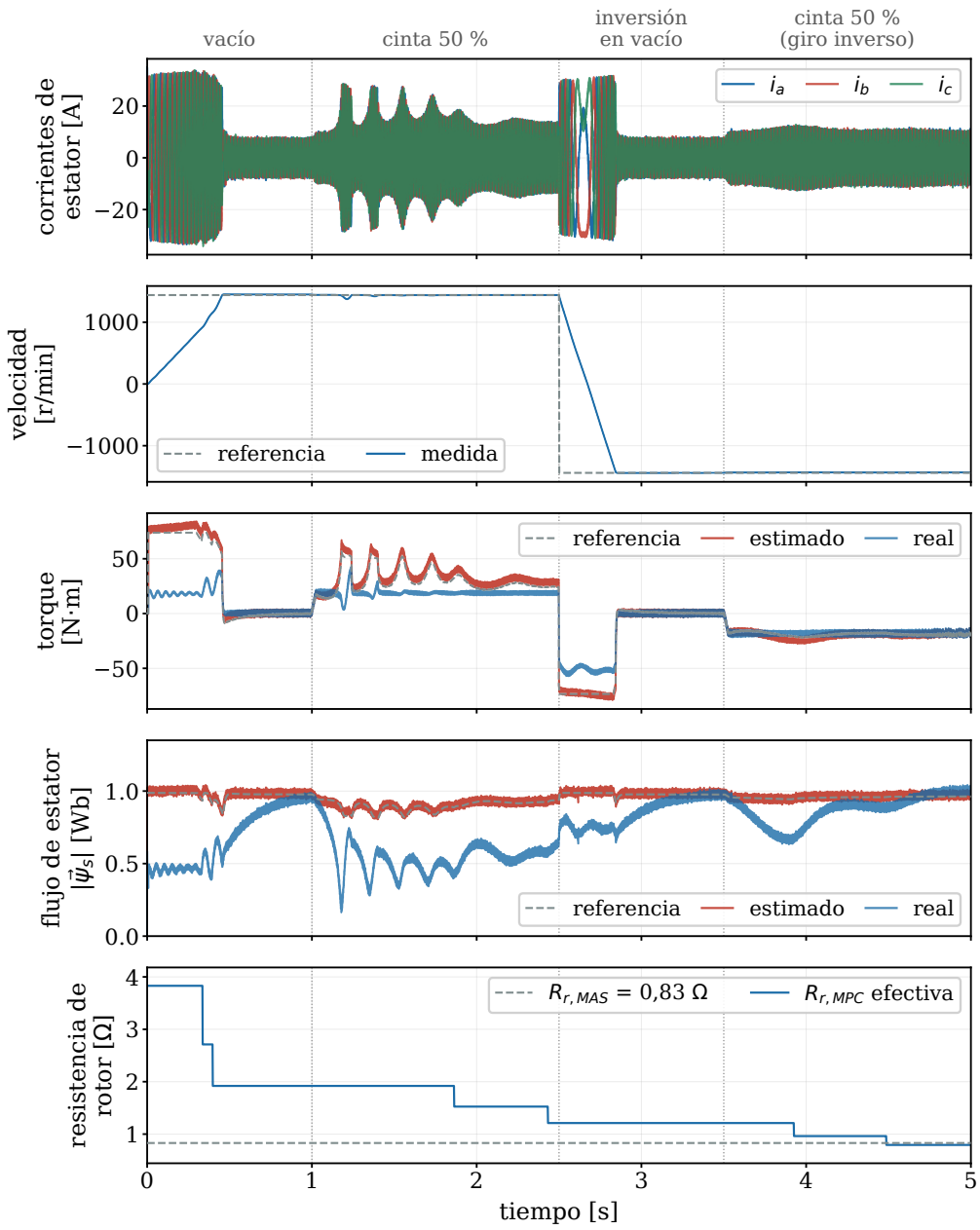


Figura 7.33: Rutina completa sobre la máquina nominal con el sistema de expertos.

La comparación con la lámina anterior es directa, porque las dos corrigen el mismo desajuste sobre la misma máquina:

1. **Interviene mucho antes.** La primera escritura ocurre a los 0,336 s, durante el

arranque, con la máquina acelerando a 897 r/min y la premagnetización ya terminada, sin esperar a que la carga revele el desajuste. La consecuencia se ve en el panel de velocidad: bajo carga la máquina no baja de 1367 r/min, frente a los 979 r/min de la Propuesta 2.

2. **Escribe a saltos y no en rampa.** Son seis escalones en toda la corrida, el mayor de 1,119 Ω. Es la diferencia de mecanismo entre una capa que decide por ciclo eléctrico con vías físicas de acercamiento rápido y otra que aplica una corrección proporcional a la magnitud que estima.

3. **Y converge al valor real.** La corrida termina en 0,793 Ω frente a 0,83 Ω, un residuo de 0,037 Ω. En el último tramo el torque estimado vale −18,90 N·m contra −18,61 N·m reales, con 0,28 N·m de discrepancia, y el flujo real alcanza 0,986 Wb sobre una referencia de 0,963 Wb. La inversión de marcha no interrumpe la corrección: los dos últimos escalones son de la misma dirección que los anteriores. Esos dos escalones los da la red de carga, que venía armada desde el tramo en sentido positivo y se reactiva porque el transitorio de la inversión satura el torque de referencia y se lee como un escalón de carga; el coordinador de arranque no se rearma (Sección G.3).

7.3.1.c. Propuesta 2

$R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$ con el modelo del controlador inicializado en 3,83 Ω.

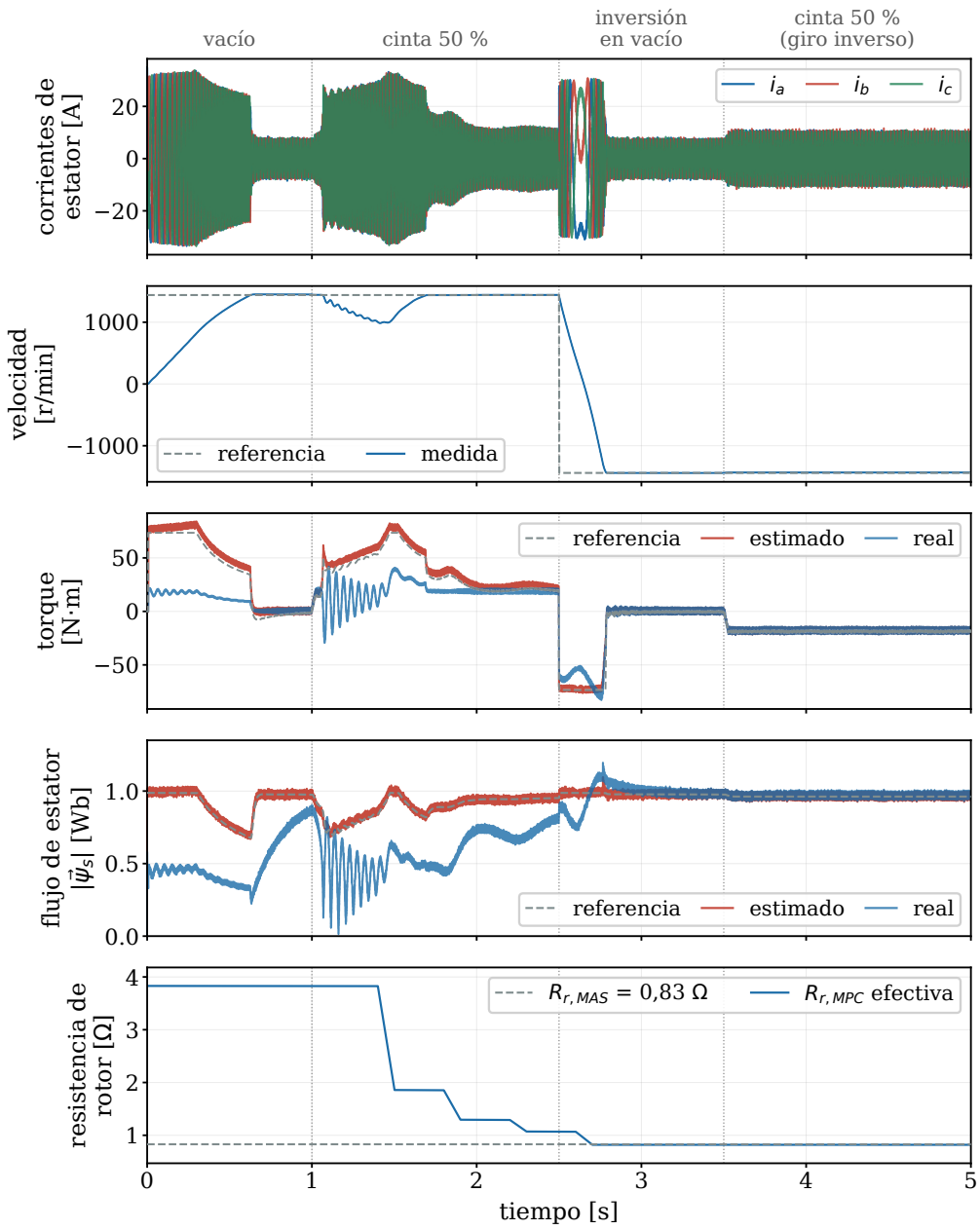


Figura 7.34: Rutina completa sobre la máquina nominal con el estimador inverso de una sola red.

La Figura 7.34 muestra tres rasgos que la vista agregada no permite observar:

1. **La capa espera a que el guardián habilite la escritura.** Durante todo el primer

tramo la resistencia aplicada permanece en 3,83 Ω. Las decisiones están programadas a partir de 1,0 s y cada 0,4 s (Subsección 6.4.1), de modo que en el tramo en vacío no hay ninguna. La primera coincide con la entrada de la carga y usa una ventana todavía en vacío: la red sí propone una corrección, $s = -0{,}616$, próxima a la necesaria, $-0{,}664$, pero declara la ventana fuera de su dominio, con $6 \cdot 10^{-5}$, y el guardián veta la escritura. La primera escritura ocurre en la decisión siguiente, a los 1,400 s, ya con la carga presente.

2. **El precio de esperar es un hundimiento transitorio.** Entre la entrada de la carga y el efecto de las primeras correcciones la velocidad cae hasta 979 r/min, y no regresa a su referencia hasta pasado el segundo 1,7.

3. **La identificación resultante es casi exacta.** La corrida termina con la resistencia aplicada en 0,819 Ω frente a 0,83 Ω reales, un residuo de 0,011 Ω. En el cuarto tramo el torque estimado vale −18,53 N·m contra −18,53 N·m reales y el flujo real recupera 0,970 Wb: el modelo interno deja de ser una aproximación y pasa a describir la máquina. Esa última corrección, sin embargo, se decidió a los 2,6 s, 0,1 s después de invertir la referencia, con una ventana en que la máquina aún giraba hacia adelante mientras frenaba, y se aplicó mientras la velocidad cruzaba por cero; desde la decisión siguiente el veto de marcha invertida bloquea toda escritura (Subsección 6.2.4.a). En la misma rutina sin inversión, la capa termina en 0,878 Ω, con un residuo de 0,048 Ω.

La corrección se aplica además de forma *continua*: la resistencia desciende por rampas, en incrementos que no superan los 0,002 Ω, y no por saltos discretos.

7.3.2. Rutinas con resistencia de rotor de 3,00 Ω

El valor de 3,00 Ω no es nuevo para la red única, que se entrenó con él: dieciocho corridas de lazo cerrado de su conjunto de entrenamiento lo usan, y otras cinco de validación y prueba. Los expertos no lo vieron exactamente, pero sí resistencias que lo rodean: 2,83 Ω, en 110 corridas de entrenamiento del experto de arranque y en 96 del de carga, además de 4,00 Ω en el de arranque y 3,13 Ω en el de carga; para ellos, el valor es una interpolación. Lo nuevo para las dos es la **trayectoria**: ninguna corrida de entrenamiento reproduce la rutina de cinco segundos. Los expertos nunca vieron una referencia de velocidad negativa; la red única vio inversiones de giro, pero en trayectorias distintas, con la cinta al 50 % en lazo cerrado o sin carga en tramos de 1,05 s. También es nueva la **combinación** con el modelo en 7,00 Ω: la red única, con esta resistencia, vio el modelo a lo sumo en 6,00 Ω; los expertos sí se entrenaron con una combinación vecina, la máquina de 2,83 Ω con el modelo en 7,075 Ω. La repetición de la rutina pregunta, por tanto, por la generalización a una trayectoria nueva, no a una resistencia fuera de lo entrenado.

7.3.2.a. Sin corrección

$R_{r,\mathrm{MAS}} = 3{,}00\ \Omega$ con el modelo del controlador inicializado en 7,00 Ω.

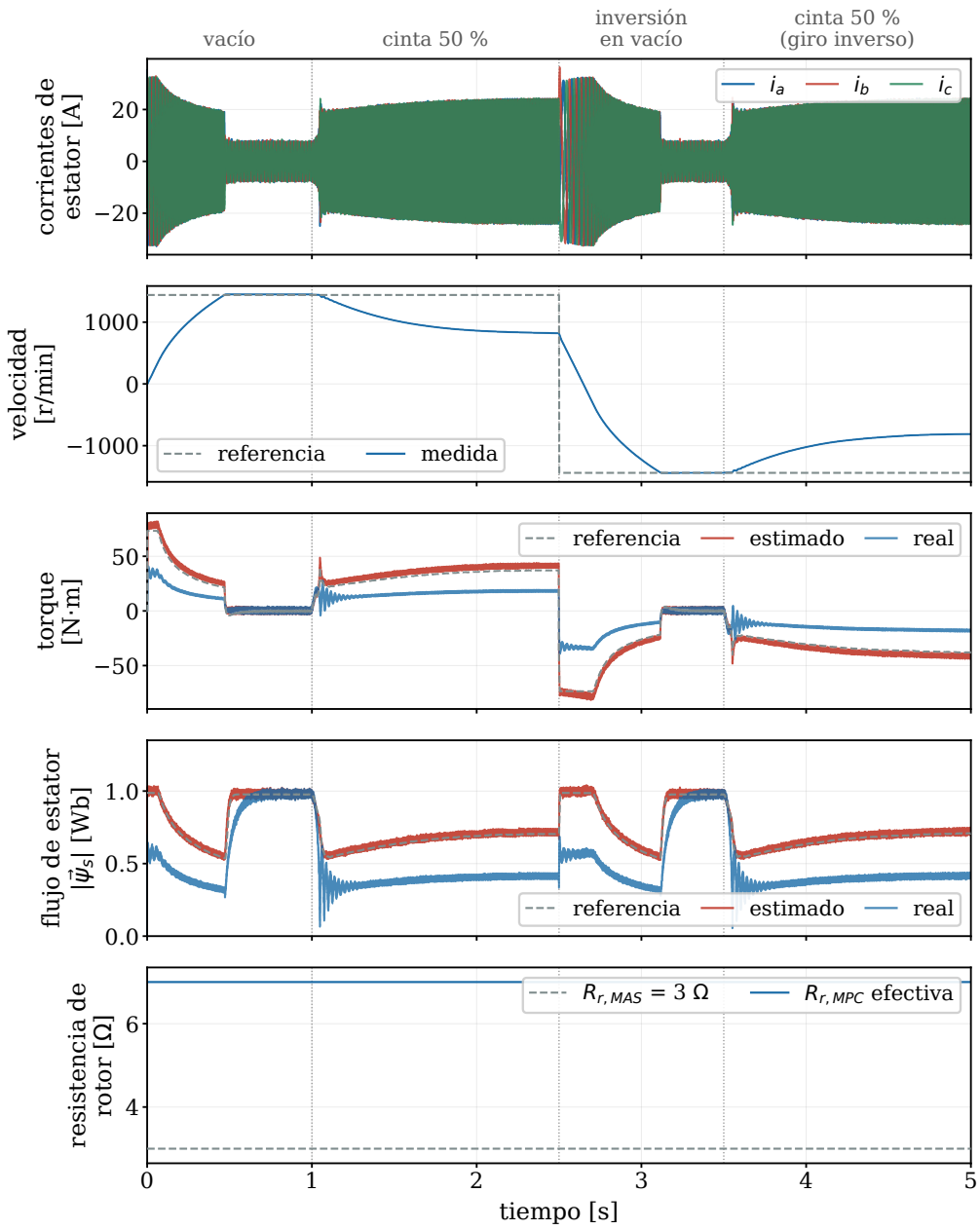


Figura 7.35: Rutina completa sobre la máquina de 3,00 Ω con el modelo sobreestimado en un factor 2,3 y sin capa correctiva.

El comportamiento reproduce el de la máquina nominal, lo que confirma que el fenómeno depende del desajuste y no del valor concreto de la resistencia: los tramos en vacío siguen la referencia con error nulo, y los cargados se estabilizan en 822 y −818 r/min, con 618 y

622 r/min de error. El torque estimado vale 40,80 N·m contra 17,92 N·m reales.

Hay, sin embargo, una diferencia que conviene registrar. Con este desajuste, menor en términos relativos, la inversión en vacío del tercer tramo *sí* cierra, alcanzando la referencia con 0,1 r/min de error, mientras que en la máquina nominal quedaba a 67 r/min. El arrastre entre tramos que allí se observaba no es, por tanto, una propiedad de la maniobra sino una consecuencia de la severidad del desajuste.

7.3.2.b. Propuesta 1

$R_{r,\mathrm{MAS}} = 3{,}00\ \Omega$ con el modelo del controlador inicializado en 7,00 Ω.

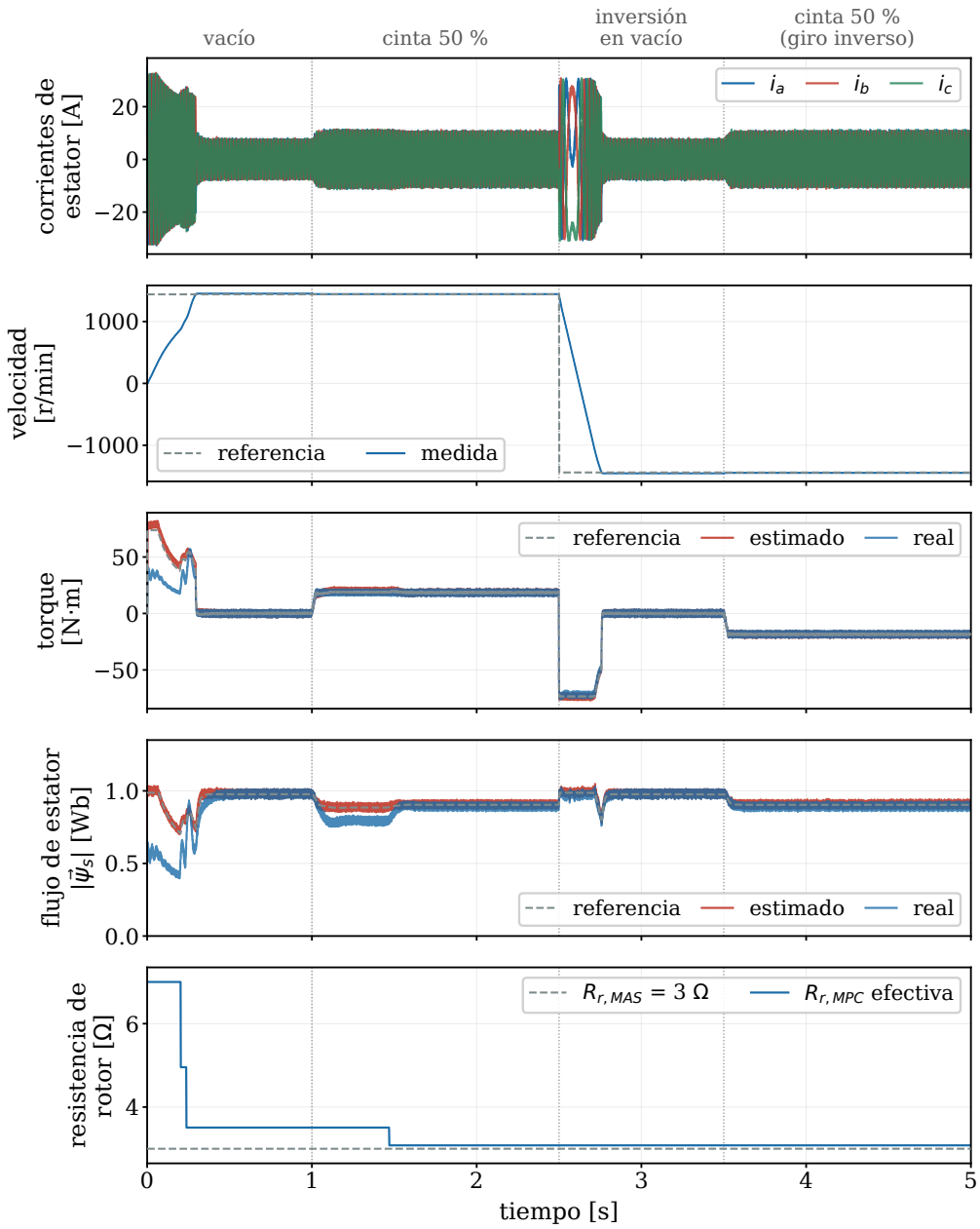


Figura 7.36: Rutina completa sobre la máquina de 3,00 Ω con el sistema de expertos.

Sobre esta máquina el sistema de expertos entrega su mejor resultado de la sección:

1. **Corrige antes y en menos pasos.** Primera escritura a los 0,203 s y tres escalones en total, el mayor de 2,044 Ω. La velocidad no baja de 1434 r/min en ningún momento de la rutina.

2. **Converge en el primer segundo y medio, y después se abstiene.** El residuo final es de 0,081 Ω sobre 3,00 Ω, esto es, un 2,7 %, frente al 4,4 % que la misma capa

deja sobre la máquina nominal. Alcanzado ese valor la capa no vuelve a escribir en los 3,5 s restantes, inversión de giro incluida: el error remanente queda por debajo de la banda muerta relativa que sus coordinadores imponen, y no actuar es entonces la conducta correcta.

3. **Y con ella el modelo interno.** El torque estimado bajo carga vale 18,61 N·m contra 18,46 N·m reales, con 0,16 N·m de discrepancia, y el flujo real alcanza 0,888 Wb frente a una referencia de 0,904 Wb.

Este comportamiento no es una particularidad de la rutina: coincide con lo que la Subsección 7.1.3 mide sobre la campaña completa, donde los indicadores que dependen del modelo interno mejoran al crecer la resistencia real de la máquina. La rutina lo exhibe en una sola corrida y sobre las señales mismas.

7.3.2.c. Propuesta 2

$R_{r,\mathrm{MAS}} = 3{,}00\ \Omega$ con el modelo del controlador inicializado en 7,00 Ω.

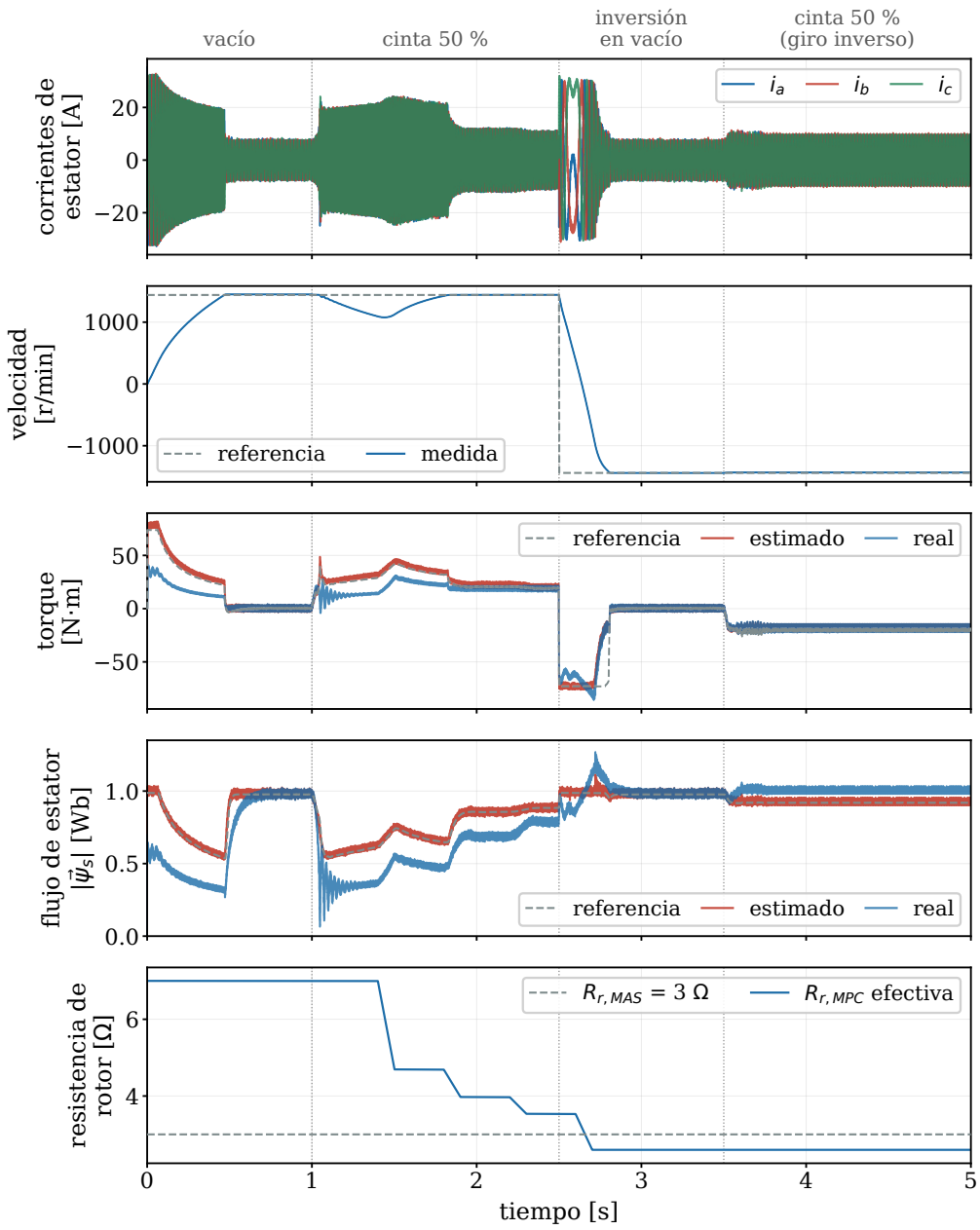


Figura 7.37: Rutina completa sobre la máquina de 3,00 Ω con el estimador inverso.

El mecanismo se conserva sobre una rutina y un punto de partida que la red no vio, aunque sí vio esta resistencia: la capa permanece inactiva durante el tramo en vacío, escribe por primera vez a los 1,400 s (la decisión de 1,0 s vuelve a quedar vetada por dominio) y aplica la corrección por rampas. La operación queda restituida, con 5,6 y 5,9 r/min de

error en los dos tramos cargados.

Lo que cambia es la *precisión* de la identificación. La corrida termina en 2,597 Ω frente a 3,00 Ω, un residuo de 0,403 Ω: treinta y siete veces mayor que el que la misma capa alcanzaba con $R_{r,\mathrm{MAS}} = 0{,}83$ Ω. El residuo cambia además de signo, pues aquí la capa se pasa de largo y termina subestimando. Ese paso de largo ocurre en la última escritura, decidida a los 2,6 s, al comienzo de la inversión: lleva la resistencia de 3,536 a 2,597 Ω mientras la velocidad cruza por cero, y el veto de marcha invertida conserva después ese valor. Esta corrida muestra que la precisión de la red única no es uniforme dentro del intervalo de $R_{r,\mathrm{MAS}}$ explorado; no demuestra, por sí sola, transferencia a otra máquina, porque las inductancias y las bases del accionamiento permanecen invariantes.

7.3.3. Lectura conjunta

La Tabla 7.20 reúne las seis corridas.

Tabla 7.20: Las seis corridas de la rutina. El error de velocidad y el de estimación de torque corresponden al segundo tramo, con la máquina cargada; el residuo de resistencia, al final de la corrida.

Máquina	**Configuración**	**Residuo de R_r** [Ω]	**Error de velocidad** [r/min]	**Error estim. de torque** [N·m]	**1ª acción** [s]	**Pico de corriente** [A]
0,83 Ω	Solo MPC	3,000	672,7	62,35	—	33,2
	Propuesta 1	0,037	7,3	10,58	**0,336**	14,4
	Propuesta 2	**0,011**	**6,1**	4,80	1,400	**11,9**
3,00 Ω	Solo MPC	4,000	618,1	22,91	—	23,8
	Propuesta 1	**0,081**	**5,5**	**0,16**	**0,203**	**10,5**
	Propuesta 2	0,403	5,6	1,67	1,400	11,4

Cuatro comentarios se siguen de la tabla.

Primera: las dos propuestas restituyen la operación por igual. El error de seguimiento bajo carga pasa de más de 600 r/min a menos de 8 en las cuatro corridas asistidas, sin que la diferencia entre propuestas supere los 2 r/min. Si el criterio es entregar el punto de operación solicitado, ambas resuelven el problema y la elección entre ellas debe apoyarse en otra cosa.

Segunda: difieren en cuándo actúan, y eso se paga en el transitorio. El sistema de expertos escribe su primera corrección durante el arranque, a los 0,336 s, con la máquina acelerando a 897 r/min; la premagnetización había terminado a los 0,012 s, y sus redes no se evalúan antes de que concluya. La escribe antes de que la carga entre, y la máquina no llega a hundirse. El estimador inverso no escribe hasta los 1,4 s: su primera decisión está programada a 1,0 s, con una ventana todavía en vacío que el guardián veta por dominio, y la siguiente llega 0,4 s después, y en ese intervalo la velocidad cae hasta 979 r/min en la máquina nominal. La diferencia de instantes depende, por tanto, de la programación de las decisiones y del veto, y no solo de lo que cada red infiere; no se ensayó otra programación.

Tercera: la precisión de identificación depende de la configuración de $R_{r,\mathrm{MAS}}$. Con el valor nominal de placa, la red única deja un residuo de 0,011 Ω frente a los 0,037 Ω del sistema de expertos. Esa diferencia depende de la inversión: la última escritura de la red

única se decidió al comienzo de la maniobra, y en la misma rutina sin inversión los residuos son 0,048 y 0,056 Ω, prácticamente iguales. Con $R_{r,\mathrm{MAS}} = 3{,}00\ \Omega$ la relación se invierte con holgura: el sistema de expertos cierra en 0,081 Ω y la red única en 0,403 Ω, cinco veces más, y es también el sistema de expertos el que entrega el mejor modelo interno, con 0,16 N·m de error de estimación de torque frente a 1,67. La comparación sólo demuestra una variación de desempeño frente a $R_{r,\mathrm{MAS}}$ dentro del mismo modelo de máquina.

Cuarta: la inversión de giro no perturba la operación, pero ninguna de las dos propuestas identifica en marcha inversa. Los tramos en vacío posteriores a una inversión cierran con menos de 0,2 r/min de error en las cuatro corridas asistidas, y ninguna de las dos capas deshace ni interrumpe la corrección ya realizada al cambiar el sentido de marcha. En el sistema de expertos solo la red de carga sigue actuando, porque ya estaba armada y el transitorio de la inversión se lee como un escalón de carga; el coordinador de arranque no se rearma. Sobre la máquina nominal, los dos escalones posteriores a la inversión continúan la progresión en la misma dirección que los anteriores; sobre la de 3,00 Ω, donde el residuo remanente ya está bajo la banda muerta, la capa no vuelve a escribir. La red única conserva la corrección que traía: el veto de marcha invertida bloquea toda escritura desde la primera decisión cuya ventana es mayoritariamente inversa. Corregir antes de invertir y conservar el parámetro no demuestra, por tanto, identificación en sentido negativo, y desde reposo con referencia negativa ninguna de las dos corrige: en el sistema de expertos las redes no actúan y la vía de *breakaway* se activa sin que el rotor esté trancado, y la red única queda vetada en todas sus decisiones (Sección G.3 y Subsección 6.2.4.a).

7.4. Síntesis comparativa de las dos propuestas

Las dos campañas de evaluación comparten un subconjunto de 104 condiciones de la configuración nominal: 91 desajustadas y 13 ajustadas. La Tabla 7.21 contrasta ambas propuestas sobre ese subconjunto y sólo mediante indicadores cuya definición es homogénea.

Tabla 7.21: Comparación sobre las condiciones desajustadas del subconjunto nominal común. Se muestran las medias de las 63 condiciones de cinta y las 28 de bomba.

	Cinta, 1440 **r/min**			**Bomba**		
Indicador	Solo MPC	Expertos	iPINN	Solo MPC	Expertos	iPINN
Residuo de R_r [Ω]	3,000	**0,124**	0,153	3,000	0,802	**0,103**
Error de velocidad [r/min]	612,7	29,0	**5,7**	260,0	8,9	**6,7**
Distorsión armónica [%]	5,72	2,10	**2,09**	5,12	**1,07**	1,12
Error estim. de torque [N·m]	30,27	1,18	**0,48**	29,51	10,59	**0,90**
Ripple de torque [N·m]	4,03	1,57	**1,56**	1,62	**1,20**	1,48
Ripple de flujo [Wb]	0,0273	0,0137	**0,0135**	0,0188	0,0179	**0,0135**
Alcanza su consigna	44,4 %	93,7 %	**100** %	42,9 %	**100** %	**100** %
Protege el caso ajustado	—	**9 de 9**	**9 de 9**	—	**4 de 4**	3 de 4

Sobre esta tabla **la red única obtiene mejores cifras en la mayoría de los indicadores**, y de forma marcada en el error de seguimiento de velocidad de la cinta y en el residuo paramétrico de la bomba. El sistema de expertos conserva la ventaja en el residuo de la cinta y, sobre todo, en la condición de rechazo, que cumple sin excepciones.

Pero **las dos propuestas no son lo mismo** y el número no dice por sí solo cuál conviene.

1. **El sistema de expertos aprende a corregir y calificar un estimador físico.** La proyección $\hat{d}$ del residuo de predicción aporta una estimación analítica inicial. A partir de mediciones del accionamiento, las redes corrigen su sesgo y producen indicadores de confianza y dominio. La resistencia aplicada no procede ciegamente de una salida neuronal: el árbitro combina esas salidas con la evidencia física antes de autorizar una escritura.

2. **La red única estima directamente el parámetro.** Recibe una ventana de señales y entrega la corrección de la resistencia sin pasar por $\hat{d}$. Sus pesos se ajustaron con datos normalizados mediante las bases de corriente, tensión y velocidad del accionamiento simulado. Por ello, los resultados sólo respaldan su desempeño en ese modelo y en las configuraciones evaluadas; el traslado a una máquina con otros parámetros no fue ensayado.

De esa diferencia se siguen dos asimetrías que la tabla exhibe, y una tercera que solo es aparente.

Primera: la red única gana en precisión. Estimar directamente el parámetro, sin pasar por un observable intermedio, evita heredar las limitaciones de ese observable. El sistema de expertos está acotado por la resolución de $\hat{d}$ y por las bandas muertas relativas que sus coordinadores imponen para no actuar sobre ruido; la red única no tiene ese piso, porque no arbitra sobre una estimación sino que la produce. Es coherente que en la bomba, donde el residuo pierde calidad y los expertos se vuelven conservadores, la diferencia sea de casi un orden de magnitud.

Segunda: el sistema de expertos protege mejor el caso ajustado. Su abstención no depende de que una red lo decida: antes de consultarlas, el coordinador comprueba con física que exista desajuste, y solo entonces pregunta. La red única no tiene esa instancia previa, de modo que su condición de rechazo descansa enteramente en lo aprendido, y la carga bomba no figura en ningún ejemplo de su entrenamiento: con ella, a velocidad reducida, aparece el único falso positivo de la campaña. Los expertos, en cambio, sí se entrenaron con carga bomba, también a 0,40 y 0,70 veces la velocidad nominal, de modo que esta asimetría no puede atribuirse solo a la arquitectura.

Tercera: el rango de resistencia no las separa. La tabla podría leerse como una diferencia de alcance entre ambas propuestas, pero no lo es. **Las dos se entrenaron sobre un rango amplio de resistencia de rotor**: el experto de arranque recorre de 0,30 a 10,00 Ω y el de carga de 0,83 a 6,33 Ω (Subsecciones 5.3.3.a y 5.3.3.b), y la red única, diecinueve valores de 0,10 a 10,00 Ω en entrenamiento, cuarenta y dos contando la validación, la prueba y los conjuntos heredados (Subsección 6.3.3). La comparación directa se hace sobre la configuración nominal, 0,83 Ω, que está en el entrenamiento de los dos expertos y de la red única. Para esta última, los puntos de partida de 5,83 y 9,83 Ω, 7,0 y 11,8 veces la resistencia real, superan el desajuste de todos sus ejemplos de regresión, que no pasa de cinco veces, y el de 9,83 Ω supera además el mayor desajuste inicial de todo su entrenamiento, diez veces. Las cinco resistencias de la campaña de la Propuesta 1, en cambio, no caen todas en su dominio: 0,10 Ω queda por debajo de los dos expertos y 0,50 Ω, del de carga (Subsección 7.1.3).

En síntesis, y sobre el dominio que ambas declaran, **las dos propuestas resuelven el problema y ninguna domina a la otra en todos los indicadores**: la red única identifica con mayor precisión y el sistema de expertos protege sin excepciones la condición de no actuar sobre un modelo correcto. El eje que las separa no es el rango de resistencias que cubren (que es comparable) sino **cuánto del resultado descansa en lo aprendido**: en una, la estimación entera; en la otra, únicamente el juicio sobre una estimación que la física ya proporciona.

Capítulo 8

Conclusiones

Esta memoria abordó la posibilidad de compensar, mediante redes neuronales informadas por física, el desajuste de la resistencia de rotor empleada por el modelo interno de un controlador predictivo. La intervención se restringió a la actualización supervisada de $R_{r,\mathrm{MPC}}^{\mathrm{eff}}$: ninguna red actuó directamente sobre las compuertas del inversor ni sustituyó la selección de vectores del FCS-MPC.

8.1. Síntesis y discusión

La evaluación se organizó en dos campañas de distinto alcance. La **Propuesta 1**, correspondiente al sistema de expertos, comprendió 2418 corridas: 806 condiciones de operación evaluadas en tres modos de ablación. La **Propuesta 2**, correspondiente al estimador inverso de una sola red, se evaluó sobre el subconjunto nominal común, compuesto por 91 condiciones desajustadas y 13 ajustadas; al ejecutar cada condición con y sin la capa se obtuvieron 208 corridas. Por consiguiente, la comparación directa entre propuestas se limita a esas 104 condiciones y no se extiende a toda la campaña de expertos (Sección 7.4).

En la rutina de cinta, con $R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$ y velocidad nominal, el error medio de seguimiento sobre las condiciones desajustadas disminuyó desde 612,7 r/min sin corrección hasta 29,0 r/min con el sistema de expertos y 5,7 r/min con la red única. En ese mismo subconjunto, el residuo paramétrico medio se redujo desde 3,00 Ω hasta 0,124 Ω y 0,153 Ω, respectivamente. Estas cifras caracterizan una rutina y una configuración concretas; no representan el promedio de las 2418 corridas ni prueban transferencia a una máquina distinta.

La campaña amplia de la Propuesta 1 mostró que la corrección recupera operaciones que el controlador desajustado no puede sostener, pero también delimitó su alcance. La Propuesta 2 recuperó las 91 condiciones desajustadas de su campaña nominal, aunque presentó un falso positivo en una de las 13 condiciones ajustadas. Por ello, ninguna conclusión debe formularse como protección absoluta de todos los escenarios.

Los indicadores separan además dos efectos. El error de estimación de torque y flujo sigue al residuo paramétrico, mientras el *ripple* y la distorsión deben compararse sólo una vez restituido el mismo punto de operación. Una máquina que no alcanza su referencia de velocidad puede exhibir valores artificialmente bajos de rizado y THD. En consecuencia, la capa no elimina el piso de calidad de onda propio del conjunto finito de vectores; recupera

el punto de operación y, con él, la calidad correspondiente al controlador ajustado. El error de estimación continúa dependiendo de cuánto residuo de R_r permanece.

8.2. Ventajas y desventajas de cada propuesta

Las dos propuestas comparten la interfaz con el controlador, pero distribuyen de manera distinta el trabajo entre el modelo físico, las redes y la lógica de seguridad. La Tabla 8.1 resume las diferencias respaldadas por la evaluación.

Tabla 8.1: Síntesis comparativa.

	Propuesta 1: expertos	**Propuesta 2: una red**
Magnitud aprendida	Corrección y confiabilidad de una estimación física	Corrección directa de R_r desde una ventana de señales
Núcleo de identificación	Proyección física $\hat{d}$ y cuatro redes especializadas	Pesos de una única red
Residuo en cinta nominal	0,124 Ω	0,153 Ω
Residuo en bomba nominal	0,802 Ω	0,103 Ω
Condición de rechazo evaluada	Sin falsos positivos en las configuraciones ensayadas	Doce aciertos en trece condiciones; un falso positivo
Momento de actuación	Desde el arranque, una vez terminada la premagnetización	Requiere una ventana informativa, normalmente con carga, y solo en las decisiones programadas: desde 1,0 s, cada 0,4 s
Seguridad en línea	Árbitros, guardián, veto/rollback local y respaldo físico	Compuerta y límites de escritura, sin respaldo físico equivalente
Complejidad	Cuatro redes y dos coordinadores	Una red y un algoritmo de escritura
Cómputo por inferencia	53 248 multiplicaciones en cada ciclo eléctrico; no medido en hardware	1 789 824 multiplicaciones cada 0,4 s; no medido en hardware
Marcha en sentido inverso	Las redes no actúan desde reposo; la red de carga sigue tras invertir	Escritura vetada; conserva el valor
Transferencia a otra máquina	No evaluada	No evaluada

Propuesta 1: sistema de expertos

Su principal ventaja es que parte de un observable físico. La proyección $\hat{d}$ entrega una estimación inicial a partir del residuo del predictor y de su sensibilidad analítica; las redes corrigen el sesgo de esa estimación y producen indicadores de confianza y dominio. Los árbitros incorporan esta evidencia antes de autorizar una escritura, y las rutas de

breakaway y emergencia proporcionan pasos físicos acotados cuando las redes se abstienen. Esta separación permite inspeccionar qué componente propuso, aceptó o rechazó cada corrección.

El costo de esa trazabilidad es una mayor complejidad. El sistema reúne cuatro redes, dos coordinadores con máquinas de estados y rutas físicas de respaldo. Su precisión queda limitada por la calidad de $\hat{d}$, por las bandas muertas y por la observabilidad disponible. El residuo se comportó como un piso absoluto: resultó pequeño frente a $R_{r,\mathrm{MAS}} = 5\ \Omega$, pero excesivo frente a 0,10 Ω, valor que además queda por debajo de la menor resistencia real con que se entrenaron los expertos. La consistencia observada entre 0,50 y 5,00 Ω no autoriza a afirmar que el mismo conjunto de pesos se transferirá sin cambios a otra máquina. Además, el vector de entrada que el `C-Script` entrega a sus redes no coincide exactamente con el que produjo el extractor de entrenamiento: difieren la historia de las redes de carga, el instante de las variables internas del controlador, dos definiciones y algunas fronteras de ciclo (Sección G.1). Sus métricas fuera de línea describen, por ello, a las redes sobre la distribución de entrenamiento; su desempeño desplegado es el que mide la campaña en lazo cerrado. Su alcance, por último, es la marcha en sentido positivo. El ajuste de la entrada al sentido de giro alcanza a las redes pero no a los coordinadores: desde reposo con referencia negativa ninguna red corrige y la vía de *breakaway* interpreta la marcha inversa como un rotor trancado. Tras un arranque en sentido positivo, la red de carga sigue actuando después de invertir (Sección G.3).

Propuesta 2: estimador inverso de una sola red

La red única evita el estimador intermedio y entrega directamente una corrección del parámetro. Dentro del subconjunto nominal común fue más precisa en la mayoría de los indicadores y, en particular, en la carga de bomba. Su estructura es más simple: concentra la inferencia y la escritura en una sola rama, con menos piezas que coordinar. Eso no implica un menor costo de cómputo. Contadas sobre el código, su red hace 1,79 millones de multiplicaciones por inferencia, treinta y cuatro veces las de las cuatro redes de la Propuesta 1 en un ciclo, y a 50 Hz, 1,7 veces más por segundo. Ninguna de las dos se midió en un procesador de control, y el período de 100 μs es el del controlador simulado: su cumplimiento en hardware queda pendiente.

Su principal debilidad es que la condición de rechazo depende de lo aprendido. La campaña registró un falso positivo con el modelo ajustado, y el barrido adicional de validez mostró que, fuera de la región en que la ventana contiene suficiente información, la red puede corregir en la dirección equivocada. Sus entradas se normalizan con las bases del accionamiento simulado; por ello, aplicar la red a otra máquina exigiría como mínimo recalcular esa interfaz y volver a validar el modelo, y probablemente regenerar los datos y reentrenar. Este comportamiento esperado no se comprobó en otra máquina durante el trabajo. Tampoco identifica en marcha inversa: su guardián veta la escritura en ese régimen y conserva la corrección obtenida antes (Subsección 6.2.4.a).

Criterio de elección

Estas conclusiones describen a cada propuesta como el sistema completo que se evaluó: redes, coordinadores o guardián, bandas muertas y rutas físicas. La campaña de la Propuesta 1 separa el experto de arranque de la capa completa, pero ninguna campaña

apaga las reglas físicas manteniendo las redes, ni al revés, y la Propuesta 2 se evaluó siempre con su guardián. El análisis de la Subsección 7.1.6 dice qué componente escribió cada corrección, no qué habría ocurrido sin él. Los resultados no permiten, por tanto, repartir la mejora entre lo aprendido y lo programado.

Sobre el subconjunto evaluado, la red única ofrece una estructura más simple, aunque no un menor costo de cómputo, y mayor precisión, mientras el sistema de expertos ofrece una decisión más trazable y una condición de rechazo respaldada por evidencia física. No existe evidencia suficiente para recomendar una de ellas para una familia de máquinas: la elección para ese propósito requiere primero ensayar cambios de inductancias, bases eléctricas y potencia. Una extensión razonable consiste en combinar ambas: usar la estimación directa cuando su dominio esté respaldado y conservar el estimador físico como condición de rechazo y ruta auxiliar.

8.3. Contribuciones

Las contribuciones principales del trabajo son:

1. **Un estimador físico en línea del desajuste de R_r**, deducido de la afinidad del predictor de corriente del MPC y calculable sin utilizar $R_{r,\mathrm{MAS}}$ durante la operación.
2. **Una arquitectura de expertos por maniobra y dirección de corrección**, con salidas acotadas de corrección, confianza y dominio, combinadas con características deducidas del modelo.
3. **Un estimador inverso alternativo de una sola red**, utilizado para cuantificar las ventajas y limitaciones de prescindir del observable físico intermedio.
4. **Una metodología de validación por escenario**, con partición de datos por corrida o trayectoria completa, métricas por zona de operación, ablaciones, condiciones ajustadas y pruebas en lazo cerrado con aplicación real de la corrección.

8.4. Trabajo futuro

El desarrollo realizado abre las siguientes líneas de continuación:

- **Sistematizar la evaluación fuera del dominio entrenado.** La exploración del Anexo F utiliza cuatro resistencias y una sola rutina; convertirla en evidencia concluyente exige una malla comparable a la campaña principal.
- **Trasladar ambas propuestas a una máquina distinta**, con otras inductancias, potencia y bases eléctricas. Esta es la prueba necesaria para estudiar transferencia entre máquinas y no debe sustituirse por un barrido aislado de $R_{r,\mathrm{MAS}}$.
- **Mejorar la observabilidad a baja velocidad**, donde el residuo aumenta por el menor contenido informativo de las señales. Esto puede requerir señales adicionales o excitación deliberada, no sólo más entrenamiento con las mismas entradas.

- **Unificar el extractor de entrenamiento con el constructor desplegado** de la Propuesta 1: leer las variables internas del controlador en su propio instante, adoptar las mismas definiciones y el mismo orden de la memoria, regenerar los datos y reentrenar, de modo que las métricas fuera de línea describan exactamente la entrada en línea.
- **Buscar de forma sistemática los pesos de las funciones de pérdida**, por ejemplo con optimización bayesiana, eligiéndolos sobre la validación con varias semillas y reservando la prueba para la medida final. En este trabajo se fijaron al diseñar las redes, y solo uno por propuesta se ajustó con ensayos (Tablas 5.8 y 6.5).
- **Extender la corrección a la marcha inversa.** En la Propuesta 1 exige referir al sentido positivo también las comparaciones de los coordinadores (el progreso del arranque, el rearme, la firma y la salida del estado inicial del coordinador de carga), y validar después el arranque inverso desde reposo. En la Propuesta 2 exige entregar a la red la ventana en espejo, con el signo de $i_{s\beta}$, $v_{s\beta}$ y las dos velocidades cambiado, que fuera de línea ya reproduce la estimación de la marcha directa (Subsección 6.2.4.a), y validarla en lazo cerrado antes de retirar el veto.
- **Extender el método a otros parámetros**, en particular a la inductancia magnetizante L_m bajo saturación.
- **Realizar validación experimental** en un banco de pruebas real para contrastar el sesgo de $\hat{d}$, el costo de inferencia, su cabida en el período de control y la robustez frente a ruido, retardos y no idealidades.

En síntesis, ambas propuestas demostraron que es posible modificar de forma supervisada la resistencia usada por el modelo interno del FCS-MPC a partir de señales disponibles en línea. El sistema de expertos aportó trazabilidad, arbitraje y respaldo físico. La red única fue estructuralmente más simple y recuperó las 91 condiciones desajustadas de su campaña nominal, aunque produjo un falso positivo en una condición ajustada. Estos resultados respaldan la viabilidad del enfoque dentro del dominio ensayado.

Apéndice A

Parámetros e Inicialización de la Simulación

Este apéndice documenta la inicialización de la plataforma de simulación definitiva descrita en el Capítulo 4. El Listado A.1 reproduce los comandos de inicialización del modelo PLECS: los parámetros nominales de la máquina y del controlador (Tabla 4.1); las dos resistencias independientes —`Rr_MAS`, aplicada a la máquina simulada, y `Rr_MPC`, la resistencia inicial del predictor, cuyo valor en el listado corresponde al escenario guardado y no al nominal del sistema—; los interruptores de los modos de evaluación; la rutina parametrizada de nueve períodos; y los parámetros del modelo de carga física (selector de modo, saturación, regularización y rampa de material). Estos valores constituyen los ajustes por defecto; en las campañas automatizadas son sobrescritos por RPC.

El orden de los parámetros del bloque `C-Script` principal es:

```
Ts; Rs; Rr_MPC; Lm; Ls; Lr; p; psi_s_nom; lambda_ptc; kw; cw1; cw2;
                                T_ref_max;
PINN_arranque; PINN_carga; PINN_inversion; CLK_ENABLE; CLK_WARMUP_STEPS;
    PINN_APPLY; PINN_CARGA_APPLY; i_s_max; w_s; k_vlim; PINN_V2_APPLY
```

El renombre de `psi_s_nom` —que antes se llamaba `psi_s_ref`, nombre reservado ahora para la referencia *efectiva*— y los parámetros `i_s_max`, `w_s` y `k_vlim` corresponden a las correcciones del controlador descritas en el Capítulo 3. El primero es el valor de pico nominal de la corriente de estator; el doble de ese valor es el límite que se penaliza solo durante la magnetización (Sección 3.4.4); el segundo define la pulsación de alimentación con que se construye la velocidad nominal; y `k_vlim` fija el margen de tensión respecto del círculo inscrito del hexágono del inversor (Subsección 3.5.2). El parámetro final, `PINN_V2_APPLY`, habilita la escritura del sistema direccional de expertos evaluado. Los índices existentes no se reordenaron al incorporar estos parámetros. Los parámetros derivados del modelo (τ_r, R_σ, τ_σ) no forman parte de esta lista: se recalculan en cada período dentro de la *Update Function* a partir de la resistencia efectiva vigente $R_{r,\mathrm{MPC}}^{\mathrm{eff}}$, de modo que la corrección en línea se propaga a toda la cadena de estimación y predicción (Subsección 4.1.3).

```
% ======================================================================
% PARAMETROS DE SIMULACION PLECS - PTC con predicción k+2
% ======================================================================

```

```
% Frecuencia y tiempo de muestreo
frec_conm = 10000;
Ts = 1/frec_conm;

% Parámetros de la máquina de inducción
Rs = 1.02;
Rr = 0.83;
Lm = 153.6e-3;
Ls = 160.3e-3;
Lr = 160.3e-3;
J  = 0.06;
p  = 2;
n_nom = 1440; % RPM
w_nom = n_nom*2*pi/60;
T_nom = 5500/(w_nom);
s_nom = (2*pi*50-w_nom*p)/(2*pi*50);

% Corriente nominal de placa: 11,81 A de linea EFICACES. La Clarke del C-Script
% es de amplitud invariante, luego el limite que ve el predictor es el valor de
% PICO. Solo al magnetizar: la penalizacion acota |i_s[k+2]| en 2*i_s_max.
i_linea_nominal = 11.81;
i_s_max = i_linea_nominal*1.414213562;

% Tensión bus DC
V_dc = 380*sqrt(2);

pi = 3.14159265358979323846;
f_red = 50;
w_s = 2*pi*f_red;

% Referencia flujo estator
% Referencia de flujo de estator (CORREGIDA en v31).
% La Clarke del C-Script es de amplitud invariante, luego en regimen
% |psi_s| = V_pico_fase/w_s = sqrt(2/3)*V_LL/w_s, con V_LL la tension nominal
% linea-linea. Es la ec. (2.14) de Mora escrita en flujo, ya que en vacio y en
% regimen psi_s = Ls*i_mr, de donde i_mrN = psi_s_nom/Ls. Equivale ademas a
% V_dc/(sqrt(3)*w_s), el circulo inscrito del hexagono del inversor.
% v30 usaba (sqrt(2)/3)*V_dc/w_s = 0.8064 Wb, un factor sqrt(3/2) = 1.2247
% menor: la maquina operaba sub-fluxada un 18 %.
V_sN = 380;           % tension nominal linea-linea [V]
% v34: es el valor NOMINAL. La referencia efectiva la calcula el C-Script
% en cada periodo, debilitando el campo por encima de w_s (Mora ec. 2.15).
psi_s_nom = sqrt(2/3)*V_sN/w_s;
% v35: margen de tension del inversor. El limite lineal es Vdc/sqrt(3) (circulo
% inscrito del hexagono); la tension que la maquina exige supera a w_s*|psi_s|
% en la caida resistiva y el termino de dispersion, de modo que k<1 evita rozar
% el borde. Parametro del INVERSOR: no depende de la maquina.
k_vlim = 0.95;

% Factor de peso función de costo
lambda_ptc = 2*T_nom / psi_s_nom;


% Controlador PI velocidad v2- Anti-Windup 15khz
%kw  = 21.532466235954939;
%cw1 = 0.995085137953149;
%cw2 = -2.282535587421024e-04;


% Controlador PI velocidad v2- Anti-Windup 10khz
kw  = 16.101262536094396;
cw1 = 0.999999999300000;
cw2 = -4.347484834352330e-11;

% Controlador PI velocidad v2- Anti-Windup 10khz LIACE
%kw  = 1.663193148700254e+02;
%cw1 = 0.999400000000000;
%cw2 = -3.607518468128194e-06;
```

```
73
74 %kw  = 1.5426502611702 70e+02;
75 %cw1 = 0.99930;
76 %cw2 = -4.537645489840366e-06;
77
78
79 T_ref_max = 2*T_nom;
80
81 % Deltas que multiplican solo los parametros de la Maquina Real
82 delta_rs = 1;
83 %delta_rr = 1; %YA NO SE USA ESTE DELTA_RR, AHORA DIRECTO RR_MAS
84 delta_ls = 1;
85 delta_lr = 1;
86 delta_lm = 1;
87
88 % Delta para cambiar el valor inicial del Rr utilizado por el MPC
89 delta_rr_MPC= 1; %YA NO SE USA ESTE DELTA_RR_MPC, AHORA DIRECTO RR_MPC
90
91 % --- Las DOS resistencias (lo unico que cambias para probar) ---
92 Rr_MAS = 5;      % Rr REAL de la maquina
93 Rr_MPC = 0.1;      % Rr que cree el MPC
94
95
96 % ===== EXPERTOS PINN: solo activar(1)/desactivar(0) =====
97 PINN_arranque=1;  PINN_carga=1;  PINN_inversion=0;
98 % ===== Clock (medicion de costo computacional) =====
99 CLK_ENABLE=1;  CLK_WARMUP_STEPS=500;
100
101 PINN_APPLY = 1;
102
103 PINN_CARGA_APPLY = 1;
104
105 % Habilitación de escritura de los expertos direccionales v2.
106 % Con valor 0, las redes y los coordinadores operan sólo en diagnóstico.
107 PINN_V2_APPLY = 1;
108
109
110 % ===== Maniobra (caso C3 validado): arranque SIN carga -> wref -> ENTRADA de carga
      sostenida =====
111 % Periodo 1: 0.7 s a wref SIN carga (arranque). Periodo 2: carga -0.45*T_nom
    SOSTENIDA (StopTime=2 s la corta).
112 % NO arrancar cargado (con Rr sobreestimado se vuelve inestable): la carga entra
    recien cuando ya gira a wref.
113 % Diagnostico (PINN_APPLY=0 y PINN_CARGA_APPLY=0): Rr_MPC_effect (Output 40) queda
    PLANO.
114 % Esperado: Output(76) detecta escalon (~0.7 s); FSM (Output 67) avanza a
    LOAD_STEADY_ANALYSIS->DONE;
115 %           Output(66)=1 (carga_active); Output(65) ~ Rr_MAS=1.33 (candidato, NO se
      aplica con APPLY=0).
116 t_dt1=3;  t_dt2=3;  t_dt3=3;  t_dt4=3;  t_dt5=3;  t_dt6=3;  t_dt7=3;  t_dt8=3;
    t_dt9=0.0;
117 w_lvl1=2*pi*50*(1-s_nom)/p;  T_lvl1=-1*T_nom;
118 w_lvl2=2*pi*50*(1-s_nom)/p;  T_lvl2=-1*T_nom;
119 w_lvl3=2*pi*50*(1-s_nom)/p;  T_lvl3=-1*T_nom;
120 w_lvl4=2*pi*50*(1-s_nom)/p;  T_lvl4=-1*T_nom;
121 w_lvl5=2*pi*50*(1-s_nom)/p;  T_lvl5=-1*T_nom;
122 w_lvl6=2*pi*50*(1-s_nom)/p;  T_lvl6=-1*T_nom;
123 w_lvl7=2*pi*50*(1-s_nom)/p;  T_lvl7=-1*T_nom;
124 w_lvl8=2*pi*50*(1-s_nom)/p;  T_lvl8=-1*T_nom;
125 w_lvl9=2*pi*50*(1-s_nom)/p;  T_lvl9=-1*T_nom;
126
127
128 % ===== CARGA FISICA (planta v25): fuente de torque REACTIVA seleccionable =====
129 % Reemplaza a "Escalones multiples". El par se OPONE a la velocidad mecanica real
130 % y vale ~0 en w=0 -> la maquina NUNCA acelera hacia atras desde reposo.
131 %   LOAD_MODE = 0  sin carga  (T = 0)
132 %   LOAD_MODE = 1  bomba       (T = -K*w*|w|,  K = T_nom/w_nom^2 ; ley de afinidad)
133 %   LOAD_MODE = 2  cinta       (T = -(T_BASE + material)*tanh(w/W_EPS))
```

```
% En modo 2 el material usa el cronograma t_dt_i / |T_lvl_i| (magnitud), con rampa.
LOAD_MODE  = 2;
T_LOAD_MAX = 1.5*T_nom;    % saturacion de proteccion [N*m]
B_VISC     = 0.0;          % friccion viscosa (idlers/rodamientos) [N*m*s/rad]
W_EPS      = 1.5;          % regularizacion de friccion cerca de w=0 [rad/s]
T_BASE     = 0.*T_nom;  % friccion de la correa vacia (modo cinta) [N*m]
RAMP_MAT   = T_nom/0.05;    % rate-limit de entrada/salida de material [N*m/s]
```

Listing A.1: Comandos de inicialización del modelo PLECS de la plataforma definitiva: parámetros de la máquina y del controlador, resistencias independientes, modos de evaluación, rutina de nueve períodos y parámetros de la carga física. Los valores de `Rr_MPC` y de la rutina corresponden al escenario guardado y se sobrescriben por RPC en las campañas.

Coeficientes del controlador de velocidad. El juego activo (k_w, c_{w1}, c_{w2} bajo el comentario ≪10khz≫) es el del primer ajuste, hecho para una máquina de inercia $J = 0{,}0055$ kg·m^2; con la inercia simulada, $J = 0{,}06$ kg·m^2, se comporta como un proporcional. Los dos juegos comentados bajo ≪10khz LIACE≫ son los ajustados para $J = 0{,}06$ kg·m^2, y el de ≪15khz≫, uno anterior para otra frecuencia de conmutación. Ninguno de los tres se emplea en los resultados. La transferencia, su verificación y la comparación con los ajustados para $J = 0{,}06$ kg·m^2 están en la Sección 3.3.

Apéndice B

Código de los Bloques C-Script

Este apéndice documenta el código correspondiente a la configuración evaluada `MPC_v36_v4.plecs` (Capítulo 4). Los cuatro primeros listados fueron extraídos de las secciones *Code Declarations*, *Start Function*, *Output Function* y *Update Function* del bloque principal. La lógica ejecutable se conserva sin modificaciones; únicamente se normalizaron algunos comentarios históricos que todavía mencionaban dimensiones de versiones anteriores. El bloque posee seis entradas, 58 estados discretos y 101 salidas de control y diagnóstico. La campaña del Capítulo 7 se corrió con `MPC_v36_v4_campana.plecs`, cuyo código difiere del impreso solo en el ajuste de la entrada por sentido de giro (Subsección 5.2.4.c), que en sentido positivo, el de todas sus corridas, es la identidad; la rutina de la Sección 7.3 usa el modelo impreso. Los dos comparten la cabecera del coordinador de carga. Los dos archivos se entregan en la carpeta reproducible de la Sección C.4.

Los pesos no se reproducen porque corresponden a matrices extensas generadas automáticamente. Los archivos vigentes son `pinn_v2_subir.h`, `pinn_v2_bajar.h`, `pinn_v2_csubir.h` y `pinn_v2_cbajar.h`; cada uno implementa una red $141 \to 64 \to 64 \to 3$. La memoria temporal compartida se encuentra en `pinn_v2_coordinator.h`, mientras que las decisiones de arranque y carga se implementan, respectivamente, en `pinn_arranque_coord_v2.h` y `pinn_carga_coord_v2.h`.

En las declaraciones se conserva la macro histórica `PINN_V2_NB=29`, que dimensiona el búfer local. No debe interpretarse como la dimensión activa de la red: la función vigente `v2_build_x` utiliza 27 características base en sus tres versiones (actual, previa y diferencia), más 60 estadísticos de historia, para formar los 141 componentes consumidos por cada cabecera neuronal. Se mantiene la declaración original porque este apéndice reproduce la configuración efectivamente evaluada.

B.1. C-Script principal: Code Declarations

```
/* ===== VERSION LIMPIA (v3 definitiva, 2026-08-08) =====
 *  Se elimino el codigo heredado que NO se ejecuta: los coordinadores de
 *  v1 (dc_step de v26 y cc22_step de v24) con sus cuatro redes
 *  (pinn_arr_1/2.h, pinn_carga_up/dn.h), y la FSM del coordinador v2
 *  rechazado. Verificado antes de cortar: solo se llaman ac2_step y
 *  k2_step. SE CONSERVA pinn_v2_coordinator.h porque aporta v2_push y
 *  v2_build_x, que SI usa v3 para armar el vector de 141 features.
 *  Las salidas y el Mux quedan intactos: las columnas de diagnostico de
 *  v1 leen 0, pero las posiciones se conservan y el utillaje sigue valido.
```

```
 * ========================================== */
/* ==========================================
 *  SECCION 1: CODE DECLARATIONS
 *  C-Script principal: PTC/FCS-MPC + expertos PINN (arranque y carga)
 * ==========================================
 *  Linaje  : base v13 (MPC k+2 + instrumentacion) -> v16 (arranque dual)
 *            -> v17..v22 (experto de carga dual) -> v24 (rectificacion)
 *            -> v25 (planta de carga fisica) -> v26 (breakaway arranque).
 *  Setup   : 6 entradas | 101 salidas | 58 estados discretos | 0 continuos
 *            Direct feedthrough = 1 | Sample time = Ts | C99 | checks ON.
 *  Headers : pinn_arranque_coord_v2.h  (+ pinn_v2_subir/bajar.h)
 *            pinn_carga_coord_v2.h     (+ pinn_v2_csubir/cbajar.h)
 *            Los .h deben estar JUNTO al .plecs (include relativo).
 *
 *  PARAMETROS DEL BLOQUE (indice ParamRealData -> nombre):
 *     0 Ts    [s]       periodo de muestreo del controlador
 *     1 Rs    [ohm]     resistencia de estator (conocida)
 *     2 Rr_MPC [ohm]    resistencia de rotor BASE del modelo del MPC
 *     3 Lm    [H]       inductancia mutua
 *     4 Ls    [H]       inductancia propia de estator
 *     5 Lr    [H]       inductancia propia de rotor
 *     6 p     [-]       pares de polos
 *     7 psi_s_nom [Wb]    magnitud NOMINAL de flujo de estator (la
 *                         referencia efectiva se debilita en la Update)
 *     8 lambda_ptc [Nm/Wb] peso del termino de flujo en la funcion de costo
 *     9 kw    [Nm s/rad]  ganancia del PI de velocidad
 *    10 cw1   [-]       coeficiente 1 del PI discreto (anti-windup)
 *    11 cw2   [-]       coeficiente 2 del PI discreto (anti-windup)
 *    12 T_ref_max [Nm]  saturacion de la referencia de torque (+/-)
 *    13 PINN_arranque   RESERVADO: leido pero SIN EFECTO en v26 (ver nota)
 *    14 PINN_carga      ENABLE del experto de carga (arg 'enable' de k2_step)
 *    15 PINN_inversion  RESERVADO: experto de inversion (futuro, sin efecto)
 *    16 CLK_ENABLE      1 = medir costo de CPU con clock()
 *    17 CLK_WARMUP_STEPS pasos ignorados antes de promediar el clock
 *    18 PINN_APPLY      arranque: 0 = solo diagnostico / 1 = aplica correccion
 *    19 PINN_CARGA_APPLY carga:   0 = solo diagnostico / 1 = aplica correccion
 *    20 i_s_max [A]    limite de PICO de la corriente de estator
 *                      (11,81 A de linea eficaces x raiz(2)). En v33
 *                      solo acota la fase de magnetizacion, al doble
 *
 *  NOTA flags (auditoria v26): el experto de ARRANQUE no tiene enable de
 *  ejecucion; con PINN_APPLY=0 corre igual (inferencia + diagnosticos) y
 *  solo se bloquea la ESCRITURA de k_rr_arr_state. PINN_arranque y
 *  PINN_inversion se conservan por compatibilidad de interfaz con los
 *  scripts RPC (ModelVars) y para el experto futuro de inversion.
 * ========================================== */

#include <math.h>
#include <float.h>
#include <stdio.h>
#define PINN_USE_CLOCK   /* mide costo computacional dentro de Ts */

/* ==========================================
 *  ENTRADAS (6)  -  Input(i)
 *  0..2 : corrientes de fase medidas ia, ib, ic          [A]
 *  3    : velocidad MECANICA del rotor w_mech            [rad/s]
 *  4    : referencia de velocidad MECANICA w_ref         [rad/s]
 *  5    : tension del enlace de continua Vdc             [V]
 * ========================================== */
#define is_a_in   Input(0)
#define is_b_in   Input(1)
#define is_c_in   Input(2)
#define w_mech_in Input(3)
#define w_ref_in  Input(4)
#define Vdc_in    Input(5)

/* ==========================================
 *  ESTADOS DISCRETOS 0..45 (BASE v13)
```

```
 *  Convencion *_out / *_prev: cajas que la Output publica tal cual;
 *  la Update las escribe al final de cada paso.
 *  ------------------------------------------------------------
 *   0..5   gates del inversor aplicados este periodo (par por pierna)
 *   6..10  nucleo del estimador y del PI (angulo, flujo, PI, i_sd)
 *  11..21  diagnosticos dq / alfa-beta / torque / velocidad
 *  22..25  angulo rotor estimado, tension dq aplicada, flag premag
 *  26..28  corrientes dq medidas y vector optimo (dataset)
 *  29..41  predicciones k+1 y k+2 del MPC (dataset)
 *  42..45  tension aplicada alfa-beta y prediccion de corriente
 *          alfa-beta a k+1 (base del residuo de la PINN)
 * ========================================== */
#define g1_prev   DiscState(0)
#define g2_prev   DiscState(1)
#define g3_prev   DiscState(2)
#define g4_prev   DiscState(3)
#define g5_prev   DiscState(4)
#define g6_prev   DiscState(5)
#define gamma_r_k    DiscState(6)   /* angulo de orientacion de campo [rad] */
#define psi_r_prev   DiscState(7)   /* flujo de rotor estimado k-1    [Wb]  */
#define x_w_prev     DiscState(8)   /* estado interno del PI (anti-windup)  */
#define T_ref_prev   DiscState(9)   /* T* del ciclo anterior          [Nm]  */
#define i_sd_prev    DiscState(10)  /* i_sd k-1 (entrada del estimador)     */
#define psi_s_d_out  DiscState(11)
#define psi_s_q_out  DiscState(12)
#define psi_r_d_out  DiscState(13)
#define psi_r_q_out  DiscState(14)  /* = 0 por orientacion de campo */
#define T_est_out    DiscState(15)
#define psi_s_alfa_out DiscState(16)
#define psi_s_beta_out DiscState(17)
#define psi_r_alfa_out DiscState(18)
#define psi_r_beta_out DiscState(19)
#define T_ref_out    DiscState(20)
#define w_eli_out    DiscState(21)  /* velocidad ELECTRICA del rotor [rad/s] */
#define theta_r_prev DiscState(22)  /* angulo MECANICO estimado (NO el real) */
#define v_sd_k       DiscState(23)  /* tension dq APLICADA este periodo [V]  */
#define v_sq_k       DiscState(24)
#define magnetized_flag DiscState(25)  /* 0 = pre-magnetizando, 1 = listo */
#define is_d_out      DiscState(26)
#define is_q_out      DiscState(27)
#define idx_opt_out   DiscState(28)  /* indice 0..7 del vector optimo */
#define i_sd_k1_out      DiscState(29)
#define i_sq_k1_out      DiscState(30)
#define psi_sd_k1_out    DiscState(31)
#define psi_sq_k1_out    DiscState(32)
#define i_sd_k2_opt_out      DiscState(33)
#define i_sq_k2_opt_out      DiscState(34)
#define psi_sd_k2_opt_out    DiscState(35)
#define psi_sq_k2_opt_out    DiscState(36)
#define T_k2_opt_out         DiscState(37)
#define psi_s_mag_k2_opt_out DiscState(38)
#define g_opt_out        DiscState(39)  /* costo del vector ganador */
#define v_sd_k1_opt_out  DiscState(40)
#define v_sq_k1_opt_out  DiscState(41)
#define v_s_alfa_k_out DiscState(42)   /* tension aplicada en alfa-beta [V] */
#define v_s_beta_k_out DiscState(43)
#define i_s_alfa_k1_prev DiscState(44) /* i_alfa predicha para el paso SIGUIENTE */
#define i_s_beta_k1_prev DiscState(45) /* (se compara con la medicion en k+1)    */

/* ==========================================
 *  CONSTANTES NUMERICAS
 * ========================================== */
#define PI_VAL     3.14159265358979323846
#define SQRT3_INV  0.57735026918962576451
#define PSI_R_MIN  0.0005   /* piso del flujo estimado [Wb]: evita la
                             * singularidad de w_slip en el arranque */

/* --- Pre-magnetizacion (v31) ------------------------------------------
```

```
 * El par lo produce el flujo de ROTOR, no el de estator. Se declara la
 * maquina lista cuando la corriente de magnetizacion de rotor estimada
 * i_mr = psi_r/Lm alcanza MAGN_FRAC veces su valor nominal, deducido del
 * flujo de estator nominal (en vacio y en regimen psi_s = Ls*i_mr).
 * El tope de tiempo NO es un valor elegido: es la cota superior teorica del
 * propio retardo de primer orden, -tau_r*ln(1-MAGN_FRAC), que se evalua en
 * linea porque tau_r depende de Rr_MPC. */
#define MAGN_FRAC     0.95

/* ============================================
 *  PARAMETROS DEL BLOQUE (copias static, se leen UNA VEZ en Start)
 *  Los macros de contexto (ParamRealData, Input, ...) solo son validos
 *  dentro de las funciones de seccion; por eso se copian a static.
 * ============================================ */
static double Ts, Rs, Rr_MPC, Lm, Ls, Lr, p, psi_s_nom, lambda_ptc, kw, cw1, cw2,
    T_ref_max;
/* Flags de experto + clock (ver tabla de parametros en la cabecera). */
static double PINN_arranque, PINN_carga, PINN_inversion, CLK_ENABLE,
    CLK_WARMUP_STEPS;
static double i_s_max;   /* limite de pico de |i_s| [A] */
static double w_s_nom;   /* frecuencia sincronica nominal [rad/s el] */
static double k_vlim;    /* v35: margen sobre el circulo inscrito (0..1) */
static double w_sync_f;  /* v35: frecuencia sincronica estimada, filtrada */
static double g_dir_ref = 1.0;   /* canonizacion de giro: +1 directa, -1 inversa */
static double g_dir_prev = 1.0;  /* para detectar el cambio de sentido */
static double psi_s_ref_out;  /* referencia de flujo EFECTIVA [Wb],
                                 * publicada en la salida 80 */
static double isq_max_out;    /* limite de i_sq vigente [A],
                                 * publicado en la salida 81 */
static double w_slip_out;     /* deslizamiento estimado [rad/s el],
                                 * publicado en la salida 82 */
static double w_sync_f_out;   /* frecuencia sincronica filtrada [rad/s el],
                                 * publicada en la salida 83 */
static double Rr_MPC_base;   /* Rr base del MPC: INMUTABLE durante la corrida */

/* ============================================
 *  ESTADOS DISCRETOS 46..57 (CAPA PINN + CLOCK)
 *  k_rr_arr_state es EL estado de correccion: Rr_eff = Rr_base * k.
 *  Lo escriben (con exclusion mutua) el arbitro de arranque y el de
 *  carga; persiste cuando ningun experto esta activo.
 * ============================================ */
#define k_rr_arr_state       DiscState(46)  /* factor k = Rr_eff/Rr_base [-] */
#define k_hat_arr_target     DiscState(47)  /* diagnostico: rr2/Rr_base      */
#define PINN_arr_armed       DiscState(48)  /* FSM arranque: armado          */
#define PINN_arr_active      DiscState(49)  /* FSM arranque: en ventana      */
#define PINN_arr_done        DiscState(50)  /* FSM arranque: finalizado      */
#define shape_metric_cv_ds   DiscState(51)  /* d_hat parcial del ciclo (B/A) */
#define Rr_MPC_effect_ds     DiscState(52)  /* Rr efectiva del modelo [ohm]  */
#define clk_steps            DiscState(53)  /* clock: pasos contados         */
#define clk_mpc_sum_ticks    DiscState(54)  /* clock: ticks acumulados MPC   */
#define clk_total_sum_ticks  DiscState(55)  /* clock: ticks acumulados total */
#define t_CPU_MPC_us_ds      DiscState(56)  /* promedio CPU del MPC   [us]   */
#define t_CPU_TOTAL_us_ds    DiscState(57)  /* promedio CPU total     [us]   */

#ifdef PINN_USE_CLOCK
#include <time.h>
#endif

/* Utilidad de seguridad: recorta x al rango [lo, hi]. */
static double clamp_d(double x, double lo, double hi){ return x<lo?lo:(x>hi?hi:x);
    }

/* ============================================
 *  TABLA DE VECTORES DEL INVERSOR 2L-VSI
 *  Indice i = 0..7 -> estados de conmutacion (Sa,Sb,Sc) de cada pierna.
 *  i=0 y i=7 son los vectores nulos; 1..6 los activos.
 * ============================================ */
static const int Sa_table[8] = {0, 1, 1, 0, 0, 0, 1, 1};
```

```
211 static const int Sb_table[8] = {0, 0, 1, 1, 1, 0, 0, 1};
212 static const int Sc_table[8] = {0, 0, 0, 0, 1, 1, 1, 1};
213
214 /* ============================================
215  *  FUNCIONES AUXILIARES DE TRANSFORMACION
216  *  Clarke (abc -> alfa-beta), Park (alfa-beta -> dq con angulo g) y
217  *  sus inversas. mag2 = modulo; norm_angle = envuelve a (-pi, pi].
218  * ============================================ */
219 static double mag2(double a, double b){ return sqrt(a*a + b*b); }
220 static double norm_angle(double x){ x=fmod(x+PI_VAL,2.0*PI_VAL); if(x<0.0)x+=2.0*
        PI_VAL; return x-PI_VAL; }
221 static double abc_to_alfa(double va,double vb,double vc){ return (2.0/3.0)*(va-0.5*
        vb-0.5*vc); }
222 static double abc_to_beta(double va,double vb,double vc){ return (vb-vc)*SQRT3_INV;
         }
223 static double alfabeta_to_d(double a,double b,double g){ return  a*cos(g)+b*sin(g);
         }
224 static double alfabeta_to_q(double a,double b,double g){ return -a*sin(g)+b*cos(g);
         }
225 static double dq_to_alfa(double d,double q,double g){ return d*cos(g)-q*sin(g); }
226 static double dq_to_beta(double d,double q,double g){ return d*sin(g)+q*cos(g); }
227
228 /* ============================================
229  *  EXPERTO PINN DE ARRANQUE v3 (direccional: arr_subir + arr_bajar)
230  *  El header trae las dos redes, el coordinador (FSM + arbitro con via
231  *  de breakaway) ahora vive en pinn_arranque_coord_v2.h.
232  * ============================================ */
233
234 /* ============================================
235  *  EXPERTO PINN DE CARGA v3 (direccional: carga_subir + carga_bajar)
236  *  v24 = emergencia fisica con pacing anti-sobrepaso + re-latch
237  *  bidireccional. ELIMINADO en la version limpia (no se ejecutaba).
238  * ============================================ */
239 #include "pinn_v2_coordinator.h"
240 #include "pinn_arranque_coord_v2.h"
241 #include "pinn_carga_coord_v2.h"        /* v24 RECTIFICACION: emergencia con pacing
         + re-latch (incluye up/dn) */
242
243 /* ---- Normalizaciones y umbrales de la capa PINN (maquina nominal) ---- */
244 #define PINN_WNOM_EL    (2.0 * (1440.0 * 2.0 * PI_VAL / 60.0))  /* w nominal
        ELECTRICA [rad/s] (p=2) */
245 #define PINN_TNOM       (5500.0 / (1440.0 * 2.0 * PI_VAL / 60.0)) /* torque nominal
        [Nm] */
246 #define PINN_J          0.06      /* inercia [kg m^2] para el residuo de torque */
247 #define PINN_T_SKIP     0.10      /* [s] ignora el inicio (flujo sin asentar)   */
248 #define PINN_AMP_MIN    2.0       /* [A] amplitud minima para segmentar ciclos  */
249 #define PINN_MIN_CYCLE 25         /* muestras minimas para aceptar un ciclo     */
250 /* v4: TRIGGER DE SCHMITT. El detector ingenuo disparaba tambien en el
251  * cruce DESCENDENTE, donde el rizado de conmutacion hace oscilar la
252  * corriente alrededor de cero: medido, el 57 % de los ciclos duraba MEDIO
253  * periodo electrico y la DFT quedaba calculada sobre media onda. Con
254  * histeresis el detector solo se rearma tras bajar de -h, y h es una
255  * fraccion de la amplitud del ciclo ANTERIOR: se autoescala y no depende
256  * de w_sync ni de w_slip, que estan contaminadas por Rr. */
257 #define PINN_ZC_FRAC    0.10      /* histeresis, en fraccion de la amplitud    */
258 #define PINN_NBUF       2304      /* capacidad del buffer de un ciclo           */
259 #define PINN_NFEAT      21        /* features base por ciclo (incl. torque_resid en
        17) */
260 #define PINN_V2_PSI_NOM 0.987600   /* v2: psi_s_nom, para psi_s_ref/psi_s_nom */
261
262 /* ============================================
263  *  ARRANQUE DUAL: switch de aplicacion, coordinador y diagnosticos
264  *  Las cajas dbg_* solo alimentan la Output (scopes/dataset); no
265  *  intervienen en el control.
266  * ============================================ */
267 static double PINN_APPLY;                 /* 0 = solo diagnostico ; 1 = aplica la
        correccion */
```

```
static double Rr_MPC_effect_candidate;    /* Rr_MPC_effect que se APLICARIA (diagnostico) */
static double dbg_p_act1, dbg_p_act2, dbg_conf1, dbg_conf2, dbg_rr1, dbg_rr2;
static double dbg_delta1, dbg_delta2, dbg_decision, dbg_fsm, dbg_conf, dbg_delta_hat;
static double dbg_armed, dbg_active, dbg_done, dbg_safe;

/* ---- Limites de seguridad del factor de correccion k = Rr_eff/Rr_base ---- */
#define POL_KMIN       0.005    /* clamp inferior de k */
#define POL_KMAX       200.0    /* clamp superior de k */
#define PINN_WRITE_FEATURES 0   /* 1 = volcar features a CSV (paridad) */

/* ==========================================
 *  ACUMULADORES POR CICLO DE CORRIENTE (segmentacion de la PINN)
 *  Un "ciclo" = entre cruces por cero ascendentes de i_alfa. Durante el
 *  ciclo se acumulan sumas; al cerrarlo se calculan las features f[].
 *  ac_A/ac_B/ac_C: sumas del residuo (d_hat = B/A); ac_i2: suma i^2;
 *  ac_w/ac_wref/ac_tref/ac_test/ac_test2/ac_psir: medias de contexto;
 *  ac_idxsw: conmutaciones del vector optimo; pb_*: buffer del ciclo.
 * ========================================== */
static double pb_al[PINN_NBUF], pb_be[PINN_NBUF];
static int    pb_n;
static double ac_A, ac_B, ac_C, ac_i2, ac_w, ac_tref, ac_test, ac_test2, ac_psir, ac_idxsw, pidx_cyc;
static double ac_wref;   /* v3: suma de w_ref*p por muestra del ciclo (para wreg_err) */
/* ==== v2: acumuladores propios (flujo de estator, i_sq, limite e_slip) ==== */
static double ac_psis, ac_isq, ac_isqmax, ac_wslip, ac_wsyncf;
static double cyc_ipeak, cyc_w0, cyc_t0, cyc_wlast, cyc_tlast;
static int    pinn_in_cyc;
/* pv_* = muestra ANTERIOR (para residuo eps = i_med - i_pred y sensibilidad) */
static double pv_is_al, pv_is_be, pv_psir_al, pv_psir_be, pv_ipred_al, pv_ipred_be;
static int    pinn_have_prev;
/* v4: estado del trigger de Schmitt y amplitud de referencia */
static int    pinn_zc_hi; static double pinn_amp_prev;
static FILE  *pinn_fp; static int pinn_cyc_idx;
/* v3d: TRAZA DE LOS COORDINADORES. Es una simulacion: nada deberia
 * tener que inferirse. Una linea por ciclo cerrado. */
#define V3_TRACE 1
static FILE *v3_fp;
static double dbg_a_ssub, dbg_a_csub, dbg_a_dsub, dbg_a_ssob, dbg_a_csob, dbg_a_dsob;
static double dbg_k_ssub, dbg_k_csub, dbg_k_dsub, dbg_k_ssob, dbg_k_csob, dbg_k_dsob;

/* ==========================================
 *  EXPERTO DE CARGA: estado propio (NO comparte nada con el arranque)
 *  El estado vive en la struct static g_coord_carga: NO se agregaron
 *  DiscState (NumDiscStates sigue en 58). Diagnosticos en Output 65..79.
 * ========================================== */
static double PINN_CARGA_APPLY;            /* 0 = diagnostico (no toca Rr_MPC_effect) ; 1 = aplica */

/* ===================== SISTEMA DE EXPERTOS PINN v2 =====================
 *  Dos expertos DIRECCIONALES + tres veedores fisicos + guardian.
 *  Convive con v1 sin tocarlo: escribe en la misma k_rr_arr_state y solo
 *  si PINN_V2_APPLY > 0.5. Con los flags de v1 apagados no hay dos escritores.
 *  Vector de red de 141 componentes: el constructor vigente usa 27
 *  caracteristicas base x {actual, previo, delta} + 10 escalares
 *  clave x {2,5,10} x {media, desviacion}. PINN_V2_NB=29 se
 *  conserva como dimension historica del buffer local. */
#define PINN_V2_NB  29
static double PINN_V2_APPLY;
static ArrCoordV2 g_ac2;   /* coordinador de ARRANQUE v2 (port de v26) */
static CargaCoordV2 g_k2;  /* coordinador de CARGA v2 (port de v24) */
static double dbg_k2_cand, dbg_k2_s, dbg_k2_dec, dbg_k2_fsm, dbg_k2_route, dbg_k2_comm;
static double fv_prev[PINN_V2_NB]; static int have_fv_prev;
```

```
static double dbg_v2_s, dbg_v2_conf, dbg_v2_dom, dbg_v2_cand;
static double dbg_v2_dec, dbg_v2_rama, dbg_v2_fsm, dbg_v2_veredicto, dbg_v2_rho2;
/* cajas de diagnostico de carga (la Output las publica en 65..78) */
static double dbg_c_cand, dbg_c_dec, dbg_c_fsm, dbg_c_pact, dbg_c_conf, dbg_c_delta
    , dbg_c_rrtgt;
static double dbg_c_armed, dbg_c_active, dbg_c_done, dbg_c_safe, dbg_c_stepdet,
    dbg_c_nconf, dbg_c_ood;

/* ============================================
 *  MODO GENERACION DE DATOS (build separado, INERTE en deploy)
 *  Solo se compila con -DPINN_GENDATA: Rr_eff salta de Rr_MPC a
 *  Rr_MPC*GEN_KMULT en t=GEN_TJUMP para logear el transitorio.
 * ============================================ */
#ifdef PINN_GENDATA
static double GEN_TJUMP, GEN_KMULT;
#endif

/* pinn_comp: magnitud del armonico h de la trayectoria (al,be) del ciclo
 * via DFT directa sobre N muestras (h=-1 -> secuencia negativa). */
static double pinn_comp(double *al, double *be, int N, int h){
    double re=0.0, im=0.0, th; int n;
    for(n=0;n<N;n++){ th=2.0*PI_VAL*(double)h*(double)n/(double)N;
        re += al[n]*cos(th)+be[n]*sin(th); im += be[n]*cos(th)-al[n]*sin(th); }
    return (2.0/(double)N)*sqrt(re*re+im*im);
}
/* pinn_reset_cycle: pone a cero los acumuladores al abrir un ciclo nuevo. */
static void pinn_reset_cycle(void){
    pb_n=0; ac_A=ac_B=ac_C=ac_i2=0.0;
    ac_w=ac_tref=ac_test=ac_test2=ac_psir=ac_idxsw=0.0; cyc_ipeak=0.0; ac_wref=0.0;
    ac_psis=ac_isq=ac_isqmax=ac_wslip=ac_wsyncf=0.0;   /* v2 */
}
```

Listing B.1: Code Declarations del C-Script principal de la configuración v36_v4 evaluada.

B.2. C-Script principal: Start Function

```
/* ============================================
 *  SECCION 2: START FUNCTION
 *  Se ejecuta UNA VEZ al inicio de la simulacion:
 *   (1) copia los 20 parametros del bloque a variables static;
 *   (2) inicializa los 58 DiscState en valores seguros (vector nulo,
 *       flujo cero, PI descargado, correccion neutra k=1);
 *   (3) resetea coordinadores y buffers de ambos expertos PINN.
 *  El ORDEN de lectura de parametros ES el mapa de indices: no moverlo.
 * ============================================ */

/* ---- (1) Parametros del MPC y del PI de velocidad (indices 0..12) ---- */
Ts        = ParamRealData(0, 0);
Rs        = ParamRealData(1, 0);
Rr_MPC    = ParamRealData(2, 0);
Lm        = ParamRealData(3, 0);
Ls        = ParamRealData(4, 0);
Lr        = ParamRealData(5, 0);
p         = ParamRealData(6, 0);
psi_s_nom = ParamRealData(7, 0);
lambda_ptc= ParamRealData(8, 0);
kw        = ParamRealData(9, 0);
cw1       = ParamRealData(10, 0);
cw2       = ParamRealData(11, 0);
T_ref_max = ParamRealData(12, 0);

/* ---- Flags de experto + clock (indices 13..17).
 *      PINN_carga (idx 14) = ENABLE del experto de carga.
 *      PINN_arranque (13) y PINN_inversion (15) se leen pero NO se
 *      consultan en v26 (reservados; ver cabecera de Declarations). ---- */
```

```
PINN_arranque    = ParamRealData(13, 0);
PINN_carga       = ParamRealData(14, 0);
PINN_inversion   = ParamRealData(15, 0);
CLK_ENABLE       = ParamRealData(16, 0);
CLK_WARMUP_STEPS = ParamRealData(17, 0);

/* MODO GENERACION DE DATOS: 2 params extra (indices 18,19) SOLO en el build
 * con PINN_GENDATA. En deploy (este), 18=PINN_APPLY y 19=PINN_CARGA_APPLY. */
#ifdef PINN_GENDATA
GEN_TJUMP = ParamRealData(18, 0);   /* instante del salto [s] */
GEN_KMULT = ParamRealData(19, 0);   /* Rr_eff_destino / Rr_MPC  (=B/A) */
#endif

/* ---- (2) Gates iniciales: vector nulo v0 (piernas en 0/1 complementario) ---- */
g1_prev = 0.0;  g2_prev = 1.0;
g3_prev = 0.0;  g4_prev = 1.0;
g5_prev = 0.0;  g6_prev = 1.0;

/* ---- Estimador de flujo y controlador PI: todo desde cero ---- */
gamma_r_k = 0.0; psi_r_prev = 0.0; x_w_prev = 0.0; T_ref_prev = 0.0; i_sd_prev =
    0.0;

/* ---- Cajas de diagnostico dq / alfa-beta / dataset: a cero ---- */
psi_s_d_out = 0.0;  psi_s_q_out = 0.0;  psi_r_d_out = 0.0;  psi_r_q_out = 0.0;
    T_est_out = 0.0;
psi_s_alfa_out = 0.0; psi_s_beta_out = 0.0; psi_r_alfa_out = 0.0; psi_r_beta_out =
    0.0;
T_ref_out = 0.0; w_eli_out = 0.0; theta_r_prev = 0.0;
v_sd_k = 0.0; v_sq_k = 0.0; magnetized_flag = 0.0;
is_d_out = 0.0; is_q_out = 0.0; idx_opt_out = 0.0;
i_sd_k1_out = 0.0; i_sq_k1_out = 0.0; psi_sd_k1_out = 0.0; psi_sq_k1_out = 0.0;
i_sd_k2_opt_out = 0.0; i_sq_k2_opt_out = 0.0; psi_sd_k2_opt_out = 0.0;
    psi_sq_k2_opt_out = 0.0;
T_k2_opt_out = 0.0; psi_s_mag_k2_opt_out = 0.0;
g_opt_out = 0.0; v_sd_k1_opt_out = 0.0; v_sq_k1_opt_out = 0.0;
v_s_alfa_k_out = 0.0; v_s_beta_k_out = 0.0;
i_s_alfa_k1_prev = 0.0; i_s_beta_k1_prev = 0.0;

Rr_MPC_base = Rr_MPC;   /* base inmutable: la correccion solo reescala k */

/* ---- DiscState 46..57: capa PINN en estado neutro + clock a cero ---- */
k_rr_arr_state = 1.0;          /* arranca en la base: Rr_eff = Rr_MPC */
k_hat_arr_target = 1.0;
PINN_arr_armed = 0.0; PINN_arr_active = 0.0; PINN_arr_done = 0.0;
shape_metric_cv_ds = 0.0;
Rr_MPC_effect_ds = Rr_MPC_base;
clk_steps = 0.0; clk_mpc_sum_ticks = 0.0; clk_total_sum_ticks = 0.0;
t_CPU_MPC_us_ds = 0.0; t_CPU_TOTAL_us_ds = 0.0;

/* ---- (3) Segmentador de ciclos y memoria de features (static) ---- */
pinn_reset_cycle();
pinn_in_cyc = 0; pinn_have_prev = 0; pinn_zc_hi = 0; pinn_amp_prev = 0.0;
pv_is_al = pv_is_be = pv_psir_al = pv_psir_be = pv_ipred_al = pv_ipred_be = 0.0;
pidx_cyc = 0.0; cyc_w0 = cyc_t0 = cyc_wlast = cyc_tlast = 0.0;
pinn_cyc_idx = 0; pinn_fp = NULL;
#if V3_TRACE
v3_fp = fopen("traza_v3.csv", "w");
if (v3_fp) fprintf(v3_fp, "t,rr_eff,k_rr,a_fsm,a_steps,a_brk,a_manio,a_pend,
    a_sinmej,a_dec,a_rama,a_ver,a_s,a_conf,a_cand,a_ssub,a_csub,a_dsub,a_ssob,
    a_csob,a_dsob,k_fsm,k_motivo,k_ncom,k_bufn,k_rback,k_ood,k_lastdir,k_contn,
    k_espera,k_rei,k_sdet,k_comm,k_emgst,k_emgev,k_emgfl,k_epspre,k_epspost,k_dec,
    k_route,k_s,k_cand,k_ssub,k_csub,k_dsub,k_ssob,k_csob,k_dsob,eps_n,rho2,
    psi_refn,t_est,t_ref,rms_i,d_hat,w_mean,accel,k_vpre,k_via\n");
#endif
#if PINN_WRITE_FEATURES
pinn_fp = fopen("features_cscript.csv", "w");
if (pinn_fp) fprintf(pinn_fp,
  "cyc,rr_eff,rr_base,d_hat,rms_eps_norm,i_peak,I1,thd,h3,h5,h7,neg_seq,"
```

```
  "w_mean,accel,tref_mean,test_mean,test_ripple,psir_mean,torque_resid,idx_switch,
      cyc_freq,cyc_nsamp\n");
#endif

/* ---- Coordinador de ARRANQUE: reset de FSM, anillos y candidato ---- */
PINN_APPLY = ParamRealData(18, 0);          /* arranque: indice 18 */
Rr_MPC_effect_candidate = Rr_MPC_base;
dbg_p_act1 = dbg_p_act2 = dbg_conf1 = dbg_conf2 = dbg_rr1 = dbg_rr2 = 0.0;
dbg_delta1 = dbg_delta2 = dbg_decision = dbg_fsm = dbg_conf = dbg_delta_hat = 0.0;
dbg_armed = dbg_active = dbg_done = dbg_safe = 0.0;

/* ---- Coordinador de CARGA: reset de FSM, anillos y diagnosticos ---- */
PINN_CARGA_APPLY = ParamRealData(19, 0);    /* carga: indice 19 */
i_s_max = ParamRealData(20, 0);             /* limite de |i_s| [A] */
w_s_nom = ParamRealData(21, 0);             /* frecuencia sincronica nominal */
k_vlim  = ParamRealData(22, 0);             /* v35: margen de tension */
w_sync_f = 0.0;                             /* v35: arranca sin debilitar */
psi_s_ref_out = psi_s_nom;                  /* valor inicial de la salida 80 */
isq_max_out = 0.0;                          /* valor inicial de la salida 81 */
w_slip_out = 0.0;                           /* valor inicial de la salida 82 */
w_sync_f_out = 0.0;                         /* valor inicial de la salida 83 */
PINN_V2_APPLY = ParamRealData(23, 0);        /* v2: indice 23 */
ac2_reset(&g_ac2, Rr_MPC_base);
k2_reset(&g_k2, Rr_MPC_base);
dbg_k2_cand = Rr_MPC_base; dbg_k2_s = dbg_k2_dec = dbg_k2_fsm = 0.0;
dbg_k2_route = dbg_k2_comm = 0.0;
v2_mp_reset();
have_fv_prev = 0;
{ int q; for (q = 0; q < PINN_V2_NB; q++) fv_prev[q] = 0.0; }
dbg_v2_s = dbg_v2_conf = dbg_v2_dom = dbg_v2_cand = 0.0;
dbg_v2_dec = dbg_v2_rama = dbg_v2_fsm = dbg_v2_veredicto = dbg_v2_rho2 = 0.0;
dbg_c_cand = dbg_c_dec = dbg_c_fsm = dbg_c_pact = dbg_c_conf = dbg_c_delta =
    dbg_c_rrtgt = 0.0;
dbg_c_armed = dbg_c_active = dbg_c_done = dbg_c_safe = dbg_c_stepdet = dbg_c_nconf
    = dbg_c_ood = 0.0;
```

Listing B.2: Start Function del C-Script principal de la configuración v36_v4 evaluada.

B.3. C-Script principal: Output Function

```
/* ==========================================
 *  SECCION 3: OUTPUT FUNCTION
 *  SIN CALCULOS: solo publica DiscState y cajas static escritas por la
 *  Update del paso anterior (patron "salida = memoria", coherente con
 *  la compensacion de retardo: lo aplicado se decidio el ciclo previo).
 *  Number of outputs = 84. NO cambiar orden ni indices: el mux de
 *  medicion y las columnas de out.csv dependen de este mapa.
 *
 *  NOMENCLATURA ANGULAR:
 *   - Output(10) theta_r_est : angulo MECANICO ESTIMADO del rotor (NO el real).
 *   - Output(17) gamma_r_est : angulo de orientacion del flujo de rotor (Park).
 *   - El angulo REAL del rotor NO sale del C-Script (se mide del bloque IM).
 * ========================================== */

/* ---- 0..5: gates del inversor (pares complementarios por pierna) ---- */
Output(0)  = g1_prev;
Output(1)  = g2_prev;
Output(2)  = g3_prev;
Output(3)  = g4_prev;
Output(4)  = g5_prev;
Output(5)  = g6_prev;

/* ---- 6..10: diagnostico dq (flujos [Wb], angulo [rad]) ---- */
Output(6)  = psi_s_d_out;
Output(7)  = psi_s_q_out;
```

```
Output(8)  = psi_r_d_out;
Output(9)  = psi_r_q_out;
Output(10) = theta_r_prev;     /* theta_r_est (NO es el theta_r real) */

/* ---- 11..14: diagnostico alfa-beta (flujos [Wb]) ---- */
Output(11) = psi_s_alfa_out;
Output(12) = psi_s_beta_out;
Output(13) = psi_r_alfa_out;
Output(14) = psi_r_beta_out;

/* ---- 15..17: torque estimado/referencia [Nm] y orientacion [rad] ---- */
Output(15) = T_est_out;
Output(16) = T_ref_out;
Output(17) = gamma_r_k;        /* gamma_r_est: orientacion de campo */

/* ---- 18..20: dataset (corrientes dq [A] y vector optimo 0..7) ---- */
Output(18) = is_d_out;         /* corriente estator dq, eje d */
Output(19) = is_q_out;         /* corriente estator dq, eje q */
Output(20) = idx_opt_out;      /* indice 0..7 del vector optimo */

/* ---- 21..33: dataset predictivo del MPC (k+1 y k+2 del vector ganador) ---- */
Output(21) = i_sd_k1_out;
Output(22) = i_sq_k1_out;
Output(23) = psi_sd_k1_out;
Output(24) = psi_sq_k1_out;

Output(25) = i_sd_k2_opt_out;
Output(26) = i_sq_k2_opt_out;
Output(27) = psi_sd_k2_opt_out;
Output(28) = psi_sq_k2_opt_out;
Output(29) = T_k2_opt_out;
Output(30) = psi_s_mag_k2_opt_out;

Output(31) = g_opt_out;
Output(32) = v_sd_k1_opt_out;
Output(33) = v_sq_k1_opt_out;

/* ---- 34..35: tension APLICADA reconstruida en alfa-beta [V] ---- */
Output(34) = v_s_alfa_k_out;
Output(35) = v_s_beta_k_out;

/* ---- 36..37: corriente predicha el paso anterior [A]
 *      (base del residuo fisico eps = i_med - i_pred de la PINN) ---- */
Output(36) = i_s_alfa_k1_prev;
Output(37) = i_s_beta_k1_prev;

/* ---- 38..50: capa v02 (correccion de Rr + clock). 0..37 = BASE v13. ---- */
Output(38) = magnetized_flag;
Output(39) = Rr_MPC_base;          /* Rr base del MPC [ohm] (inmutable) */
Output(40) = Rr_MPC_effect_ds;     /* Rr EFECTIVA del modelo [ohm]      */
Output(41) = k_rr_arr_state;       /* factor k = Rr_eff/Rr_base [-]     */
Output(42) = 0.0;                  /* RESERVADO (antes Rr_ref_arranque, eliminado)
    */
Output(43) = k_hat_arr_target;
Output(44) = PINN_arr_armed;
Output(45) = PINN_arr_active;
Output(46) = PINN_arr_done;
Output(47) = 0.0;                  /* RESERVADO (antes PINN_arr_done_reason,
    eliminado) */
Output(48) = shape_metric_cv_ds;   /* shape_metric_cv (provisional) */
Output(49) = t_CPU_MPC_us_ds;      /* costo CPU medio del MPC   [us] */
Output(50) = t_CPU_TOTAL_us_ds;    /* costo CPU medio total     [us] */

/* ---- 51..64: diagnosticos del experto de ARRANQUE ---- */
Output(51) = Rr_MPC_effect_candidate;   /* Rr_MPC_effect que se aplicaria (arranque
    ) */
Output(52) = dbg_decision;              /* 0 none / 1 arr_1 / 2 arr_2 */
Output(53) = dbg_fsm;                   /* estado FSM arranque 0..8 */
```

```
Output(54) = dbg_p_act1;                /* prob. de actuar red arr_1 */
Output(55) = dbg_p_act2;                /* prob. de actuar red arr_2 */
Output(56) = dbg_conf1;                 /* confianza arr_1 */
Output(57) = dbg_conf2;                 /* confianza arr_2 */
Output(58) = dbg_delta1;                /* rr1 - Rr_MPC_effect [ohm] */
Output(59) = dbg_delta2;                /* rr2 - Rr_MPC_effect [ohm] */
Output(60) = dbg_armed;
Output(61) = dbg_active;
Output(62) = dbg_done;
Output(63) = dbg_safe;
Output(64) = dbg_delta_hat;             /* delta aplicado por el coordinador de
    arranque */

/* ---- 65..79: diagnosticos del experto de CARGA ---- */
Output(65) = dbg_c_cand;     /* Rr_MPC_effect_candidate_carga (Rr propuesto; vivo
    aun en diagnostico) */
Output(66) = dbg_c_dec;      /* decision carga: 0 none / 1 carga_active / 2
    out_of_domain / 3 safe */
Output(67) = dbg_c_fsm;      /* estado FSM carga (0..9) */
Output(68) = dbg_c_pact;     /* p_act_carga (max de up/dn) */
Output(69) = dbg_c_conf;     /* confidence_carga */
Output(70) = dbg_c_delta;    /* delta_Rr_hat_carga [ohm] */
Output(71) = dbg_c_rrtgt;    /* Rr_target_carga [ohm] */
Output(72) = dbg_c_armed;    /* armed_carga */
Output(73) = dbg_c_active;   /* active_carga */
Output(74) = dbg_c_done;     /* done_carga */
Output(75) = dbg_c_safe;     /* safe_carga */
Output(76) = dbg_c_stepdet;  /* deteccion de escalon de carga (0/1) */
Output(77) = dbg_c_nconf;    /* contador de confirmacion */
Output(78) = dbg_c_ood;      /* indicador out_of_domain (0/1) */
Output(79) = PINN_CARGA_APPLY;  /* eco del switch de aplicacion de carga */

/* ---- 80: referencia de flujo EFECTIVA (debilitada por encima de la
 *      velocidad sincronica nominal). APENDADA: 0..79 no cambian. ---- */
Output(80) = psi_s_ref_out;

/* ---- 81: limite de i_sq vigente, que sigue al flujo efectivo y por
 *      tanto cambia en zona de campo debilitado. ---- */
Output(81) = isq_max_out;

/* ---- 82: deslizamiento estimado [rad/s el]. w_sync = w_eli + w_slip. ----
 *      APENDADA: 0..81 no cambian de indice. */
Output(82) = w_slip_out;

/* ---- 83: frecuencia sincronica FILTRADA (tau = 10 ms): la que fija la
 *      referencia de flujo de v35/v36 y el techo de par de v36. ---- */
Output(83) = w_sync_f_out;

/* ---- 84..90: diagnosticos del sistema de expertos v2. APENDADOS. ---- */
Output(84) = dbg_v2_cand;      /* Rr_MPC_effect propuesto por v2 [ohm]   */
Output(85) = dbg_v2_s;         /* s = log10(Rr_est / rr_eff)             */
Output(86) = dbg_v2_conf;      /* confiabilidad                          */
Output(87) = dbg_v2_dom;       /* pertenencia a la maniobra              */
Output(88) = dbg_v2_dec;       /* 0 none 1 aplica 2 ignora 3 revierte    */
Output(89) = dbg_v2_rama;      /* +1 subestimado / -1 sobreestimado      */
Output(90) = dbg_v2_veredicto; /* veedor 1: 0 no tocar 1 medible 2 ignora */
Output(91) = dbg_v2_rho2;      /* rho2 = B^2/(A C)                       */

/* ---- 92..97: diagnosticos del experto de CARGA v2. ---- */
Output(92) = dbg_k2_cand;   /* Rr propuesto por carga [ohm]              */
Output(93) = dbg_k2_s;      /* s = log10(Rr_est / rr_eff)                */
Output(94) = dbg_k2_dec;    /* 0 none 1 activa 2 OOD 3 safe 4 ignora     */
Output(95) = dbg_k2_fsm;    /* estado de la FSM de carga (0..9)          */
Output(96) = dbg_k2_route;  /* +1 subir / -1 bajar                       */
Output(97) = dbg_k2_comm;   /* committed: el arranque cede la pluma      */

/* ---- 98: MOTIVO de la decision de carga (K2_MOT_*). Sirve para saber
 *      POR QUE no actuo, en vez de suponerlo. ---- */
```

```
Output(98) = (double)g_k2.motivo;

/* ---- 99..100: v3, diagnostico del REINICIO DE ANALISIS. Sirve para ver
 *      si el reinicio llega a dispararse y en que orden ocurren las cosas,
 *      en vez de deducirlo. ---- */
Output(99)  = (double)g_k2.reinicios;
Output(100) = (double)g_k2.step_detected;
```

Listing B.3: Output Function del C-Script principal: 101 salidas de control y diagnóstico.

B.4. C-Script principal: Update Function

```
/* build-tag v26.1-arranque-breakaway-signfix | base v25 planta fisica | carga_v24
    v24.0-rectificacion-pacing */
/* ==========================================
 *  SECCION 4: UPDATE FUNCTION   [v30: limite de i_sq]
 *  Se ejecuta una vez por periodo Ts, DESPUES de la Output. Contiene:
 *   A) correccion de Rr y constantes derivadas del modelo;
 *   B) MPC completo (PASO 1..13): medicion -> estimacion -> PI ->
 *      prediccion k+1 -> optimizacion k+2 -> gates -> diagnosticos;
 *   C) capa PINN: predictor alfa-beta, residuo, segmentacion por ciclo,
 *      features, inferencia y arbitros de ARRANQUE y CARGA;
 *   D) clock de costo computacional (opcional).
 *  El ORDEN de los pasos es parte del algoritmo validado (compensacion
 *  de retardo y alineacion temporal del residuo): NO reordenar.
 * ========================================== */
{
    /* ---- Variables locales del paso (mediciones, predicciones, costo) ---- */
    double ia, ib, ic;
    double w_eli, w_ref, Vdc;
    double is_alfa, is_beta;
    double is_d, is_q;
    double gamma_r_k1;
    double psi_r;
    double w_slip, w_sync;
    double psi_s_d, psi_s_q;
    double T_est;
    double e_w, x_w, T_ref_local;
    double psi_r_k1;
    double i_sd_k1, i_sq_k1;
    double psi_sd_k1, psi_sq_k1;
    int i;
    int idx_opt;
    double g_opt;
    double v_an_k1_i, v_bn_k1_i, v_cn_k1_i;
    double v_alfa_k1_i, v_beta_k1_i;
    double v_sd_k1_i, v_sq_k1_i;
    double psi_sd_k2, psi_sq_k2;
    double i_sd_k2, i_sq_k2;
    double T_k2;
    double psi_s_mag_k2;
    double g_cost;
    double psi_r_alfa, psi_r_beta;
    double psi_s_alfa, psi_s_beta;
    double theta_r;
    double v_sd_k1_opt, v_sq_k1_opt;
    double i_sd_k2_opt, i_sq_k2_opt;
    double psi_sd_k2_opt, psi_sq_k2_opt;
    double T_k2_opt, psi_s_mag_k2_opt;
    double tau_r, kr, sigma, R_sigma, L_sigma, tau_sigma;
    double isq_max, k_isq_pen;   /* v30: limite de corriente de par */
    double i_s_mag_k2, i_s_lim, k_is_pen;   /* v33: limite de |i_s| */
    double psi_s_ref, w_ref_el;            /* v34: campo debilitado */
    double T_ref_max_eff;                  /* v36: techo de par vigente */
    double v_lim, psi_lim, psi_piso;       /* v35: limite de flujo por tension */
```

```
    double r_psi;                          /* v36: psi_s_ref / psi_s_nom       */
    double i_mr_ref, i_mr_est, t_magn_max; /* criterio de pre-magnetizacion    */
    int    lim_i_activo;                   /* solo al magnetizar   */
    double k_rr_total;
#ifdef PINN_USE_CLOCK
    clock_t clk_us=0, clk_ps=0, clk_pe=0, clk_ue=0;
    if (CLK_ENABLE > 0.5) clk_us = clock();
#endif

#ifdef PINN_GENDATA
    /* GENERACION DE DATOS: salto programado de Rr_eff (politica PINN apagada).
     * k=1 antes de GEN_TJUMP (Rr_eff=Rr_MPC), k=GEN_KMULT despues (Rr_eff=B). */
    k_rr_arr_state = (CurrentTime >= GEN_TJUMP) ? GEN_KMULT : 1.0;
#endif

    /* ============================================
     *  A) CORRECCION DE Rr Y CONSTANTES DERIVADAS
     *  k_rr_arr_state es lo UNICO que escriben los expertos PINN.
     *  Aqui se satura a [POL_KMIN, POL_KMAX] y se reconstruyen las
     *  constantes del modelo con la Rr efectiva: la correccion se
     *  propaga ASI a estimador, orientacion y predictor a la vez.
     * ============================================ */
    k_rr_total = clamp_d(k_rr_arr_state, POL_KMIN, POL_KMAX);
    Rr_MPC_effect_ds = Rr_MPC_base * k_rr_total;
    tau_r     = Lr / Rr_MPC_effect_ds;           /* cte. de tiempo rotor [s] */
    kr        = Lm / Lr;                         /* factor de acoplamiento   */
    sigma     = 1 - (Lm * Lm) / (Ls * Lr);       /* coef. de dispersion      */
    R_sigma   = Rs + kr * kr * Rr_MPC_effect_ds; /* resistencia equivalente  */
    L_sigma   = sigma * Ls;                      /* inductancia transitoria  */
    tau_sigma = L_sigma / R_sigma;               /* cte. transitoria [s]     */


    /* ============================================
     *  B) MPC / PTC  (PASO 1..13)
     * ============================================ */

    /* PASO 1: mediciones (corrientes [A], velocidades [rad/s], Vdc [V]).
     * w_eli = velocidad ELECTRICA = w_mech * p. */
    ia = is_a_in; ib = is_b_in; ic = is_c_in;
    w_eli = w_mech_in * p; w_ref = w_ref_in; Vdc = Vdc_in;

    /* ---- v30: LIMITE DE LA CORRIENTE DE PAR ---------------------------
     * Regular |psi_s| constante impone un maximo de par: como
     * psi_sq = L_sigma*i_sq, subir i_sq hunde psi_r y el par cae. El
     * optimo esta en i_sq = psi_s_ref/(sqrt(2)*L_sigma). El horizonte de
     * 2*Ts no lo ve (tau_r es ~1000 veces mayor), asi que el optimizador
     * se pasa. Se penaliza el exceso sobre ese punto.
     *
     * Ganancia DERIVADA, sin constantes libres: pasarse en un paso
     * discreto completo debe costar el rango entero de la referencia:
     *   delta_isq_max = (2/3)*Vdc*Ts/L_sigma   (lo que un vector mueve i_sq)
     *   k_isq_pen * delta_isq_max = T_ref_max
     * Todo son parametros del bloque: vale para cualquier maquina. */
    /* v34: REFERENCIA DE FLUJO CON DEBILITAMIENTO DE CAMPO (Mora ec. 2.15).
     * Por encima de la velocidad sincronica nominal la tension disponible no
     * alcanza para sostener el flujo nominal, de modo que la referencia se
     * reduce en proporcion inversa a la velocidad pedida. Se emplea la
     * REFERENCIA de velocidad y no la medida, para que el flujo no dependa del
     * transitorio mecanico ni realimente al lazo.
     * psi_s_ref es LOCAL de esta funcion: al calcularse aqui, antes de isq_max,
     * todo lo que viene despues --limite de i_sq, referencia de magnetizacion
     * i_mr_ref = psi_s_ref/Ls y termino de flujo de la funcion de costo-- se
     * adapta solo al flujo debilitado sin cambiar ni una linea. */
    w_ref_el  = p * fabs(w_ref_in);
    /* v35: el limite lo impone la TENSION, y la tension que la maquina exige es
     * ~ w_sync*|psi_s|. v34 comparaba contra la REFERENCIA de velocidad, que
     * ignora el deslizamiento; como el deslizamiento crece con Rr, una jaula
     * resistiva sobremodulaba sin que el criterio lo detectara. Se usa aqui la
```

```
121      * frecuencia sincronica estimada del periodo anterior (filtrada mas abajo)
122      * y el enlace de continua MEDIDO, de modo que el limite se adapta solo.
123      * Es la misma ley de v34: psi_s_nom*w_s_nom = Vdc/sqrt(3) es el numerador. */
124     /* 1) tension que el inversor puede sintetizar sin salir del circulo
125      *    inscrito del hexagono, con el margen k_vlim */
126     v_lim = k_vlim * Vdc * 0.57735026919;      /* = k_vlim * Vdc/sqrt(3) */
127
128     /* 2) flujo maximo que esa tension sostiene a la frecuencia sincronica
129      *    vigente. Al arrancar w_sync_f vale ~0: se usa el nominal. */
130     if (w_sync_f > 1.0)
131         psi_lim = v_lim / w_sync_f;
132     else
133         psi_lim = psi_s_nom;
134
135     /* 3) la referencia es la menor de las dos: nunca por encima del nominal */
136     if (psi_lim < psi_s_nom)
137         psi_s_ref = psi_lim;
138     else
139         psi_s_ref = psi_s_nom;
140
141     /* 4) piso: por debajo de el la maquina queda desmagnetizada y el
142      *    estimador de deslizamiento, que divide por psi_r, se dispara */
143     psi_piso = 0.30 * psi_s_nom;
144     if (psi_s_ref < psi_piso)
145         psi_s_ref = psi_piso;
146     psi_s_ref_out = psi_s_ref;   /* se publica en la salida 80 */
147
148     /* v36: TECHO DE PAR COHERENTE CON EL FLUJO VIGENTE.
149      * El par maximo a flujo regulado va con psi^2, de modo que en campo
150      * debilitado la maquina ya no puede entregar 2*T_nom: pedirselo satura el
151      * PI de forma permanente e inyecta en la funcion de costo un error de par
152      * enorme y constante, que ademas resta peso relativo al termino de flujo.
153      * Escalar por (psi_s_ref/psi_s_nom)^2 equivale a (w_nom/w)^2 cuando la ley
154      * de flujo es psi = psi_nom*w_nom/w, pero al derivarlo del flujo vigente
155      * sigue solo al margen k_vlim, al piso y al min(), y por debajo de la
156      * nominal el cociente vale 1 (sin efecto, sin disparador).
157      * T_ref_max es un static de la Start: NO se escribe aqui, o se compondria
158      * cada periodo hasta anularse. */
159     r_psi         = psi_s_ref / psi_s_nom;
160     T_ref_max_eff = T_ref_max * r_psi * r_psi;
161
162     isq_max   = psi_s_ref / (1.41421356237 * L_sigma);
163     k_isq_pen = 1.5 * T_ref_max_eff * L_sigma / (Vdc * Ts);
164     isq_max_out = isq_max;   /* se publica en la salida 81 */
165
166     /* v33: LIMITE DE LA CORRIENTE DE ESTATOR, SOLO AL MAGNETIZAR.
167      * Mientras el rotor esta desmagnetizado, psi_r tarda tau_r en establecerse
168      * y el lazo empuja i_sd sin freno: sin este tope se alcanzan picos de unos
169      * 69 A. Se admite el doble del pico nominal de placa porque en esa fase no
170      * se produce par y conviene establecer el flujo cuanto antes.
171      * Una vez magnetizada, el limite se RETIRA: los dos terminos de la funcion
172      * de costo --seguimiento de par y regulacion de |psi_s|-- ya fijan i_sq e
173      * i_sd, y la corriente se mantiene por si sola en valores razonables.
174      * Mantenerlo tambien en operacion (v32) hundia el par disponible de 72,4 a
175      * 37,5 N.m y dejaba T*_max = 2*T_nom inalcanzable.
176      * La ganancia usa el mismo criterio que k_isq_pen --pasarse del limite en
177      * un paso discreto completo cuesta el rango entero de la referencia de
178      * par-- y coincide con ella, porque el paso de corriente que produce un
179      * vector, (2/3)*Vdc*Ts/L_sigma, es el mismo para cualquier componente. */
180     lim_i_activo = (magnetized_flag < 0.5);
181     i_s_lim      = 2.0 * i_s_max;
182     k_is_pen     = 1.5 * T_ref_max_eff * L_sigma / (Vdc * Ts);
183
184     /* PASO 2: Clarke abc -> alfa-beta */
185     is_alfa = abc_to_alfa(ia, ib, ic);
186     is_beta = abc_to_beta(ia, ib, ic);
187
188     /* PASO 3: Park alfa-beta -> dq con el angulo de campo gamma_r_k */
```

```
189     is_d = alfabeta_to_d(is_alfa, is_beta, gamma_r_k);
190     is_q = alfabeta_to_q(is_alfa, is_beta, gamma_r_k);
191
192     /* PASO 4: estimacion del flujo de rotor (modelo de corriente,
193      * Forward Euler con i_sd del paso ANTERIOR). El piso PSI_R_MIN
194      * evita dividir por ~0 en w_slip durante el arranque. */
195     psi_r = (Lm / tau_r) * Ts * i_sd_prev + (1.0 - Ts / tau_r) * psi_r_prev;
196     if (psi_r < PSI_R_MIN) psi_r = PSI_R_MIN;
197
198     /* PASO 5: deslizamiento, velocidad sincronica y avance del angulo
199      * de campo (condicion de orientacion; integracion rectangular). */
200     w_slip = (Lm * is_q) / (tau_r * psi_r);
201     w_sync = w_eli + w_slip;
202     /* v35: filtro de primer orden (tau = 10 ms) sobre |w_sync|. Sin el, la
203      * referencia de flujo seguiria el rizado de i_sq y pelearia contra su
204      * propio lazo; 10 ms es lento frente al periodo de muestreo y rapido
205      * frente a la dinamica mecanica. */
206     w_sync_f += (Ts / 0.010) * (fabs(w_sync) - w_sync_f);
207     /* v36: copia a cajas static para poder publicarlas en la Output.
208      * Ninguna de las dos es reconstruible exactamente fuera del bloque:
209      * w_slip divide por psi_r YA saturado por PSI_R_MIN, y w_sync_f es el
210      * estado de un filtro. No se altera ningun calculo. */
211     w_slip_out = w_slip;
212     w_sync_f_out = w_sync_f;
213     gamma_r_k1 = norm_angle(gamma_r_k + Ts * w_sync);
214
215     /* PASO 6: posicion MECANICA estimada del rotor (solo diagnostico) */
216     theta_r = norm_angle(theta_r_prev + Ts * w_eli / p);
217
218     /* PASO 7: flujo de estator dq (psi_s = kr*psi_r + L_sigma*i_s) y
219      * torque estimado T = 1.5*p*(psi x i). */
220     psi_s_d = kr * psi_r + L_sigma * is_d;
221     psi_s_q = L_sigma * is_q;
222     T_est = 1.5 * p * (psi_s_d * is_q - psi_s_q * is_d);
223
224     /* PASO 7b: pre-magnetizacion (CORREGIDA en v31). Hasta que el ROTOR
225      * este magnetizado se mantiene T*=0 (el MPC solo magnetiza). El flag
226      * queda en 1 PERMANENTE la primera vez que se cumple la condicion.
227      *
228      * v30 comparaba |psi_s| contra su referencia. Es incorrecto: el par
229      * vale T = 1.5*p*kr*psi_r*i_sq --el termino de dispersion de
230      * psi_s = kr*psi_r + L_sigma*i_sd se cancela y no produce par-- y la
231      * corriente de irrupcion llena ese termino en pocos ciclos. Medido:
232      * el flag se activaba a los 2,5 ms con psi_r al 6,2 % de su valor
233      * final. Ahora se espera a la corriente de magnetizacion de rotor
234      *   i_mr = psi_r/Lm        (tau_r*di_mr/dt + i_mr = i_sd)
235      * cuya referencia sale del flujo de estator nominal, porque en vacio
236      * y en regimen psi_s = Ls*i_mr:
237      *   i_mr_ref = psi_s_ref/Ls
238      * El tope temporal es la COTA SUPERIOR del propio transitorio: si i_sd
239      * se mantuviera exactamente en su valor final i_mr_ref --el caso mas
240      * lento posible, sin sobrecorriente de magnetizacion-- el retardo de
241      * primer orden alcanzaria MAGN_FRAC en -tau_r*ln(1-MAGN_FRAC). Con
242      * cualquier i_sd mayor tarda menos, de modo que rebasar esa cota
243      * significa que el lazo no logra sostener i_sd y conviene continuar
244      * antes que bloquear el arranque. Se evalua en linea porque tau_r
245      * depende de Rr_MPC. */
246     if (magnetized_flag < 0.5) {
247         i_mr_ref   = psi_s_ref / Ls;          /* corriente de magnetizacion pedida
                         */
248         i_mr_est   = psi_r / Lm;              /* la que el estimador ve
                         */
249         t_magn_max = -tau_r * log(1.0 - MAGN_FRAC);   /* tope de seguridad [s]
                         */
250
251         /* se da por magnetizada por cualquiera de las dos vias */
252         if (i_mr_est >= MAGN_FRAC * i_mr_ref)
253             magnetized_flag = 1.0;
```

```
        else if (CurrentTime >= t_magn_max)
            magnetized_flag = 1.0;
    }

    /* PASO 8: PI de velocidad con anti-windup. Congelado (T*=0, estados
     * en 0) durante la pre-magnetizacion para no acumular error. La
     * referencia se satura a +/- T_ref_max_eff (v36: el techo vigente,
     * reducido en campo debilitado). */
    if (magnetized_flag < 0.5) {
        T_ref_local = 0.0; e_w = 0.0; x_w = 0.0;
    } else {
        e_w = w_ref * p - w_eli;
        x_w = cw1 * x_w_prev + cw2 * T_ref_prev;
        T_ref_local = kw * (e_w - x_w);
        if (T_ref_local >  T_ref_max_eff) T_ref_local =  T_ref_max_eff;
        else if (T_ref_local < -T_ref_max_eff) T_ref_local = -T_ref_max_eff;
    }

    /* PASO 9: prediccion a k+1 (compensacion de retardo) con la tension
     * dq APLICADA este periodo (v_sd_k, v_sq_k, elegida el ciclo previo).
     * Forward Euler de las ecuaciones dq del modelo. */
    psi_r_k1 = (Lm / tau_r) * Ts * is_d + (1.0 - Ts / tau_r) * psi_r;
    if (psi_r_k1 < PSI_R_MIN) psi_r_k1 = PSI_R_MIN;

    i_sd_k1 = (1.0 - Ts / tau_sigma) * is_d
            + (Ts / L_sigma) * (v_sd_k + (kr / tau_r) * psi_r)
            + Ts * w_sync * is_q;
    i_sq_k1 = (1.0 - Ts / tau_sigma) * is_q
            + (Ts / L_sigma) * (v_sq_k - kr * w_eli * psi_r)
            - Ts * w_sync * is_d;
    psi_sd_k1 = psi_s_d + Ts * (v_sd_k - Rs * is_d + w_sync * psi_s_q);
    psi_sq_k1 = psi_s_q + Ts * (v_sq_k - Rs * is_q - w_sync * psi_s_d);

    /* PASO 9b: predictor AUXILIAR de corriente en alfa-beta (no toca la
     * funcion de costo). Reconstruye la tension aplicada en alfa-beta y
     * predice i(k+1|k); la prediccion queda en DiscState y en el paso
     * siguiente se compara con la medicion -> residuo fisico de la PINN.
     * Este predictor es AFIN en Rr_eff: base de d_hat. */
    v_s_alfa_k_out = dq_to_alfa(v_sd_k, v_sq_k, gamma_r_k);
    v_s_beta_k_out = dq_to_beta(v_sd_k, v_sq_k, gamma_r_k);
    {
        double i_s_alfa_k1, i_s_beta_k1, psi_r_alfa_k, psi_r_beta_k;
        psi_r_alfa_k = dq_to_alfa(psi_r, 0, gamma_r_k);   /* psi_rq = 0 (
            orientacion) */
        psi_r_beta_k = dq_to_beta(psi_r, 0, gamma_r_k);
        i_s_alfa_k1 = (1.0 - Ts / tau_sigma) * is_alfa
                    + (Ts / L_sigma) * (v_s_alfa_k_out
                      + kr * ((psi_r_alfa_k / tau_r) + w_eli * psi_r_beta_k));
        i_s_beta_k1 = (1.0 - Ts / tau_sigma) * is_beta
                    + (Ts / L_sigma) * (v_s_beta_k_out
                      + kr * ((psi_r_beta_k / tau_r) - w_eli * psi_r_alfa_k));
        i_s_alfa_k1_prev = i_s_alfa_k1;
        i_s_beta_k1_prev = i_s_beta_k1;
    }

    /* PASO 10: FCS-MPC sobre k+2. Para cada uno de los 8 vectores:
     * tension de fase -> alfa-beta -> dq (con el angulo PREDICHO
     * gamma_r_k1), prediccion de corrientes/flujos/torque a k+2 y
     * funcion de costo g = |T*-T| + lambda*|psi*-|psi||. Gana el menor. */
    g_opt = 1e10; idx_opt = 0; v_sd_k1_opt = 0.0; v_sq_k1_opt = 0.0;
    i_sd_k2_opt = i_sq_k2_opt = psi_sd_k2_opt = psi_sq_k2_opt = T_k2_opt =
        psi_s_mag_k2_opt = 0.0;
    for (i = 0; i < 8; i++) {
        v_an_k1_i = Vdc * ( 2.0 * Sa_table[i] - Sb_table[i] - Sc_table[i]) / 3.0;
        v_bn_k1_i = Vdc * (-Sa_table[i] + 2.0 * Sb_table[i] - Sc_table[i]) / 3.0;
        v_cn_k1_i = Vdc * (-Sa_table[i] - Sb_table[i] + 2.0 * Sc_table[i]) / 3.0;
        v_alfa_k1_i = abc_to_alfa(v_an_k1_i, v_bn_k1_i, v_cn_k1_i);
        v_beta_k1_i = abc_to_beta(v_an_k1_i, v_bn_k1_i, v_cn_k1_i);
```

```
320         v_sd_k1_i = alfabeta_to_d(v_alfa_k1_i, v_beta_k1_i, gamma_r_k1);
321         v_sq_k1_i = alfabeta_to_q(v_alfa_k1_i, v_beta_k1_i, gamma_r_k1);
322         i_sd_k2 = (1.0 - Ts / tau_sigma) * i_sd_k1
323                 + (Ts / L_sigma) * (v_sd_k1_i + (kr / tau_r) * psi_r_k1)
324                 + Ts * w_sync * i_sq_k1;
325         i_sq_k2 = (1.0 - Ts / tau_sigma) * i_sq_k1
326                 + (Ts / L_sigma) * (v_sq_k1_i - kr * w_eli * psi_r_k1)
327                 - Ts * w_sync * i_sd_k1;
328         psi_sd_k2 = psi_sd_k1 + Ts * (v_sd_k1_i - Rs * i_sd_k1 + w_sync * psi_sq_k1
                );
329         psi_sq_k2 = psi_sq_k1 + Ts * (v_sq_k1_i - Rs * i_sq_k1 - w_sync * psi_sd_k1
                );
330         T_k2 = 1.5 * p * (psi_sd_k2 * i_sq_k2 - psi_sq_k2 * i_sd_k2);
331         psi_s_mag_k2 = mag2(psi_sd_k2, psi_sq_k2);
332         g_cost = fabs(T_ref_local - T_k2) + lambda_ptc * fabs(psi_s_ref -
                psi_s_mag_k2);
333         /* v30: exceso de i_sq sobre el optimo. Penalizacion BLANDA (no
334          * descarte) para que siempre exista un vector ganador. */
335         if (fabs(i_sq_k2) > isq_max)
336             g_cost += k_isq_pen * (fabs(i_sq_k2) - isq_max);
337         /* v33: exceso de la MAGNITUD de corriente, solo mientras magnetiza.
338          * Tambien blanda: si ningun vector cumpliera, gana el que menos lo
339          * viola y el algoritmo nunca se queda sin actuacion. */
340         if (lim_i_activo) {
341             i_s_mag_k2 = mag2(i_sd_k2, i_sq_k2);
342             if (i_s_mag_k2 > i_s_lim)
343                 g_cost += k_is_pen * (i_s_mag_k2 - i_s_lim);
344         }
345         if (g_cost < g_opt) {
346             g_opt = g_cost; idx_opt = i; v_sd_k1_opt = v_sd_k1_i; v_sq_k1_opt =
                    v_sq_k1_i;
347             i_sd_k2_opt = i_sd_k2; i_sq_k2_opt = i_sq_k2;
348             psi_sd_k2_opt = psi_sd_k2; psi_sq_k2_opt = psi_sq_k2;
349             T_k2_opt = T_k2; psi_s_mag_k2_opt = psi_s_mag_k2;
350         }
351     }
352
353     /* PASO 11: gates del vector ganador (cada pierna con su complemento).
354      * La Output los publicara en el paso siguiente. */
355     g1_prev = (double)Sa_table[idx_opt]; g2_prev = 1.0 - (double)Sa_table[idx_opt];
356     g3_prev = (double)Sb_table[idx_opt]; g4_prev = 1.0 - (double)Sb_table[idx_opt];
357     g5_prev = (double)Sc_table[idx_opt]; g6_prev = 1.0 - (double)Sc_table[idx_opt];
358
359     /* PASO 12: memoria del paso: angulo, flujo, PI, i_sd y la tension dq
360      * del vector ganador (sera "lo aplicado" del proximo ciclo). */
361     gamma_r_k = gamma_r_k1; psi_r_prev = psi_r; x_w_prev = x_w;
362     T_ref_prev = T_ref_local; i_sd_prev = is_d; theta_r_prev = theta_r;
363     v_sd_k = v_sd_k1_opt; v_sq_k = v_sq_k1_opt;
364
365     /* PASO 13: volcado de diagnosticos a las cajas que publica Output */
366     psi_s_d_out = psi_s_d; psi_s_q_out = psi_s_q; psi_r_d_out = psi_r; psi_r_q_out
            = 0.0; T_est_out = T_est;
367     psi_r_alfa = dq_to_alfa(psi_r_d_out, psi_r_q_out, gamma_r_k);
368     psi_r_beta = dq_to_beta(psi_r_d_out, psi_r_q_out, gamma_r_k);
369     psi_s_alfa = dq_to_alfa(psi_s_d_out, psi_s_q_out, gamma_r_k);
370     psi_s_beta = dq_to_beta(psi_s_d_out, psi_s_q_out, gamma_r_k);
371     psi_s_alfa_out = psi_s_alfa; psi_s_beta_out = psi_s_beta;
372     psi_r_alfa_out = psi_r_alfa; psi_r_beta_out = psi_r_beta;
373     T_ref_out = T_ref_local; w_eli_out = w_eli;
374     is_d_out = is_d; is_q_out = is_q; idx_opt_out = (double)idx_opt;
375     i_sd_k1_out = i_sd_k1; i_sq_k1_out = i_sq_k1; psi_sd_k1_out = psi_sd_k1;
            psi_sq_k1_out = psi_sq_k1;
376     i_sd_k2_opt_out = i_sd_k2_opt; i_sq_k2_opt_out = i_sq_k2_opt;
377     psi_sd_k2_opt_out = psi_sd_k2_opt; psi_sq_k2_opt_out = psi_sq_k2_opt;
378     T_k2_opt_out = T_k2_opt; psi_s_mag_k2_opt_out = psi_s_mag_k2_opt;
379     g_opt_out = g_opt; v_sd_k1_opt_out = v_sd_k1_opt; v_sq_k1_opt_out = v_sq_k1_opt
            ;
380
```

```
#ifdef PINN_USE_CLOCK
    if (CLK_ENABLE > 0.5) clk_ps = clock();   /* marca: fin del MPC, inicio PINN */
#endif

    /* ==========================================
     *  C) CAPA PINN (arranque + carga)
     *  Gate de actividad: maquina magnetizada, t > PINN_T_SKIP y
     *  amplitud de corriente suficiente. Si no, se descarta el ciclo.
     *  Estructura: (c1) residuo y sensibilidad por muestra ->
     *  (c2) cierre de ciclo: features f[0..20] -> inferencia arr_1/arr_2
     *  -> arbitro de arranque (breakaway v26) -> features de carga
     *  fc[0..23] -> inferencia carga_up/dn -> arbitro de carga (v24) ->
     *  (c3) acumulacion de la muestra en el ciclo abierto.
     * ========================================== */
    {
        double zmag   = mag2(is_alfa, is_beta);
        int    active = (magnetized_flag > 0.5) && (CurrentTime > PINN_T_SKIP) && (
            zmag > PINN_AMP_MIN);
        double Rr_eff_now = Rr_MPC_effect_ds;

        if (!active) {
            pinn_in_cyc = 0; pinn_zc_hi = 0; pinn_reset_cycle();
        } else {
            /* (c1) Residuo eps = i_med - i_pred(k|k-1) y sensibilidad phi
             * (derivada analitica del predictor respecto de Rr_eff),
             * AMBOS con las senales del instante de la prediccion (k-1):
             * salvo el angulo de psi_r en phi, que es gamma[k+1]: ver cap. 5 */
            double eps_al = 0.0, eps_be = 0.0, pa = 0.0, pb = 0.0, cc;
            int crossing;
            if (pinn_have_prev) {
                eps_al = is_alfa - pv_ipred_al; eps_be = is_beta - pv_ipred_be;
                cc = Ts / L_sigma;
                pa = cc * (-kr * kr * pv_is_al + (kr / Lr) * pv_psir_al);
                pb = cc * (-kr * kr * pv_is_be + (kr / Lr) * pv_psir_be);
            }
            /* Frontera de ciclo: cruce ascendente de i_alfa CON HISTERESIS.
             * Sin ella el rizado alrededor del cruce descendente abria un
             * ciclo espurio y la ventana quedaba en medio periodo (v4). */
            {   double zh = PINN_ZC_FRAC * pinn_amp_prev;
                if (zh < 1e-9) {          /* aun sin amplitud de referencia */
                    crossing = (pinn_have_prev && pv_is_al < 0.0 && is_alfa >= 0.0)
                        ;
                } else if (pinn_zc_hi) {  /* ya disparo: espera a bajar de -h */
                    if (is_alfa < -zh) pinn_zc_hi = 0;
                    crossing = 0;
                } else {                  /* armado: dispara al superar +h  */
                    crossing = (pinn_have_prev && is_alfa >= zh);
                    if (crossing) pinn_zc_hi = 1;
                }
            }

            if (crossing) {
                /* (c2) CIERRE DE CICLO: si junto suficientes muestras,
                 * calcula las 21 features base y corre ambos expertos. */
                if (pinn_in_cyc && pb_n >= PINN_MIN_CYCLE) {
                    int    N    = (pb_n < PINN_NBUF) ? pb_n : PINN_NBUF;
                    double invn = 1.0 / (double)pb_n;
                    double dt   = cyc_tlast - cyc_t0; if (dt < 1e-9) dt = 1e-9;
                    int hs;
                    /* Canonizacion de giro: la maquina es simetrica, de modo que
                     * el mismo experto sirve en los dos sentidos si se presentan
                     * las entradas en el sentido positivo. Histeresis del 10 %% de
                     * la nominal para no conmutar al cruzar cero. Solo entradas de las
                         redes, no los coordinadores: ver cap. 5 */
                    if      (ac_wref * invn >  0.10 * PINN_WNOM_EL) g_dir_ref =
                        1.0;
                    else if (ac_wref * invn < -0.10 * PINN_WNOM_EL) g_dir_ref =
                        -1.0;
```

```
444                     hs = (g_dir_ref < 0.0) ? -1 : 1;
445                     /* Al invertir el giro el vector espacial se conjuga y las
446                      * secuencias positiva y negativa se INTERCAMBIAN. Como
447                      * |X(-h)| = |X(h)| bajo conjugacion, basta invertir el ORDEN.
                            */
448                     double I1   = pinn_comp(pb_al, pb_be, N, hs); double I1d = I1 +
                            1e-9;
449                     double he   = 0.0; int hh;
450                     double rms_eps, rms_i, tmean, tvar, f[PINN_NFEAT];
451                     for (hh = 2; hh <= 10; hh++) { double c = pinn_comp(pb_al,
                            pb_be, N, hh*hs); he += c*c; }
452                     rms_eps = sqrt(ac_C * invn); rms_i = sqrt(ac_i2 * invn) + 1e-9;
453                     tmean = ac_test * invn; tvar = ac_test2 * invn - tmean*tmean;
                            if (tvar < 0.0) tvar = 0.0;
454                     /* --- FEATURES BASE f[0..20] (por ciclo) ---
455                      *  0 Rr_eff  1 Rr_base  2 d_hat=B/A [ohm]  3 rms_eps/rms_i
456                      *  4 i_peak  5 I1 (fundamental)  6 THD(2..10)  7 h3  8 h5
457                      *  9 h7  10 sec. negativa  11 w_media/wnom  12 acel/wnom
458                      * 13 T*_medio/Tnom  14 T_est/Tnom  15 rizado T/Tnom
459                      * 16 psi_r medio  17 torque_resid  18 idx_switch
460                      * 19 frecuencia del ciclo [Hz]  20 num. muestras */
461                     f[0]=Rr_eff_now; f[1]=Rr_MPC_base; f[2]=ac_B/(ac_A+1e-12);
462                     f[3]=rms_eps/rms_i; f[4]=cyc_ipeak; f[5]=I1; f[6]=sqrt(he)/I1d;
463                     f[7]=pinn_comp(pb_al,pb_be,N,3*hs)/I1d; f[8]=pinn_comp(pb_al,
                            pb_be,N,5*hs)/I1d;
464                     f[9]=pinn_comp(pb_al,pb_be,N,7*hs)/I1d; f[10]=pinn_comp(pb_al,
                            pb_be,N,-1*hs)/I1d;
465                     f[11]=(ac_w*invn)/PINN_WNOM_EL; f[12]=(cyc_wlast-cyc_w0)/dt/
                            PINN_WNOM_EL;
466                     f[13]=(ac_tref*invn)/PINN_TNOM; f[14]=tmean/PINN_TNOM; f[15]=
                            sqrt(tvar)/PINN_TNOM;
467                     f[16]=ac_psir*invn;
468                     /* torque_resid = (T_est_med - (J/p)*dw_el/dt) / T_nom  (T_real
                            desde la
469                      * aceleracion medida, sin carga). ~0 en matched, grande si
                            Rr_MPC>>Rr_MAS. */
470                     f[17]=(tmean - PINN_J * (cyc_wlast - cyc_w0) / (p * dt)) /
                            PINN_TNOM;
471                     f[18]=(pb_n>1)?ac_idxsw/(double)(pb_n-1):0.0;   /* idx_switch
                            */
472                     f[19]=1.0/dt;                                    /* cyc_freq
                            */
473                     f[20]=(double)pb_n;                              /* cyc_nsamp
                            */
474 #if PINN_WRITE_FEATURES
475                     if (pinn_fp) { int q; fprintf(pinn_fp,"%d",pinn_cyc_idx);
476                         for(q=0;q<PINN_NFEAT;q++) fprintf(pinn_fp,",%.8g",f[q]);
                                fprintf(pinn_fp,"\n"); }
477 #endif
478                     pinn_cyc_idx++;
479
480
481
482                     /* ---- SISTEMA v2: dos expertos direccionales ----
483                      * fv[0..26] contiene las 27 caracteristicas base de la version
                            v4. */
484                     {
485                         double fv[PINN_V2_NB];
486                         double psis_m = ac_psis * invn;
487                         double isqmax_m = ac_isqmax * invn;
488                         /* v4: 27 caracteristicas, orden de BASE_V4 en
489                          * dataset_v4.py. Salen rr_base, w_mean, accel,
490                          * cyc_freq y slip_norm (duplicados / no informativa);
491                          * entran phys_C, w_ref y tpa. */
492                         fv[ 0] = f[0];                  /* rr_eff       */
493                         fv[ 1] = f[2];                  /* d_hat        */
494                         fv[ 2] = f[3];                  /* rms_eps_norm */
495                         fv[ 3] = ac_C;                  /* phys_C       */
```

```
496                         fv[ 4] = f[4];                  /* i_peak       */
497                         fv[ 5] = f[5];                  /* I1           */
498                         fv[ 6] = f[6];                  /* thd          */
499                         fv[ 7] = f[7];                  /* h3           */
500                         fv[ 8] = f[8];                  /* h5           */
501                         fv[ 9] = f[9];                  /* h7           */
502                         fv[10] = f[10];                 /* neg_seq      */
503                         fv[11] = f[13] * g_dir_ref;     /* tref_mean   ESPEJO */
504                         fv[12] = f[14] * g_dir_ref;     /* test_mean   ESPEJO */
505                         fv[13] = f[15];                 /* test_ripple  */
506                         fv[14] = f[16];                 /* psir_mean    */
507                         fv[15] = f[17] * g_dir_ref;     /* torque_resid ESPEJO */
508                         fv[16] = f[18];                 /* idx_switch   */
509                         fv[17] = f[20];                 /* cyc_nsamp    */
510                         fv[18] = rms_i;                 /* rms_i        */
511                         fv[19] = ((ac_wref - ac_w) * invn) / PINN_WNOM_EL *
                                g_dir_ref;
512                         fv[20] = (ac_wref * invn) / PINN_WNOM_EL * g_dir_ref;  /*
                                w_ref ESPEJO */
513                         fv[21] = f[13] / (rms_i + 1e-9) * g_dir_ref;   /* tpa
                                ESPEJO */
514                         fv[22] = (psi_s_ref_out) / PINN_V2_PSI_NOM;
515                         fv[23] = psi_s_ref_out - psis_m;
516                         fv[24] = (ac_isq * invn) / (isqmax_m + 1e-9); /* isq_frac invariante */
517                         fv[25] = ac_wslip * invn * g_dir_ref;
518                         fv[26] = (ac_B*ac_B) / (ac_A*ac_C + 1e-30);
519                         /* Un cambio de sentido invalida los agregados multi-
                                periodo: dos de
520                          * las diez claves de ventana movil son ESPEJO. */
521                         if (g_dir_ref != g_dir_prev) {
522                             v2_mp_reset(); have_fv_prev = 0; g_dir_prev = g_dir_ref
                                    ;
523                         }
524                         v2_push(fv);
525                         if (have_fv_prev) {
526                             double xv[PINN_V2_BAJAR_N_IN];
527                             double s_sub, c_sub, d_sub, s_sob, c_sob, d_sob;
528                             double cand2, s_o, conf_o; int rama_o, fsm_o, ver_o,
                                    dec2;
529                             v2_build_x(xv, fv, fv_prev);
530                             pinn_v2_subir_forward(xv, &s_sub, &c_sub, &d_sub);
531                             pinn_v2_bajar_forward(xv, &s_sob, &c_sob, &d_sob);
532                             dbg_a_ssub = s_sub; dbg_a_csub = c_sub; dbg_a_dsub =
                                    d_sub;
533                             dbg_a_ssob = s_sob; dbg_a_csob = c_sob; dbg_a_dsob =
                                    d_sob;
534                             /* ventana de arranque, identica a la de v26 */
535                             double wrs = (w_ref > 1e-6) ? w_ref : 1e-6; /* con w_ref
                                    < 0 la ventana no se abre; ver cap. 5 */
536                             double prog = (w_eli / p) / wrs;
537                             int stg2 = (prog < 0.20) ? 0 : (prog < 0.90 ? 1 : 2);
538                             int sa2 = (magnetized_flag > 0.5) && (CurrentTime >
                                    PINN_T_SKIP)
539                                   && (zmag > PINN_AMP_MIN) && (stg2 >= 1);
540                             int se2 = (prog >= 0.97);
541                             dec2 = ac2_step(&g_ac2, s_sub, c_sub, d_sub, s_sob,
                                    c_sob, d_sob,
542                                             (magnetized_flag > 0.5), sa2, se2, stg2
                                                ,
543                                             f[2], f[3], fv[26], fv[22], f[14], f
                                                [13], f[12],
544                                             w_ref,   /* v3: re-arme por maniobra */
545                                             prog,    /* v3h: detector de
                                                estancamiento */
546                                             dt,      /* v4b: techo en tiempo del
                                                calado */
547                                             (PINN_V2_APPLY > 0.5),
```

```
548                                         &cand2, &s_o, &conf_o, &rama_o, &fsm_o,
                                                &ver_o);
549                         dbg_v2_s = s_o; dbg_v2_conf = conf_o; dbg_v2_cand =
                                cand2;
550                         dbg_v2_dom = (rama_o == V2_RAMA_SUB) ? d_sub : d_sob;
551                         dbg_v2_dec = (double)dec2; dbg_v2_rama = (double)rama_o
                                ;
552                         dbg_v2_fsm = (double)fsm_o; dbg_v2_veredicto = (double)
                                ver_o;
553                         dbg_v2_rho2 = fv[26];
554                         if (PINN_V2_APPLY > 0.5 && dec2 != AC2_SAFE_D
555                             && g_k2.committed == 0)   /* carga comiteo ->
                                    arranque cede */
556                             k_rr_arr_state = g_ac2.rr_effect / Rr_MPC_base;
557                     }
558                     { int qv2; for (qv2 = 0; qv2 < PINN_V2_NB; qv2++) fv_prev[
                            qv2] = fv[qv2]; }
559                     have_fv_prev = 1;
560
561                     /* ---- EXPERTO DE CARGA v2: vector x[141] y sus dos redes.
562                      * El coordinador es el port de v24: conserva deteccion de
563                      * escalon, espera de transitorio, re-arme multi-evento y
564                      * la via de emergencia. */
565                     if (have_fv_prev) {
566                         double xk[PINN_V2_CSUBIR_N_IN];
567                         double ks_u, kc_u, kd_u, ks_o, kc_o, kd_o;
568                         double kc, ks, krt, kcf; int kf, ka, kac, kdn, ksf, ksd
                                , knc, kod, kdec;
569                         v2_build_x(xk, fv, fv_prev);
570                         pinn_v2_csubir_forward(xk, &ks_u, &kc_u, &kd_u);
571                         pinn_v2_cbajar_forward(xk, &ks_o, &kc_o, &kd_o);
572                         dbg_k_ssub = ks_u; dbg_k_csub = kc_u; dbg_k_dsub = kd_u
                                ;
573                         dbg_k_ssob = ks_o; dbg_k_csob = kc_o; dbg_k_dsob = kd_o
                                ;
574                         kdec = k2_step(&g_k2, rms_i, f[13], f[11], f[12], fv
                                [19], f[2],
575                                        f[3], fv[26], fv[22],
576                                        ks_u, kc_u, kd_u, ks_o, kc_o, kd_o,
577                                        Rr_MPC_effect_ds,
578                                        w_ref,   /* v3f: re-arme por maniobra */
579                                        (PINN_carga > 0.5), (PINN_V2_APPLY >
                                            0.5),
580                                        &kc, &ks, &krt, &kcf,
581                                        &kf, &ka, &kac, &kdn, &ksf, &ksd, &knc,
                                            &kod);
582                         dbg_k2_cand = kc; dbg_k2_s = ks; dbg_k2_dec = (double)
                                kdec;
583                         dbg_k2_fsm = (double)kf; dbg_k2_route = (double)g_k2.
                                route;
584                         dbg_k2_comm = (double)g_k2.committed;
585                         if (PINN_carga > 0.5 && PINN_V2_APPLY > 0.5 && kdec ==
                                K2_ACTIVE) /* tambien la reversion entrega K2_ACTIVE; ver
                                anexo G */
586                             k_rr_arr_state = g_k2.rr_effect / Rr_MPC_base;
587 #if V3_TRACE
588                         if (v3_fp) fprintf(v3_fp, "%.6g,%.6g,%.6g,%.6g,%.6g,%.6
                                g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g
                                ,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g
                                ,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g
                                ,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g
                                ,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g,%.6g
                                ,%.6g,%.6g,%.6g\n",
589                             CurrentTime, Rr_MPC_effect_ds, k_rr_arr_state,
590                             (double)g_ac2.state, (double)g_ac2.steps, (double)
                                    g_ac2.brk_steps,
591                             (double)g_ac2.maniobras, (double)g_ac2.pend_n, (
                                    double)g_ac2.sin_mejora,
```

```
                                dbg_v2_dec, dbg_v2_rama, dbg_v2_veredicto, dbg_v2_s
                                    , dbg_v2_conf, dbg_v2_cand,
                                dbg_a_ssub, dbg_a_csub, dbg_a_dsub, dbg_a_ssob,
                                    dbg_a_csob, dbg_a_dsob,
                                (double)g_k2.state, (double)g_k2.motivo, (double)
                                    g_k2.n_commits,
                                (double)g_k2.buf_n, (double)g_k2.rolled_back, (
                                    double)g_k2.ood_flag,
                                (double)g_k2.last_dir, (double)g_k2.cont_n, (double
                                    )g_k2.espera,
                                (double)g_k2.reinicios, (double)g_k2.step_detected,
                                     (double)g_k2.committed,
                                (double)g_k2.emg_steps, (double)g_k2.emg_events, (
                                    double)g_k2.emg_flips,
                                g_k2.eps_pre,
                                (g_k2.post_n > 0) ? g_k2.eps_post_sum / (double)
                                    g_k2.post_n : 0.0,
                                (double)kdec, (double)g_k2.route, ks, kc,
                                dbg_k_ssub, dbg_k_csub, dbg_k_dsub, dbg_k_ssob,
                                    dbg_k_csob, dbg_k_dsob,
                                f[3], fv[26], fv[22], f[14], f[13], rms_i, f[2], f
                                    [11], f[12], (double)g_k2.vpre, (double)g_k2.
                                    via);
#endif
                        }
                    }
                }
                /* v4: amplitud de referencia para la histeresis del ciclo
                 * siguiente. Se reutiliza la compuerta de amplitud que ya
                 * existia, para no introducir un umbral nuevo. */
                if (cyc_ipeak > PINN_AMP_MIN) pinn_amp_prev = cyc_ipeak;
                /* Abrir el ciclo nuevo (el cruce actual es su primera muestra) */
                pinn_reset_cycle();
                pinn_in_cyc = 1;
                cyc_w0 = w_eli; cyc_t0 = CurrentTime; pidx_cyc = (double)idx_opt;
            }

            /* (c3) Acumulacion de la muestra actual en el ciclo abierto:
             * sumas del residuo (A=phi.phi, B=eps.phi, C=eps.eps), contexto
             * (w, T*, T_est, psi_r), conmutaciones y buffer alfa-beta. */
            if (pinn_in_cyc) {
                if (pinn_have_prev) {
                    ac_A += pa*pa + pb*pb; ac_B += eps_al*pa + eps_be*pb;
                    ac_C += eps_al*eps_al + eps_be*eps_be;
                }
                ac_i2 += is_alfa*is_alfa + is_beta*is_beta;
                /* v2: flujo de estator estimado, corriente de par y su tope */
                ac_psis += sqrt(psi_s_d*psi_s_d + psi_s_q*psi_s_q);
                ac_isq += fabs(is_q); ac_isqmax += isq_max;
                ac_wslip += w_slip; ac_wsyncf += w_sync_f;
                ac_w += w_eli; ac_tref += T_ref_local; ac_wref += w_ref * p;   /*
                    v3 */
                ac_test += T_est; ac_test2 += T_est*T_est;
                ac_psir += sqrt(psi_r_alfa_out*psi_r_alfa_out + psi_r_beta_out*
                    psi_r_beta_out);
                if (pb_n > 0 && (double)idx_opt != pidx_cyc) ac_idxsw += 1.0;
                pidx_cyc = (double)idx_opt;
                if (zmag > cyc_ipeak) cyc_ipeak = zmag;
                if (pb_n < PINN_NBUF) { pb_al[pb_n] = is_alfa; pb_be[pb_n] =
                    is_beta; }
                cyc_wlast = w_eli; cyc_tlast = CurrentTime; pb_n++;
            }
        }
        /* Memoria de la muestra para el residuo del paso siguiente */
        pv_is_al = is_alfa; pv_is_be = is_beta;
        pv_psir_al = psi_r_alfa_out; pv_psir_be = psi_r_beta_out; /* gamma[k+1] */
        pv_ipred_al = i_s_alfa_k1_prev; pv_ipred_be = i_s_beta_k1_prev;
        pinn_have_prev = 1;
```

```
        shape_metric_cv_ds = (pb_n > 0) ? (ac_B / (ac_A + 1e-12)) : 0.0;
    }

    /* ========================================
     *  D) CLOCK DE COSTO COMPUTACIONAL (opcional, CLK_ENABLE)
     *  Promedia ticks de CPU del MPC y del total tras el warmup.
     * ======================================== */
#ifdef PINN_USE_CLOCK
    if (CLK_ENABLE > 0.5) {
        double d_total, d_pinn, d_mpc, n;
        clk_pe = clock(); clk_ue = clock();
        d_total = (double)(clk_ue - clk_us);
        d_pinn  = (double)(clk_pe - clk_ps);
        d_mpc   = d_total - d_pinn;
        if (clk_steps < CLK_WARMUP_STEPS) {
            clk_steps += 1.0;
        } else {
            clk_mpc_sum_ticks   += d_mpc;
            clk_total_sum_ticks += d_total;
            clk_steps += 1.0;
            n = clk_steps - CLK_WARMUP_STEPS; if (n < 1.0) n = 1.0;
            t_CPU_MPC_us_ds   = 1.0e6 * clk_mpc_sum_ticks   / (double)
                CLOCKS_PER_SEC / n;
            t_CPU_TOTAL_us_ds = 1.0e6 * clk_total_sum_ticks / (double)
                CLOCKS_PER_SEC / n;
        }
    }
#endif
}
```

Listing B.4: Update Function del C-Script principal: PTC, extracción cíclica y sistema de expertos.

B.5. C-Script de carga física

El segundo bloque `C-Script` de la plataforma implementa el modelo de carga mecánica reactiva de la Subsección 4.1.2. Su código completo es compacto y se reproduce íntegro: selector de modo (sin carga / bomba / cinta), cronograma de material con rampa limitada en pendiente, y protecciones de saturación y finitud.

```
/* ================ Code Declarations ================ */
/* ========================================
 *  Carga_fisica (planta v25) - carga mecanica REACTIVA seleccionable
 * ========================================
 *  Reemplaza a "Escalones multiples" como fuente del torque de carga.
 *  A diferencia de aquel (torque PRESCRITO de signo fijo desde t=0, que
 *  empujaba la maquina hacia atras en arranques cargados), este par es
 *  REACTIVO: se OPONE a la velocidad mecanica real y vale ~0 en w_m=0,
 *  de modo que NUNCA impulsa la maquina -> no hay aceleracion negativa
 *  no fisica desde el reposo.
 *
 *  SETUP:
 *    Number of inputs   : 1   (Input(0) = w_m [rad/s mecanico], via From[w])
 *    Number of outputs  : 1   (Output(0) = T_cmd [N*m] -> ControlledTorque)
 *    Number of cont.st. : 0
 *    Number of disc.st. : 1   (DiscState(0) = nivel de material rate-limited)
 *    Direct feedthrough : 1   (la salida depende de w_m)
 *    Sample time        : Ts
 *
 *  PARAMETROS (indice ParamRealData -> nombre):
 *    0 LOAD_MODE   selector de modo (0 / 1 / 2)
 *    1 T_nom [Nm]  torque nominal de la bomba a w_nom
 *    2 w_nom [rad/s] velocidad nominal (define K_pump)
```

```
 *    3 T_LOAD_MAX [Nm] saturacion de proteccion |T| <= T_LOAD_MAX
 *    4 B_VISC [Nm s/rad] coeficiente viscoso adicional
 *    5 W_EPS  [rad/s] ancho de la tanh (regularizacion de Coulomb)
 *    6 T_BASE [Nm] friccion base de la cinta (correa vacia)
 *    7 RAMP_MAT [Nm/s] rate-limit del material sobre la correa
 *    8..16  t_dt1..t_dt9  [s]  duracion de cada tramo del cronograma
 *   17..25  T_lvl1..T_lvl9 [Nm] material objetivo por tramo (se usa |.|)
 *
 *  MODOS (LOAD_MODE):
 *    0 sin carga : T = 0.
 *    1 bomba     : T = -K*w*|w| - B*w,  K = T_nom/w_nom^2  (ley de
 *                  afinidad; T(0)=0, C^1 suave; par ~T_nom a w_nom).
 *    2 cinta     : T = -(T_base + T_mat)*tanh(w/w_eps) - B*w. Friccion
 *                  reactiva regularizada (Coulomb suavizado): se opone
 *                  al movimiento y tiende a 0 en el reposo. T_mat =
 *                  material sobre la correa (rampa suave segun el
 *                  cronograma t_dt_i/|T_lvl_i|; entrada/salida mineral).
 *
 *  Convencion de signo = la del ControlledTorque existente (valor
 *  negativo FRENA la rotacion positiva). Proteccion: |T| <= T_LOAD_MAX
 *  y NaN/Inf -> 0. La carga fisica JAMAS debe impulsar la maquina.
 *
 *  NOTA: los macros de contexto (ParamRealData, Input, DiscState,
 *  CurrentTime...) solo son validos DENTRO de las funciones de seccion;
 *  por eso los parametros se leen a variables static en Start (igual
 *  que el bloque MPC del proyecto) y el helper del cronograma usa esas
 *  static, no ParamRealData.
 * ========================================== */

/* === Code Declarations === */
#include <math.h>

/* Parametros persistentes (copiados en Start) y ganancia de bomba cf_K */
static double cf_mode, cf_tnom, cf_wnom, cf_tmax, cf_bvisc, cf_weps, cf_tbase,
    cf_ramp, cf_K;
static double cf_dt[9], cf_lv[9];

/* cf_sched_target: material objetivo [Nm] segun el cronograma.
 * Recorre los 9 tramos acumulando duraciones; devuelve |T_lvl_i| del
 * tramo que contiene a t, o 0 tras el ultimo (correa descargada). */
static double cf_sched_target(double t) {
    double acc = 0.0;
    int i;
    for (i = 0; i < 9; i++) {
        acc += cf_dt[i];
        if (t < acc) return fabs(cf_lv[i]);
    }
    return 0.0;   /* tras el ultimo tramo: correa descargada */
}

/* ================= Start Function ================= */
/* === Start Function ===
 * Lee los 26 parametros a static (el orden ES el mapa de indices),
 * precalcula la ganancia de bomba K = T_nom/w_nom^2 con guarda de
 * division por cero, sanea W_EPS y arranca con la correa vacia. */
{
    int i;
    cf_mode  = ParamRealData(0, 0);
    cf_tnom  = ParamRealData(1, 0);
    cf_wnom  = ParamRealData(2, 0);
    cf_tmax  = ParamRealData(3, 0);
    cf_bvisc = ParamRealData(4, 0);
    cf_weps  = ParamRealData(5, 0);
    cf_tbase = ParamRealData(6, 0);
    cf_ramp  = ParamRealData(7, 0);
    for (i = 0; i < 9; i++) {
        cf_dt[i] = ParamRealData(8 + i, 0);    /* t_dt1..t_dt9   */
        cf_lv[i] = ParamRealData(17 + i, 0);   /* T_lvl1..T_lvl9 */
```

```
    }
    cf_K = (cf_wnom > 1e-9) ? cf_tnom / (cf_wnom * cf_wnom) : 0.0;
    if (cf_weps <= 1e-9) cf_weps = 1.0;   /* evita tanh(w/0) */
    DiscState(0) = 0.0;   /* material inicial = 0 (correa vacia) */
}

/* ================= Output Function ================= */
/* === Output Function ===
 * Calcula el torque de carga REACTIVO segun LOAD_MODE:
 *   modo 0 sin carga : T = 0.
 *   modo 1 BOMBA     : T = -K*w*|w| - B*w. Par cuadratico (ley de
 *                      afinidad); w*|w| conserva el signo => siempre
 *                      OPONE al giro y T(0)=0.
 *   modo 2 CINTA     : T = -(T_base+T_mat)*tanh(w/w_eps) - B*w.
 *                      Friccion de Coulomb regularizada con tanh:
 *                      opone al movimiento, ~0 cerca del reposo (sin
 *                      discontinuidad en w=0). T_mat = DiscState(0).
 * Proteccion final: saturacion |T| <= T_LOAD_MAX y NaN/Inf -> 0.
 * Signo: T negativo FRENA rotacion positiva (conv. ControlledTorque). */
{
    double w    = Input(0);        /* velocidad mecanica [rad/s] */
    double Tmat = DiscState(0);    /* material sobre la correa [Nm] */
    int    mode = (int)(cf_mode + 0.5);
    double T    = 0.0;
    if (mode == 1)      T = -cf_K * w * fabs(w) - cf_bvisc * w;                  /*
        BOMBA */
    else if (mode == 2) T = -(cf_tbase + Tmat) * tanh(w / cf_weps) - cf_bvisc * w;
        /* CINTA */
    else                T = 0.0;                                                 /*
        SIN CARGA */
    if (T >  cf_tmax) T =  cf_tmax;   /* saturacion de proteccion */
    if (T < -cf_tmax) T = -cf_tmax;
    if (!(T == T))    T = 0.0;        /* NaN/Inf -> 0 (T==T falla solo con NaN) */
    Output(0) = T;
}

/* ================= Update Function ================= */
/* === Update Function ===
 * Dinamica del material sobre la correa (solo modo cinta):
 *  1) lee el objetivo del cronograma cf_sched_target(t) [Nm];
 *  2) rate-limit simetrico RAMP_MAT [Nm/s]: la carga y la descarga
 *     de mineral son rampas suaves, no escalones;
 *  3) DiscState(0) converge al objetivo paso a paso.
 * En modos 0 y 1 el objetivo queda en 0 (material se vacia si lo hubo). */
{
    double target = 0.0;
    int    mode   = (int)(cf_mode + 0.5);
    double step, cur, d;
    if (mode == 2) target = cf_sched_target(CurrentTime);   /* solo cinta usa
        material */
    step = cf_ramp * SampleTimePeriod(0);   /* avance maximo por paso [Nm] */
    cur  = DiscState(0);
    d    = target - cur;
    if      (d >  step) cur += step;    /* subiendo: entrada de material  */
    else if (d < -step) cur -= step;    /* bajando: descarga de material  */
    else                cur  = target;  /* dentro del paso: clava exacto  */
    DiscState(0) = cur;
}
```

Listing B.5: C-Script de la carga mecánica reactiva de la configuración v36_v4.

B.6. Integración de los expertos en el C-Script principal

El fragmento siguiente muestra la integración completa de los cuatro expertos dentro de la *Update Function*: construcción de las 27 características base, actualización de las

ventanas multiperíodo, formación del vector de 141 componentes, inferencia de las dos redes de arranque y las dos redes de carga, y llamada a sus coordinadores. El orden de actualización de `fv_prev` que caracteriza a la rama de carga se conserva porque forma parte de la implementación con que se obtuvieron los resultados; su efecto se explica en el Capítulo 5.

```
/* ---- SISTEMA v2: dos expertos direccionales ----
                     * fv[0..26] contiene las 27 caracteristicas base de la version
                         v4. */
                    {
                        double fv[PINN_V2_NB];
                        double psis_m = ac_psis * invn;
                        double isqmax_m = ac_isqmax * invn;
                        /* v4: 27 caracteristicas, orden de BASE_V4 en
                         * dataset_v4.py. Salen rr_base, w_mean, accel,
                         * cyc_freq y slip_norm (duplicados / no informativa);
                         * entran phys_C, w_ref y tpa. */
                        fv[ 0] = f[0];                  /* rr_eff        */
                        fv[ 1] = f[2];                  /* d_hat         */
                        fv[ 2] = f[3];                  /* rms_eps_norm  */
                        fv[ 3] = ac_C;                  /* phys_C        */
                        fv[ 4] = f[4];                  /* i_peak        */
                        fv[ 5] = f[5];                  /* I1            */
                        fv[ 6] = f[6];                  /* thd           */
                        fv[ 7] = f[7];                  /* h3            */
                        fv[ 8] = f[8];                  /* h5            */
                        fv[ 9] = f[9];                  /* h7            */
                        fv[10] = f[10];                 /* neg_seq       */
                        fv[11] = f[13] * g_dir_ref;     /* tref_mean   ESPEJO */
                        fv[12] = f[14] * g_dir_ref;     /* test_mean   ESPEJO */
                        fv[13] = f[15];                 /* test_ripple   */
                        fv[14] = f[16];                 /* psir_mean     */
                        fv[15] = f[17] * g_dir_ref;     /* torque_resid ESPEJO */
                        fv[16] = f[18];                 /* idx_switch    */
                        fv[17] = f[20];                 /* cyc_nsamp     */
                        fv[18] = rms_i;                 /* rms_i         */
                        fv[19] = ((ac_wref - ac_w) * invn) / PINN_WNOM_EL *
                            g_dir_ref;
                        fv[20] = (ac_wref * invn) / PINN_WNOM_EL * g_dir_ref;  /*
                            w_ref ESPEJO */
                        fv[21] = f[13] / (rms_i + 1e-9) * g_dir_ref;   /* tpa
                            ESPEJO */
                        fv[22] = (psi_s_ref_out) / PINN_V2_PSI_NOM;
                        fv[23] = psi_s_ref_out - psis_m;
                        fv[24] = (ac_isq * invn) / (isqmax_m + 1e-9); /* isq_frac invariante */
                        fv[25] = ac_wslip * invn * g_dir_ref;
                        fv[26] = (ac_B*ac_B) / (ac_A*ac_C + 1e-30);
                        /* Un cambio de sentido invalida los agregados multi-
                            periodo: dos de
                         * las diez claves de ventana movil son ESPEJO. */
                        if (g_dir_ref != g_dir_prev) {
                            v2_mp_reset(); have_fv_prev = 0; g_dir_prev = g_dir_ref
                                ;
                        }
                        v2_push(fv);
                        if (have_fv_prev) {
                            double xv[PINN_V2_BAJAR_N_IN];
                            double s_sub, c_sub, d_sub, s_sob, c_sob, d_sob;
                            double cand2, s_o, conf_o; int rama_o, fsm_o, ver_o,
                                dec2;
                            v2_build_x(xv, fv, fv_prev);
                            pinn_v2_subir_forward(xv, &s_sub, &c_sub, &d_sub);
                            pinn_v2_bajar_forward(xv, &s_sob, &c_sob, &d_sob);
                            dbg_a_ssub = s_sub; dbg_a_csub = c_sub; dbg_a_dsub =
                                d_sub;
                            dbg_a_ssob = s_sob; dbg_a_csob = c_sob; dbg_a_dsob =
                                d_sob;
```

```
53                          /* ventana de arranque, identica a la de v26 */
54                          double wrs = (w_ref > 1e-6) ? w_ref : 1e-6; /* con w_ref
                                < 0 la ventana no se abre; ver cap. 5 */
55                          double prog = (w_eli / p) / wrs;
56                          int stg2 = (prog < 0.20) ? 0 : (prog < 0.90 ? 1 : 2);
57                          int sa2 = (magnetized_flag > 0.5) && (CurrentTime >
                                PINN_T_SKIP)
58                                && (zmag > PINN_AMP_MIN) && (stg2 >= 1);
59                          int se2 = (prog >= 0.97);
60                          dec2 = ac2_step(&g_ac2, s_sub, c_sub, d_sub, s_sob,
                                c_sob, d_sob,
61                                          (magnetized_flag > 0.5), sa2, se2, stg2
                                              ,
62                                          f[2], f[3], fv[26], fv[22], f[14], f
                                              [13], f[12],
63                                          w_ref,   /* v3: re-arme por maniobra */
64                                          prog,    /* v3h: detector de
                                              estancamiento */
65                                          dt,      /* v4b: techo en tiempo del
                                              calado */
66                                          (PINN_V2_APPLY > 0.5),
67                                          &cand2, &s_o, &conf_o, &rama_o, &fsm_o,
                                               &ver_o);
68                          dbg_v2_s = s_o; dbg_v2_conf = conf_o; dbg_v2_cand =
                                cand2;
69                          dbg_v2_dom = (rama_o == V2_RAMA_SUB) ? d_sub : d_sob;
70                          dbg_v2_dec = (double)dec2; dbg_v2_rama = (double)rama_o
                                ;
71                          dbg_v2_fsm = (double)fsm_o; dbg_v2_veredicto = (double)
                                ver_o;
72                          dbg_v2_rho2 = fv[26];
73                          if (PINN_V2_APPLY > 0.5 && dec2 != AC2_SAFE_D
74                              && g_k2.committed == 0)   /* carga comiteo ->
                                    arranque cede */
75                              k_rr_arr_state = g_ac2.rr_effect / Rr_MPC_base;
76                      }
77                      { int qv2; for (qv2 = 0; qv2 < PINN_V2_NB; qv2++) fv_prev[
                            qv2] = fv[qv2]; }
78                      have_fv_prev = 1;
79
80                      /* ---- EXPERTO DE CARGA v2: vector x[141] y sus dos redes.
81                       * El coordinador es el port de v24: conserva deteccion de
82                       * escalon, espera de transitorio, re-arme multi-evento y
83                       * la via de emergencia. */
84                      if (have_fv_prev) {
85                          double xk[PINN_V2_CSUBIR_N_IN];
86                          double ks_u, kc_u, kd_u, ks_o, kc_o, kd_o;
87                          double kc, ks, krt, kcf; int kf, ka, kac, kdn, ksf, ksd
                                , knc, kod, kdec;
88                          v2_build_x(xk, fv, fv_prev);
89                          pinn_v2_csubir_forward(xk, &ks_u, &kc_u, &kd_u);
90                          pinn_v2_cbajar_forward(xk, &ks_o, &kc_o, &kd_o);
91                          dbg_k_ssub = ks_u; dbg_k_csub = kc_u; dbg_k_dsub = kd_u
                                ;
92                          dbg_k_ssob = ks_o; dbg_k_csob = kc_o; dbg_k_dsob = kd_o
                                ;
93                          kdec = k2_step(&g_k2, rms_i, f[13], f[11], f[12], fv
                                [19], f[2],
94                                         f[3], fv[26], fv[22],
95                                         ks_u, kc_u, kd_u, ks_o, kc_o, kd_o,
96                                         Rr_MPC_effect_ds,
97                                         w_ref,   /* v3f: re-arme por maniobra */
98                                         (PINN_carga > 0.5), (PINN_V2_APPLY >
                                               0.5),
99                                         &kc, &ks, &krt, &kcf,
100                                        &kf, &ka, &kac, &kdn, &ksf, &ksd, &knc,
                                               &kod);
101                         dbg_k2_cand = kc; dbg_k2_s = ks; dbg_k2_dec = (double)
```

```
                                kdec;
                            dbg_k2_fsm = (double)kf; dbg_k2_route = (double)g_k2.
                                route;
                            dbg_k2_comm = (double)g_k2.committed;
                            if (PINN_carga > 0.5 && PINN_V2_APPLY > 0.5 && kdec ==
                                K2_ACTIVE) /* tambien la reversion entrega K2_ACTIVE; ver
                                anexo G */
                                k_rr_arr_state = g_k2.rr_effect / Rr_MPC_base;
                        }
                    }
                }
```

Listing B.6: Construcción de características, inferencia de las cuatro redes y llamada a los coordinadores en v36_v4.

El extracto del coordinador de arranque muestra el enrutamiento direccional, los umbrales de dominio y confianza, la firma física, la confirmación temporal, la banda muerta y la escritura acotada:

```
            if (!hay_desajuste) { c->pend = AC2_RAMA_NONE; c->pend_n = 0; break; }

            rama_red = AC2_RAMA_NONE;
            if (dom_sub >= dom_min && conf_sub >= conf_min && s_sub > 0.0)
                rama_red = AC2_RAMA_SUB;
            if (dom_sob >= dom_min && conf_sob >= conf_min && s_sob < 0.0) {
                if (rama_red != AC2_RAMA_NONE)
                    rama_red = (firma != AC2_RAMA_NONE) ? firma : AC2_RAMA_NONE;
                else rama_red = AC2_RAMA_SOBRE;
            }
            if (rama_red == AC2_RAMA_NONE) {
                veredicto = 3; dec = AC2_IGNORA; c->pend_n = 0; break;
            }
            if (firma != AC2_RAMA_NONE && firma != rama_red) { c->pend_n = 0; break
                ; }
            if (c->last_dir != 0 && rama_red != c->last_dir && c->flips >=
                AC2_MAX_FLIPS) {
                c->state = AC2ST_DONE; c->done = 1; break;
            }
            if (rama_red == c->pend) c->pend_n++;
            else { c->pend = rama_red; c->pend_n = 1; }
            if (c->pend_n < n_conf) break;

            c->rama = rama_red;
            c->state = AC2ST_ACTIVE;
            s_hat = (c->rama == AC2_RAMA_SUB) ? s_sub : s_sob;
            conf  = (c->rama == AC2_RAMA_SUB) ? conf_sub : conf_sob;
            /* v3: deadband MIXTO -- el mayor entre el absoluto y el relativo,
             * expresado en log-razon sobre el Rr vigente. */
            {   double dead_ohm = (AC2_DEAD_REL * c->rr_effect > AC2_DEAD_OHM)
                                  ? AC2_DEAD_REL * c->rr_effect : AC2_DEAD_OHM;
                double s_dead = log10(1.0 + dead_ohm / c->rr_effect);
#if AC2_DEAD_ADAPT
                {   /* 3*sigma de la salida CRUDA de la red enrutada, sobre las
                     * AC2_N_MED ventanas. El mixto queda de PISO. */
                    const double *ring = (c->rama == AC2_RAMA_SUB) ? c->m_ssub : c
                        ->m_ssob;
                    int nn = (c->m_n < AC2_N_MED) ? c->m_n : AC2_N_MED, q;
                    if (nn >= 2) {
                        double mu = 0.0, va = 0.0, s_ad;
                        for (q = 0; q < nn; q++) mu += ring[q];
                        mu /= (double)nn;
                        for (q = 0; q < nn; q++) va += (ring[q]-mu)*(ring[q]-mu);
                        s_ad = AC2_DEAD_KSIG * sqrt(va / (double)(nn - 1));
                        if (s_ad > s_dead) s_dead = s_ad;
                    }
                }
#endif
```

```
                if (fabs(s_hat) < s_dead) { c->pend_n = 0; c->state = AC2ST_DONE;
                                            c->done = 1; break; }
            }
            /* (aqui iba el criterio de TENDENCIA; eliminado 2026-08-08, ver la
             * nota junto a AC2_MAX_STEPS arriba) */

            paso = ac2_clamp(AC2_GAMMA * s_hat * frac, -AC2_S_RATE, AC2_S_RATE);
            rr_new = ac2_clamp(c->rr_effect * pow(10.0, paso), AC2_RR_MIN,
                AC2_RR_MAX);
            if (apply) { c->rr_effect = rr_new; c->steps++; }
            if (c->last_dir != 0 && rama_red != c->last_dir) c->flips++;
            c->last_dir = rama_red;
            dec = AC2_APPLY;
            c->state = AC2ST_ANALISIS;     /* v26: re-analizar tras aplicar */
```

Listing B.7: Árbitro direccional de arranque: dominio, confianza, firma física, confirmación y escritura acotada.

B.7. Vías de respaldo físicas

Los dos extractos siguientes documentan las vías de respaldo descritas en el Capítulo 5. El coordinador de carga incluye el veto de coherencia, el *rollback* del último candidato cuando aumenta el residuo normalizado y la vía de emergencia con *pacing* y rectificación. La rama del *rollback* entrega una decisión activa, de modo que el candidato restaurado llega al controlador (Sección G.4). La vía de *breakaway* del coordinador de arranque emplea el signo de la mediana de $\hat{d}$ y pasos acotados para abandonar una condición de rotor trancado. Estas rutas son mecanismos físicos independientes de las salidas neuronales.

```
/* Enrutamiento, veto y propuesta de escritura. */
                    int route = K2_RUTA_NONE;
                    /* v4b: la firma solo vale donde el debilitamiento de campo
                     * es posible (ver K2_PSI_WMIN). Fuera de esa banda no dice
                     * "ajustado" ni "sobreestimado": dice NADA, y no veta. */
                    int firma = (w_mean >= K2_PSI_WMIN && psi_ref_n >= K2_PSI_SUB)
                                ? K2_RUTA_SUB : K2_RUTA_NONE;
                    if (sub_on && !sob_on)      { route = K2_RUTA_SUB; }
                    else if (sob_on && !sub_on) { route = K2_RUTA_SOBRE; }
                    else if (sub_on && sob_on)  {
                        route = (med_dh > K2_DHAT_THR) ? K2_RUTA_SUB :
                                ((med_dh < -K2_DHAT_THR) ? K2_RUTA_SOBRE :
                                    K2_RUTA_NONE);
                        if (route == K2_RUTA_NONE) c->motivo = K2_MOT_CONFLICTO;
                    } else c->motivo = K2_MOT_NINGUNA;
                    /* v2: la firma fisica veta la rama incoherente */
                    if (route != K2_RUTA_NONE && firma != K2_RUTA_NONE && firma !=
                        route) {
                        route = K2_RUTA_NONE; c->motivo = K2_MOT_FIRMA;
                    }
                    {
                        double s_hat = (route == K2_RUTA_SUB) ? msu :
                                       ((route == K2_RUTA_SOBRE) ? mso : 0.0);
                        double cf    = (route == K2_RUTA_SUB) ? mcu :
                                       ((route == K2_RUTA_SOBRE) ? mco : 0.0);
                        double est   = rr_eff * pow(10.0, s_hat);
                        double delta = est - rr_eff, ad = fabs(delta);
                        /* v3: deadband MIXTO (mayor entre absoluto y relativo) */
                        double dead_mix = (K2_DEAD_REL*rr_eff > K2_DEAD_OHM)
                                          ? K2_DEAD_REL*rr_eff : K2_DEAD_OHM;
#if K2_DEAD_ADAPT
                        {   /* 3*sigma de la salida CRUDA de la red enrutada sobre
                             * las K2_N_CONFIRM ventanas. En OHM, para comparar con
```

```
                                * 'ad', que tambien esta en ohm. El mixto queda de
                                    PISO. */
                               const double *ring = (route == K2_RUTA_SUB) ? c->b_ssub
                                    : c->b_ssob;
                               int nn = c->buf_n, q;
                               if (nn >= 2) {
                                   double mu = 0.0, va = 0.0, sig, d_ad;
                                   for (q = 0; q < nn; q++) mu += ring[q];
                                   mu /= (double)nn;
                                   for (q = 0; q < nn; q++) va += (ring[q]-mu)*(ring[q
                                       ]-mu);
                                   sig = sqrt(va / (double)(nn - 1));
                                   /* de log-razon a ohm alrededor del punto vigente
                                       */
                                   d_ad = rr_eff * (pow(10.0, K2_DEAD_KSIG*sig) - 1.0)
                                       ;
                                   if (d_ad > dead_mix) dead_mix = d_ad;
                               }
                           }
#endif
                           *s_o = s_hat; *rr_target_o = est; *conf_o = cf;
                           c->route = route;
                           if (route == K2_RUTA_NONE) {
                               c->state=K2ST_DONE; c->rms_done=rms_i; c->tref_done=
                                   atref;
                               decision = K2_IGNORA;
                           } else if (ad > K2_DOMAIN_MAX + K2_OOD_MARGIN) {
                               decision=K2_OUT_OF_DOMAIN; c->ood_flag=1; c->state=
                                   K2ST_DONE;
                               c->rms_done=rms_i; c->tref_done=atref; c->motivo=
                                   K2_MOT_OOD;
                           } else if (delta*(double)route < 0.0) {
                               c->state=K2ST_DONE; c->rms_done=rms_i; c->tref_done=
                                   atref;
                               c->motivo=K2_MOT_INCOHER;
                           } else if (c->last_dir != 0 && route != c->last_dir) {
                               c->state=K2ST_DONE; c->rms_done=rms_i; c->tref_done=
                                   atref;
                               c->motivo=K2_MOT_MONOTONIA;
                           } else if (cf >= K2_CONF_MIN && ad >= dead_mix && ad <=
                               K2_DOMAIN_MAX) {
                               double paso = k2_clamp(K2_GAMMA*s_hat, -K2_S_RATE,
                                   K2_S_RATE);
                               double rr_new = k2_clamp(rr_eff*pow(10.0,paso),
                                   K2_RR_MIN, K2_RR_MAX);
                               c->rr_prev = rr_eff;
                               /* guardian v2: se guarda el RESIDUO previo, no el
                                   e_reg */
                               c->eps_pre = (c->post_n>0)? c->eps_post_sum/(double)c->
                                   post_n : eps_n;
                               c->rms_pre = (c->post_n>0)? c->rms_post_sum/(double)c->
                                   post_n : rms_i;
                               c->cand = rr_new;
                               if (apply) c->rr_effect = rr_new;
                               c->committed=1; c->n_commits++; c->last_dir = route;
                               decision=K2_ACTIVE; c->motivo=K2_MOT_COMITEO;
                               c->state=K2ST_APPLY; c->settle=0;
                               c->eps_post_sum=0; c->rms_post_sum=0; c->post_n=0;
                           } else {
                               /* se distingue POR QUE no se aplico */
                               c->motivo = (ad < dead_mix)         ? K2_MOT_DEADBAND :
                                           ((ad > K2_DOMAIN_MAX)   ? K2_MOT_DOMINIO  :
                                                                     K2_MOT_CONF);
                               c->state=K2ST_DONE; c->rms_done=rms_i; c->tref_done=
                                   atref;
                           }

/* Guardián: comparación del residuo antes y después. */
```

```
83              case K2ST_APPLY:
84                  if (rms_i > c->baseline_loaded*(1.0+K2_LOAD_RISE_FRAC) ||
85                      atref > c->tref_loaded*(1.0+K2_TREF_RISE_FRAC) + 0.02) {
86                      k2_new_event(c, 3); break;   /* otro escalon: el veto quedaria
                            sucio */
87                  }
88                  /* v3e: SEGUNDA copia de la prueba de descarga. Necesita el mismo
89                   * `step_detected` que la de STEADY: sin el, la propia mejora que
90                   * produce el commit baja la corriente y se lee como descarga, y
91                   * el coordinador engancha DONE al ciclo siguiente de comitear
92                   * (visto en traza_v3.csv: APLICA en t=2,201 -> DONE en t=2,211
93                   * con n_commits=1 de 3). */
94                  if (c->step_detected && c->have_baseline &&
95                      rms_i < c->baseline*(1.0+K2_LOAD_RISE_FRAC) &&
96                      atref < c->tref_baseline*(1.0+K2_TREF_RISE_FRAC) + 0.02 &&
97                      fabs(wreg) < 0.02) { c->state=K2ST_DONE; c->motivo=
                            K2_MOT_DESCARGA;
98                      c->rms_done=rms_i; c->tref_done=atref; break; }
99                  c->settle++;
100                 if (c->settle > K2_N_SETTLE/2 && a < K2_ACCEL_SETTLE) {
101                     c->eps_post_sum += eps_n; c->rms_post_sum += rms_i; c->post_n
                            ++;
102                 }
103                 if (c->settle >= K2_N_SETTLE) {
104                     double eps_post = (c->post_n>0)? c->eps_post_sum/(double)c->
                            post_n : eps_n;
105                     double rms_post = (c->post_n>0)? c->rms_post_sum/(double)c->
                            post_n : rms_i;
106                     /* ---- GUARDIAN v2: ?mejoro el RESIDUO? Solo vale si la carga
107                      * no cambio entre el antes y el despues y el PI no satura.
                             ---- */
108                     int valido = (c->rms_pre > 1e-6) &&
109                                  (fabs(rms_post - c->rms_pre) < K2_LOAD_TOL*c->
                                      rms_pre) &&
110                                  (atref < 1.9);
111                     if (apply && valido && eps_post > c->eps_pre*(1.0+
                            K2_GUARD_MARGIN)) {
112                         c->rr_effect = c->rr_prev; c->cand = c->rr_prev; c->
                                rolled_back = 1;
113                         c->motivo = K2_MOT_REVERTIDO;
114                         decision = K2_ACTIVE; /* la reversion llega al controlador */
115                         c->rr_visto = c->rr_prev; /* no se lee como correccion ajena */
116                         c->state = K2ST_DONE; c->rms_done=rms_i; c->tref_done=atref
                                ;
117                     } else if (c->n_commits >= K2_MAX_COMMITS) {
118                         c->motivo = K2_MOT_MAXCOMMIT;
119                         c->state = K2ST_DONE; c->rms_done=rms_i; c->tref_done=atref
                                ;
120                     } else {
121                         c->baseline_loaded = rms_i; c->tref_loaded = atref;
122                         c->buf_n = 0; c->settle = 0;
123                         c->eps_post_sum=0; c->rms_post_sum=0; c->post_n=0;
124                         c->state = K2ST_STEADY;
125                     }
126                 }
127                 break;
128
129 /* Vía física de emergencia con pacing y rectificación. */
130             /* ==== EMERGENCIA (v23 + pacing y rectificacion de v24): intacta ====
                    */
131             case K2ST_EMERGENCY:
132                 /* ================== v4: LAS REDES TIENEN PRIORIDAD
                        ================== */
133                  * La via de emergencia existe para devolver el modelo al dominio
134                  * donde los expertos pueden pronunciarse, NO para acertar el valor
135                  * --el mismo papel que cumple el breakaway en el arranque--. Por
                        eso
136                  * cede en cuanto una red reclama dominio con confianza suficiente.
```

```
137                  *
138                  * Sin esta salida, la emergencia se sostenia sola: su unica
                         condicion
139                  * de salida era que la regulacion mejorase (wreg < 0,04), pero al
140                  * zarandear Rr impedia que la maquina regulara, de modo que la
141                  * condicion nunca se cumplia. Medido en la rutina PF: 79 % del
                         tiempo
142                  * en EMERGENCIA, 27 pasos fisicos con 10 cambios de sentido, y la
                         red
143                  * de carga con CERO commits pese a proponer con conf 0,88 y dom
                         1,00.
144                  * ========================================== */
145                 /* v4b: SE ENTRA DE DOS MANERAS, SE SALE DE DOS MANERAS.
146                  * Ceder no debe significar mandar a la cola. `k2_new_event` deja
147                  * la FSM en TRANSIENT con trans = 0, asi que el experto que acaba
148                  * de reclamar debe esperar K2_N_TRANS ciclos MAS los K2_N_CONFIRM
149                  * de confirmacion. Eso es correcto si la emergencia movio Rr --hay
150                  * un transitorio fisico real que esperar, y medir sobre el daria
151                  * una estimacion sucia-- y es puro retraso si no aplico ni un paso
                         .
152                  * Mismo patron que `brk_modo` en el coordinador de arranque.
153                  * La via 10 marca en la traza la cesion sin transitorio. */
154                 if ((d_sub >= K2_DOM_MIN && c_sub >= K2_CONF_MIN) ||
155                     (d_sob >= K2_DOM_MIN && c_sob >= K2_CONF_MIN)) {
156                     c->emg_wreg_n = 0;
157                     if (c->emg_applied > 0.0) { k2_new_event(c, 5); break; }
158                     /* no toco Rr: se cede al analisis con linea base fresca. Se
159                      * conservan las exigencias normales --K2_N_CONFIRM ciclos de
160                      * buffer y el resto de los veedores--, solo se ahorra la
161                      * espera de un transitorio que no existe. */
162                     c->state = K2ST_STEADY;
163                     c->baseline_loaded = rms_i; c->tref_loaded = atref;
164                     c->buf_n = 0; c->n_commits = 0; c->last_dir = 0;
165                     c->route = K2_RUTA_NONE; c->step_detected = 1;
166                     c->eps_post_sum = 0; c->rms_post_sum = 0; c->post_n = 0;
167                     c->via = 10;
168                     break;
169                 }
170                 if (wreg < K2_EMG_WREG_OFF) { c->emg_wreg_n = 0; k2_new_event(c, 6)
                        ; break; }
171                 if (wreg > K2_EMG_WREG_ON && fabs(med_edh) >= K2_EMG_DHAT_MIN &&
172                     ((med_edh > 0.0) ? +1 : -1) != c->emg_dir) c->emg_flip_n++;
173                 else c->emg_flip_n = 0;
174                 if (c->emg_flip_n >= K2_EMG_N_ON) {
175                     c->emg_flips++; c->emg_flip_n = 0;
176                     if (c->emg_flips > K2_EMG_MAX_FLIPS) {
177                         c->state = K2ST_DONE; c->rms_done=rms_i; c->tref_done=atref
                                ; break;
178                     }
179                     c->emg_dir = -c->emg_dir; c->emg_applied = 0.0;
180                     c->emg_cyc = K2_EMG_PERIOD; c->emg_wreg_last = wreg;
181                 }
182                 c->emg_cyc++;
183                 if (c->emg_cyc >= K2_EMG_PERIOD) {
184                     int improving;
185                     c->emg_cyc = 0;
186                     improving = (wreg < c->emg_wreg_last * (1.0 -
                            K2_EMG_IMPROVE_TOL));
187                     {
188                     /* v4: paso RELATIVO al Rr vigente, con piso en la resolucion
189                      * del estimador. El presupuesto por episodio sigue siendo
190                      * absoluto y acotando el recorrido total. */
191                     /* v4 (corregido): relativo a rr_BASE, no a rr_eff. Referirlo
                            al
192                      * valor VIGENTE realimenta positivamente --la emergencia infla
                            Rr,
193                      * el paso siguiente crece, infla mas--: medido, los pasos
                            llegaron
```

```
                     * a 0,654 ohm, MAS grandes que los 0,4 fijos que reemplazaron.
                     * rr_base es el valor que puso el usuario y no se mueve. */
                    /* v4b: contra el MENOR entre el vigente y el base. Referirlo
                     * solo a rr_base lo deja escalando con el ERROR INICIAL: tras
                     * converger a 0,856 desde un base de 2,33, el paso seguia
                     * valiendo 0,25*2,33 = 0,582 ohm --el 68 % del valor ya
                     * corregido--, y un solo disparo lo mandaba a 0,274 (medido).
                     * Con el minimo: al SUBIR queda anclado al menor, que es lo
                     * que evita la realimentacion positiva ya medida (pasos de
                     * hasta 0,654 ohm cuando se referia a rr_eff); al BAJAR se
                     * encoge con rr_eff, de modo que no puede destruir una
                     * convergencia. Cuando rr_eff >= rr_base es identico al
                     * anterior: el cambio SOLO afecta a la bajada, que es donde
                     * se midio el dano. */
                    double rr_anc = (rr_eff < c->rr_base) ? rr_eff : c->rr_base;
                    double emg_step = K2_EMG_STEP_REL * rr_anc;
                    if (emg_step < K2_EMG_STEP_MIN) emg_step = K2_EMG_STEP_MIN;
                    if (!improving && !ref_jump && wreg > K2_EMG_WREG_ON &&
                        fabs(med_edh) >= K2_EMG_DHAT_MIN &&
                        ((med_edh > 0.0) ? +1 : -1) == c->emg_dir &&
                        c->emg_applied + emg_step <= K2_EMG_BUDGET + 1e-9) {
                        double rr_new = k2_clamp(rr_eff + (double)c->emg_dir *
                            emg_step,
                                                 K2_RR_MIN, K2_RR_MAX);
                        c->cand = rr_new;
                        if (apply) c->rr_effect = rr_new;
                        c->emg_applied += emg_step; c->emg_steps++;
                        c->committed = 1;             /* el arranque cede la pluma
                            */
                        decision = K2_ACTIVE;
                    }
                    }   /* v4: cierra el bloque del paso relativo */
                    c->emg_wreg_last = wreg;
                }
                break;
```

Listing B.8: Coordinador de carga: veto, rollback por residuo y vía física de emergencia.

```
        case AC2ST_MAGNETIZED:
            /* v26 intacto: magnetizado y sin arrancar. En vacio dura pocos
             * ciclos; un CALADO cargado persiste -> breakaway. */
            if (startup_active) { c->state = AC2ST_ANALISIS; c->stall_n = 0; c->
                stall_t = 0.0; }
            else {
                c->stall_n++;
                c->stall_t += dt_cyc;          /* v4b: y tambien en TIEMPO */
                /* v4b: dos caminos al mismo disparo. El de ciclos es el de v26;
                 * el de tiempo solo puede ADELANTARLO, y exige `stall_n >= 1`,
                 * es decir UN CICLO ELECTRICO COMPLETO ya cerrado: por lento que
                 * sea el ciclo, nunca se actua dentro de uno. La condicion sobre
                 * el residuo es comun a los dos y no se afloja. */
                if ((c->stall_n >= AC2_STALL_N ||
                     (c->stall_n >= 1 && c->stall_t >= AC2_STALL_T)) &&
                    fabs(med_dh) >= AC2_BRK_DEADBAND) {
                    c->state = AC2ST_BREAKAWAY;
                    c->brk_modo = 0;                  /* v3h: modo CALADO (v26) */
                    c->brk_cyc = AC2_BRK_PERIOD;      /* primer paso pronto */
                }
            }
            break;

        case AC2ST_BREAKAWAY:
            /* v26 intacto: paso FISICO fijo en el signo del residuo. La magnitud
             * de la red no sirve aqui (fuera de distribucion en calado); el
             * signo si. Freno natural: |d_hat| -> 0 al acercarse a Rr_MAS. */
            /* v3h: la salida depende del MODO. Del calado se sale en cuanto la
             * maquina arranca; del estancado, solo cuando de verdad ALCANZA la
             * referencia (stage 2), porque se entro con startup_active ya en 1. */
```

```
            if (c->brk_modo == 0) {
                if (startup_active) { c->state = AC2ST_ANALISIS; c->stall_n = 0;
                                      c->stall_t = 0.0; break; }
            } else {
                if (stage >= AC2_STAGE_MAX || startup_ended) {
                    c->state = AC2ST_ANALISIS; c->prog_n = 0; c->brk_modo = 0;
                        break;
                }
            }
            if (fabs(med_dh) < AC2_BRK_DEADBAND) break;            /* HOLD */
            c->brk_cyc++;
            if (c->brk_cyc >= AC2_BRK_PERIOD) {
                int dir_nuevo = (med_dh > 0.0) ? +1 : -1;
                c->brk_cyc = 0;
                /* v3i: BISECCION AL CAMBIAR DE SENTIDO. v26 confiaba en que
                 * |d_hat| -> 0 al acercarse a Rr_MAS, y en la rama SOBREestimada
                 * (rho2 ~ 0,98) eso se cumple. En la SUBestimada no: alli la
                 * magnitud del residuo es poco fiable y no se anula, asi que el
                 * paso fijo hacia pasar de largo y volver, con una banda medida
                 * de 4,0 ohm. Un cambio de sentido significa que ya se ENCERRO
                 * el valor: se parte el paso al medio. */
                if (c->brk_dir != 0 && dir_nuevo != c->brk_dir) {
                    c->brk_flips++;
                    c->brk_step *= 0.5;
                    if (c->brk_step < AC2_BRK_STEP_MIN)
                        c->brk_step = AC2_BRK_STEP_MIN;
                    /* Tras varios cruces ya estamos DENTRO del dominio util de
                     * las redes: ese es el objetivo del breakaway, no acertar
                     * el valor. Se cede y que afinen ellas. */
                    if (c->brk_flips >= AC2_BRK_MAX_FLIPS) {
                        c->state = AC2ST_ANALISIS; c->brk_modo = 0;
                        c->prog_n = 0; c->stall_n = 0; c->stall_t = 0.0;
                        break;
                    }
                }
                c->brk_dir = dir_nuevo;
                rr_new = ac2_clamp(c->rr_effect + (double)c->brk_dir * c->brk_step,
                                   AC2_RR_MIN, AC2_RR_MAX);
                if (apply) { c->rr_effect = rr_new; c->brk_steps++; }
                dec = AC2_BRK;
            }
            break;
```

Listing B.9: Vía física de breakaway: detección de rotor trancado y aproximación acotada por el signo de la mediana de d_hat.

Apéndice C

Implementación en Python: Núcleo Físico y Redes

Este apéndice reproduce las piezas centrales de la implementación en Python de la Propuesta 1: el predictor $\alpha\beta$ de un paso, su sensibilidad analítica y la proyección $\hat{d}$ empleada para construir observables físicos; la arquitectura compartida por los cuatro expertos direccionales; y la pérdida híbrida utilizada durante el entrenamiento. La física no se introduce minimizando directamente el residuo de corriente, cuyo mínimo presentó un sesgo sistemático, sino mediante ρ^2 como ponderación de la restricción direccional y mediante las variables físicas incorporadas al vector de entrada. El pipeline completo —generación de escenarios por RPC, extracción de ciclos, etiquetado, entrenamiento, métricas y exportación a C— permanece en el repositorio del proyecto.

C.1. Núcleo físico compartido

```
def predict_alphabeta(i_a, i_b, v_a, v_b, psir_a, psir_b, w_el, Rr_eff, m):
    kr   = m.Lm / m.Lr
    sig  = 1.0 - (m.Lm*m.Lm)/(m.Ls*m.Lr); Lsig = sig*m.Ls
    Rsig = m.Rs + kr*kr*Rr_eff;           tau_r = m.Lr/Rr_eff
    a = 1.0 - m.Ts/(Lsig/Rsig); b = m.Ts/Lsig
    i_a_pred = a*i_a + b*(v_a + kr*(psir_a/tau_r + w_el*psir_b))
    i_b_pred = a*i_b + b*(v_b + kr*(psir_b/tau_r - w_el*psir_a))
    return i_a_pred, i_b_pred

def sensitivity_alphabeta(i_a, i_b, psir_a, psir_b, Rr_eff, m):  # phi = d(i_pred)/
    dRr_eff
    kr = m.Lm/m.Lr; sig = 1.0-(m.Lm*m.Lm)/(m.Ls*m.Lr); Lsig = sig*m.Ls
    c = m.Ts/Lsig
    return c*(-kr*kr*i_a + (kr/m.Lr)*psir_a), c*(-kr*kr*i_b + (kr/m.Lr)*psir_b)

def project_error(eps_a, eps_b, phi_a, phi_b):     # <eps,phi> / <phi,phi> ~ (
    Rr_MAS - Rr_eff)
    num = eps_a*phi_a + eps_b*phi_b
    den = phi_a*phi_a + phi_b*phi_b
    return num/(den + 1e-12)
```

Listing C.1: Núcleo físico: predictor $\alpha\beta$ a un paso, sensibilidad analítica y proyección $\hat{d}$ (`src/physics.py`).

C.2. Arquitectura compartida de los expertos

```python
class ExpertoV2(nn.Module):
    """MLP tanh con tres cabezas. La primera sale acotada en log-razon."""

    def __init__(self, n_in, s_out=1.0, hidden=(64, 64)):
        super().__init__()
        capas, d = [], n_in
        for h in hidden:
            lin = nn.Linear(d, h)
            nn.init.xavier_uniform_(lin.weight)
            nn.init.zeros_(lin.bias)
            capas += [lin, nn.Tanh()]
            d = h
        self.cuerpo = nn.Sequential(*capas)
        self.h_s = nn.Linear(d, 1)
        self.h_conf = nn.Linear(d, 1)
        self.h_dom = nn.Linear(d, 1)
        for h in (self.h_s, self.h_conf, self.h_dom):
            nn.init.xavier_uniform_(h.weight)
            nn.init.zeros_(h.bias)
        self.s_out = s_out

    def forward(self, x):
        z = self.cuerpo(x)
        s = self.s_out * torch.tanh(self.h_s(z)).squeeze(-1)
        conf = torch.sigmoid(self.h_conf(z)).squeeze(-1)
        dom = torch.sigmoid(self.h_dom(z)).squeeze(-1)
        return s, conf, dom
```

Listing C.2: Arquitectura MLP compartida por los cuatro expertos direccionales.

Las cuatro redes emplean dos capas ocultas de 64 neuronas con activación hiperbólica y tres cabezas de salida: corrección logarítmica s, confianza y pertenencia al dominio. Los expertos de arranque utilizan una cota de salida $S_{\text{out}} = 1{,}6$, mientras que los de carga utilizan $S_{\text{out}} = 0{,}7$; estas constantes se verifican en los encabezados exportados que acompañan al modelo PLECS.

C.3. Pérdida híbrida compartida

```python
def perdida(s, conf, dom, *, s_star, act, dom_t, matched, fuera, rho2, peso,
            w_data=1.0, w_dom=0.5, w_conf=0.1, w_dir=0.2, w_quiet=1.5,
            w_dom0=3.0, w_conf0=0.5, conf_esc=0.05, pos_w=1.0):
    n_act = act.sum().clamp_min(1.0)
    n_p = peso.sum().clamp_min(1.0)

    # datos: MSE en log-razon, solo donde le toca actuar
    data = (((s - s_star) ** 2) * act * peso).sum() / (act * peso).sum().clamp_min
        (1.0)

    # dominio: BCE sobre "esta es mi maniobra".
    # pos_w > 1 pesa la clase POSITIVA (dom_t = 1). Existe porque el empuje neto
    # hacia 0 es fuerte: w_dom0 = 3,0 sobre 'fuera' contra w_dom = 0,5 aqui.
    # OJO al usarlo: se midio que los fallos de recall que quedan estan
    # concentrados en VACIO (100 % en bajar, 94 % en subir), que es justo donde
    # el residuo no informa. Subir pos_w empuja a reclamar dominio ahi.
    bce = F.binary_cross_entropy(dom.clamp(1e-6, 1 - 1e-6), dom_t, reduction="none"
        )
    w_cls = 1.0 + (pos_w - 1.0) * dom_t
    l_dom = (bce * peso * w_cls).sum() / (peso * w_cls).sum().clamp_min(1.0)

    # confianza: autocalibrada contra el error propio
    with torch.no_grad():
```

```
        conf_t = torch.exp(-(s - s_star).abs() / conf_esc)
    l_conf = ((conf - conf_t) ** 2 * act).sum() / n_act

    # SIGNO fisico: penaliza corregir en sentido opuesto al del residuo,
    # ponderado por rho2 (donde el residuo no informa, no restringe)
    l_dir = (torch.relu(-s * torch.sign(s_star)) * act * rho2).sum() / n_act

    # no perturbar el caso ajustado
    l_quiet = ((s ** 2) * matched * peso).sum() / (matched * peso).sum().clamp_min
        (1.0)

    # callarse fuera de dominio: ajustado U rama equivocada U OOD U region ciega
    n_f = (fuera * peso).sum().clamp_min(1.0)
    l_dom0 = ((dom ** 2) * fuera * peso).sum() / n_f
    l_conf0 = ((conf ** 2) * fuera * peso).sum() / n_f

    total = (w_data * data + w_dom * l_dom + w_conf * l_conf + w_dir * l_dir +
             w_quiet * l_quiet + w_dom0 * l_dom0 + w_conf0 * l_conf0)
    return total, {"data": float(data), "dom": float(l_dom), "dir": float(l_dir),
                   "quiet": float(l_quiet), "dom0": float(l_dom0)}
```

Listing C.3: Pérdida híbrida compartida: ajuste supervisado, dominio, confianza, dirección y protección del caso ajustado.

Los pesos de la firma de la función son los valores por omisión del programa. Las redes desplegadas se entrenaron con $w_{\mathrm{dom0}} = 48$ las de arranque y 24 las de carga, pasados por línea de órdenes (Tabla 5.8).

El término direccional compara el signo de la salida s con el de la etiqueta supervisada $s^\star$ y se pondera mediante ρ^2. Por tanto, actúa como una penalización tipo bisagra que se atenúa cuando el residuo posee baja capacidad explicativa; no corresponde a un error cuadrático ni utiliza directamente $\mathrm{sign}(\hat{d})$ como etiqueta.

C.4. Flujo de reproducción

Para reproducir la Propuesta 1 sobre la plataforma y el espacio paramétrico estudiados, el orden de ejecución es el siguiente:

1. **Generación de escenarios** (requiere PLECS con el servidor XML-RPC activo): la rejilla de pares $(R_{r,\mathrm{MPC}}, R_{r,\mathrm{MAS}})$, niveles de carga y rutinas de maniobra se simula de forma automatizada; cada escenario produce un registro CSV con la totalidad de las salidas del C-Script.

2. **Construcción del conjunto de datos**: detección de ciclos por *trigger*, extracción de características, etiquetado por regímenes y partición por corrida completa, con las validaciones automáticas de variables prohibidas y rangos físicos.

3. **Entrenamiento** de cada red con su función de pérdida híbrida, y evaluación con las métricas por zona.

4. **Exportación a C**: generación de los archivos de cabecera con pesos y normalización, y verificación de paridad numérica red por red.

5. **Integración en PLECS**: incorporación de los encabezados y coordinadores en el C-Script, verificación de paridad de las inferencias y batería de regresión en modo diagnóstico antes de habilitar la escritura de $R^{\mathrm{eff}}_{r,\mathrm{MPC}}$. La paridad de las inferencias

se verifica sobre vectores ya preparados; la del vector de entrada completo, entre el extractor de entrenamiento y el `C-Script`, se midió aparte y no es exacta (Sección G.1).

6. **Evaluación de desempeño**: generación automatizada de escenarios sobre el modelo de la campaña: rutinas de bomba y cinta, barrido de valores de $R_{r,\mathrm{MAS}}$ y modos de operación, y cálculo reproducible de los indicadores desde los registros de simulación.

Carpeta reproducible. Los archivos necesarios para identificar y ejecutar la configuración final de las dos propuestas se entregan aparte, con una guía de ejecución: la carpeta `entrega_reproducible` y su archivo `LEEME.md`. Contiene los modelos de PLECS con que se obtuvieron las tablas del Capítulo 7, las cabeceras con los pesos y los coordinadores, los pesos entrenados y su configuración, los programas de entrenamiento, de campaña, de cálculo de indicadores y de figuras, y los registros de las comprobaciones del Anexo G. La copia del modelo con la variante de la Sección G.2 se incluye marcada como no evaluada en la campaña. La guía identifica las versiones de las herramientas —PLECS 5.0.3, Python 3.13 con PyTorch 2.11, R 4.6.1— y un caso pequeño por propuesta, con su resultado esperado y un programa que lo ejecuta y lo compara. No incluye los registros completos de simulación, de varios gigabytes.

Apéndice D

Figuras Complementarias de la Campaña de Evaluación

El Capítulo 7 presenta, para cada bloque de la campaña, los indicadores que responden directamente a la pregunta de si el accionamiento recupera el punto de operación solicitado: el residuo de la resistencia de rotor y el error de seguimiento de velocidad. Este anexo reúne los indicadores restantes —distorsión armónica de corriente, error de estimación de torque y de flujo, y sus respectivos *ripple*— para los mismos escenarios, de modo que la evidencia quede completa sin recargar la lectura del capítulo.

Las figuras conservan las convenciones descritas en la Subsección 7.1.1: eje horizontal con la resistencia de rotor del modelo en escala logarítmica, línea vertical discontinua en el valor real de la máquina, columnas por carga de arranque, filas por nivel del escalón, y una curva por modo de ablación. En todos los indicadores, menos es mejor.

D.1. Rutina completa por resistencia real de rotor

Complemento de la Subsección 7.1.3, a velocidad nominal.

D.1.1. Máquina de 0,10 Ω

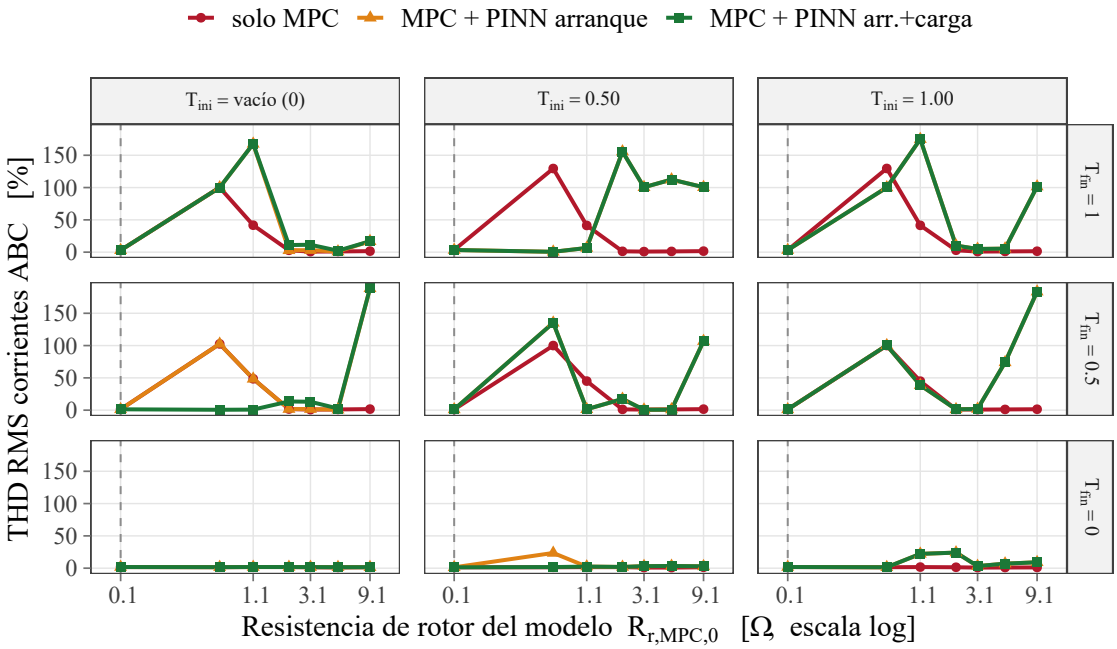


Figura D.1: Distorsión armónica de las corrientes de estator. Máquina de 0,10 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

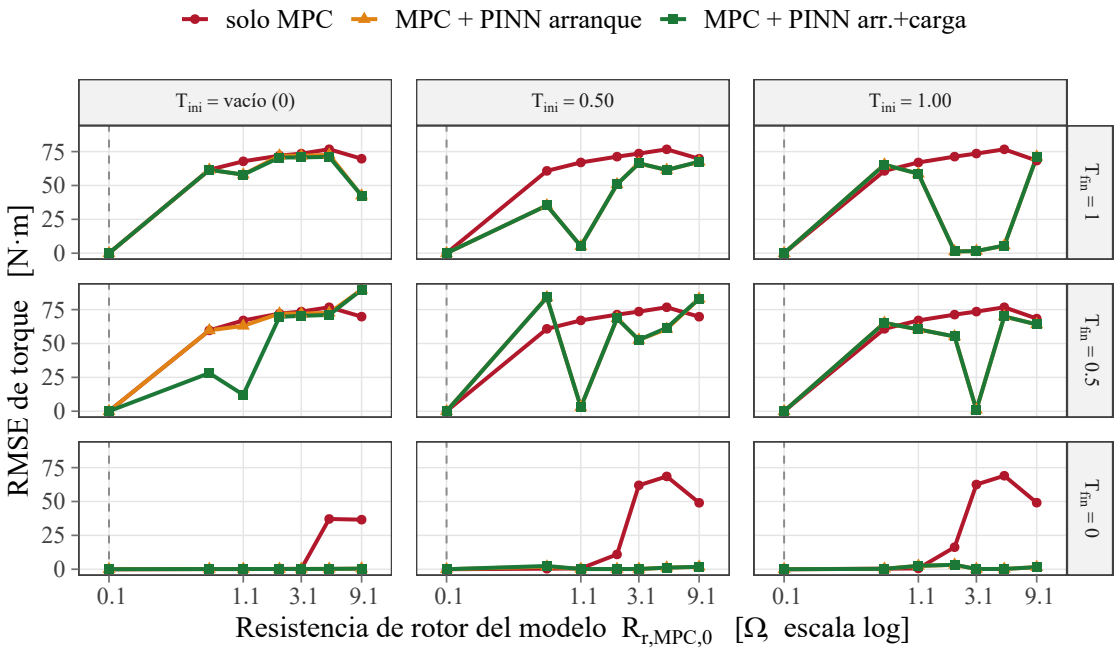


Figura D.2: Error de estimación de torque: discrepancia entre el torque real y el que estima el observador del controlador. Máquina de 0,10 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

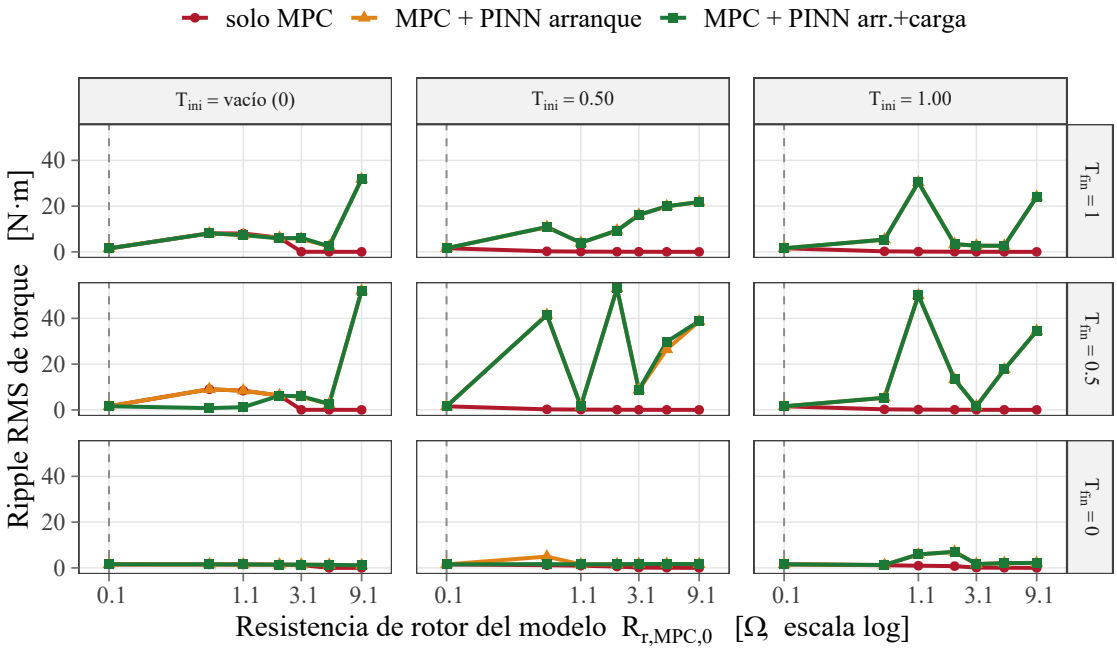


Figura D.3: *Ripple* RMS del torque entregado por la máquina. Máquina de 0,10 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

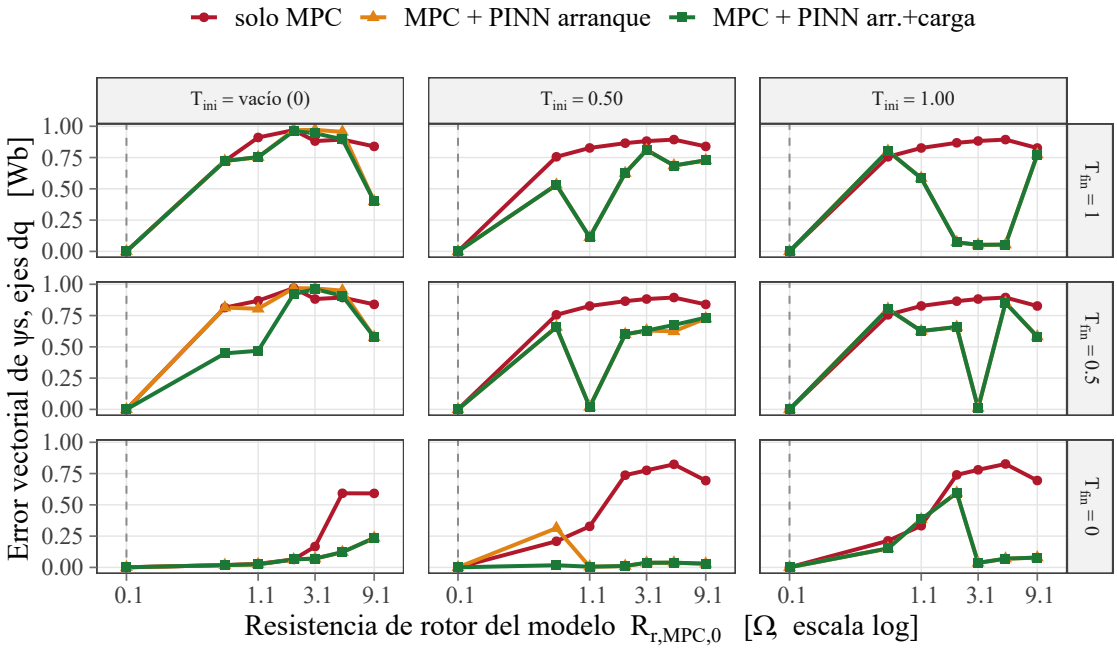


Figura D.4: Error vectorial de estimación del flujo de estator, en ejes *dq* (Tabla 7.1). Máquina de 0,10 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

D.1.2. Máquina de 0,50 Ω

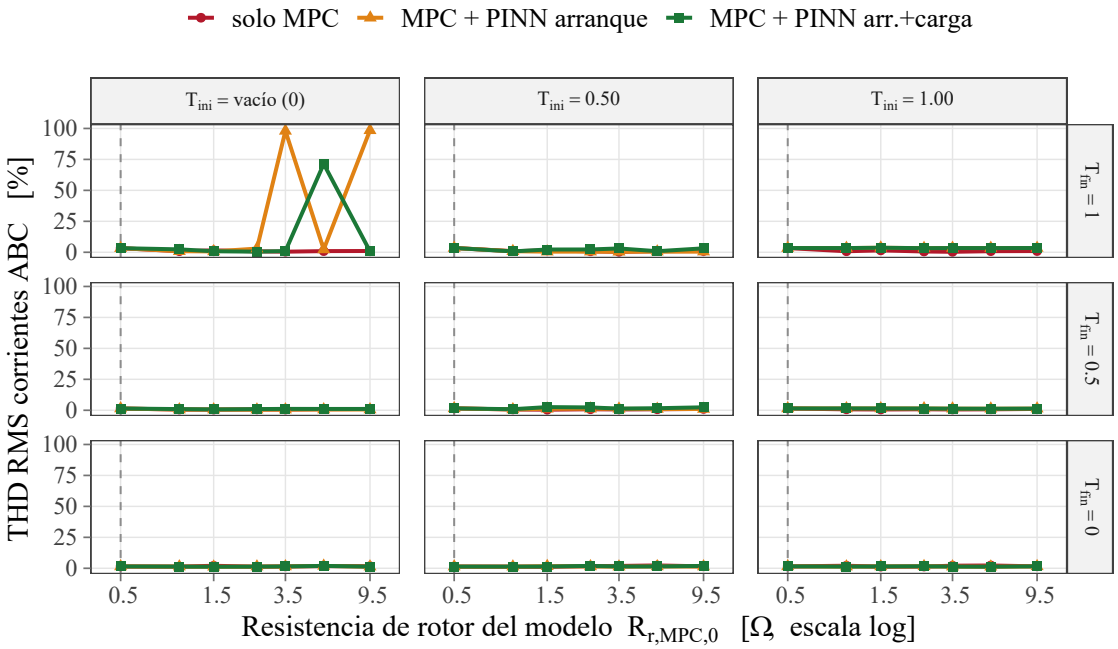


Figura D.5: Distorsión armónica de las corrientes de estator. Máquina de 0,50 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

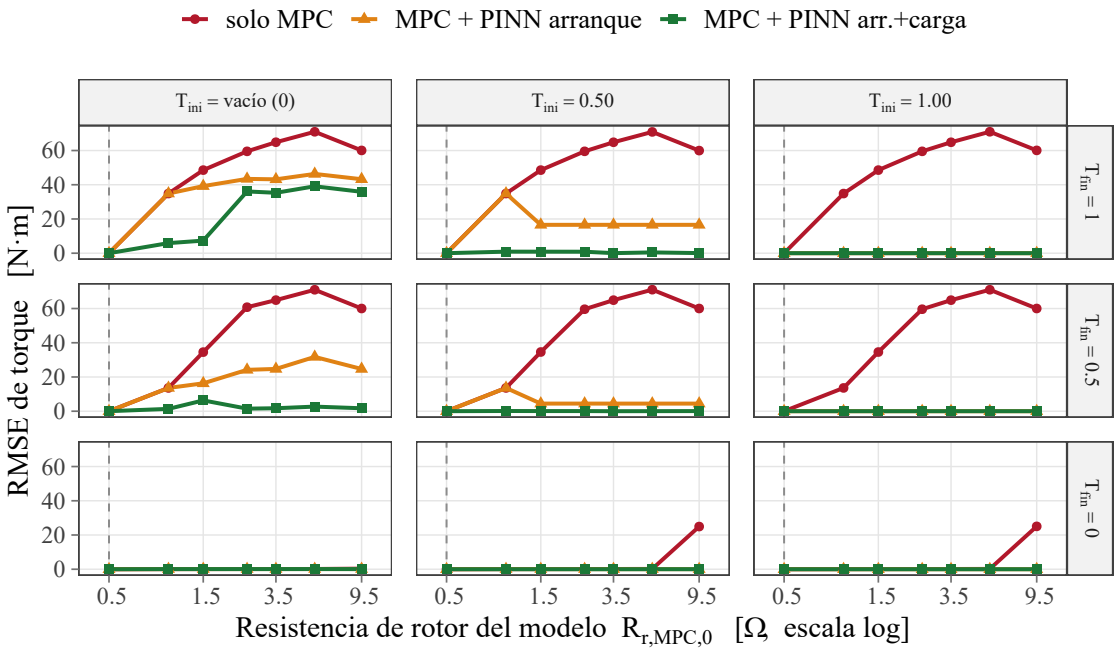


Figura D.6: Error de estimación de torque: discrepancia entre el torque real y el que estima el observador del controlador. Máquina de 0,50 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

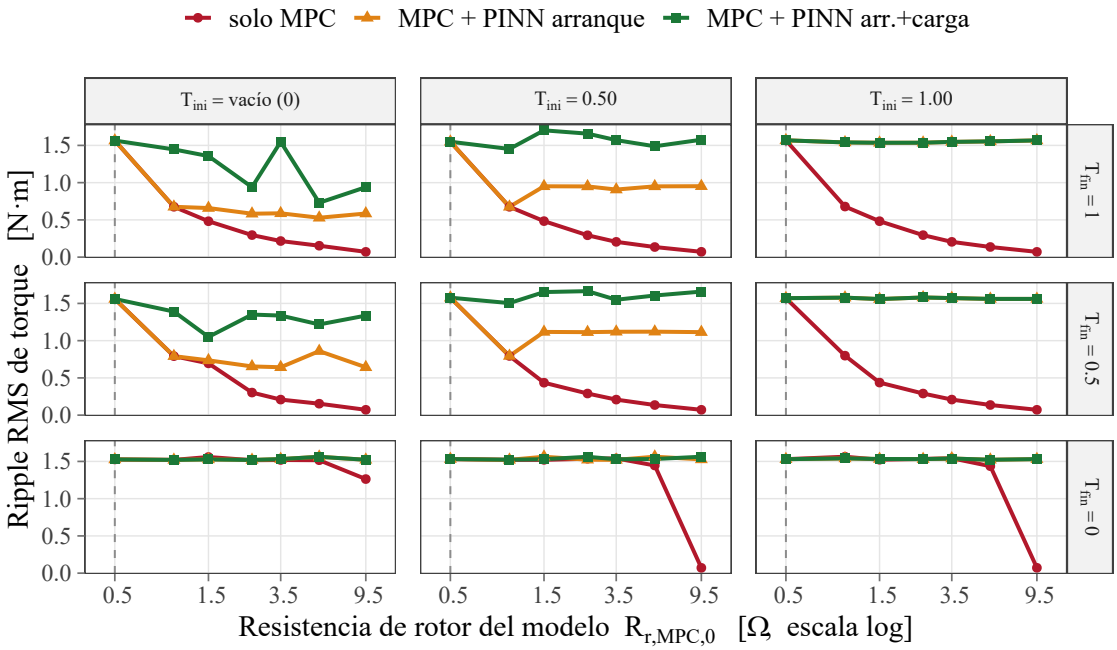


Figura D.7: *Ripple* RMS del torque entregado por la máquina. Máquina de 0,50 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

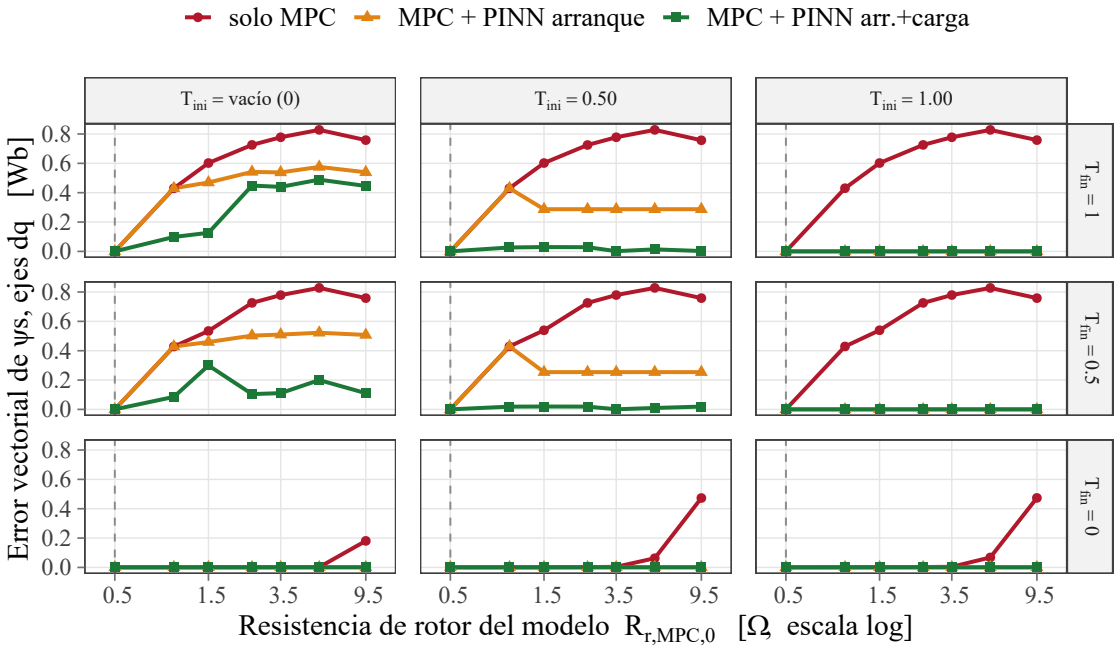


Figura D.8: Error vectorial de estimación del flujo de estator, en ejes *dq* (Tabla 7.1). Máquina de 0,50 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

D.1.3. Máquina de 3,00 Ω

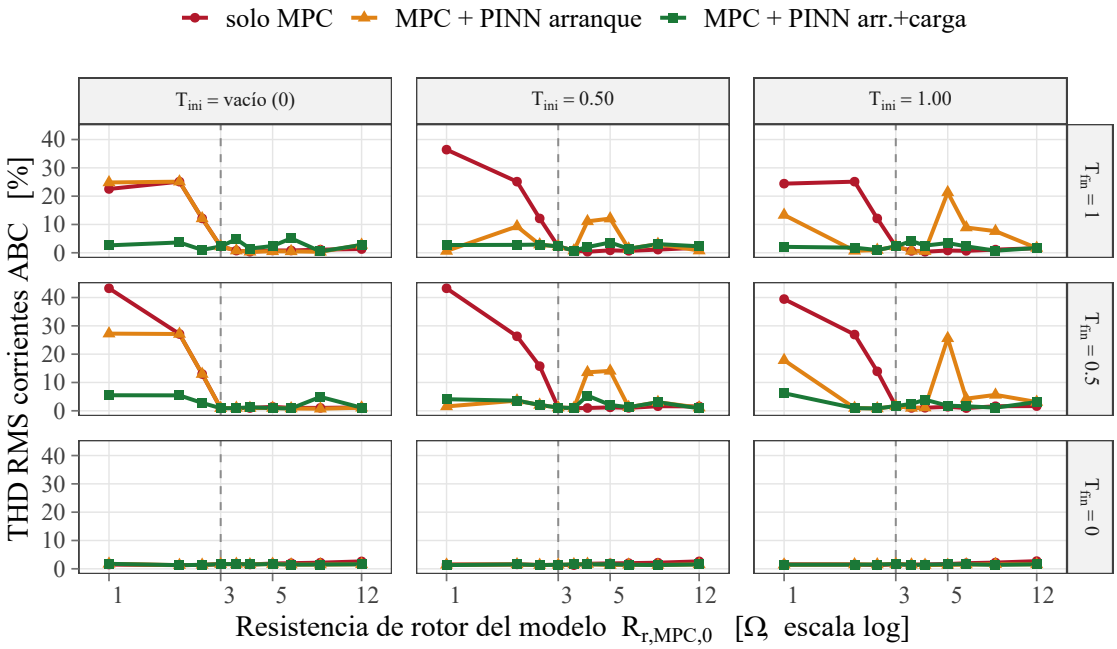


Figura D.9: Distorsión armónica de las corrientes de estator. Máquina de 3,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

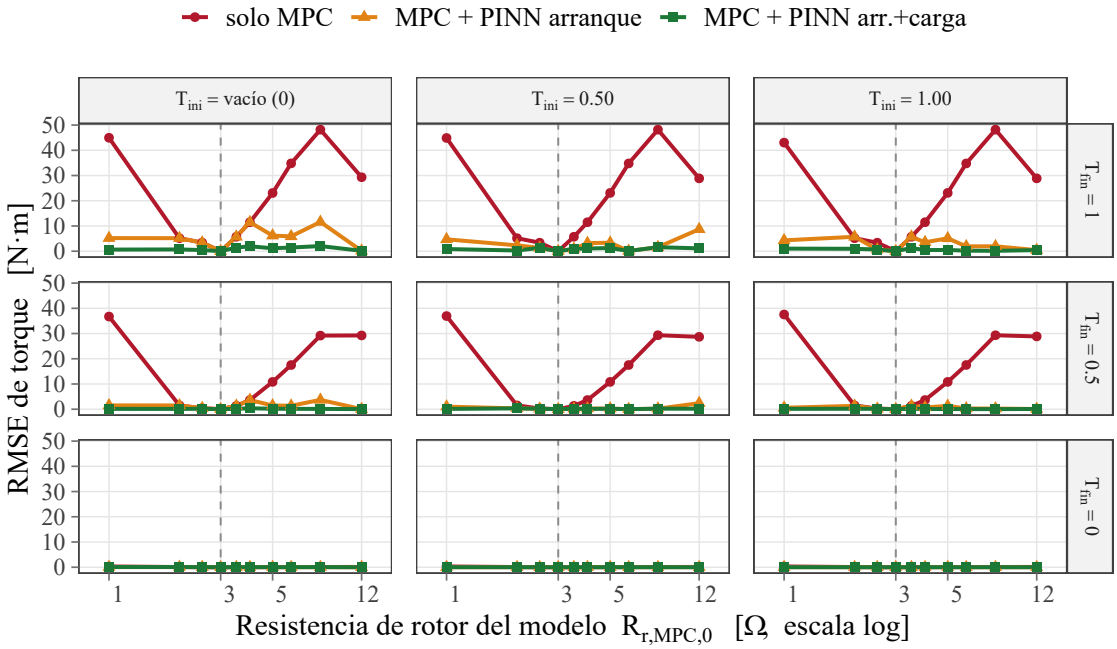


Figura D.10: Error de estimación de torque: discrepancia entre el torque real y el que estima el observador del controlador. Máquina de 3,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

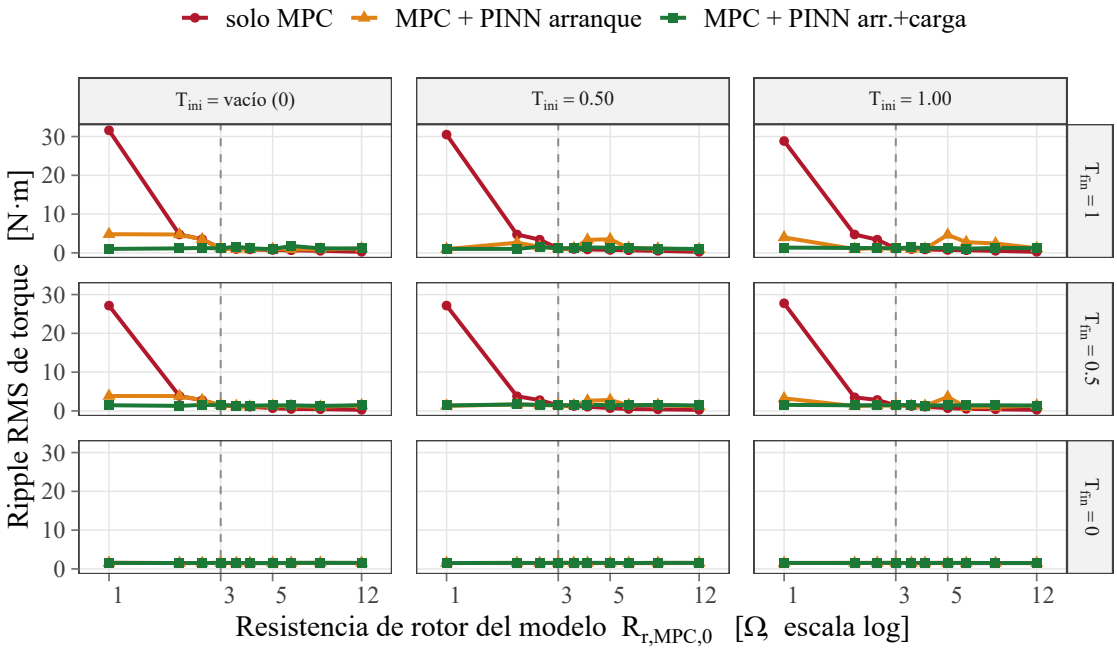


Figura D.11: *Ripple* RMS del torque entregado por la máquina. Máquina de 3,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

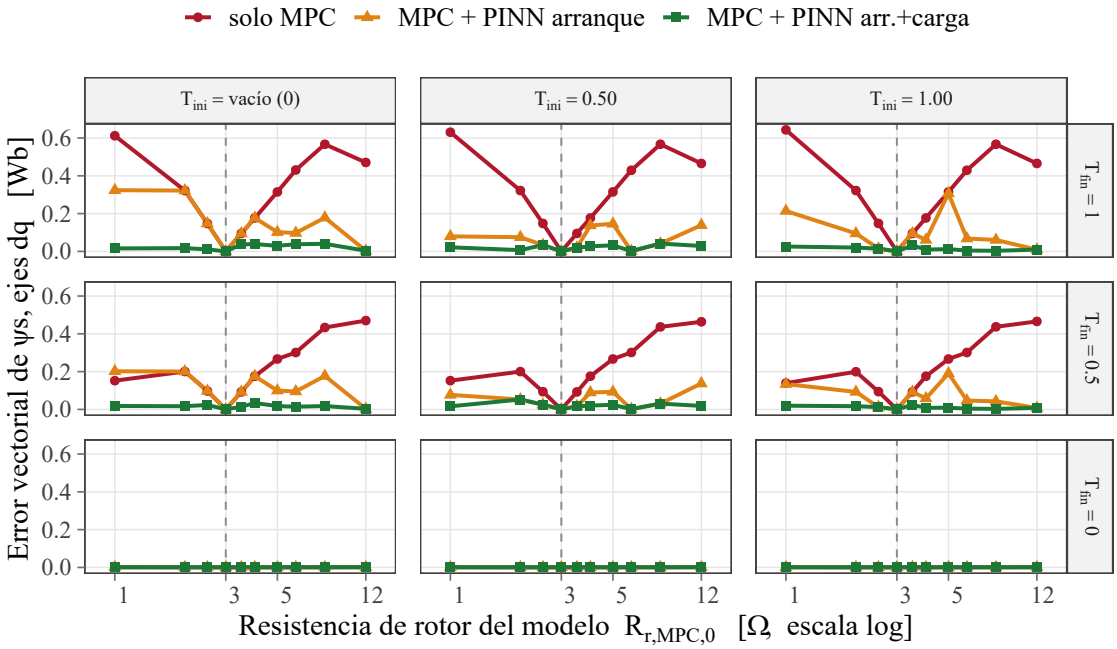


Figura D.12: Error vectorial de estimación del flujo de estator, en ejes *dq* (Tabla 7.1). Máquina de 3,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

D.1.4. Máquina de 5,00 Ω

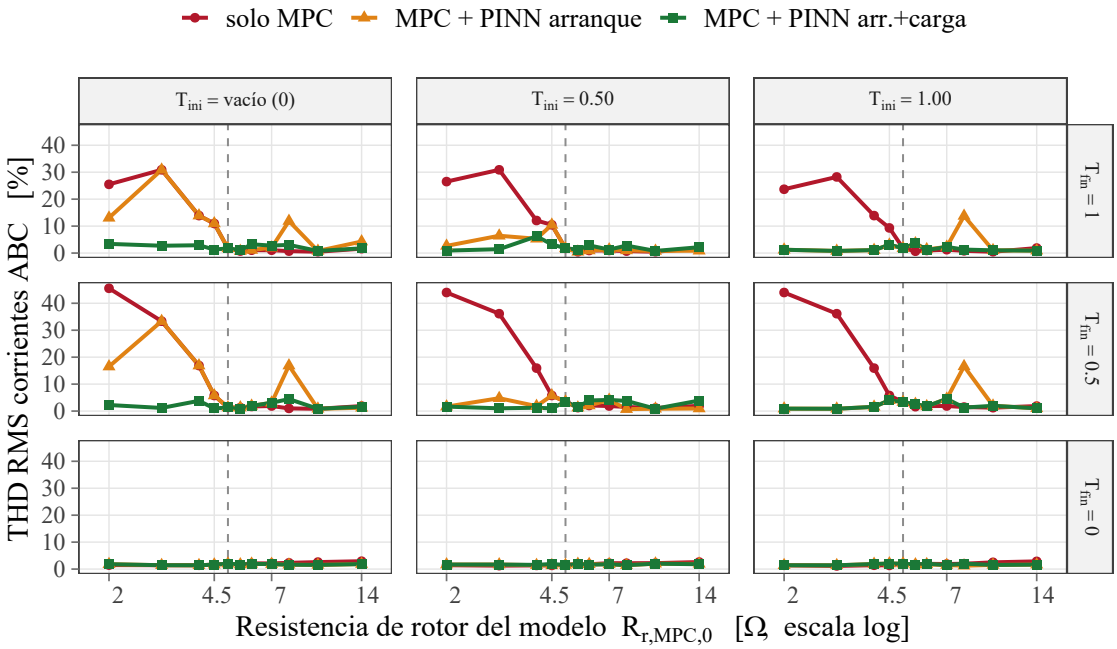


Figura D.13: Distorsión armónica de las corrientes de estator. Máquina de 5,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

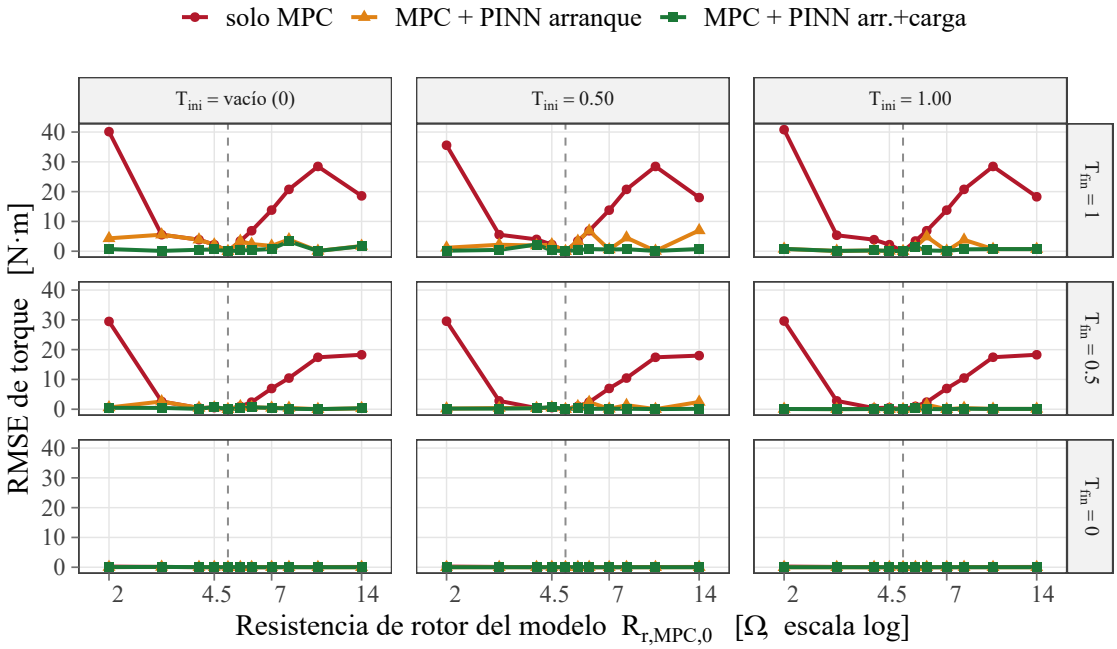


Figura D.14: Error de estimación de torque: discrepancia entre el torque real y el que estima el observador del controlador. Máquina de 5,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

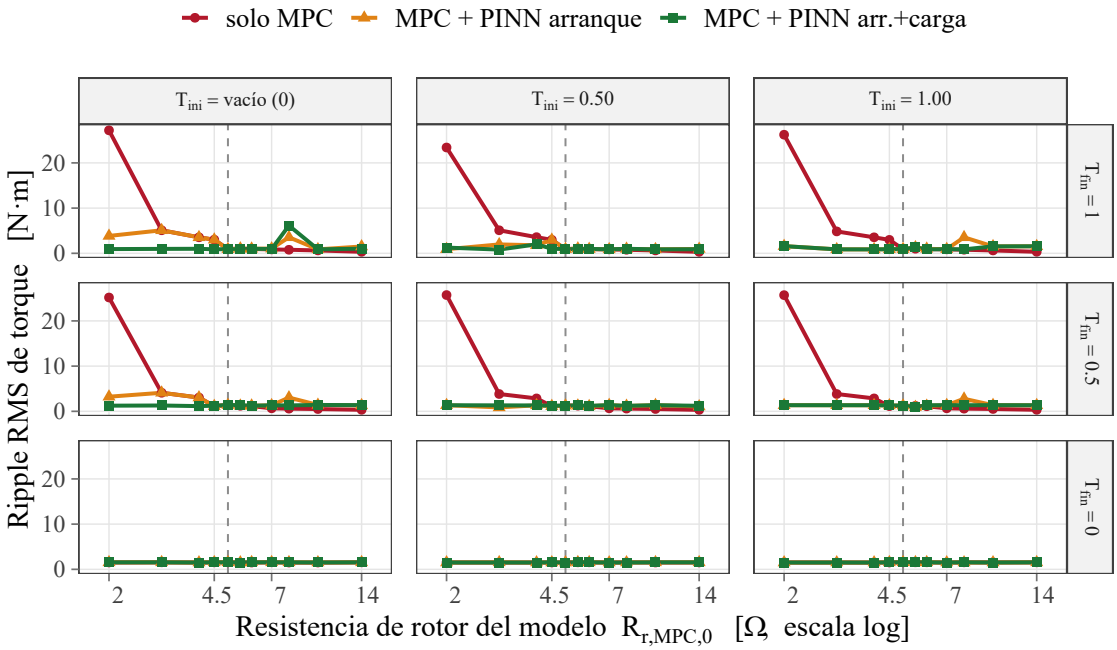


Figura D.15: *Ripple* RMS del torque entregado por la máquina. Máquina de 5,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

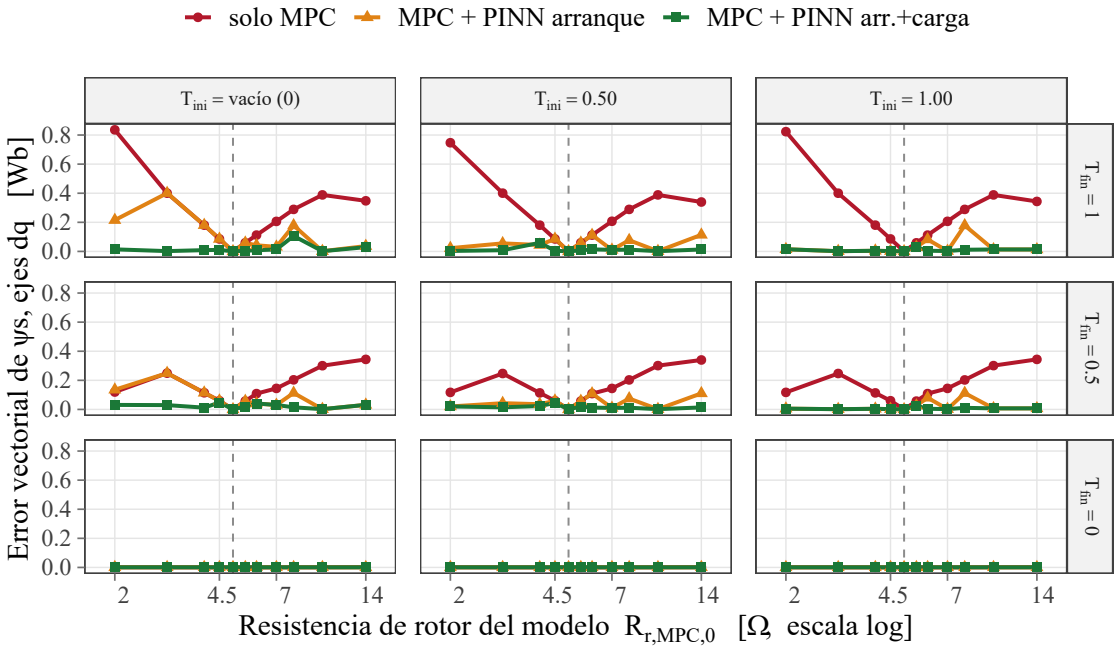


Figura D.16: Error vectorial de estimación del flujo de estator, en ejes *dq* (Tabla 7.1). Máquina de 5,00 Ω; velocidad de referencia 1440 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

D.2. Carga de bomba

Complemento de la Subsección 7.1.4. Con esta carga los tres indicadores sólo se apartan de su valor nominal en el extremo de subestimación.

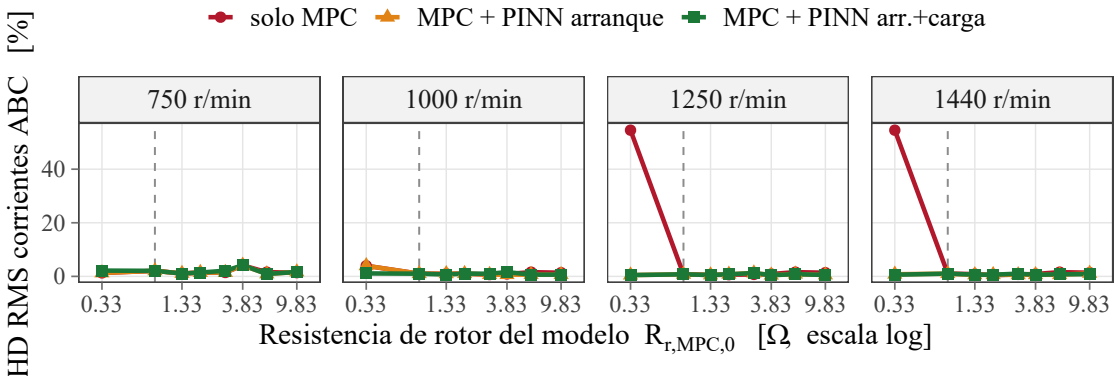


Figura D.17: Distorsión armónica de las corrientes de estator. Carga de bomba; las facetas recorren la referencia de velocidad, que en esta carga fija también el par solicitado. Menor es mejor.

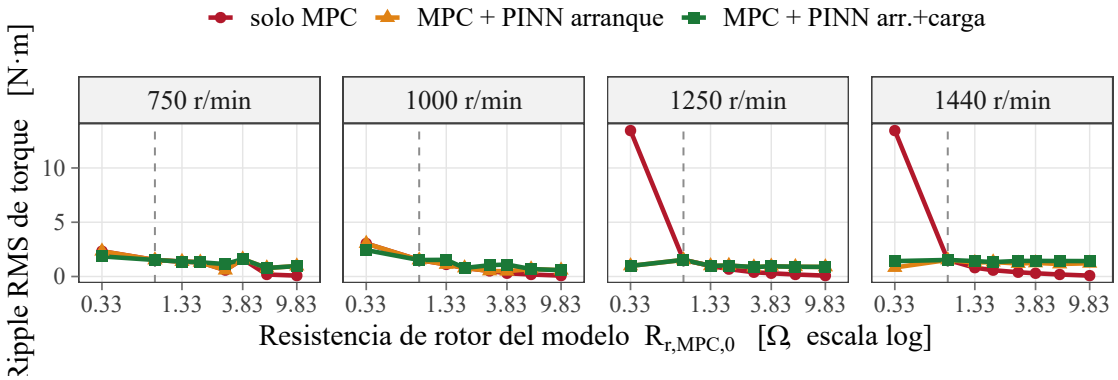


Figura D.18: *Ripple* RMS del torque entregado por la máquina. Carga de bomba; las facetas recorren la referencia de velocidad, que en esta carga fija también el par solicitado. Menor es mejor.

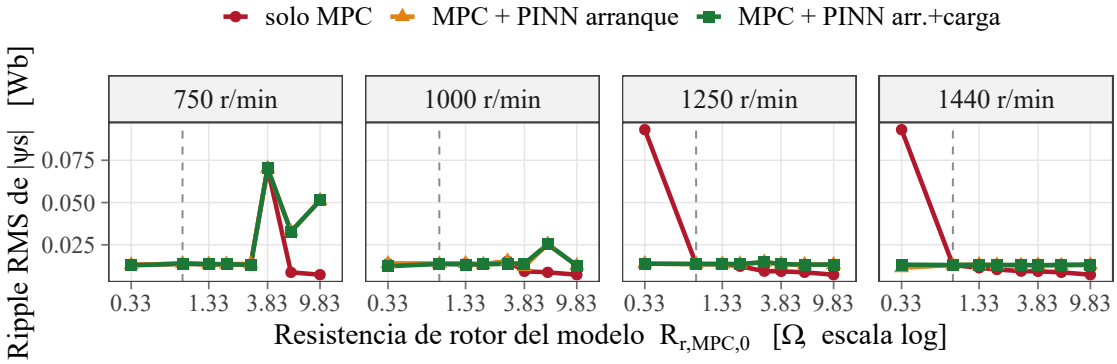


Figura D.19: *Ripple* RMS de la magnitud del flujo de estator. Carga de bomba; las facetas recorren la referencia de velocidad, que en esta carga fija también el par solicitado. Menor es mejor.

D.3. Velocidad reducida

Esta sección complementa la Subsección 7.1.5. Se considera la máquina de resistencia nominal a 1000 r/min.

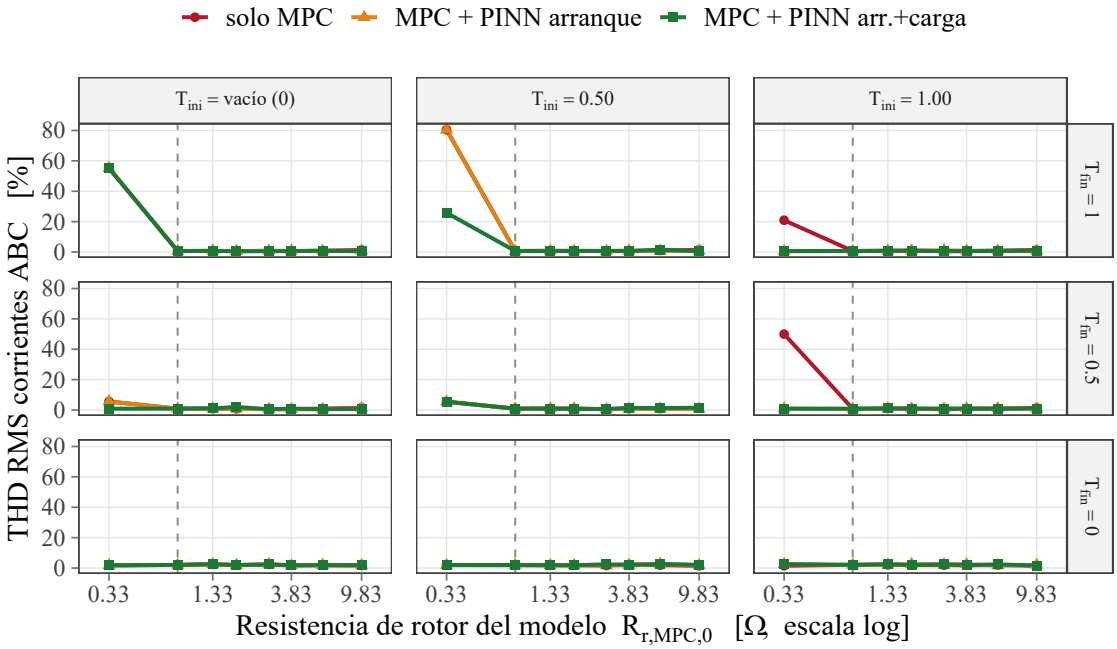


Figura D.20: Distorsión armónica de las corrientes de estator. Velocidad de referencia 1000 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

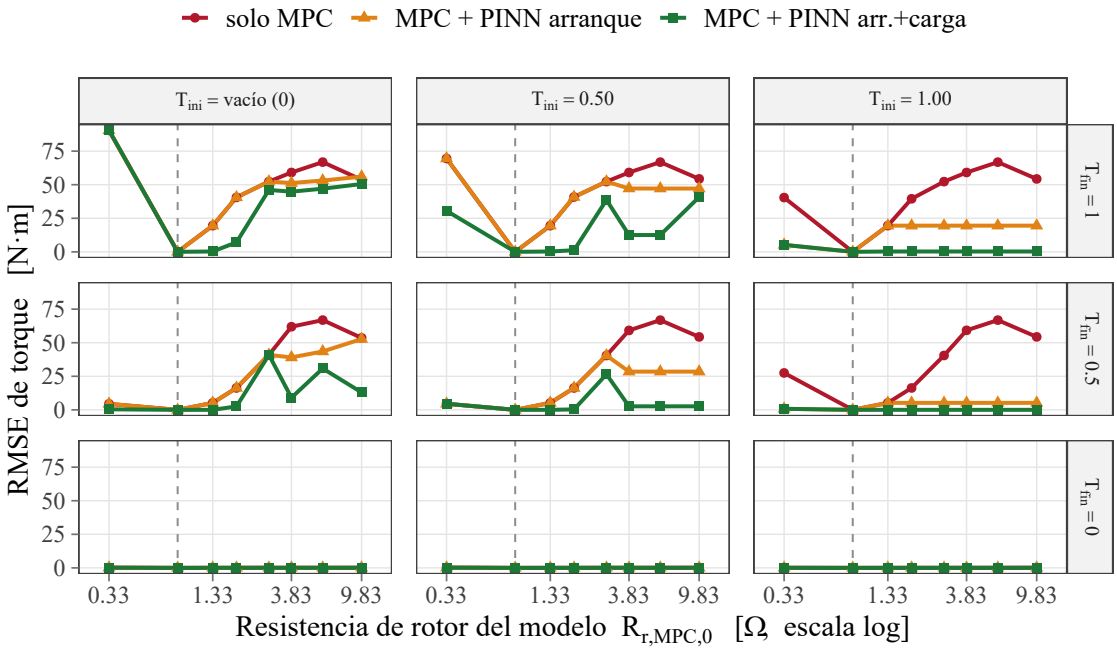


Figura D.21: Error de estimación de torque: discrepancia entre el torque real y el que estima el observador del controlador. Velocidad de referencia 1000 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

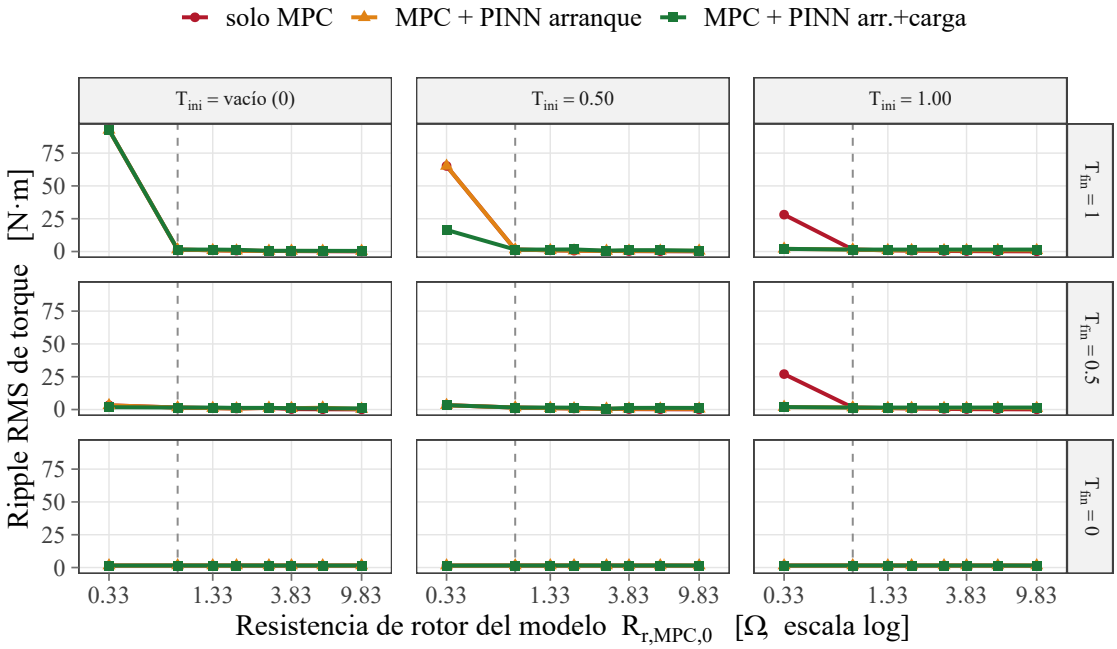


Figura D.22: *Ripple* RMS del torque entregado por la máquina. Velocidad de referencia 1000 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

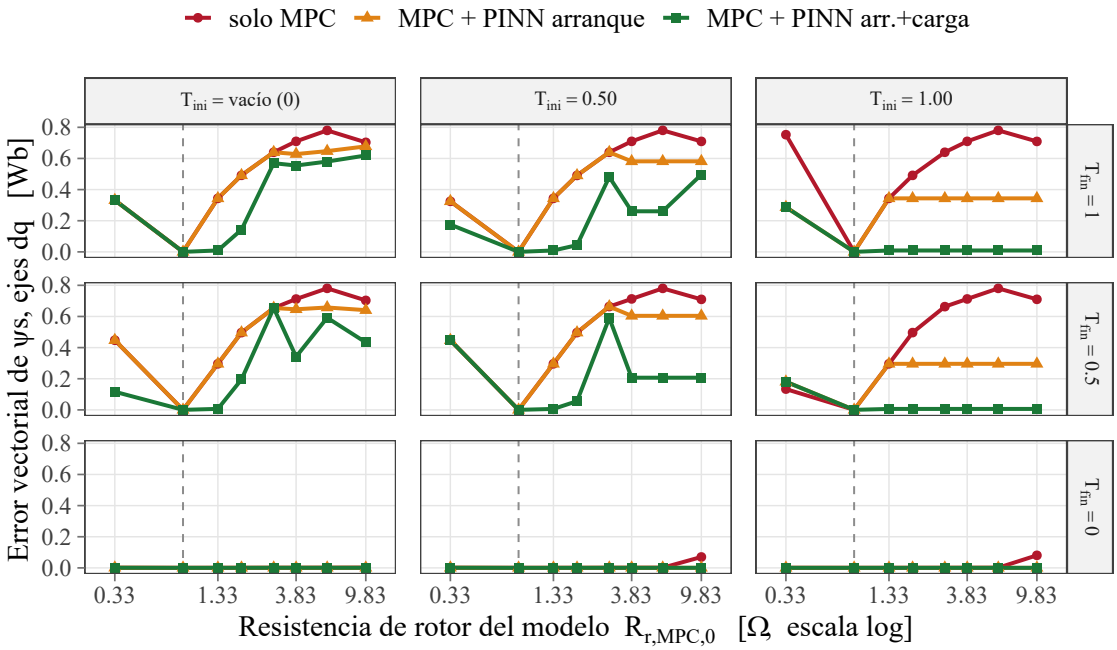


Figura D.23: Error vectorial de estimación del flujo de estator, en ejes *dq* (Tabla 7.1). Velocidad de referencia 1000 r/min; rutina de arranque con carga y escalón posterior. Menor es mejor.

Apéndice E

Tasa de Corrección frente a Ventana de Arranque

La escalera de la máquina nominal presenta un punto que rompe la monotonía: el escalón $R_{r,\mathrm{MPC},0} = 3{,}83\ \Omega$ deja un residuo mayor que el de 5,83 y el de 9,83 Ω, que son desajustes considerablemente más severos (Tabla 7.2). Este anexo documenta su origen, que no es un límite del alcance de las redes sino una interacción entre dos restricciones del coordinador de arranque.

E.1. El síntoma

Sobre la máquina de $R_{r,\mathrm{MAS}} = 0{,}83\ \Omega$, carga de cinta y velocidad nominal, el residuo que deja el experto de arranque tras un arranque en vacío recorre la escalera de forma no monótona:

Tabla E.1: Residuo tras el arranque en vacío sobre la máquina nominal. El máximo no está en el desajuste mayor sino en 3,83 Ω.

$R_{r,\mathrm{MPC},0}$ [Ω]	0,33	1,33	1,83	2,83	**3,83**	5,83	9,83
Residuo [Ω]	0,186	0,500	0,466	0,588	**1,090**	0,634	0,651

Un límite del rango aprendido produciría el patrón contrario, con el residuo creciendo al alejarse del valor de entrenamiento. Aquí el peor caso está en el interior de la escalera.

E.2. Las dos restricciones

El coordinador de arranque impone un **límite de tasa** a cada escritura: la corrección aplicada se satura en AC2_S_RATE = 0,15 en logaritmo decimal, esto es, un factor $10^{0{,}15} = 1{,}4125$ por decisión. Su **ventana de actuación se cierra cuando la máquina alcanza régimen**, momento a partir del cual la maniobra de arranque ha terminado y el experto cede.

Ambas restricciones son deliberadas. La primera existe porque la corrección *no tiene vuelta atrás*: se consolida escalón a escalón, de modo que un paso demasiado grande sobre

una estimación equivocada dejaría el modelo peor sin posibilidad de deshacerlo. La segunda delimita el dominio del experto, que fue entrenado sobre transitorios de arranque.

Su interacción, en cambio, no es evidente. La duración del arranque *crece* con el desajuste, porque un modelo más equivocado acelera peor; un desajuste mayor concede por tanto más ciclos y más decisiones. Pero el número de decisiones *necesarias* crece también, y más deprisa. La Tabla E.2 contrapone ambas cantidades.

Tabla E.2: Decisiones necesarias para cerrar el desajuste a la tasa máxima, frente a las que la ventana de arranque concede. El déficit alcanza su máximo en 3,83 Ω, que es donde el residuo se dispara.

$R_{r,\mathrm{MPC},0}$ [Ω]	0,33	1,83	2,83	**3,83**	5,83	9,83
Decisiones necesarias	2,67	2,29	3,55	**4,43**	5,64	7,16
Decisiones disponibles	2	1	2	**2**	4	8
Déficit	0,67	1,29	1,55	**2,43**	1,64	−0,84
Duración del arranque [s]	0,589	0,269	0,366	**0,449**	0,544	0,779
Residuo [Ω]	0,186	0,466	0,588	**1,090**	0,634	0,651

El cierre de la ventana es verificable sobre las corridas: la última escritura ocurre, en todos los escalones, a milisegundos del instante en que la máquina alcanza el 95 % de su referencia de velocidad. El escalón 3,83 Ω es el punto en que ambas curvas se cruzan de la peor manera: el arranque es todavía lo bastante rápido como para conceder solo dos decisiones, mientras el desajuste ya requiere cuatro o cinco.

E.3. La estimación era correcta

Conviene separar lo que la red estima de lo que el coordinador aplica, porque solo lo segundo está limitado. Reproducido el escenario sobre el modelo definitivo, que publica la corrección propuesta antes de saturarla, se obtiene la Tabla E.3.

Tabla E.3: Escalón $R_{r,\mathrm{MPC},0} = 3{,}83\ \Omega$: corrección que la red propone en cada decisión, desajuste realmente pendiente y corrección que el coordinador aplica, todas en logaritmo decimal. La red identifica el desajuste con error inferior al 6 %; lo aplicado es en ambos casos el tope de tasa.

Decisión	**Desajuste pendiente**	**Propuesta de la red**	**Aplicado**	**Confianza**
1	0,664	0,672	0,150	0,657
2	0,514	0,540	0,150	0,723

La red opera además dentro de su dominio y sobre evidencia de calidad: la pertenencia al dominio vale 1,000 en ambas decisiones y el coeficiente de determinación del ajuste físico, ρ^2, se mantiene entre 0,988 y 0,999. El mismo comportamiento se observa en los escalones vecinos, donde la propuesta reproduce el desajuste con error inferior al 1 %.

El residuo remanente no procede, por tanto, de un fallo de estimación sino del límite de actuación. De haberse aplicado íntegra la primera propuesta, la resistencia habría pasado de 3,83 a 0,815 Ω en una sola decisión, con un residuo de 0,015 Ω en lugar de los 1,090 Ω que quedan tras dos decisiones saturadas.

E.4. Por qué se manifiesta en una sola faceta

El residuo de 1,090 Ω no degrada por igual las nueve facetas de carga del escalón. Ocho de ellas terminan con menos de 12 r/min de error; una sola, la que arranca *en vacío* y recibe después un escalón a *par nominal*, colapsa a 1284 r/min y arrastra por sí sola el promedio del escalón hasta los 154 r/min que la Tabla 7.2 recoge.

La razón es que el residuo determina si la máquina **sobrevive al escalón de carga**, y esa es una condición de umbral y no una degradación gradual:

- con $R^{\text{eff}}_{r,\text{MPC}} = 1{,}92\ \Omega$ —lo que deja el arranque en 3,83— la entrada de par nominal deja el rotor trancado, y una máquina con el rotor trancado no entrega al experto de carga ningún ciclo válido sobre el que decidir: no vuelve a escribir;
- con $R^{\text{eff}}_{r,\text{MPC}} = 1{,}46\ \Omega$ —lo que deja el arranque en 5,83, gracias a sus cuatro decisiones— la máquina resiste el escalón, y el experto de carga corrige entonces con normalidad hasta 0,860 Ω.

Es también la maniobra más desfavorable por construcción: durante un arranque en vacío el desajuste apenas se manifiesta en las señales, de modo que el experto dispone de la peor información justo cuando dispone del menor tiempo.

Apéndice F

Comportamiento Fuera del Rango de Entrenamiento

La red única se entrena sobre resistencias reales de rotor comprendidas entre 0,10 y 10 Ω; el experto de arranque, entre 0,30 y 10 Ω, y el de carga, entre 0,83 y 6,33 Ω. La campaña de la Propuesta 1 en el Capítulo 7 evalúa cinco valores: cuatro caen en el rango del experto de arranque y tres en el del de carga, y el de 0,10 Ω queda fuera de los dos. La de la Propuesta 2 evalúa solo 0,83 Ω. Salvo esas excepciones, el valor de la resistencia real está dentro de lo entrenado, aunque algunos puntos de partida del modelo no lo están: los de 12 y 14 Ω de la Propuesta 1 superan el mayor valor visto, 10 Ω, y los de 5,83 y 9,83 Ω de la Propuesta 2 implican desajustes iniciales mayores que los de sus ejemplos de regresión. Este anexo recoge una exploración complementaria que sale deliberadamente del rango de entrenamiento, y que permite contrastar empíricamente una diferencia que el cuerpo del texto solo puede argumentar por construcción: qué ocurre cuando el estimador opera donde no fue entrenado, según proceda de un cálculo o de unos pesos.

Se trata de una exploración deliberadamente pequeña: cuatro resistencias, 0,06; 0,08; 11 y 12 Ω, cada una con el modelo desajustado y con el modelo ajustado, es decir, ocho casos por propuesta; una rutina, arranque con cinta al 50 % sostenida durante 7 s, y una velocidad, la nominal; y sus conclusiones deben leerse con ese alcance.

F.1. Un obstáculo previo: el techo de escritura

La primera tentativa no midió lo que pretendía. Los dos algoritmos de escritura saturan la resistencia que entregan al controlador, y lo hacen *en el mismo valor superior*:

Tabla F.1: Límites que cada algoritmo impone al valor de resistencia que escribe sobre el controlador. El techo, 10 Ω en ambas, coincide con el extremo superior del rango de entrenamiento de la red única y del experto de arranque; el experto de carga se entrenó solo hasta 6,33 Ω.

	Límite inferior	**Límite superior**
Propuesta 1: expertos	0,1 Ω	10 Ω
Propuesta 2: una sola red	0,001 Ω	10 Ω

La consecuencia es que **el sistema tal como se despliega no puede expresar un valor superior a** 10 Ω, de modo que cualquier corrección por encima de esa cifra mide la saturación y no la red: en esa primera tentativa, los casos desajustados de las dos propuestas y los ajustados de la Propuesta 2 terminaron en 10,000 Ω —la Propuesta 1 no escribió sobre sus modelos ajustados, que conservaron 11 y 12 Ω—, y el residuo resultante no es más que la distancia entre el techo y la resistencia real. Por debajo la situación es asimétrica: el suelo de la Propuesta 1 le impide escribir valores bajo 0,1 Ω, mientras que el de la Propuesta 2, dos órdenes de magnitud menor, no interviene.

Para observar el comportamiento de las redes y no el de los limitadores se prepararon **copias exploratorias** de ambos modelos con el techo elevado a 20 Ω. Conviene subrayar que se trata de una modificación del diseño y no de la configuración desplegada: lo que sigue caracteriza a los estimadores, no al sistema tal como se entrega.

F.2. Por encima del rango, con el techo elevado

Tabla F.2: Resultado sobre dos máquinas cuya resistencia real excede el rango de entrenamiento, con el techo de escritura elevado a 20 Ω. Carga de cinta al 50 % y velocidad nominal. En el caso ajustado la conducta correcta es no escribir.

Propuesta	**Caso**	$R_{r,\mathrm{MAS}}$	$R_{r,\mathrm{MPC},0}$	$R^{\mathrm{eff}}_{r,\mathrm{MPC}}$ **final**	**Residuo**
1: expertos	desajustado	11	3,67	10,973	**0,2** %
	ajustado	11	11,00	11,000	**0,0** %
	desajustado	12	4,00	11,021	8,2 %
	ajustado	12	12,00	12,000	**0,0** %
2: una sola red	desajustado	11	3,67	9,641	12,4 %
	ajustado	11	11,00	10,253	6,8 %
	desajustado	12	4,00	10,261	14,5 %
	ajustado	12	12,00	10,340	13,8 %

El contraste es nítido y admite una lectura directa.

En los dos casos ensayados por encima del rango, el sistema de expertos sigue corrigiendo. Sobre la máquina de 11 Ω lleva el modelo desde 3,67 hasta 10,973 Ω en ocho escrituras, con un residuo del 0,2 %, pese a que la resistencia real, 11 Ω, y el valor final, 10,973 Ω, quedan fuera del rango sobre el que se entrenó; el punto de partida, 3,67 Ω, sí está dentro. Sobre la de 12 Ω la corrección queda incompleta: termina en 11,021 Ω, con un 8,2 % de residuo. Y, lo que importa más, **no toca ninguno de los dos modelos que ya estaban ajustados**: en ambos casos no realiza una sola escritura. Es coherente con que su estimador se calcule: la proyección $\hat{d}$ se obtiene con la misma fórmula del modelo del predictor tanto dentro como fuera del rango visto, y las redes conservan su papel, que es juzgarla. Lo que se conserva fuera del rango es esa construcción, no la exactitud: la identidad (5.12) vale entre predicciones, y la relación con la corriente medida arrastra el sesgo de la ecuación (5.16).

La red única, en cambio, converge hacia el borde de lo que aprendió. Las cuatro corridas terminan entre 9,64 y 10,34 Ω con independencia del punto de partida, cerca de 10 Ω, que es exactamente el extremo superior de su conjunto de entrenamiento. Sucede incluso cuando el modelo era correcto: desde 11 y desde 12 Ω lo arrastra hacia abajo, degradándolo un 6,8 y un 13,8 % respectivamente. En estos cuatro casos se comporta como

un aproximador fuera de su dominio, que satura en el extremo del recorrido que sus datos cubren.

F.3. Por debajo del rango

Tabla F.3: Resultado sobre dos máquinas cuya resistencia real queda por debajo del rango de entrenamiento de las dos propuestas, con los límites de escritura originales. Carga de cinta al 50 % y velocidad nominal. En el caso ajustado la conducta correcta es no escribir.

Propuesta	**Caso**	$R_{r,\mathrm{MAS}}$	$R_{r,\mathrm{MPC},0}$	$R^{\mathrm{eff}}_{r,\mathrm{MPC}}$ **final**	**Residuo** [Ω]	[%]
1: expertos	desajustado	0,06	0,18	0,1000	0,0400	66,7
	ajustado	0,06	0,06	0,0600	0,0000	0,0
	desajustado	0,08	0,24	0,1203	0,0403	50,4
	ajustado	0,08	0,08	0,0800	0,0000	0,0
2: una sola red	desajustado	0,06	0,18	0,0690	0,0090	14,9
	ajustado	0,06	0,06	0,0598	0,0002	0,3
	desajustado	0,08	0,24	0,0836	0,0036	4,5
	ajustado	0,08	0,08	0,0716	0,0084	10,5

Por debajo la comparación es incompleta, porque el suelo de 0,1 Ω impide a la Propuesta 1 expresar el valor buscado. Sus dos residuos de 0,040 Ω (Tabla F.3) coinciden en la cifra, pero no en la causa. En los dos casos corrige solo la red de carga, con tres compromisos, cada uno limitado al factor 1,26 por aplicación de su coordinador, y después se detiene porque agotó los tres compromisos que admite por evento de carga (Subsección 5.2.5.c), aunque el residuo siga indicando sobreestimación. Sobre la máquina de 0,06 Ω el tercer paso queda recortado por el suelo, en 0,100 Ω, y el residuo es la distancia hasta él. Sobre la de 0,08 Ω los tres pasos completos llevan el modelo de 0,24 a 0,120 Ω, 0,020 Ω por encima del suelo: allí el valor final lo fija el límite de compromisos, no el suelo. En ninguno de los dos casos un compromiso más habría cambiado la cifra: la red de carga estima entonces 0,066 y 0,083 Ω, cerca del valor real, pero la corrección que falta, 0,034 y 0,037 Ω, queda bajo la banda muerta absoluta de 0,05 Ω del coordinador (ecuación (5.34)). Ninguna de las dos cifras de residuo mide, por tanto, la capacidad de estimación de la red, sino los límites de escritura que la rodean. Sí conserva, en cambio, la condición de rechazo: no escribe sobre ninguno de los dos modelos ajustados.

La Propuesta 2 sí puede escribir en esa región, y allí **se acerca al valor real**: corrige de 0,18 a 0,069 Ω sobre la máquina de 0,06, con un 14,9 % de error, y de 0,24 a 0,084 sobre la de 0,08, esta última con un 4,5 % de error. El resultado matiza la lectura anterior: en estos dos casos la red no converge hacia el borde de su entrenamiento, 0,10 Ω, como sí lo hace por encima del rango; con dos puntos no puede decirse a qué se debe esa diferencia. Su condición de rechazo, sin embargo, vuelve a fallar: sobre la máquina de 0,08 Ω desplaza un modelo correcto un 10,5 %, aunque sobre la de 0,06 Ω lo deja prácticamente intacto, con un 0,3 %.

F.4. Qué se concluye y qué no

Las conclusiones se apoyan en los ocho casos por propuesta de las Tablas F.2 y F.3, y valen solo para ellos. Sobre la condición de rechazo, la Propuesta 1 no escribe sobre

ninguno de sus cuatro modelos ajustados; la Propuesta 2 desplaza tres, un 6,8; un 13,8 y un 10,5 %, y deja el cuarto casi intacto. Sobre la corrección de los modelos desajustados, por encima del rango la Propuesta 1 termina con un 0,2 % de residuo en 11 Ω y un 8,2 % en 12 Ω, y la Propuesta 2, con un 12,4 y un 14,5 %. Por debajo, la Propuesta 1 se detiene a 0,040 Ω del valor real en los dos casos, por su suelo de escritura en uno y por su límite de compromisos en el otro, y en los dos la banda muerta le impide seguir, aunque su red estime un valor cercano al real; la Propuesta 2 termina con un 14,9 y un 4,5 %. Estos resultados son coherentes con la diferencia de construcción entre un estimador que se calcula y uno que se aprende, pero no la demuestran: la configuración de 0,10 Ω del Capítulo 7, que también queda bajo el rango de los dos expertos, deja con la Propuesta 1 un residuo medio de 0,708 Ω, siete veces la resistencia real (Subsección 7.1.3).

Debe declararse también lo que esta exploración *no* establece. Son cuatro resistencias, una sola rutina y una sola velocidad. El ensayo por encima del rango exigió elevar un límite del diseño, de modo que caracteriza a los estimadores y no al sistema desplegado, que nunca escribiría por encima de 10 Ω. Y por debajo la comparación queda incompleta por el suelo, el límite de compromisos y la banda muerta de la Propuesta 1. Las conclusiones del cuerpo del texto se refieren al dominio declarado, salvo las excepciones señaladas al comienzo de este anexo: la configuración de 0,10 Ω, bajo el rango de los dos expertos; la de 0,50 Ω, bajo el del experto de carga, y los puntos de partida que exceden lo visto en entrenamiento. Y esta exploración no puede sustituir a una evaluación sistemática fuera de ese dominio.

Apéndice G

Verificaciones de la Implementación de la Propuesta 1

Este anexo reúne las comprobaciones de la Propuesta 1 a las que remite el Capítulo 5: cuatro sobre el código desplegado y una medición del sesgo del estimador físico. Cada sección dice qué se detectó, lo muestra en una tabla y resume su efecto.

Varias usan la misma **corrida de referencia**: una corrida de entrenamiento del experto de carga con el perfil creciente desde vacío de la Tabla 5.11, es decir, arranque en vacío y cuatro escalones de 0,45; 0,75; 0,95 y 1,00 veces el par nominal, sobre la máquina de 0,83 Ω con el modelo en 2,33 Ω y la capa inactiva.

G.1. Vector de entrada: despliegue frente a entrenamiento

Qué se detectó. Los datos de entrenamiento se generaron con un extractor en Python, y en línea el vector lo arma el `C-Script`. Sobre una misma simulación, los dos vectores de 141 componentes no son idénticos. La paridad de la exportación no lo advierte, porque compara C con `PyTorch` sobre vectores ya preparados.

Cómo se midió. Una copia del modelo que solo agrega la escritura a archivo registra, en cada ciclo, el vector que el `C-Script` entrega a las redes; el extractor de entrenamiento forma el suyo procesando el registro de esa misma simulación. Los ciclos se emparejan por su instante de inicio y los dos vectores se restan componente a componente. Como el `C-Script` arma un vector para las dos redes de arranque y otro para las dos de carga, cada uno se compara por separado. La diferencia se mide en desviaciones estándar σ del conjunto de entrenamiento, que es la escala con que cada red normaliza sus entradas: una diferencia de $0{,}1\,\sigma$ es un décimo de la dispersión típica de esa entrada (Tabla G.1).

Tabla G.1: Diferencia entre el vector con que se entrenaron las redes, formado por el extractor en Python, y el que el `C-Script` les entrega en línea, en la corrida de referencia (177 ciclos). Se mide en desviaciones estándar σ del conjunto de entrenamiento de cada experto: porcentaje de componentes del bloque, sobre todos los ciclos, que difieren en menos de $0{,}1\,\sigma$, y diferencia que no supera el 95 % de ellas.

	Vector de arranque		**Vector de carga**	
Bloque	**Bajo** $0{,}1\,\sigma$ [%]	95 % **bajo** [σ]	**Bajo** $0{,}1\,\sigma$ [%]	95 % **bajo** [σ]
Actual (27)	98,6	0,02	98,6	0,02
Previo (27)	98,7	0,02	78,3	0,55
Diferencia (27)	94,5	0,12	55,4	3,5
Medias móviles (30)	97,1	0,04	96,7	0,04
Desviaciones móviles (30)	89,2	0,35	86,5	0,76
Vector completo (141)	95,5	0,08	83,5	0,90

En el vector de las redes de arranque, el 95,5 % de las componentes difiere en menos de $0{,}1\,\sigma$, y la diferencia mediana es del orden de $10^{-5}\,\sigma$. En el de las redes de carga, los bloques *previo* y *diferencia* se apartan más, por la primera de las causas que siguen. Las otras tres corridas revisadas, con la capa activa, con el modelo ajustado y una de prueba con $R_{r,\mathrm{MPC},0} = 3{,}33\ \Omega$, dan diferencias iguales o menores: entre el 95,5 y el 99,0 % de las componentes bajo $0{,}1\,\sigma$ en el vector de arranque, y entre el 83,5 y el 92,6 % en el de carga. Si se exige coincidencia exacta, con una diferencia relativa de a lo sumo 10^{-6}, solo coinciden el 47 % y el 29 % de las componentes, respectivamente, y ningún ciclo completo; pero la mayor parte de esas diferencias son de redondeo.

Las diferencias. Un emulador en Python del constructor de C reproduce sus 141 componentes, de modo que la diferencia queda explicada por completo. Se resume en cinco causas:

1. **El orden de** `fv_prev` (Subsección 5.4.2.b): las redes de carga reciben el ciclo previo igual al actual y la diferencia nula, 54 de sus 141 entradas, pero se entrenaron con el ciclo anterior real.

2. **El instante de las variables internas del controlador.** El extractor de entrenamiento las lee del registro con un período de atraso; en C corresponden al mismo instante que las corrientes. Afecta a $\hat{d}$, a ρ^2 y a otras nueve características.

3. **La referencia de flujo.** Para `psi_ref_norm` y `err_flujo`, C usa la del cierre del ciclo y el entrenamiento, su media sobre el ciclo.

4. **Las fronteras de ciclo.** Entre el 3,3 y el 14,4 % de los ciclos difieren en una o dos muestras, y la diferencia pasa a los agregados de hasta diez ciclos.

5. **El primer ciclo.** Las redes de carga se evalúan desde el primer ciclo válido, que no existe en los datos de entrenamiento; esos primeros vectores no abren ninguna compuerta en el sentido equivocado.

Por qué la propuesta funciona igual. Las diferencias grandes se concentran donde menos pesan. En el vector de arranque aparecen con la máquina en velocidad y sin carga: la compuerta de `arr_bajar` cambia en 14 ciclos, 12 de ellos en la región ciega, que su coordinador ignora, y con la capa activa los 8 cambios caen en ella. En el vector de carga

están en los bloques *previo* y *diferencia*: con la máquina en régimen, el ciclo anterior se parece al actual, y la red conserva la información de los bloques *actual* y de las medias, que coinciden. Medido fuera de línea, sobre la partición de prueba del experto de carga, dar a las redes la historia desplegada eleva el error absoluto medio de 0,053 a 0,084 Ω en `carga_subir` y de 0,030 a 0,040 Ω en `carga_bajar`, y los falsos positivos de esta última sobre el modelo ajustado pasan de 0 a 0,12 %: una degradación acotada, no una pérdida de la función. Y los resultados en lazo cerrado del Capítulo 7 ya la incluyen, porque se obtuvieron con el vector que construye el `C-Script`.

G.2. Instantes del predictor y de la sensibilidad

Qué se detectó. La identidad afín (5.12) exige evaluar la sensibilidad en el mismo instante que la predicción de la que procede el residuo. La corriente y la magnitud del flujo coinciden, pero el **ángulo no**: la predicción gira el flujo de rotor estimado con $\gamma_r[k]$ y la sensibilidad con $\gamma_r[k+1]$, porque toma el flujo que el bloque publica después de actualizar el ángulo (Tabla G.2). La diferencia es un paso de giro, $T_s\,\omega_s$, que a la velocidad nominal vale 1,7°.

Tabla G.2: Secuencia temporal del predictor auxiliar y de la sensibilidad en el `C-Script` de la Propuesta 1. Los pasos son los de la *Update Function* (Listado B.4).

Paso	Qué calcula	Ángulo del flujo
Período k, pasos 1 a 5	Mide $\mathbf{i}_s[k]$ y $\omega_m[k]$, estima $\hat{\psi}_r[k]$ y calcula el ángulo siguiente, $\gamma_r[k+1] = \gamma_r[k] + T_s\,\hat{\omega}_s[k]$	—
Período k, paso 9b	Predicción auxiliar $\hat{\mathbf{i}}_s[k+1\|k]$, con $\mathbf{i}_s[k]$, la tensión aplicada, $\omega_{em}[k]$ y $\hat{\psi}_r[k]$	$\gamma_r[k]$
Período k, pasos 12 y 13	Actualiza el ángulo a $\gamma_r[k+1]$ y publica el flujo $\hat{\psi}_r[k]$ en $\alpha\beta$	$\gamma_r[k+1]$
Período k, al final	Guarda $\mathbf{i}_s[k]$, el flujo publicado y la predicción	$\gamma_r[k+1]$
Período $k+1$	Residuo $\boldsymbol{\varepsilon}[k+1] = \mathbf{i}_s[k+1] - \hat{\mathbf{i}}_s[k+1\|k]$	—
Período $k+1$	Sensibilidad $\boldsymbol{\varphi}[k]$, con $\mathbf{i}_s[k]$ y el flujo guardado	$\gamma_r[k{+}1]$, debería ser $\gamma_r[k]$

Cómo se midió. Sobre las cuatro corridas de la Sección G.1 se reconstruyó el predictor a partir de las señales registradas, con un error máximo de $2 \cdot 10^{-14}$ A, y se derivó numéricamente respecto de la resistencia. La derivada coincide con (5.10)–(5.11) evaluadas con $\gamma_r[k]$, con una diferencia relativa de $4 \cdot 10^{-8}$, de modo que la fórmula es correcta; la sensibilidad del código se aparta de ella entre un 0,2 y un 1,6 % en la mediana de cada corrida. Las redes se entrenaron con la sensibilidad alineada, porque el extractor lee el flujo publicado con un período de atraso (segunda causa de la Sección G.1).

Efecto. Bajo carga, donde la capa escribe, $\hat{d}$ cambia entre un 0,4 y un 1 %, y ninguna red cambia su compuerta en 1 900 evaluaciones. En velocidad y sin carga el desfase sí invierte el signo de $\hat{d}$, pero esa es la región ciega, donde las redes se abstienen. En lazo cerrado se repitieron con la sensibilidad alineada 16 casos de la campaña del Capítulo 7 sobre la máquina de 0,83 Ω: ocho de cinta con escalón a medio par, cuatro de cinta en vacío y cuatro de bomba. La capa escribe las mismas veces y en los mismos instantes, y solo en nueve casos cambia el valor de la última escritura (Tabla G.3). Los resultados de ese capítulo corresponden, por tanto, al código tal como está desplegado.

La misma secuencia explica que el flujo publicado en $\alpha\beta$ difiera del real en $|\boldsymbol{\psi}_s|\, T_s\, \omega_s$, 0,029 Wb a 1440 r/min, aun con el modelo ajustado; el indicador de flujo del Capítulo 7 lo evita porque compara en ejes dq.

Tabla G.3: Efecto de alinear la sensibilidad sobre 16 casos de la campaña en lazo cerrado: mayor diferencia, en valor absoluto, entre la corrida con la sensibilidad alineada y la del código desplegado, y la magnitud con que conviene compararla. Indicadores del Capítulo 7, sobre los 0,5 s finales.

Indicador	**Diferencia máxima**	**Referencia**
Escrituras de $R_{r,\mathrm{MPC}}^{\mathrm{eff}}$	ninguna	Mismo número e instantes en los 16 casos
$R_{r,\mathrm{MPC}}^{\mathrm{eff}}$ final y error de resistencia	0,0051 Ω	Diez veces menor que la banda muerta de corrección, 0,05 Ω
Error de velocidad	0,027 r/min	Menos del 0,5 % del piso del controlador de velocidad, 5,5 r/min
Error de estimación de torque	0,040 N·m	De 1,17 a 1,21 N·m en el caso de mayor cambio
Error de estimación de flujo	0,0033 Wb	De 0,018 a 0,021 Wb en el caso de mayor cambio
Distorsión armónica	1,3 puntos	Dentro de su dispersión entre casos con el mismo punto final: de 2,56 a 4,93 %, con desviación de 0,82 puntos

G.3. Marcha en sentido inverso

Qué se detectó. Las entradas de las redes se refieren al sentido positivo (Subsección 5.2.4.c), pero eso no basta para corregir en marcha inversa desde reposo, porque varias condiciones que evalúan los coordinadores suponen el sentido positivo y no reciben ese ajuste. La Tabla G.4 las recorre: qué evalúa cada una y qué ocurre con referencia negativa. La fracción de corriente de eje q, en cambio, no depende del sentido: el entrenamiento y el código la calculan con $|i_{sq}|$ (Listado B.6), y en el tramo inverso con carga de la rutina de la Sección 7.3 toma los mismos valores que en el directo, con mediana de 0,140 contra 0,142.

Tabla G.4: Condiciones que evalúa el código de los coordinadores de la Propuesta 1 y que suponen que la máquina gira en sentido positivo: qué evalúa cada una y qué ocurre si la referencia de velocidad es negativa. Ninguna recibe el ajuste de giro de la Subsección 5.2.4.c. Las velocidades están normalizadas por la nominal.

Coordinador	**Condición que evalúa**	**Qué ocurre con referencia negativa**
Arranque	Progreso $\omega_m / \max(\omega_m^\star, 10^{-6})$ y etapa derivada	El progreso vale del orden de -10^8 y la etapa queda en cero: la ventana de las redes no se abre
Arranque	*Breakaway* si, magnetizada, la máquina no declara arranque activo en cinco ciclos o 150 ms	Lee la marcha inversa como un rotor trancado
Arranque	Estancamiento si el progreso no avanza un 1 % en cinco ciclos	El progreso negativo no avanza: vuelve al *breakaway*
Arranque	Firma de sobreestimación $\hat{T}_e > T^\star$	Queda invertida, aunque no llega a consultarse
Arranque y carga	Rearme si la referencia baja de 5 rad/s y luego vuelve a superarlos	No se rearma
Carga	Salida del estado inicial con velocidad media sobre 0,05, y luego sobre 0,4	No sale del estado inicial
Carga	Firma consultada solo con velocidad media sobre 0,75	No se consulta

Cómo se midió. La rutina de cinco segundos de la Sección 7.3, sobre la máquina de 0,83 Ω con el modelo en 3,83 Ω, se repitió desde reposo con la referencia negativa en sus cuatro tramos (Tabla G.5). Sin la capa, las dos corridas son simétricas, con diferencias máximas de 0,01 r/min y 0,003 N·m.

Efecto. Con la capa, en sentido negativo no escribe ninguna red: el coordinador de carga no sale de su estado inicial, y el de arranque lee la marcha como un rotor trancado y entra al *breakaway* a los 0,315 s. Desde ahí da 56 pasos físicos que no se asientan, porque en vacío el signo de $\hat{d}$ no es fiable: la resistencia baja de 3,83 a 1,33 Ω, sube a 2,83, baja a 0,33 y sigue oscilando, con seis cambios de sentido, hasta los 0,955 Ω con que termina la rutina. Bajo carga, la velocidad cae hasta 941 r/min. En la rutina con inversión, en cambio, la red de carga sigue corrigiendo después de invertir: las dos escrituras posteriores a la inversión son suyas.

Tabla G.5: Rutina de cinco segundos de la Sección 7.3 con la Propuesta 1. En las tres corridas la máquina tiene 0,83 Ω y el modelo del controlador parte en 3,83 Ω; la columna de $R^{\text{eff}}_{r,\text{MPC}}$ da el valor con que termina la rutina, a los 5 s. Escrituras: correcciones decididas por las redes y pasos de la vía física del coordinador de arranque. El error de velocidad es la mediana de los últimos 0,3 s de cada tramo cargado, y la velocidad mínima se mide en esos tramos, pasados 0,3 s de la entrada de la carga; ambas en r/min.

Rutina	**Escrituras red / física**	$R^{\text{eff}}_{r,\text{MPC}}$ **al terminar** [Ω]	**Error de velocidad, 2.º / 4.º tramo**	**Velocidad mínima**
Sentido positivo	6 / 0	0,774	7,3 / 5,8	1411
Sentido negativo, desde reposo	0 / 56	0,955, sin asentarse: oscila entre 0,33 y 2,83	343 / 5,6	941
Con inversión (Sección 7.3)	6 / 0	0,793	7,3 / 5,8	1411

G.4. Aplicación de la reversión del coordinador de carga

Qué se comprobó. Que la reversión de la rectificación llega al controlador. El coordinador de carga entrega una decisión activa en tres ramas: el compromiso de una corrección de red, el paso de la vía de emergencia y la reversión (Listado B.8). La puerta de carga copia su valor al controlador con cualquiera de ellas (Listado B.6), y en la reversión el coordinador registra además el valor restaurado como propio, para no tomarlo por una corrección ajena.

Cómo se midió. Sobre la campaña del Capítulo 7. En 390 de las 806 corridas con el experto de carga, la resistencia aplicada es idéntica a la de la misma condición con solo el experto de arranque: su coordinador nunca escribió, y no hay nada que revertir. Las otras 416 se simularon con la traza de los coordinadores (Tabla G.6).

Tabla G.6: Corridas de la campaña del Capítulo 7 con una reversión del coordinador de carga, por resistencia de la máquina. Ninguna registra más de una, y todas son con cinta.

Máquina [Ω]	0,10	0,50	0,83	3,00	5,00	**Total**
Corridas con reversión	3	1	10	22	12	**48**

Resultado. En las 48, la resistencia aplicada cambia en el período siguiente a la reversión y toma exactamente el valor que tenía antes de la escritura revertida. Las demás simulaciones de la Propuesta 1 que usa este documento (la rutina de la Sección 7.3, el reparto de la Tabla 7.13a y el Anexo F) se revisaron con sus trazas, y ninguna registra una reversión.

G.5. El sesgo de la proyección y el alcance de ρ^2

Qué se midió. El sesgo de la proyección (5.16) y lo que ρ^2 no certifica (Subsección 5.2.3), sobre la corrida de referencia, donde el desajuste real se conoce: $R_{r,\text{MAS}} - R^{\text{eff}}_{r,\text{MPC}} = 0{,}83 - 2{,}33 = -1{,}50\ \Omega$. Se usa el $\hat{d}$ que calcula el propio `C-Script` (Tabla G.7).

Tabla G.7: El sesgo de la proyección y el alcance de ρ^2 en la corrida de referencia. Las tres primeras filas son medianas sobre sus 102 ciclos de régimen de carga; la del modelo ajustado es la misma rutina con el modelo en 0,83 Ω, sobre 145 ciclos. Las tres últimas recorren los ciclos de la corrida con $\rho^2 > 0{,}95$.

Magnitud	**Valor**
$\hat{d}$, frente a un desajuste de $-1{,}50$ Ω	$-5{,}75$ Ω, con el signo correcto en todos los ciclos
ρ^2	0,996
$\hat{d}$ con el modelo ajustado	$-0{,}010$ Ω
Variación de la energía de sensibilidad A	Un factor de 15
Correlación de rangos entre A y la magnitud del sesgo	$-0{,}98$
$\hat{d}$ en el quinto de ciclos con menor A; distancia al desajuste en el quinto de mayor A	$-8{,}8$ Ω; 0,56 Ω

Lectura. El sesgo aparece con el desajuste y desaparece con el modelo ajustado, como corresponde al término de estado de (5.16): $\hat{d}$ acierta el sentido del error, pero no su tamaño. Un residuo casi perfectamente alineado, con $\rho^2 = 0{,}996$, convive con un sesgo de 4,25 Ω, de modo que la alineación no prueba que la resistencia sea la única causa. Y el sesgo crece cuando baja A, que ρ^2 no mide: en los ciclos menos excitados, $\hat{d}$ se aleja más del desajuste.

Bibliografía

[1] J. Müller, "Apuntes para la asignatura máquinas eléctricas," Departamento de Ingeniería Eléctrica, Universidad Técnica Federico Santa María, 2003.

[2] W. Leonhard, *Control of Electrical Drives*, 3rd ed. Springer, 2001.

[3] J. Holtz, "The representation of AC machine dynamics by complex signal flow graphs," *IEEE Transactions on Industrial Electronics*, vol. 42, no. 3, pp. 263–271, Jun. 1995.

[4] A. Mora, "Control vectorial de máquinas de inducción," Apuntes para la asignatura de Accionamientos Eléctricos, DIE, Universidad Técnica Federico Santa María, Oct. 2021.

[5] J. Rodríguez and P. Cortés, *Predictive Control of Power Converters and Electrical Drives*. John Wiley & Sons, Ltd, 2012.

[6] J. Rodriguez, C. Garcia, A. Mora, F. Flores-Bahamonde, P. Acuna, M. Novak, Y. Zhang, L. Tarisciotti, S. A. Davari, F. Wang, Z. Zhang, M. Norambuena, T. Dragičević, T. Geyer, F. Blaabjerg, R. Kennel, D. A. Khaburi, M. Abdelrahem, N. Mijatovic, Z. Zhang, and R. P. Aguilera, "Latest advances of model predictive control in electrical drives—Part I: Basic concepts and advanced strategies," *IEEE Transactions on Power Electronics*, vol. 37, no. 4, pp. 3927–3942, Apr. 2022.

[7] J. Rodríguez, R. M. Kennel, J. R. Espinoza, M. Trincado, C. A. Silva, and C. A. Rojas, "High-performance control strategies for electrical drives: An experimental assessment," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 812–820, Feb. 2012.

[8] M. Norambuena, J. Rodriguez, C. Garcia, Z. Zhang, F. Wang, and R. Kennel, "A very simple strategy for high-quality performance of AC machines using model predictive control," *IEEE Transactions on Power Electronics*, vol. 34, no. 1, pp. 794–803, Jan. 2019.

[9] A. Mora, "Modelo dinámico de máquinas de inducción," Apuntes para la asignatura de Máquinas Eléctricas, DIE, Universidad Técnica Federico Santa María, Mar. 2022.

[10] J. Holtz, "Methods for speed sensorless control of AC drives," in *Sensorless Control of AC Motors*, K. Rajashekara, Ed. IEEE Press, 1996.

[11] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear

partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[12] D. Barrios, "Control digital de velocidad para un banco de pruebas," Memoria de Titulación, Departamento de Electricidad, Universidad Técnica Federico Santa María, 2007.

[13] I. Urrutia, "Control FOC aplicado a una máquina de inducción utilizando un inversor ANPC de cinco niveles," Memoria de Título, Departamento de Ingeniería Eléctrica, Universidad Técnica Federico Santa María, 2025.

[14] D. Gálvez, "Compensación de R_s usando mediciones de corriente de estator para una máquina de inducción controlada por PTC," Tesis de Título y Grado de Magíster, Departamento de Ingeniería Eléctrica, Universidad Técnica Federico Santa María, 2026.

[15] A. Mora, "Modelo de máquinas de campo giratorio: la máquina generalizada," Apuntes para la asignatura de Accionamientos Eléctricos, DIE, Universidad Técnica Federico Santa María, Sep. 2024.

[16] S. Kouro, P. Cortés, R. Vargas, U. Ammann, and J. Rodriguez, "Model predictive control—a simple and powerful method to control power converters," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 1826–1838, Jun. 2009.

[17] S. Vazquez, J. Rodriguez, M. Rivera, L. G. Franquelo, and M. Norambuena, "Model predictive control for power converters and drives: Advances and trends," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 2, pp. 935–947, Feb. 2017.

[18] F. Wang, S. Li, X. Mei, W. Xie, J. Rodríguez, and R. M. Kennel, "Model-based predictive direct control strategies for electrical drives: An experimental evaluation of PTC and PCC methods," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 671–681, Jun. 2015.

[19] G. Goodwin, S. Graebe, and M. Salgado, *Control System Design*. Prentice Hall, 2000.

[20] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems: Theory and Design*, 3rd ed. Prentice Hall, 1996.

[21] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Pearson, 2009.

[22] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015, disponible en http://neuralnetworksanddeeplearning.com.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. on Learning Representations (ICLR)*, 2015.

[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[25] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Physics-informed neural networks for power systems," in *Proc. IEEE PES General Meeting*, 2020.

[26] A. F. Psaros, K. Kawaguchi, and G. E. Karniadakis, "Meta-learning PINN loss functions," *Journal of Computational Physics*, vol. 458, p. 111121, 2022.

[27] M. Berardi, F. V. Difonzo, and M. Icardi, "Inverse physics-informed neural networks for transport models in porous materials," *Computer Methods in Applied Mechanics and Engineering*, vol. 435, p. 117628, 2025.

[28] J. Morrow, "Inverse physics-informed neural network," Morrow Consultants, 2024, disponible en https://www.morrowconsultants.com/inverse-physics-informed-neural-network-2.

[29] W. Wojtak, "PINNs and iPINNs in PyTorch," Repositorio de código, 2024, disponible en https://github.com/w-wojtak/PINNs-and-iPINNs-Pytorch.

[30] Departamento de Electrónica, UTFSM, "Capítulo 5: Inversores," Apuntes ELO381 – Electrónica Industrial.

[31] J. Holtz, "Pulsewidth modulation for electronic power conversion," *Proceedings of the IEEE*, vol. 82, no. 8, pp. 1194–1214, Aug. 1994.

[32] F. Fröhr and F. Orttenburger, *Introducción al Control Electrónico.* Siemens Aktiengesellschaft – Marcombo, 1986.

[33] P. Lezana, "Manejo de la plataforma digital DSP+FPGA para el control de accionamientos," Apéndice A, Departamento de Ingeniería Eléctrica, Universidad Técnica Federico Santa María.

[34] J. Juliet, "Nociones sobre sistemas de tiempo discreto y aplicación en accionamientos eléctricos," Apuntes para la asignatura de Accionamientos Eléctricos, DIE, Universidad Técnica Federico Santa María, Oct. 2022.

[35] P. Cortés, J. Rodríguez, C. Silva, and A. Flores, "Delay compensation in model predictive current control of a three-phase inverter," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 2, pp. 1323–1325, Feb. 2012.