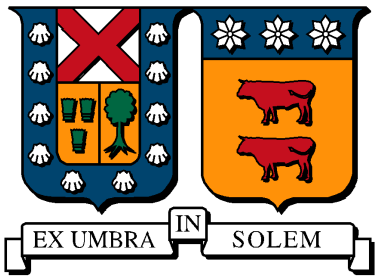
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA
VALPARAÍSO - CHILE



"DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR PREDICTIVO EN UN SISTEMA DE PÉNDULO DE FURUTA."

DANIEL ANDRÉS BUNSTER CRAVERO

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
Ingeniero Civil Electrónico

PROFESOR GUÍA
JUAN CARLOS AGÜERO VASQUEZ, Ph.D.

PROFESOR CORREFERENTE
CÉSAR SILVA JIMÉNEZ, Ph.D.

AGOSTO 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: Diseño e implementación de un controlador predictivo en un sistema de péndulo de Furuta.

Nombre del candidato(a): Daniel Andrés Bunster Cravero.

Carrera / Grado: Ingeniería Civil Electrónica.

Campus: Casa Central. **Departamento:** Electrónica.

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Juan C. Agüero, 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: 02/11/2026 **Firma:** _____

Estudiante o Candidato(a):

Fecha: 02-11-2026 **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.

Agradecimientos

Quiero agradecer a los profesores del Departamento de Electrónica por su contribución a mi formación profesional, especialmente a los del área de Control, ya que sus clases me inspiraron a tomar esta mención. En particular, agradezco a mi profesor guía, Juan Carlos Agüero, por brindarme este tema de título, su tiempo y el material necesario para llevarlo a cabo.

Agradezco también a las amistades que me dejó la universidad y a mi pololа por su compañía y por estar presente en estos últimos años de carrera. Un reconocimiento especial a los amigos que conocí al llegar a la región en la pensión, un vínculo que mantengo intacto hasta el día de hoy.

Por sobre todo, quiero darle las gracias a lo más importante de mi vida: mi familia. A mis padres y mi hermana, por su amor incondicional y respaldo constante a pesar de la distancia. A mis abuelos, tanto a los que me acompañan hoy como a los que me cuidan desde el recuerdo, y a cada familiar que siempre me tiene presente en sus oraciones. Sin el respaldo de cada uno de ustedes, jamás habría llegado hasta aquí. Muchas gracias.

Por último agradecer al Centro Avanzado de Ingeniería Eléctrica y Electrónica ANID AC3E CIA250006, al proyecto Fondecyt ANID 1261290 del profesor Juan C. Agüero, y al Proyecto Interno USM 2026 PI_LIR_26_28.

Resumen

En este trabajo de titulación se presenta el diseño e implementación de un método avanzado de control de procesos, el Control Predictivo por Modelo (MPC), aplicado a un sistema de péndulo invertido rotacional. Esta técnica consiste en implementar un esquema de control que utiliza un modelo matemático de la planta para predecir su comportamiento futuro y, en función de ello, optimizar las acciones de control sujetas a restricciones operacionales. La planta bajo estudio corresponde a un Péndulo de Furuta de la plataforma Quanser QUBE-Servo 2, un sistema mecánico subactuado, altamente no lineal e inherentemente inestable, lo que lo convierte en un excelente caso de interés para la validación de estrategias de control avanzadas. Para el desarrollo del controlador, se realiza la modelación del sistema seguida de un proceso de linealización en torno al punto de equilibrio inestable para analizar sus propiedades dinámicas. El método de optimización seleccionado para resolver el problema de MPC es el algoritmo Active-Set, implementado en el solver quadprog de MATLAB. Se desarrollan simulaciones en MATLAB/Simulink con el fin de verificar el correcto funcionamiento del bloque de control. Posteriormente, en la implementación física en tiempo real, se evalúa el desempeño del controlador utilizando el hardware de adquisición de datos y bloques de Quanser (QUARC), analizando la respuesta del sistema ante variaciones en el horizonte de predicción, el periodo de muestreo y las restricciones de actuación en el motor.

Palabras clave: MPC, Péndulo de Furuta, Active-Set, Quadprog, MATLAB/Simulink, QUBE-Servo 2, QUARC.

Abstract

This undergraduate thesis presents the design and implementation of an advanced process control method, Model Predictive Control (MPC), applied to a rotary inverted pendulum system. This technique consists of implementing a control scheme that uses a mathematical model of the plant to predict its future behavior and, based on this, optimize the control actions subject to operational constraints. The plant under study is a Furuta Pendulum on the Quanser QUBE-Servo 2 platform, an underactuated, highly nonlinear, and inherently unstable mechanical system, which makes it an excellent case of interest for validating advanced control strategies. To develop the controller, the system is modeled and subsequently linearized around the unstable equilibrium point in order to analyze its dynamic properties. The optimization method selected to solve the MPC problem is the Active-Set algorithm, implemented in MATLAB's `quadprog` solver. Simulations are carried out in MATLAB/Simulink to verify the correct operation of the control block. Subsequently, in the real-time physical implementation, the controller's performance is evaluated using Quanser's data acquisition hardware and blocks (QUARC), analyzing the system's response to variations in the prediction horizon, the sampling period, and the actuator's constraints.

Keywords: MPC, Furuta Pendulum, Active-Set, Quadprog, MATLAB/Simulink, QUBE-Servo 2, QUARC.

Símbolos y acrónimos

Esta sección presenta una lista de los símbolos y acrónimos relevantes para el desarrollo de esta memoria.

ACRÓNIMO	EXPLICACIÓN
MPC	Control Predictivo por Modelo (*Model Predictive Control*)
LQR	Regulador Cuadrático Lineal (*Linear Quadratic Regulator*)
DLQR	Regulador Cuadrático Lineal Discreto (*Discrete Linear Quadratic Regulator*)
PID	Proporcional-Integral-Derivativo (*Proportional-Integral-Derivative*)
QP	Programación Cuadrática (*Quadratic Programming*)
DARE	Ecuación de Riccati Algebraica Discreta (*Discrete Algebraic Riccati Equation*)
ZOH	Retenedor de Orden Cero (*Zero-Order Hold*)
A/D	Convertidor análogo-digital
D/A	Convertidor digital-análogo
ISE	Integral del Error Cuadrático (*Integral Square Error*)
IAE	Integral del Error Absoluto (*Integral Absolute Error*)

SÍMBOLO	EXPLICACIÓN
θ	Ángulo del brazo horizontal del péndulo de Furuta
α	Ángulo del péndulo respecto a la posición vertical invertida
$\dot{\theta},\ \dot{\alpha}$	Velocidades angulares de θ y α
u	Voltaje de control aplicado al motor
x	Vector de estados, $x = [\theta, \alpha, \dot{\theta}, \dot{\alpha}]^T$
$\hat{x}$	Vector de estados estimado
y	Vector de salidas medidas
$A_c,\ B_c$	Matrices del modelo en espacio de estados en tiempo continuo
$A_d,\ B_d$	Matrices del modelo en espacio de estados discretizado
$C,\ D$	Matrices de salida del modelo en espacio de estados
$\Phi,\ \Gamma$	Matrices de transición de estado y de entrada obtenidas mediante discretización ZOH
$Q,\ R$	Matrices de ponderación de estados y de la acción de control en la función de costo del MPC
Q_N	Matriz de costo terminal, solución de la ecuación de Riccati discreta (DARE)
N	Horizonte de predicción del controlador MPC
T_s	Período de muestreo
t_s	Tiempo de establecimiento
τ	Variable de integración temporal
J	Función de costo del problema de optimización del MPC
U	Secuencia de acciones de control futuras sobre el horizonte de predicción
$M_p,\ L_p$	Masa y longitud del péndulo
L_r	Longitud del brazo rotatorio
D_r	Coeficiente de fricción viscosa del brazo
$J_t,\ J_p,\ J_r$	Momentos de inercia total, del péndulo y del brazo
g	Aceleración de gravedad
$k_t,\ k_m$	Constante de torque y constante de fuerza contraelectromotriz del motor
R_m	Resistencia de armadura del motor

Contenido

1 Introducción **1**
- 1.1 Motivaciones . . . 2
- 1.2 Estado del arte . . . 2
  - 1.2.1 Control Predictivo Basado en Modelo . . . 2
  - 1.2.2 Algoritmo de optimización Active-Set . . . 3
  - 1.2.3 Estimación de estados en sistemas subactuados . . . 3
- 1.3 Objetivos del trabajo . . . 4

2 Fundamentos Teóricos **5**
- 2.1 Control Predictivo por Modelo . . . 5
  - 2.1.1 Fundamentos de MPC . . . 6
  - 2.1.2 Definición del problema de control . . . 6
  - 2.1.3 Formulación densa . . . 7
  - 2.1.4 Seguimiento de referencia . . . 8
  - 2.1.5 Restricción de desigualdad . . . 8
  - 2.1.6 Planteamiento final del problema QP . . . 9
  - 2.1.7 Resultado final y horizonte deslizante . . . 9
- 2.2 Método de optimización . . . 10
  - 2.2.1 Método de optimización Active-Set de quadprog . . . 10
- 2.3 Muestreo del sistema . . . 11
  - 2.3.1 Fundamentos del muestreo en sistemas de control . . . 11
  - 2.3.2 Muestreo utilizando Retenedor de Orden Cero (ZOH) . . . 11
  - 2.3.3 Derivación de las matrices discretas Φ y Γ . . . 12
- 2.4 Estimación de estados . . . 13
  - 2.4.1 Fundamentos de la estimación de estados . . . 13
  - 2.4.2 Técnica de estimación utilizada: enfoque cinemático determinista . . . 13
  - 2.4.3 Diferencias respecto a un observador de Luenberger . . . 14

3 Planta de interés: Péndulo de Furuta **15**
- 3.1 Descripción del sistema . . . 15
  - 3.1.1 Componentes mecánicos y grados de libertad . . . 16
  - 3.1.2 Sensores y actuador . . . 16
- 3.2 Modelo matemático del péndulo de Furuta . . . 17
  - 3.2.1 Ecuaciones de movimiento (formulación de Euler-Lagrange) . . . 17
  - 3.2.2 Modelo no lineal completo . . . 19

3.3 Linealización del modelo . . . 21
3.3.1 Punto de operación (péndulo invertido) . . . 21
3.3.2 Modelo linealizado . . . 21
3.4 Modelo en espacio de estados . . . 23
3.4.1 Representación en tiempo continuo . . . 23
3.4.2 Discretización del modelo . . . 24
3.5 Análisis de controlabilidad y observabilidad . . . 25
3.5.1 Controlabilidad . . . 25
3.5.2 Observabilidad . . . 25
3.5.3 Verificación en MATLAB . . . 26
3.6 Formulación del MPC sobre el sistema . . . 26
3.6.1 Ley de control y arquitectura del controlador . . . 26
3.6.2 Ausencia de acción integral . . . 26
3.7 Integración de la estimación de estados al MPC . . . 27
3.7.1 Estados estimados como entrada al controlador . . . 27
3.7.2 Diseño del estimador . . . 27

4 Simulación del sistema de control **29**
4.1 Entorno MATLAB y Simulink . . . 29
4.2 Bloques utilizados para la simulación . . . 30
4.3 Preparación de la simulación . . . 33
4.4 Pruebas de simulación sin estimador . . . 33
4.4.1 Respuesta al seguimiento de referencia . . . 33
4.4.2 Tiempos de ejecución del solver quadprog . . . 36
4.5 Pruebas de simulación con estimación de estados . . . 37
4.5.1 Validación del estimador (estados reales vs. estimados) . . . 37
4.5.2 Desempeño del MPC con estados estimados . . . 39
4.6 Análisis comparativo . . . 40

5 Implementación física del sistema de control **42**
5.1 Hardware utilizado . . . 42
5.1.1 Péndulo de Furuta (plataforma Quanser) . . . 42
5.1.2 Interfaz de adquisición de datos (USB / QUARC) . . . 43
5.2 Software: Quanser y Simulink en tiempo real . . . 43
5.3 Configuración del MPC . . . 45
5.4 Metodología de inicio: liberación manual en el ángulo de trabajo . . . 45
5.5 Resultados experimentales . . . 46
5.5.1 Efecto del horizonte de predicción . . . 47
5.5.2 Efecto del período de muestreo . . . 49
5.5.3 Efecto de las restricciones de actuación . . . 51
5.5.4 Efecto del ángulo inicial . . . 53
5.6 Tablas de resultados . . . 54
5.7 Comparación con controlador LQR . . . 55
5.8 Respuesta frente a perturbaciones . . . 59
5.9 Demostración de swing-up con MPC estándar . . . 61

6 Conclusiones y trabajos futuros **65**
6.1 Conclusiones 65
6.2 Trabajos futuros 67

BIBLIOGRAFÍA **68**

ANEXO — CÓDIGOS MATLAB **69**

Lista de Figuras

2.1 Lazo de control predictivo por modelo. Fuente: MathWorks (2025) 6

3.1 Coordenadas generalizadas y parámetros geométricos del péndulo de Furuta: ángulo del brazo θ (respecto al eje x_0, plano horizontal) y ángulo del péndulo α (respecto a la vertical z_0), ambos positivos en sentido antihorario (CCW). El equilibrio de interés en este trabajo es $\alpha = 0$ (péndulo invertido). 15
3.2 Lazo cerrado con estados estimados como entrada al controlador MPC. . . . 27

4.1 Entorno MATLAB. 30
4.2 Entorno Simulink: modelo `mpc_furuta_test_sin_pendulo.slx`. 30
4.3 Bloques utilizados en Simulink. 32
4.4 Subsistema **State X**: estimador de velocidades angulares por derivación filtrada. 32
4.5 Bloque `mpc_block`: resolución del problema QP mediante `quadprog`. 32
4.6 Respuesta temporal de θ_k y α_k ante la referencia, con y sin estimador, partiendo de $\alpha_0 = 30°$. 35
4.7 Señal de control u_k, con y sin estimador. 35
4.8 Velocidades angulares $\dot{\theta}_k$ y $\dot{\alpha}_k$ de la planta simulada, con y sin estimador. . . 36
4.9 Tiempos de ejecución del solver `quadprog` por paso de muestreo, con y sin estimador. 37
4.10 Comparación de las velocidades angulares reales y estimadas, ensayo con estimador. 38
4.11 Error de estimación de las velocidades angulares (estimado − real). 39

5.1 Sistema de péndulo de Furuta y configuración de hardware. 43
5.2 Configuración del bloque HIL Initialize para el QUBE-Servo 2. 44
5.3 Diagrama completo de Simulink para la implementación en tiempo real (configuración con liberación manual, sin swing-up automático). 45
5.4 Respuesta temporal (θ_k, α_k) para distintos valores de N. 48
5.5 Señal de control u_k para distintos valores de N. 48
5.6 Tiempos de ejecución del solver para distintos valores de N. 49
5.7 Respuesta temporal (θ_k, α_k) para distintos valores de T_s. 50
5.8 Señal de control u_k para distintos valores de T_s. 50
5.9 Tiempos de ejecución del solver para distintos valores de T_s. 51
5.10 Respuesta temporal (θ_k, α_k) para distintas restricciones de actuación $u_{\max}$. . . 52
5.11 Señal de control u_k para distintas restricciones de actuación, mostrando la saturación activa bajo $u_{\max} = 6$ V. 52

5.12 Tiempos de ejecución del solver para distintas restricciones de actuación. . . 53
5.13 Respuesta temporal (θ_k, α_k) para distintos ángulos iniciales α_0. 54
5.14 Señal de control u_k para distintos ángulos iniciales α_0. 54
5.15 Respuesta angular (θ_k, α_k) del MPC en su configuración estándar, medida desde la activación del balance. 56
5.16 Señal de control del MPC en su configuración estándar, saturada en -10 V (línea discontinua) durante el transitorio inicial sin sobrepasar el límite. . . . 57
5.17 Tiempo de ejecución del solver `quadprog` durante la fase de balance del MPC. 57
5.18 Respuesta angular (θ_k, α_k) del DLQR, medida desde la activación del balance. 58
5.19 Señal de control del DLQR, que excede ampliamente el límite de ± 10 V (línea discontinua) durante el transitorio inicial. 58
5.20 Respuesta temporal ante perturbaciones manuales, MPC vs DLQR. 60
5.21 Señal de control u_k ante perturbaciones manuales, MPC vs DLQR. 60
5.22 Diagrama completo de Simulink para la implementación en tiempo real, incluyendo el subsistema de swing-up automático. 62
5.23 Respuesta angular durante la maniobra de swing-up y la posterior estabilización con MPC estándar. 63
5.24 Señal de control durante swing-up y balance con MPC estándar. 64

CAPÍTULO

1 Introducción

En el área del control automático, el diseño de estrategias capaces de garantizar estabilidad y desempeño en sistemas no lineales e inherentemente inestables constituye uno de los problemas más desafiantes y de mayor interés tanto académico como industrial. El péndulo invertido rotatorio, conocido como péndulo de Furuta, es uno de los sistemas de referencia más utilizados para la validación de técnicas avanzadas de control. Este sistema está compuesto por un brazo horizontal accionado por un motor de corriente continua y un péndulo articulado en el extremo del brazo, libre de oscilar en el plano vertical. Su punto de equilibrio de interés, el péndulo en posición vertical invertida, es naturalmente inestable: ante cualquier perturbación, el sistema tiende a alejarse de dicho equilibrio si no se aplica una acción de control adecuada.

A las dificultades propias de su inestabilidad se suma el carácter subactuado del sistema: posee dos grados de libertad —el ángulo del brazo θ y el ángulo del péndulo α— pero solo una entrada de control, el voltaje aplicado al motor V_m. Esta combinación de no linealidad, inestabilidad y subactuación lo convierte en un banco de pruebas exigente y representativo para evaluar controladores avanzados.

En este contexto, el Control Predictivo Basado en Modelo (MPC, del inglés *Model Predictive Control*) surge como una alternativa atractiva frente a técnicas clásicas como el PID o el regulador cuadrático lineal (LQR). A diferencia de estos, el MPC utiliza explícitamente un modelo matemático del sistema para predecir su comportamiento futuro y resuelve en línea, en cada instante de muestreo, un problema de optimización que minimiza una función de costo sujeta a restricciones. Esta capacidad de incorporar restricciones de forma explícita —tales como límites de voltaje del actuador— y de anticipar la evolución del sistema lo distingue de las metodologías convencionales.

Para la resolución del problema de optimización cuadrática (QP) resultante, se emplea el solver `quadprog` de MATLAB con el algoritmo Active-Set, implementado de forma que cada iteración del controlador se ejecuta dentro del período de muestreo del sistema. Adicionalmente, dado que los encoders del sistema QUBE-Servo 2 de Quanser miden

únicamente las posiciones angulares θ y α, las velocidades angulares $\dot{\theta}$ y $\dot{\alpha}$, necesarias para el controlador, se obtienen mediante una estimación de estados implementada con bloques de función de transferencia en Simulink que realizan una derivación filtrada de las señales del encoder. Este enfoque, recomendado por el fabricante (Quanser Inc., 2014), fue adoptado por su simplicidad de implementación y su robustez ante el ruido de medición intrínseco de los encoders.
El trabajo se valida primero en simulación mediante MATLAB/Simulink y posteriormente en la plataforma física QUBE-Servo 2, logrando mantener el péndulo en la posición vertical invertida ($\alpha = 0$) de forma indefinida, sujeta únicamente a restricciones de voltaje en el actuador.

1.1 Motivaciones

El péndulo de Furuta es ampliamente reconocido en la literatura de control como un sistema de laboratorio de alto valor pedagógico e investigativo. Su dinámica concentra en un solo dispositivo compacto las principales dificultades del control moderno: inestabilidad intrínseca, no linealidad, subactuación y acoplamiento entre variables. La plataforma QUBE-Servo 2 de Quanser, utilizada en este trabajo, añade la restricción práctica de disponer únicamente de encoders de posición, sin medición directa de velocidades angulares, lo que hace necesaria una etapa de estimación de estados previa al controlador.
Desde el punto de vista del controlador, las técnicas clásicas presentan limitaciones concretas en este escenario. El control PID, diseñado para sistemas SISO, no gestiona de manera natural el acoplamiento entre θ y α. El LQR, aunque permite un diseño multivariable sistemático, no incorpora restricciones sobre la señal de control, lo que puede resultar en voltajes que excedan la capacidad del actuador. El MPC, en cambio, gestiona estas restricciones de forma explícita en su formulación, garantizando que la acción de control permanezca dentro de los límites físicos del sistema en todo instante.
La elección del algoritmo Active-Set de `quadprog` como método de optimización responde a su eficiencia computacional en problemas de tamaño reducido como el que se presenta en este trabajo, donde el costo de resolver el QP debe ser inferior al período de muestreo para garantizar la factibilidad en tiempo real.

1.2 Estado del arte

1.2.1 Control Predictivo Basado en Modelo

El Control Predictivo Basado en Modelo es una metodología de control desarrollada a finales de los años setenta (Richalet et al., 1978), cuyo principio fundamental es el uso explícito

de un modelo matemático del proceso para predecir su evolución futura y optimizar las acciones de control en un horizonte de tiempo finito. En cada instante de muestreo, el MPC resuelve un problema de optimización que minimiza una función de costo cuadrática sujeta a restricciones, aplica únicamente la primera acción de control de la secuencia óptima y repite el procedimiento en el siguiente instante (principio de horizonte deslizante).
Desde sus primeras aplicaciones industriales en procesos petroquímicos, el MPC ha evolucionado hacia sistemas de mayor velocidad y menor escala, incluyendo sistemas mecánicos inestables como el péndulo invertido (Seman et al., 2013). Su capacidad para manejar sistemas multivariable con restricciones lo posiciona como una alternativa superior a los controladores clásicos en aplicaciones donde los límites físicos del actuador son determinantes para la estabilidad del lazo cerrado.

1.2.2 Algoritmo de optimización Active-Set

Para la resolución del problema de programación cuadrática (QP) resultante del MPC, MATLAB ofrece el solver `quadprog`, cuyo algoritmo Active-Set resuelve una secuencia de subproblemas de igualdad considerando en cada iteración únicamente el conjunto de restricciones activas, o candidatas a estar activas, en el óptimo. Esta estrategia es especialmente ventajosa en el contexto del MPC, donde las restricciones activas tienden a variar poco entre instantes de muestreo consecutivos, lo que permite aprovechar la solución anterior como punto de partida (*warm-start*) y reducir el costo computacional (Nocedal and Wright, 2006). Su eficiencia en problemas de tamaño reducido, como el formulado en este trabajo, lo hace adecuado para implementaciones en tiempo real con períodos de muestreo del orden de los milisegundos.

1.2.3 Estimación de estados en sistemas subactuados

En sistemas de control como el péndulo de Furuta implementado sobre la plataforma QUBE-Servo 2, los encoders ópticos proveen mediciones directas de las posiciones angulares θ y α, pero no de las velocidades $\dot{\theta}$ y $\dot{\alpha}$. Para que el controlador MPC pueda operar con el vector de estados completo $x = [\theta,\, \alpha,\, \dot{\theta},\, \dot{\alpha}]^\top$, es necesaria una etapa de estimación de las velocidades no medidas.
Una alternativa clásica es el observador de Luenberger, que estima los estados no medidos a partir de un modelo dinámico del sistema y de la comparación entre la salida medida y la predicha (Luenberger, 1971). Sin embargo, en este trabajo se optó por una aproximación más directa y de menor complejidad de diseño: la derivación filtrada de las señales del encoder mediante bloques de función de transferencia en Simulink. Esta técnica, recomendada explícitamente por el fabricante en la documentación oficial del sistema (Quanser Inc., 2014), consiste en aproximar la derivada de la señal de posición

mediante un filtro que evita la amplificación del ruido de alta frecuencia inherente a una diferenciación pura, obteniendo así estimaciones estables y confiables de $\dot{\theta}$ y $\dot{\alpha}$.

1.3 Objetivos del trabajo

El objetivo general de este trabajo es diseñar e implementar un controlador predictivo basado en modelo para el sistema de péndulo de Furuta QUBE-Servo 2, logrando estabilizar el péndulo en su posición vertical invertida y mantenerlo en dicho equilibrio de forma indefinida, respetando las restricciones de voltaje del actuador.
Esto se alcanzará mediante los siguientes objetivos específicos:

- Obtener el modelo matemático no lineal del sistema de péndulo de Furuta mediante el formalismo de Euler-Lagrange, a partir de la documentación oficial del fabricante (Quanser Inc., 2014).
- Linealizar el modelo no lineal en torno al punto de equilibrio inestable ($\theta_0 = 0$, $\alpha_0 = 0$) y obtener su representación en espacio de estados en tiempo continuo y discreto.
- Verificar las propiedades de controlabilidad y observabilidad del sistema linealizado en MATLAB.
- Diseñar un regulador cuadrático lineal discreto (DLQR) sobre el modelo linealizado (A_d, B_d), utilizado para determinar matrices de ponderación Q y R que logren un desempeño de balance adecuado, y reutilizarlas directamente —junto con el costo terminal derivado de su ecuación de Riccati asociada—en la función de costo del controlador MPC.
- Implementar la estimación de velocidades angulares mediante derivación filtrada de las señales del encoder en Simulink.
- Formular el problema de Control Predictivo por Modelo como un problema de programación cuadrática y resolverlo mediante el solver `quadprog` de MATLAB con el algoritmo Active-Set.
- Validar el controlador en simulación mediante MATLAB/Simulink, analizando el efecto del horizonte de predicción y el período de muestreo sobre el desempeño del sistema.
- Implementar el controlador en la plataforma física QUBE-Servo 2 y evaluar experimentalmente su desempeño variando el horizonte de predicción N, el período de muestreo T_s y las restricciones de voltaje del actuador ($u \in [-10,\ 10]$ V).

CAPÍTULO

2 Fundamentos Teóricos

2.1 Control Predictivo por Modelo

El Control Predictivo por Modelo (MPC) constituye una de las estrategias de control óptimo más utilizadas en aplicaciones modernas debido a su capacidad para manejar de manera explícita restricciones sobre las entradas, estados y salidas del sistema. En cada instante de muestreo, MPC utiliza un modelo matemático para predecir la evolución futura del sistema y resuelve un problema de optimización que minimiza una función de costo sujeta a restricciones.

En cada instante de muestreo, el controlador resuelve el problema de optimización descrito en las subsecciones siguientes para obtener la secuencia óptima de acciones de control sobre el horizonte de predicción, pero solo aplica a la planta la primera de estas acciones, descartando el resto. En el siguiente instante de muestreo se mide o estima nuevamente el estado del sistema y se replantea el problema de optimización completo con esta información actualizada. Esta estrategia, conocida como horizonte deslizante (*receding horizon*), introduce una retroalimentación implícita que permite corregir en cada paso las desviaciones respecto de lo predicho, ya sea por errores de modelado, perturbaciones no consideradas o el hecho de que el sistema real es no lineal mientras que el modelo utilizado para predecir es lineal (Capítulo 3). Sin este mecanismo de recálculo, el controlador aplicaría en lazo abierto una secuencia de acciones calculada una única vez, lo que resultaría poco robusto ante cualquier desviación del comportamiento real respecto del modelo.

La formulación del problema de optimización puede plantearse de dos maneras equivalentes según las variables de decisión que se utilicen: si se consideran únicamente las entradas futuras como incógnitas y se eliminan los estados futuros mediante sustitución recursiva del modelo, se obtiene la *formulación densa*; si en cambio se mantienen tanto los estados como las entradas como incógnitas del problema, se obtiene la *formulación dispersa* (*sparse*). La formulación dispersa resulta más eficiente numéricamente para horizontes de predicción

largos, ya que preserva la estructura de bloques dispersos del problema, pero introduce una complejidad de implementación adicional. Dado que en este trabajo se emplea un horizonte reducido ($N = 3$), se opta por la formulación densa, presentada a continuación, la cual resulta más directa de implementar y no presenta las limitaciones de acondicionamiento numérico que la motivan para horizontes largos.

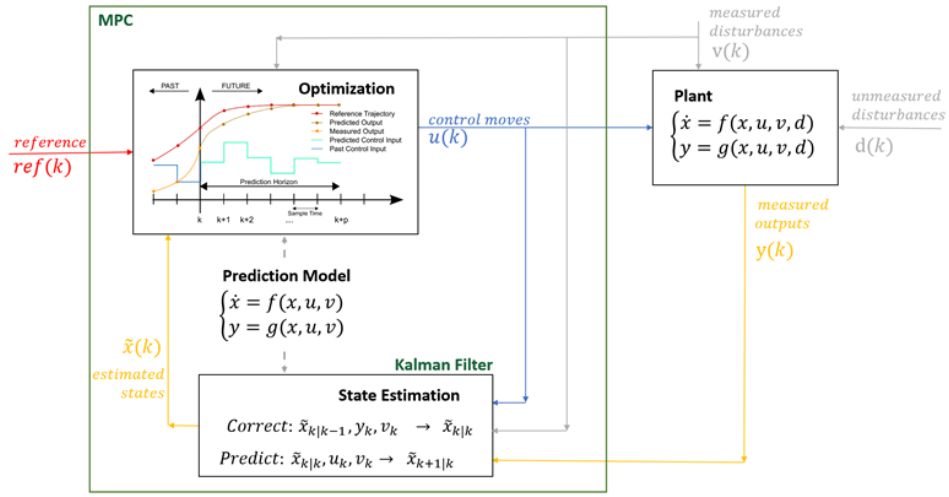


Figura 2.1: Lazo de control predictivo por modelo. Fuente: MathWorks (2025)

2.1.1 Fundamentos de MPC

En cada instante de muestreo k, el controlador MPC resuelve un problema de optimización sobre un horizonte de predicción finito de N pasos, obteniendo una secuencia óptima de acciones de control futuras $u_k, u_{k+1}, \ldots, u_{k+N-1}$. Sin embargo, solo se aplica al sistema la primera de estas acciones, u_k; en el siguiente instante de muestreo el procedimiento se repite por completo con la nueva medición del estado, principio conocido como horizonte deslizante (*receding horizon*) (MathWorks, 2025). La función de costo penaliza cuadráticamente la desviación de los estados respecto al punto de operación y el esfuerzo de control a lo largo del horizonte. En este trabajo, el horizonte de control se fija igual al horizonte de predicción N, es decir, se optimizan N acciones de control libres, siguiendo la formulación densa presentada en la siguiente subsección.

2.1.2 Definición del problema de control

Considerando un sistema lineal discreto en espacio de estados:

$$x_{k+1} = A_d x_k + B_d u_k \tag{2.1}$$

donde $x_k = [\theta, \alpha, \dot{\theta}, \dot{\alpha}]^T$ es el vector de estados del péndulo de Furuta y u_k es el voltaje

aplicado al motor. El controlador MPC busca, en cada instante k, la secuencia de entradas futuras $U = [u_k, u_{k+1}, \ldots, u_{k+N-1}]^T$ que minimiza la función de costo cuadrática

$$J = \sum_{i=0}^{N-1} \left(x_{k+i}^T Q\, x_{k+i} + u_{k+i}^T R\, u_{k+i} \right) + x_{k+N}^T Q_N\, x_{k+N} \tag{2.2}$$

sujeta a la dinámica (2.1) y a las restricciones de actuación del sistema. Las matrices de ponderación utilizadas son $Q = \mathrm{diag}(5,\ 1,\ 1,\ 20)$ y $R = 1$, que penalizan con mayor peso el error angular del péndulo α y, en menor medida, el ángulo del brazo θ y las velocidades angulares. La matriz de costo terminal Q_N corresponde a la solución de la ecuación de Riccati discreta (DARE) asociada al regulador DLQR sobre el par (A_d, B_d) con las mismas matrices Q y R.

Esta elección de Q_N no es arbitraria: en la teoría de estabilidad de MPC para sistemas lineales, si Q_N se escoge como la solución de la DARE, la función de costo óptima $J^\circ(x_k)$ del problema (2.2) puede utilizarse como función de Lyapunov del lazo cerrado. En efecto, la ley de control óptima sin restricciones activas coincide localmente con la ley LQR $u = -L_\infty x$, de modo que J° hereda la propiedad de decrecimiento estricto $J^\circ(x_{k+1}) - J^\circ(x_k) \leq -x_k^T Q\, x_k$ propia del funcional de costo del LQR infinito, lo que garantiza estabilidad asintótica (exponencial, dado que $Q, Q_N \succ 0$) del origen para el sistema en lazo cerrado, dentro de la región donde las restricciones de entrada no están activas. Esta es la motivación teórica para utilizar el costo terminal DLQR en lugar de, por ejemplo, $Q_N = Q$.

2.1.3 Formulación densa

La formulación densa consiste en eliminar los estados futuros del problema de optimización, expresándolos únicamente en función del estado actual x_k y de la secuencia de entradas futuras U. Aplicando recursivamente la ecuación (2.1) se obtiene, para el estado i pasos hacia adelante,

$$x_{k+i} = A_d^i x_k + \sum_{j=1}^{i} A_d^{i-j} B_d\, u_{k+j-1}, \qquad i = 1, \ldots, N \tag{2.3}$$

lo que permite escribir de forma compacta el vector de estados predichos como

$$X = \bar{A}\, x_k + \bar{O}\, U, \tag{2.4}$$

donde $X = [x_{k+1}^T, \ldots, x_{k+N}^T]^T$ apila los estados predichos, y las matrices $\bar{A}$ y $\bar{O}$ se construyen por bloques como

$$\bar{A} = \begin{bmatrix} A_d \\ A_d^2 \\ \vdots \\ A_d^N \end{bmatrix}, \qquad \bar{O} = \begin{bmatrix} B_d & 0 & \cdots & 0 \\ A_d B_d & B_d & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A_d^{N-1} B_d & A_d^{N-2} B_d & \cdots & B_d \end{bmatrix}. \tag{2.5}$$

Esta construcción se implementa directamente en MATLAB acumulando potencias de A_d y sus productos con B_d por bloques, lo que permite expresar el problema de optimización completo únicamente en términos de la secuencia de entradas futuras U, tal como se requiere en la formulación densa del QP. Cabe notar que, al ser el péndulo de Furuta un sistema de lazo abierto inestable (Capítulo 3), las potencias A_d^i que componen $\bar{A}$ y $\bar{O}$ crecen con i; esto no genera problemas de acondicionamiento numérico relevantes en este trabajo dado el horizonte reducido utilizado ($N = 3$), pero es una limitación conocida de la formulación densa frente a la formulación dispersa (*sparse*) para horizontes largos sobre plantas inestables.

2.1.4 Seguimiento de referencia

En su forma general, el problema de MPC se plantea en términos del error respecto a una referencia x_{ref}, definiendo el error de seguimiento como $e_{k+i} = x_{k+i} - x_{\text{ref}}$ y sustituyendo este error en la función de costo (2.2). En este trabajo, el objetivo de control es estabilizar el péndulo en su punto de equilibrio invertido, por lo que la referencia de estados es fija e igual al origen, $x_{\text{ref}} = 0$. Bajo esta condición, el problema de seguimiento se reduce a un problema de regulación: el error de seguimiento coincide con el propio vector de estados, $e_{k+i} = x_{k+i}$, y no es necesario introducir un sistema aumentado ni una acción integral adicional, ya que el equilibrio deseado corresponde directamente al origen del modelo linealizado.

2.1.5 Restricción de desigualdad

La única restricción de actuación considerada corresponde a la saturación de voltaje del amplificador que alimenta al motor del péndulo de Furuta, la cual se expresa como una cota inferior y superior sobre cada entrada del horizonte:

$$u_{\min} \leq u_{k+i} \leq u_{\max}, \qquad i = 0, \ldots, N-1, \tag{2.6}$$

con $u_{\min} = -10$ V y $u_{\max} = 10$ V. En notación matricial, esta restricción puede escribirse de forma equivalente como $G\,U \leq b$, con $G = [I_N;\ -I_N]$ y $b = [u_{\max}\mathbf{1};\ -u_{\min}\mathbf{1}]$; sin embargo, en la implementación no se construyen explícitamente las matrices G y b, sino que las cotas $u_{\min}$

y $u_{\max}$ se entregan directamente al solver `quadprog` de MATLAB a través de sus argumentos `lb` y `ub`, lo que resulta computacionalmente más eficiente que una formulación general de desigualdades. Esta simplificación es válida porque, a diferencia de formulaciones más generales de MPC que también acotan los estados predichos ($x_{k+i} \in \mathcal{X}$) o imponen un conjunto terminal ($x_{k+N} \in \mathcal{X}_f$) para garantizar factibilidad recursiva, en este trabajo no se impusieron restricciones sobre los estados, únicamente sobre la entrada.

2.1.6 Planteamiento final del problema QP

El problema de programación cuadrática resultante queda planteado como:

$$\min_{U} \ \frac{1}{2} U^T H\, U + f^T U \tag{2.7}$$

sujeto a:

$$G\, U \leq b \tag{2.8}$$

Sustituyendo la predicción densa (2.4) en la función de costo (2.2), las matrices del problema QP quedan definidas como

$$H = 2\left(\bar{O}^T \bar{Q}\, \bar{O} + \bar{R}\right), \qquad f = 2\,\bar{O}^T \bar{Q}\, \bar{A}\, e_k, \tag{2.9}$$

donde $e_k = x_k - x_{\text{ref}}$ es el error respecto a la referencia (igual a x_k en este trabajo, según lo indicado en la Sección de seguimiento de referencia), y las matrices de costo apiladas se construyen como $\bar{Q} = \text{blkdiag}(Q, \ldots, Q, Q_N)$ ($N-1$ bloques Q más el bloque terminal Q_N) y $\bar{R} = \text{blkdiag}(R, \ldots, R)$ (N bloques R). Esta expresión de H y f es exactamente la que se implementa en el bloque de control MPC en Simulink: H se calcula de forma *offline* junto con $\bar{A}$, $\bar{O}$, $\bar{Q}$ y $\bar{R}$ antes de iniciar la simulación, mientras que f se recalcula en línea en cada instante de muestreo a partir del estado (o estado estimado) actual. Nótese que la matriz H es constante (no depende de x_k) precisamente porque se calcula *offline*, mientras que f sí depende linealmente del estado y debe recalcularse en cada iteración del controlador.

2.1.7 Resultado final y horizonte deslizante

Resuelto el problema de optimización (2.7) mediante el solver `quadprog` (Sección 2.2), se obtiene la secuencia óptima de control U^* sobre todo el horizonte de predicción N. De esta secuencia, únicamente el primer elemento se aplica efectivamente a la planta; los $N-1$ elementos restantes se descartan y se recalculan en el siguiente instante de muestreo a partir del estado actualizado del sistema.
En términos del algoritmo implementado, cada ciclo de control ejecuta los siguientes pasos:

1. Medir o estimar el estado actual x_k (Sección 2.4).

2. Actualizar el vector f del problema QP a partir de x_k, manteniendo fija la matriz H (calculada una única vez, fuera de línea).

3. Resolver el problema de optimización cuadrática (2.7) mediante `quadprog`, obteniendo U^*.

4. Aplicar a la planta únicamente la primera componente de U^*.

5. Esperar al siguiente instante de muestreo y repetir el procedimiento con el nuevo estado.

Esta estrategia de horizonte deslizante es la que efectivamente confiere a MPC su carácter de controlador realimentado: aunque en cada ciclo se resuelve un problema de optimización en lazo abierto sobre una ventana de predicción finita, el recálculo continuo a partir del estado medido corrige las desviaciones entre el comportamiento predicho por el modelo lineal y el comportamiento real del péndulo de Furuta, que es no lineal. El desempeño resultante de esta estrategia, en particular la sensibilidad a los parámetros N y T_s, se evalúa experimentalmente en el Capítulo 5.

2.2 Método de optimización

2.2.1 Método de optimización Active-Set de quadprog

El problema QP (2.7)–(2.8) es un problema de optimización convexa (dado que $H \succeq 0$ por construcción, al ser $\bar{Q} \succeq 0$ y $\bar{R} \succ 0$), por lo que las condiciones de Karush-Kuhn-Tucker (KKT) son necesarias y suficientes para caracterizar su óptimo global (Nocedal and Wright, 2006). Para el problema (2.7)–(2.8), dichas condiciones se escriben como

$$HU^\star + f + G^T\lambda^\star = 0, \qquad GU^\star \leq b, \qquad \lambda^\star \geq 0, \qquad \lambda_j^\star\,(G_jU^\star - b_j) = 0 \;\; \forall j, \tag{2.10}$$

donde $\lambda^\star$ es el vector de multiplicadores de Lagrange asociado a las restricciones de desigualdad, y la última condición (holgura complementaria) establece que cada restricción está activa ($\lambda_j^\star > 0$) o inactiva ($G_jU^\star < b_j$, con $\lambda_j^\star = 0$), pero no ambas simultáneamente.
El algoritmo Active-Set resuelve (2.10) de forma iterativa: en cada iteración mantiene un *conjunto activo* $\mathcal{A}_i \subseteq \{1, \ldots, m\}$ de restricciones que se suponen activas con igualdad, y resuelve el sistema lineal de KKT restringido a dicho conjunto —equivalente a un problema de mínimos cuadrados con restricciones de igualdad—, obteniendo una dirección de búsqueda; luego actualiza $\mathcal{A}_i$ agregando restricciones que se violarían al moverse en esa dirección, o removiendo aquellas cuyo multiplicador de Lagrange resulte negativo (violando la condición $\lambda^\star \geq 0$), hasta satisfacer (2.10) por completo. Esta estrategia resulta especialmente adecuada para problemas de tamaño reducido como el formulado en este

trabajo (horizonte $N = 3$, una sola entrada de control, y por lo tanto a lo sumo $2N = 6$ restricciones de caja), en los que el número de restricciones activas es pequeño y el costo computacional por iteración es bajo, permitiendo resolver el QP dentro del período de muestreo del sistema.
En la implementación, el solver se invoca con las opciones `Algorithm = 'active-set'` y `Display = 'off'`, y se entrega como punto inicial la solución nula; adicionalmente, se incorpora un mecanismo de respaldo que, en caso de que `quadprog` no converja (`exitflag` distinto de 1), calcula la solución sin restricciones mediante $U_{\text{ns}} = -H^{-1}f$ —que corresponde exactamente a la solución de (2.10) cuando el conjunto activo es vacío ($\lambda^\star = 0$)— y satura la primera componente dentro de los límites $[u_{\min}, u_{\max}]$, garantizando que siempre se aplique una acción de control válida al sistema.

2.3 Muestreo del sistema

2.3.1 Fundamentos del muestreo en sistemas de control

De acuerdo con el criterio de Nyquist-Shannon, la frecuencia de muestreo de un sistema de control digital debe ser considerablemente mayor que el doble del ancho de banda de la dinámica que se desea controlar, con el fin de evitar pérdida de información y efectos de *aliasing*. En el caso del péndulo de Furuta, la dinámica del péndulo invertido en torno al punto de equilibrio inestable presenta constantes de tiempo del orden de decenas de milisegundos, dada la baja inercia y longitud reducida del péndulo ($L_p = 0.129$ m); de hecho, el valor propio inestable del modelo linealizado en lazo abierto, $\lambda \approx 16.17\ \mathrm{rad/s}$ (Capítulo 3), corresponde a una constante de tiempo característica de solo $1/\lambda \approx 62$ ms. Por este motivo se seleccionó un período de muestreo $T_s = 2$ ms, considerablemente más rápido que la dinámica dominante del sistema (más de 30 veces menor que la constante de tiempo ≈ 62 ms del modo inestable), lo que además resulta compatible con el tiempo de cómputo requerido por el solver `quadprog` en cada iteración del controlador. Este valor coincide, además, con el período de muestreo utilizado en el controlador LQR de referencia de Quanser para el mismo hardware (`q_qube_bal_lqr.mdl`, incluido en el *courseware* del QUBE-Servo 2), lo que respalda su elección como un punto de operación validado experimentalmente por el fabricante para esta planta.

2.3.2 Muestreo utilizando Retenedor de Orden Cero (ZOH)

El retenedor de orden cero (*Zero-Order Hold*, ZOH) es el método de discretización estándar para sistemas de control digital, y modela el comportamiento real de un convertidor digital-analógico: la señal de control calculada por el controlador se mantiene constante durante todo el período de muestreo T_s, hasta que se actualiza en el siguiente instante de muestreo.

Esta hipótesis es consistente con la forma en que el amplificador de potencia y la tarjeta de adquisición de datos aplican el voltaje al motor del péndulo de Furuta, por lo que el modelo discreto utilizado para el diseño del controlador se obtiene discretizando el modelo continuo mediante ZOH.

2.3.3 Derivación de las matrices discretas Φ y Γ

Para justificar estas expresiones a partir del modelo continuo, considérese el sistema lineal invariante en el tiempo

$$\dot{x}(t) = A_c\, x(t) + B_c\, u(t), \tag{2.11}$$

y supóngase que la entrada permanece constante entre instantes de muestreo consecutivos, tal como impone el retenedor de orden cero:

$$u(t) = u_k, \qquad t \in [kT_s,\, (k+1)T_s). \tag{2.12}$$

Multiplicando (2.11) por el factor integrante $e^{-A_c t}$ y reagrupando, el lado izquierdo se reconoce como la derivada de un producto:

$$e^{-A_c t}\dot{x}(t) - e^{-A_c t}A_c\, x(t) = \frac{d}{dt}\left(e^{-A_c t}x(t)\right) = e^{-A_c t}B_c\, u(t). \tag{2.13}$$

Integrando ambos lados entre $t_k = kT_s$ y $t_{k+1} = (k+1)T_s$, y usando que $u(t) = u_k$ es constante en este intervalo por (2.12):

$$e^{-A_c t_{k+1}}x(t_{k+1}) - e^{-A_c t_k}x(t_k) = \left(\int_{t_k}^{t_{k+1}} e^{-A_c \tau}\, d\tau\right) B_c\, u_k. \tag{2.14}$$

Multiplicando ambos lados por $e^{A_c t_{k+1}}$ y despejando $x(t_{k+1})$:

$$x(t_{k+1}) = e^{A_c T_s}\, x(t_k) + \left(\int_{t_k}^{t_{k+1}} e^{A_c (t_{k+1}-\tau)}\, d\tau\right) B_c\, u_k. \tag{2.15}$$

Realizando el cambio de variable $\eta = t_{k+1} - \tau$ en la integral (de modo que $d\eta = -d\tau$, y los límites $\tau \in [t_k, t_{k+1}]$ se transforman en $\eta \in [0, T_s]$), se obtiene finalmente $x_{k+1} = \Phi\, x_k + \Gamma\, u_k$, de donde se identifican las matrices
A partir del modelo continuo (A_c, B_c), las matrices discretas se obtienen mediante:

$$\Phi = e^{A_c T_s}, \qquad \Gamma = \int_0^{T_s} e^{A_c \tau}\, d\tau\ B_c \tag{2.16}$$

Estas expresiones se obtienen integrando la solución exacta de la ecuación de estado continua $\dot{x} = A_c x + B_c u$ sobre un período de muestreo, bajo el supuesto de entrada

constante por tramos propio del ZOH; Φ describe la evolución libre del sistema durante T_s, mientras que Γ pondera el efecto acumulado de la entrada constante u_k durante ese mismo intervalo. En la práctica, este cálculo no se realiza de forma manual: se utiliza la función `c2d` de MATLAB sobre el modelo en espacio de estados continuo del péndulo de Furuta, con el período de muestreo $T_s = 2$ ms y el método `'zoh'` por defecto, obteniendo directamente las matrices discretas $A_d = \Phi$ y $B_d = \Gamma$ utilizadas en el resto del diseño del controlador.

2.4 Estimación de estados

En sistemas reales no siempre es posible medir directamente todos los estados del sistema. Para el péndulo de Furuta, se requiere estimar los estados no medibles a partir de las salidas disponibles.

2.4.1 Fundamentos de la estimación de estados

Existen distintos enfoques para reconstruir estados no medidos de un sistema dinámico a partir de sus salidas. El observador de Luenberger es un enfoque determinístico basado en el modelo en espacio de estados del sistema: a partir de una copia del modelo y de una ganancia de corrección L, diseñada típicamente por ubicación de polos, el observador corrige su estimación en función del error entre la salida medida y la salida predicha por el modelo. El filtro de Kalman, en cambio, es un estimador estocástico óptimo bajo el supuesto de ruido gaussiano en el proceso y en la medición, que requiere caracterizar dichas incertidumbres mediante matrices de covarianza. Ambos enfoques requieren un modelo del sistema y presentan una complejidad de diseño e implementación no despreciable. En este trabajo, dado que únicamente es necesario estimar las velocidades angulares $\dot{\theta}$ y $\dot{\alpha}$ a partir de las posiciones angulares medidas directamente por los encoders θ y α, se optó por una alternativa más simple, de tipo derivación filtrada, en lugar de un observador basado en modelo, considerando además las estrictas restricciones de tiempo real impuestas por el período de muestreo de 2 ms.

2.4.2 Técnica de estimación utilizada: enfoque cinemático determinista

La técnica implementada sigue un enfoque cinemático determinista: los estados no medidos se reconstruyen únicamente a partir de la geometría del movimiento y de las mediciones directas de los sensores, sin recurrir al modelo dinámico de masas y torques del péndulo. Concretamente, se trata de una derivación filtrada de las señales de posición angular (θ, α) entregadas por los encoders, mediante bloques de función de transferencia en Simulink. Cada velocidad angular se obtiene aproximando la derivada de la señal de

posición correspondiente mediante un filtro de la forma

$$G_d(s) = \frac{s}{\tau s + 1}, \tag{2.17}$$

que corresponde a un diferenciador de primer orden con un polo de filtrado en $-1/\tau$. A diferencia de una derivación pura ($G_d(s) = s$), este filtro atenúa las componentes de alta frecuencia asociadas al ruido de cuantización propio de los encoders ópticos, evitando que dicho ruido se amplifique al estimar $\dot{\theta}$ y $\dot{\alpha}$. La constante de tiempo τ se selecciona como un compromiso entre atenuación de ruido y retardo de fase introducido por el filtro: valores de τ mayores atenúan más el ruido pero introducen mayor retardo en la estimación, mientras que valores menores se aproximan a una derivada pura y son más sensibles al ruido de medición. Este enfoque es el recomendado explícitamente por Quanser en la documentación oficial del sistema QUBE-Servo 2 para la estimación de velocidades angulares a partir de los encoders.

2.4.3 Diferencias respecto a un observador de Luenberger

La derivación filtrada utilizada en este trabajo difiere de forma fundamental de un observador de Luenberger (Luenberger, 1971) en que no emplea el modelo dinámico del sistema ni realiza una corrección basada en el error de salida: se trata de un procesamiento de señal aplicado directamente sobre las mediciones de posición, sin retroalimentación del modelo. Esto la hace considerablemente más simple de diseñar e implementar —no requiere calcular una ganancia de corrección ni verificar la observabilidad del sistema para su diseño— y computacionalmente más económica, al no involucrar operaciones matriciales en cada paso de muestreo. Como contrapartida, no ofrece garantías de optimalidad estadística ni de robustez ante incertidumbre en los parámetros del modelo, y su desempeño depende directamente de la calidad de la señal medida y de la elección adecuada de la constante de tiempo τ. Dado que los encoders ópticos del QUBE-Servo 2 presentan un nivel de ruido de medición relativamente bajo, esta aproximación resultó suficiente para obtener estimaciones de velocidad angular adecuadas para el desempeño del controlador MPC, sin la complejidad adicional de diseñar e implementar un observador basado en modelo.

CAPÍTULO

3 Planta de interés: Péndulo de Furuta

3.1 Descripción del sistema

El péndulo de Furuta (también conocido como péndulo rotatorio invertido) es un sistema mecánico subactuado compuesto por dos eslabones rígidos: un brazo horizontal accionado por un motor eléctrico que gira en el plano horizontal, y un péndulo libre articulado en el extremo del brazo que oscila en el plano vertical. El sistema posee dos grados de libertad pero solo una entrada de control, lo que lo clasifica como subactuado. La Figura 3.1 resume las coordenadas generalizadas y los parámetros geométricos que se utilizan en el resto del capítulo.

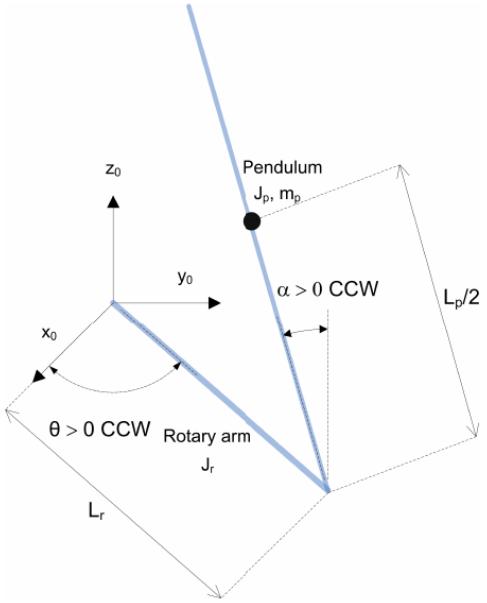


Figura 3.1: Coordenadas generalizadas y parámetros geométricos del péndulo de Furuta: ángulo del brazo θ (respecto al eje x_0, plano horizontal) y ángulo del péndulo α (respecto a la vertical z_0), ambos positivos en sentido antihorario (CCW). El equilibrio de interés en este trabajo es $\alpha = 0$ (péndulo invertido).

3.1.1 Componentes mecánicos y grados de libertad

Las variables generalizadas del sistema son:

- θ: ángulo de rotación del brazo horizontal [rad].
- α: ángulo del péndulo respecto a la vertical invertida [rad], con $\alpha = 0$ correspondiendo a la posición vertical hacia arriba.

Ambos ángulos se definen como positivos en sentido antihorario (*counter-clockwise*, CCW; ver Figura 3.1), y una tensión positiva aplicada al motor gira el brazo en ese mismo sentido, siguiendo la convención del fabricante Quanser (Quanser Inc., 2014) para la plataforma QUBE-Servo 2.

Table 3.1: Parámetros del péndulo de Furuta (plataforma QUBE-Servo 2).

Símbolo	Descripción	Valor	Unidad
R_m	Resistencia del motor	8.4	[Ω]
k_t	Constante de torque	0.042	[N·m/A]
k_m	Constante de fuerza contraelectromotriz	0.042	[V·s/rad]
M_r	Masa del brazo	0.095	[kg]
L_r	Longitud del brazo	0.085	[m]
J_r	Momento de inercia del brazo respecto a su centro	5.72×10^{-5}	[kg·m^2]
D_r	Coeficiente de fricción viscosa del brazo	0	[N·m·s/rad]
M_p	Masa del péndulo	0.024	[kg]
L_p	Longitud del péndulo	0.129	[m]
J_p	Momento de inercia del péndulo respecto a su centro	3.33×10^{-5}	[kg·m^2]
D_p	Coeficiente de fricción viscosa del péndulo	0	[N·m·s/rad]
g	Aceleración de gravedad	9.81	[m/s^2]

Los momentos de inercia J_r y J_p se calculan respecto al centro de masa de cada eslabón, modelado como una barra delgada uniforme: $J_r = M_r L_r^2/12$ y $J_p = M_p L_p^2/12$.

3.1.2 Sensores y actuador

Las posiciones θ y α se miden con encoders ópticos incrementales de cuadratura, montados en el eje del brazo y en la articulación del péndulo. Cada encoder entrega 512 líneas por revolución que, tras decodificación por cuadratura ($\times 4$), se convierten en 2048 cuentas/revolución. El bloque de lectura convierte cuentas a radianes con ganancias $\pm 2\pi/512/4$, de signo opuesto entre canales, de modo que θ y α resulten positivos en sentido antihorario (Figura 3.1).

El encoder del péndulo entrega una medición continua α_{full}, definida como cero cuando el péndulo cuelga hacia abajo. El ángulo respecto a la posición invertida usado en el modelo se obtiene con

$$\alpha = (\alpha_{\text{full}} \bmod 2\pi) - \pi, \tag{3.1}$$

de modo que $\alpha = 0$ corresponde a la vertical invertida; la función módulo mantiene la medición acotada entre 0 y 2π sin importar el sentido de giro usado para llevar el péndulo a esa posición.
El actuador es un motor DC acoplado al brazo mediante una caja reductora integrada; su par τ es proporcional a la corriente de armadura mediante la constante k_t. Se alimenta con un amplificador de potencia lineal limitado en voltaje, lo que impone la restricción $-10\ \text{V} \leq u \leq 10\ \text{V}$ (Capítulo 2). La señal de voltaje se invierte (ganancia -1) antes del amplificador, de modo que un voltaje de control positivo gire el brazo en sentido antihorario. La dinámica eléctrica del motor (R_m, k_m) se incorpora directamente al modelo mecánico en la Sección 3.3.2.

3.2 Modelo matemático del péndulo de Furuta

3.2.1 Ecuaciones de movimiento (formulación de Euler-Lagrange)

El formalismo de Euler-Lagrange permite obtener las ecuaciones de movimiento de un sistema mecánico a partir de una única función escalar, el **Lagrangiano**

$$\mathcal{L} = T - V, \tag{3.2}$$

donde T es la **energía cinética total** del sistema y V es su **energía potencial total**, ambas expresadas en función de las coordenadas generalizadas $q = [\theta,\ \alpha]^T$ y de sus velocidades $\dot{q} = [\dot{\theta},\ \dot{\alpha}]^T$. Para sistemas sometidos a fuerzas no conservativas —como el par motor y la fricción viscosa en este caso—, las ecuaciones de movimiento se obtienen de

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = Q_i, \quad i = 1, 2, \tag{3.3}$$

donde Q_i agrupa precisamente las fuerzas generalizadas que no derivan de $\mathcal{L}$: el par motor τ y la fricción viscosa $-D_r\dot{\theta}$ actuando sobre θ, y la fricción viscosa $-D_p\dot{\alpha}$ actuando sobre α (el péndulo no está actuado directamente), es decir

$$Q_\theta = \tau - D_r\dot{\theta}, \qquad Q_\alpha = -D_p\dot{\alpha}. \tag{3.4}$$

Antes de poder aplicar (3.3) es necesario construir explícitamente T y V a partir de la geometría del sistema, lo que se desarrolla a continuación.

Cinemática del sistema. El pivote del péndulo, ubicado en el extremo del brazo, gira junto con este en el plano horizontal a una distancia fija L_r del eje vertical, por lo que su posición es

$$\Big(x_P,\ y_P,\ z_P\Big) = \Big(L_r\cos\theta,\ \ L_r\sin\theta,\ \ 0\Big). \tag{3.5}$$

El centro de masa del péndulo se encuentra a una distancia $L_p/2$ de este pivote, y el péndulo está articulado de modo que su inclinación α ocurre en el plano vertical que gira junto con el brazo (es decir, el eje de rotación del péndulo es perpendicular tanto a la vertical como al brazo). Su posición en el sistema de referencia fijo resulta entonces

$$x_{cm} = L_r\cos\theta + \frac{L_p}{2}\sin\alpha\sin\theta, \tag{3.6}$$

$$y_{cm} = L_r\sin\theta - \frac{L_p}{2}\sin\alpha\cos\theta, \tag{3.7}$$

$$z_{cm} = \frac{L_p}{2}\cos\alpha. \tag{3.8}$$

Es importante notar que la altura z_{cm} depende únicamente de $\cos\alpha$ y no de θ: como el brazo se mueve enteramente en el plano horizontal, girarlo no cambia la altura del pivote del péndulo ni, en consecuencia, la de su centro de masa. Las componentes horizontales x_{cm} e y_{cm}, en cambio, sí dependen tanto de θ como de α; son estas dos componentes las que, al derivarlas respecto al tiempo, introducen el acoplamiento entre $\dot\theta$ y $\dot\alpha$ que aparecerá en la energía cinética.

Energía cinética. La energía cinética total del sistema se descompone en tres contribuciones independientes: la rotación propia del brazo, la rotación propia del péndulo en torno a su propio centro de masa, y la traslación del centro de masa del péndulo.
La rotación del brazo, con velocidad angular $\dot\theta$ y momento de inercia J_r respecto a su eje de giro, aporta

$$T_{\text{brazo}} = \frac{1}{2}J_r\dot\theta^2. \tag{3.9}$$

El péndulo gira además sobre sí mismo a razón $\dot\alpha$, con momento de inercia J_p respecto a su propio centro de masa, aportando una segunda contribución puramente rotacional:

$$T_{\text{rot},p} = \frac{1}{2}J_p\dot\alpha^2. \tag{3.10}$$

La tercera contribución es la traslación del centro de masa del péndulo (masa M_p), que se obtiene derivando (3.6)–(3.8) respecto al tiempo y sumando el cuadrado de las tres componentes de velocidad, $v^2 = \dot x_{cm}^2 + \dot y_{cm}^2 + \dot z_{cm}^2$. Tras simplificar mediante identidades

trigonométricas (los productos cruzados en $\sin\theta\cos\theta$ se cancelan exactamente), resulta

$$v^2 = L_r^2\dot{\theta}^2 + \left(\frac{L_p}{2}\right)^2 \sin^2\alpha\,\dot{\theta}^2 + \left(\frac{L_p}{2}\right)^2 \dot{\alpha}^2 - L_r L_p \cos\alpha\,\dot{\theta}\dot{\alpha}, \tag{3.11}$$

de donde la energía cinética de traslación del péndulo es

$$T_{\text{tras}} = \frac{1}{2}M_p v^2 = \frac{1}{2}M_p L_r^2\dot{\theta}^2 + \frac{1}{2}M_p\left(\frac{L_p}{2}\right)^2 \sin^2\alpha\,\dot{\theta}^2 + \frac{1}{2}M_p\left(\frac{L_p}{2}\right)^2 \dot{\alpha}^2 - M_p\frac{L_p}{2}L_r\cos\alpha\,\dot{\theta}\dot{\alpha}. \tag{3.12}$$

Sumando las tres contribuciones, $T = T_{\text{brazo}} + T_{\text{rot},p} + T_{\text{tras}}$, y agrupando los términos en $\dot{\theta}^2$ y en $\dot{\alpha}^2$, se obtiene la energía cinética total del sistema:

$$T = \frac{1}{2}\left(J_r + M_p L_r^2\right)\dot{\theta}^2 + \frac{1}{2}\left(J_p + M_p\left(\frac{L_p}{2}\right)^2\right)\dot{\alpha}^2 + \frac{1}{2}M_p\left(\frac{L_p}{2}\right)^2 \sin^2\alpha\,\dot{\theta}^2 - M_p\frac{L_p}{2}L_r\cos\alpha\,\dot{\theta}\dot{\alpha}. \tag{3.13}$$

El primer término agrupa la inercia propia del brazo, J_r, con la inercia de traslación del péndulo vista desde el eje de giro, $M_p L_r^2$ (proveniente de (3.12), no de T_{brazo}, pero agrupada aquí por comodidad de notación); el segundo corresponde a la rotación propia del péndulo respecto a su pivote; el tercero es el acoplamiento no lineal que aparece porque, al inclinarse el péndulo, su masa se aleja del eje de rotación del brazo en una cantidad que depende de $\sin\alpha$; y el cuarto es el término de acoplamiento (Coriolis/centrífugo) cruzado entre $\dot{\theta}$ y $\dot{\alpha}$, proporcional a $\cos\alpha$.

Energía potencial. Al ser el único término conservativo del sistema la gravedad, V depende únicamente de la altura del centro de masa del péndulo sobre un nivel de referencia, dada por la componente z_{cm} de (3.8) (el brazo no aporta, pues se mueve en un plano horizontal a altura constante). Con M_p la masa del péndulo y g la aceleración de gravedad, $V = M_p g z_{cm}$, es decir

$$V = M_p g\frac{L_p}{2}\cos\alpha, \tag{3.14}$$

que es mínima en $\alpha = \pi$ (péndulo colgando) y máxima en $\alpha = 0$ (péndulo invertido), consistente con que este último sea un punto de equilibrio inestable.

3.2.2 Modelo no lineal completo

Con T (3.13) y V (3.14) ya determinados, se aplica el formalismo de Euler-Lagrange (3.3) a cada coordenada generalizada por separado.

Ecuación para θ. Ni T ni V dependen explícitamente de θ (solo de $\dot{\theta}$, α y $\dot{\alpha}$): θ es una *coordenada cíclica* y, por lo tanto, $\partial\mathcal{L}/\partial\theta = 0$. Derivando $\mathcal{L} = T - V$ respecto a $\dot{\theta}$,

$$\frac{\partial\mathcal{L}}{\partial\dot{\theta}} = \Big(J_r + M_p L_r^2\Big)\dot{\theta} + M_p\Big(\frac{L_p}{2}\Big)^2 \sin^2\alpha\,\dot{\theta} - M_p\frac{L_p}{2}L_r\cos\alpha\,\dot{\alpha}, \tag{3.15}$$

y derivando nuevamente respecto al tiempo —usando $\frac{d}{dt}\sin^2\alpha = \sin(2\alpha)\dot{\alpha}$ y $\frac{d}{dt}[\cos\alpha\,\dot{\alpha}] = -\sin\alpha\,\dot{\alpha}^2 + \cos\alpha\,\ddot{\alpha}$—,

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\theta}} = \Big(J_r + M_p L_r^2 + M_p\Big(\frac{L_p}{2}\Big)^2\sin^2\alpha\Big)\ddot{\theta} + M_p\Big(\frac{L_p}{2}\Big)^2\sin(2\alpha)\dot{\alpha}\dot{\theta} - M_p\frac{L_p}{2}L_r\cos\alpha\,\ddot{\alpha} + M_p\frac{L_p}{2}L_r\sin\alpha\,\dot{\alpha}^2. \tag{3.16}$$

Como $\partial\mathcal{L}/\partial\theta = 0$, la ecuación de Euler-Lagrange para θ es directamente (3.16) igualada a $Q_\theta = \tau - D_r\dot{\theta}$ de (3.4):

$$\begin{aligned}\Big(J_r + M_p L_r^2 + M_p\Big(\tfrac{L_p}{2}\Big)^2\sin^2\alpha\Big)\ddot{\theta} - M_p\tfrac{L_p}{2}L_r\cos\alpha\,\ddot{\alpha} + M_p\Big(\tfrac{L_p}{2}\Big)^2\sin(2\alpha)\,\dot{\theta}\dot{\alpha} \\ + M_p\tfrac{L_p}{2}L_r\sin\alpha\,\dot{\alpha}^2 + D_r\dot{\theta} = \tau.\end{aligned} \tag{3.17}$$

Ecuación para α. Derivando ahora $\mathcal{L}$ respecto a $\dot{\alpha}$,

$$\frac{\partial\mathcal{L}}{\partial\dot{\alpha}} = \Big(J_p + M_p\Big(\frac{L_p}{2}\Big)^2\Big)\dot{\alpha} - M_p\frac{L_p}{2}L_r\cos\alpha\,\dot{\theta}, \tag{3.18}$$

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\alpha}} = \Big(J_p + M_p\Big(\frac{L_p}{2}\Big)^2\Big)\ddot{\alpha} - M_p\frac{L_p}{2}L_r\cos\alpha\,\ddot{\theta} + M_p\frac{L_p}{2}L_r\sin\alpha\,\dot{\alpha}\dot{\theta}. \tag{3.19}$$

A diferencia de θ, ahora α sí aparece explícitamente dentro de $\mathcal{L}$ —a través de $\sin^2\alpha$ y $\cos\alpha$ en T, y de $\cos\alpha$ en V—, por lo que $\partial\mathcal{L}/\partial\alpha \neq 0$:

$$\frac{\partial\mathcal{L}}{\partial\alpha} = M_p\Big(\frac{L_p}{2}\Big)^2\sin\alpha\cos\alpha\,\dot{\theta}^2 + M_p\frac{L_p}{2}L_r\sin\alpha\,\dot{\theta}\dot{\alpha} + M_p g\frac{L_p}{2}\sin\alpha, \tag{3.20}$$

donde el último término proviene de $-\partial V/\partial\alpha$, ya que V decrece con α mientras el péndulo pasa de invertido a colgante. Sustituyendo (3.19) y (3.20) en $\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\alpha}} - \frac{\partial\mathcal{L}}{\partial\alpha} = Q_\alpha = -D_p\dot{\alpha}$, el término $M_p\frac{L_p}{2}L_r\sin\alpha\,\dot{\alpha}\dot{\theta}$ aparece con igual valor en ambos lados y se cancela exactamente, quedando la segunda ecuación de movimiento:

$$\Big(J_p + M_p\Big(\tfrac{L_p}{2}\Big)^2\Big)\ddot{\alpha} - M_p\tfrac{L_p}{2}L_r\cos\alpha\,\ddot{\theta} - \tfrac{1}{2}M_p\Big(\tfrac{L_p}{2}\Big)^2\sin(2\alpha)\,\dot{\theta}^2 - M_p g\tfrac{L_p}{2}\sin\alpha + D_p\dot{\alpha} = 0. \tag{3.21}$$

Las ecuaciones (3.17) y (3.21) son las ecuaciones de movimiento no lineales acopladas del péndulo de Furuta, obtenidas íntegramente a partir del formalismo de Euler-Lagrange.

Estas pueden escribirse de forma compacta en la representación matricial estándar

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + G(q) = \tau_{\text{gen}}, \tag{3.22}$$

con $\tau_{\text{gen}} = [\tau,\ 0]^T$, matriz de inercia dependiente de la configuración

$$M(q) = \begin{bmatrix} J_r + M_p L_r^2 + M_p (L_p/2)^2 \sin^2\alpha & -M_p (L_p/2) L_r \cos\alpha \\ -M_p (L_p/2) L_r \cos\alpha & J_p + M_p (L_p/2)^2 \end{bmatrix},$$

y los términos centrífugos/Coriolis y gravitacionales agrupados, respectivamente, en $C(q,\dot{q})\dot{q}$ y $G(q)$ según (3.17)–(3.21) (incluyendo en $G(q)$ los términos de fricción viscosa $D_r\dot{\theta}$, $D_p\dot{\alpha}$ por simplicidad de notación). Este modelo es fuertemente no lineal —los términos de acoplamiento dependen de $\sin\alpha$, $\cos\alpha$, $\sin(2\alpha)$ y de productos cruzados de las velocidades angulares— por lo que no es directamente utilizable en el diseño de un controlador MPC lineal, motivando la linealización presentada en la siguiente sección. Este modelo corresponde a la formulación estándar utilizada por Quanser (Quanser Inc., 2014) para la familia de péndulos rotatorios QUBE-Servo, y es la que se linealiza en la Sección 3.3.2 para obtener el modelo utilizado en el diseño del controlador.

3.3 Linealización del modelo

3.3.1 Punto de operación (péndulo invertido)

El punto de equilibrio de interés es la posición vertical invertida del péndulo:

$$q_0 = \begin{bmatrix} \theta_0 \\ \alpha_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \dot{q}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad u_0 = 0 \tag{3.23}$$

Este punto corresponde a un equilibrio inestable del sistema: en la vecindad de $\alpha_0 = 0$, la gravedad genera un par que tiende a alejar al péndulo de la vertical (Ecuación (3.21), término $-M_p g (L_p/2)\sin\alpha$), por lo que sin una acción de control adecuada el sistema diverge de este punto. Se eligió este equilibrio, y no la posición colgante ($\alpha_0 = \pi$, que es estable, según (3.14)), porque el objetivo del trabajo es precisamente demostrar la capacidad del controlador MPC para estabilizar un sistema inherentemente inestable dentro de sus restricciones de actuación.

3.3.2 Modelo linealizado

Aplicando linealización por series de Taylor de primer orden en torno al punto de equilibrio (3.23) a las ecuaciones no lineales (3.17)–(3.21), se aproxima $\sin\alpha \approx \alpha$, $\cos\alpha \approx 1$,

$\sin(2\alpha) \approx 2\alpha$, y se desprecian todos los términos cuadráticos en las variables de estado y sus derivadas (por ejemplo $\alpha\,\dot\theta\dot\alpha$, $\dot\alpha^2$, $\dot\theta^2\alpha$), por corresponder a infinitésimos de orden superior respecto a la desviación del equilibrio. Bajo esta aproximación, los términos de acoplamiento no lineal se anulan exactamente y las ecuaciones (3.17)–(3.21) se reducen al sistema lineal

$$\Big(J_r + M_p L_r^2\Big)\ddot\theta - M_p \tfrac{L_p}{2} L_r\,\ddot\alpha + D_r\dot\theta = \tau, \tag{3.24}$$

$$\Big(J_p + M_p\Big(\tfrac{L_p}{2}\Big)^2\Big)\ddot\alpha - M_p \tfrac{L_p}{2} L_r\,\ddot\theta - M_p g \tfrac{L_p}{2}\alpha + D_p\dot\alpha = 0. \tag{3.25}$$

Este par de ecuaciones puede escribirse como $M_0\ddot q + D_0\dot q + G_0 q = B_0\tau$, con matriz de inercia constante

$$M_0 = \begin{bmatrix} J_r + M_p L_r^2 & -M_p \frac{L_p}{2} L_r \\ -M_p \frac{L_p}{2} L_r & J_p + M_p\Big(\frac{L_p}{2}\Big)^2 \end{bmatrix}. \tag{3.26}$$

Despejando $\ddot q = M_0^{-1}\Big(B_0\tau - D_0\dot q - G_0 q\Big)$ mediante la regla de Cramer, el determinante de M_0 resulta

$$J_t \triangleq \det(M_0) = J_r J_p + M_p\Big(\frac{L_p}{2}\Big)^2 J_r + J_p M_p L_r^2, \tag{3.27}$$

donde los términos cruzados en M_p^2 se cancelan exactamente entre el producto de la diagonal de M_0 y el cuadrado de su término fuera de la diagonal. Aplicando la inversa de M_0, se obtiene finalmente el modelo en espacio de estados continuo

$$\dot x = A_c x + B_c u, \qquad x = [\theta,\ \alpha,\ \dot\theta,\ \dot\alpha]^T, \tag{3.28}$$

con (evaluando numéricamente $J_t = 1.339 \times 10^{-8}\ \mathrm{kg^2\,m^4}$ con los parámetros de la Tabla 3.1):

$$A_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{M_p^2 (L_p/2)^2 L_r g}{J_t} & \dfrac{-D_r\Big(J_p + M_p (L_p/2)^2\Big)}{J_t} & 0 \\ 0 & \dfrac{M_p g (L_p/2)\Big(J_r + M_p L_r^2\Big)}{J_t} & \dfrac{-M_p (L_p/2) L_r D_r}{J_t} & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 149.28 & -0.0104 & 0 \\ 0 & 261.61 & -0.0103 & 0 \end{bmatrix}, \tag{3.29}$$

$$B_c = \frac{1}{J_t}\begin{bmatrix} 0 \\ 0 \\ J_p + M_p (L_p/2)^2 \\ M_p (L_p/2) L_r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 49.73 \\ 49.15 \end{bmatrix}. \tag{3.30}$$

Nótese que en esta etapa B_c pondera directamente el par motor τ, no aún el voltaje u; la relación entre ambos se incorpora en el siguiente apartado. Los valores propios de A_c son aproximadamente $\{0,\ -16.18,\ +16.17,\ -0.0046\}$: el par de valores propios reales ± 16.17

confirma la naturaleza inestable del punto de equilibrio (el polo en $+16.17$ corresponde a un modo que crece exponencialmente ante cualquier desviación de α respecto de la vertical), mientras que el valor propio en el origen refleja que el ángulo del brazo θ no tiene una dinámica restauradora propia (es libre de girar indefinidamente); el cuarto valor propio, cercano a cero, corresponde al leve amortiguamiento aportado por el término D_r (en este caso nulo, según la Tabla 3.1, por lo que su pequeño valor residual proviene de acoplamiento numérico con el resto del sistema).

Incorporación de la dinámica eléctrica del motor. El par τ no es la variable manipulada directamente: el motor DC se alimenta con el voltaje de armadura u, relacionado con el par a través de la ecuación eléctrica del motor (despreciando la inductancia de armadura, válida para las frecuencias de interés de este sistema),

$$\tau = k_t\, i_a = \frac{k_t}{R_m}\Big(u - k_m\dot{\theta}\Big), \tag{3.31}$$

donde i_a es la corriente de armadura y $k_m\dot{\theta}$ es la fuerza contraelectromotriz inducida por la rotación del brazo. Sustituyendo τ por $\frac{k_t}{R_m}u - \frac{k_t k_m}{R_m}\dot{\theta}$ en (3.24)–(3.25), y dado que τ afecta a $\ddot{\theta}$ y $\ddot{\alpha}$ a través de M_0^{-1}, se producen dos efectos: la matriz B_c se escala completa por k_t/R_m, y aparece un amortiguamiento adicional proporcional a $\dot{\theta}$ en ambas filas, de magnitud $k_t k_m/R_m$ multiplicada por la componente correspondiente de B_c (antes de escalar). Esto actualiza las matrices (3.29)–(3.30) según:

$$B_c \leftarrow \frac{k_t}{R_m}B_c, \qquad A_c(3,3) \leftarrow A_c(3,3) - \frac{k_t^2}{R_m}B_c(3), \qquad A_c(4,3) \leftarrow A_c(4,3) - \frac{k_t^2}{R_m}B_c(4), \tag{3.32}$$

donde $B_c(3)$ y $B_c(4)$ se evalúan después de la primera actualización (uso consistente con la implementación en MATLAB, donde $k_t = k_m$ por tratarse de un motor DC ideal, de modo que ambos términos usan el mismo factor k_t^2/R_m). Esta es exactamente la operación realizada por la función `QUBE_ROTPEN_ABCD_eqns.m` (ver Código 6.2), y da lugar a las matrices A_c, B_c numéricas reportadas en (3.29)–(3.30), que ya incluyen este efecto.

3.4 Modelo en espacio de estados

3.4.1 Representación en tiempo continuo

El modelo en espacio de estados en tiempo continuo del péndulo de Furuta queda dado por

$$\dot{x}(t) = A_c\, x(t) + B_c\, u(t), \qquad y(t) = C\, x(t) + D\, u(t), \tag{3.33}$$

con A_c y B_c dadas por las ecuaciones (3.29)–(3.30), ya incorporada la dinámica eléctrica del motor. Para efectos de diseño y verificación de las propiedades estructurales del sistema (controlabilidad) se utiliza

$$C = I_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.34}$$

es decir, se consideran los cuatro estados como salidas del modelo de diseño. Cabe notar que esto difiere de la instrumentación real del sistema físico, en la que únicamente θ y α son medidos directamente por los encoders (matriz C_{real}, Sección 3.5); esta distinción es la que motiva el análisis de observabilidad presentado en la Sección 3.5 y la etapa de estimación de estados descrita en el Capítulo 2.

3.4.2 Discretización del modelo

Discretizando el modelo continuo (3.29)–(3.30) mediante retenedor de orden cero con el período de muestreo $T_s = 2$ ms seleccionado en el Capítulo 2 (función `c2d` de MATLAB, Ecuación (2.16)), se obtienen las matrices discretas:

$$A_d = \begin{bmatrix} 1 & 2.99 \times 10^{-4} & 2.000 \times 10^{-3} & 1.99 \times 10^{-7} \\ 0 & 1.00052 & -2.06 \times 10^{-8} & 2.000 \times 10^{-3} \\ 0 & 0.29861 & 0.99998 & 2.99 \times 10^{-4} \\ 0 & 0.52331 & -2.06 \times 10^{-5} & 1.00052 \end{bmatrix}, \qquad B_d = \begin{bmatrix} 9.946 \times 10^{-5} \\ 9.831 \times 10^{-5} \\ 9.947 \times 10^{-2} \\ 9.832 \times 10^{-2} \end{bmatrix}. \tag{3.35}$$

Los valores propios de A_d son aproximadamente $\{1,\ 1.0329,\ 0.9682,\ 0.99999\}$, consistentes con los del modelo continuo a través de la relación $\lambda_d = e^{\lambda_c T_s}$: el valor propio en 1.0329 (fuera del círculo unitario, $e^{16.17 \times 0.002} \approx 1.0329$) refleja el modo inestable, mientras que el resto se ubica dentro o justo sobre el círculo unitario.
El modelo en espacio de estados en tiempo discreto queda entonces dado por

$$x_{k+1} = A_d\, x_k + B_d\, u_k, \qquad y_k = C_d\, x_k + D_d\, u_k, \tag{3.36}$$

con A_d y B_d dadas por la Ecuación (3.35). A diferencia de A_c y B_c, las matrices de salida no se ven afectadas por la discretización mediante retenedor de orden cero: la ecuación de salida $y = Cx + Du$ se evalúa en los mismos instantes de muestreo sin necesidad de

transformación, por lo que

$$C_d = C = I_4, \qquad D_d = D = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.37}$$

es decir, se reutilizan directamente las matrices C y D del modelo continuo (3.34).

3.5 Análisis de controlabilidad y observabilidad

3.5.1 Controlabilidad

Un sistema discreto (A_d, B_d) es completamente controlable si su matriz de controlabilidad

$$\mathcal{C} = \begin{bmatrix} B_d & A_d B_d & A_d^2 B_d & A_d^3 B_d \end{bmatrix} \tag{3.38}$$

tiene rango completo, igual a la dimensión del vector de estados ($n = 4$). Aunque el péndulo de Furuta es un sistema subactuado (una sola entrada para dos grados de libertad), el acoplamiento dinámico entre θ y α —presente tanto en A_c como, tras la linealización, en A_d, a través de los términos fuera de la diagonal derivados de M_0^{-1} en (3.26)— permite que el torque aplicado al brazo influya, de manera indirecta pero completa, sobre las cuatro variables de estado. En este trabajo se verificó que $\mathrm{rango}(\mathcal{C}) = 4$, por lo que el sistema linealizado es completamente controlable en torno al punto de operación invertido.

3.5.2 Observabilidad

De manera análoga, un sistema (A_d, C) es completamente observable si su matriz de observabilidad

$$\mathcal{O} = \begin{bmatrix} C \\ CA_d \\ CA_d^2 \\ CA_d^3 \end{bmatrix} \tag{3.39}$$

tiene rango completo. El análisis relevante para este trabajo no utiliza $C = I_4$, sino la matriz de salida real del sistema instrumentado, $C_{\text{real}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$, que refleja que únicamente θ y α son medidos directamente por los encoders. Se verificó que, incluso con esta matriz de salida reducida, $\mathrm{rango}(\mathcal{O}) = 4$: el sistema es completamente observable a partir de las mediciones de posición angular. Este resultado es el que justifica, desde el punto de vista teórico, que sea posible reconstruir las velocidades angulares $\dot{\theta}$ y $\dot{\alpha}$ a partir de las

mediciones de posición, tal como se implementa mediante la derivación filtrada descrita en el Capítulo 2.

3.5.3 Verificación en MATLAB

La verificación numérica se realiza mediante las funciones `ctrb` y `obsv` de MATLAB:

```
rank(ctrb(Ad, Bd))              % = 4 -> sistema controlable
rank(obsv(Ad, [1 0 0 0; 0 1 0 0]))   % = 4 -> sistema observable
```

En ambos casos el rango obtenido es igual a la dimensión del sistema ($n = 4$), confirmando que el modelo linealizado y discretizado del péndulo de Furuta es completamente controlable y observable en torno al punto de operación invertido, condición necesaria tanto para el diseño del controlador MPC como para la etapa de estimación de estados.

3.6 Formulación del MPC sobre el sistema

3.6.1 Ley de control y arquitectura del controlador

Una alternativa común para plantas de lazo abierto inestable es preestabilizar el sistema con una realimentación de estados $u = -K_{\text{lqr}}x + v$ y diseñar el MPC sobre la variable v, evitando así que las potencias A_d^i de la formulación densa (Capítulo 2) crezcan sin control. En este trabajo se optó por la alternativa más directa: el MPC opera sobre el par (A_d, B_d) del modelo linealizado sin transformación previa, resolviendo en línea el problema QP del Capítulo 2 a partir del estado (estimado) actual.
El regulador DLQR, diseñado sobre el mismo par (A_d, B_d) con $Q = \text{diag}(5, 1, 1, 20)$ y $R = 1$, no forma parte del lazo del MPC y cumple dos roles aparte: actúa como controlador de balance independiente, habilitado cuando $|\alpha| < 30°$ (valor adoptado en la implementación física y en las simulaciones de este trabajo; la plantilla original de Quanser para el QUBE-Servo 2 emplea $\pm 10°$, valor que finalmente no se utilizó); y su ecuación de Riccati discreta asociada se reutiliza como costo terminal Q_N del MPC (Capítulo 2), lo que acota la estabilidad del horizonte finito $N = 3$ sin imponer una restricción terminal explícita.
Se prefirió esta arquitectura, más simple que una preestabilización con sistema aumentado, porque el horizonte reducido ($N = 3$) ya mantiene acotado el crecimiento de A_d^i, y el MPC directo resultó suficiente para estabilizar el sistema dentro de las restricciones de actuación disponibles.

3.6.2 Ausencia de acción integral

Dado que la referencia de control es fija e igual al origen del modelo linealizado ($x_{\text{ref}} = 0$, Sección de Seguimiento de referencia del Capítulo 2), y que no existen perturbaciones

constantes de magnitud relevante que desplacen el punto de equilibrio real respecto del nominal, no se incorporó una acción integral ni un sistema aumentado al esquema de control. El error en régimen estacionario observado en la implementación física se atribuye principalmente a no linealidades no modeladas (fricción estática, zona muerta del actuador) y se documenta en los resultados experimentales del Capítulo 5.

3.7 Integración de la estimación de estados al MPC

3.7.1 Estados estimados como entrada al controlador

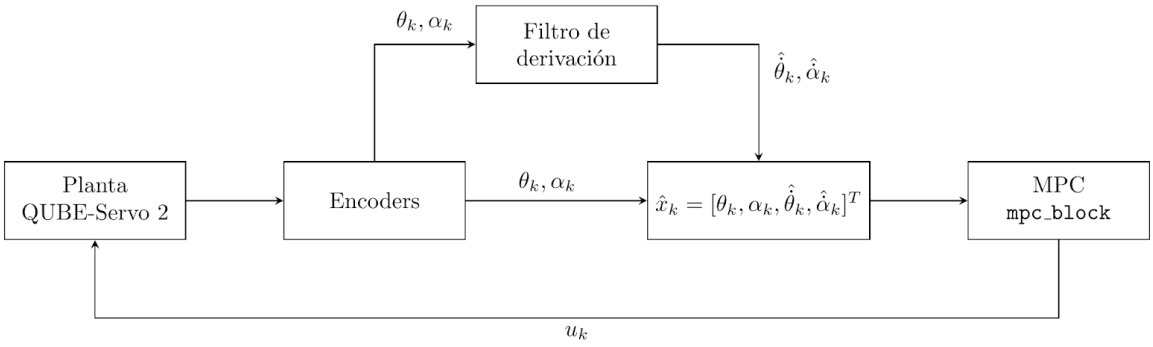


Figura 3.2: Lazo cerrado con estados estimados como entrada al controlador MPC.

En la implementación en Simulink, el vector de estados x_k que recibe el bloque de control MPC (función `mpc_block`, Capítulo 2, ver Código 6.7) no proviene de una medición directa, sino que se construye concatenando las posiciones angulares medidas por los encoders (θ_k, α_k) con las velocidades angulares estimadas mediante los filtros de derivación descritos en el Capítulo 2 ($\hat{\dot{\theta}}_k$, $\hat{\dot{\alpha}}_k$): $\hat{x}_k = [\theta_k,\ \alpha_k,\ \hat{\dot{\theta}}_k,\ \hat{\dot{\alpha}}_k]^T$. Este vector $\hat{x}_k$ es el que se utiliza para calcular el error $e_k = \hat{x}_k - x_{\text{ref}}$ y, a partir de él, el vector lineal f del problema QP en cada instante de muestreo.

3.7.2 Diseño del estimador

El estimador implementado no requiere sintonizar matrices de covarianza de proceso o de medición, a diferencia de un filtro de Kalman: su único parámetro de diseño es la constante de tiempo τ del filtro de derivación (2.17) de cada canal (θ y α), seleccionada como compromiso entre atenuación del ruido de cuantización de los encoders y el retardo de fase introducido en la estimación de velocidad, tal como se describe en el Capítulo 2. La validez de esta estimación se confirma de manera indirecta en los resultados de simulación y de implementación física presentados en los Capítulos 4 y 5, donde el desempeño del

controlador MPC operando con los estados estimados $\hat{x}_k$ se compara con el caso de referencia en que se asume acceso completo a los estados reales.

CAPÍTULO

4 Simulación del sistema de control

Antes de implementar el controlador en el QUBE-Servo 2 físico, se valida su desempeño en un entorno de simulación puramente software, reemplazando la planta real por el modelo discreto (A_d, B_d) obtenido en el Capítulo 3. Esto permite verificar que el MPC estabiliza el sistema y respeta las restricciones de actuación antes de exponer el hardware a un controlador aún no probado sobre él.

4.1 Entorno MATLAB y Simulink

MATLAB es la plataforma de cálculo numérico en la que se implementaron los scripts de este trabajo: `qube_rotpen_param.m` y `setup_qube_rotpen.m` (parámetros físicos y modelo en espacio de estados de la planta; ver Códigos 6.1 y 6.3), `find_dlqr.m` (diseño del controlador de balance DLQR; ver Código 6.4), `build_mpc_matrices.m` (construcción de las matrices densas del problema QP; ver Código 6.5) y `mpc_furuta.m` (script principal que integra lo anterior y abre el modelo de Simulink; ver Código 6.6). Estos scripts se ejecutan íntegramente fuera de línea (*offline*), antes de correr la simulación: calculan las matrices H, $\bar{A}$, $\bar{O}$, $\bar{Q}$ y $\bar{R}$ del problema QP y las dejan disponibles en el *workspace* de MATLAB, desde donde el bloque MPC de Simulink las utiliza en cada paso de muestreo. Simulink es el entorno de modelado por diagrama de bloques donde se implementa el lazo de control completo. El modelo de simulación (`mpc_furuta_test_sin_pendulo.slx`) reutiliza exactamente el mismo bloque de control (`mpc_block`, ver Código 6.7) que el modelo de implementación física (Capítulo 5), reemplazando únicamente los bloques de entrada/salida de hardware (HIL) por un bloque de planta simulada.

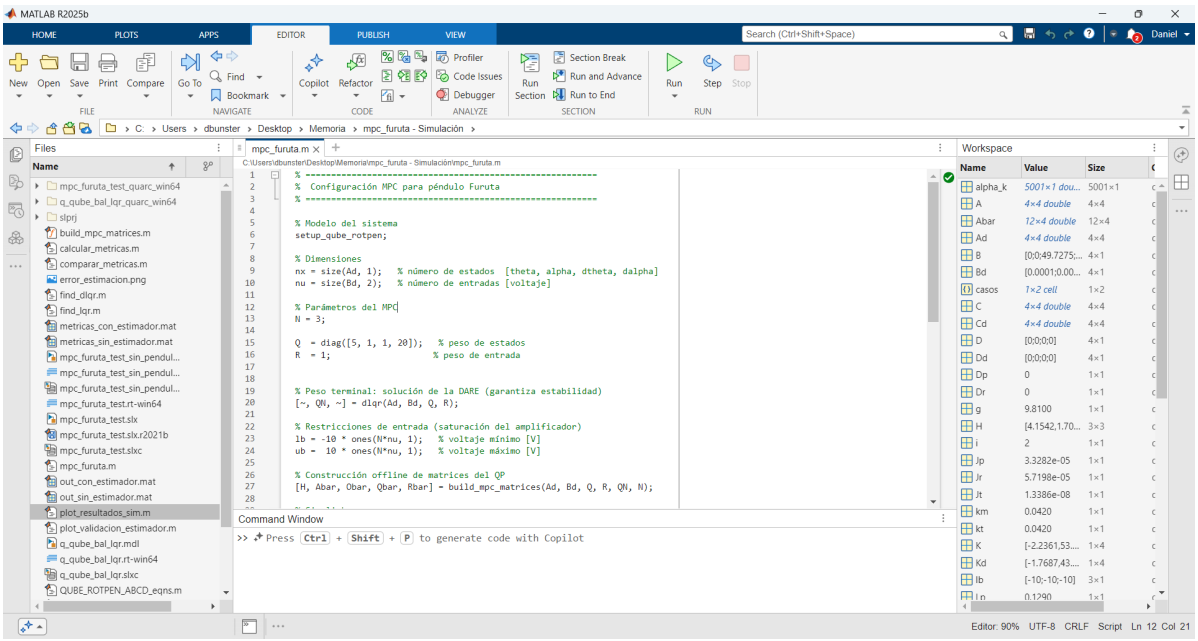


Figura 4.1: Entorno MATLAB.

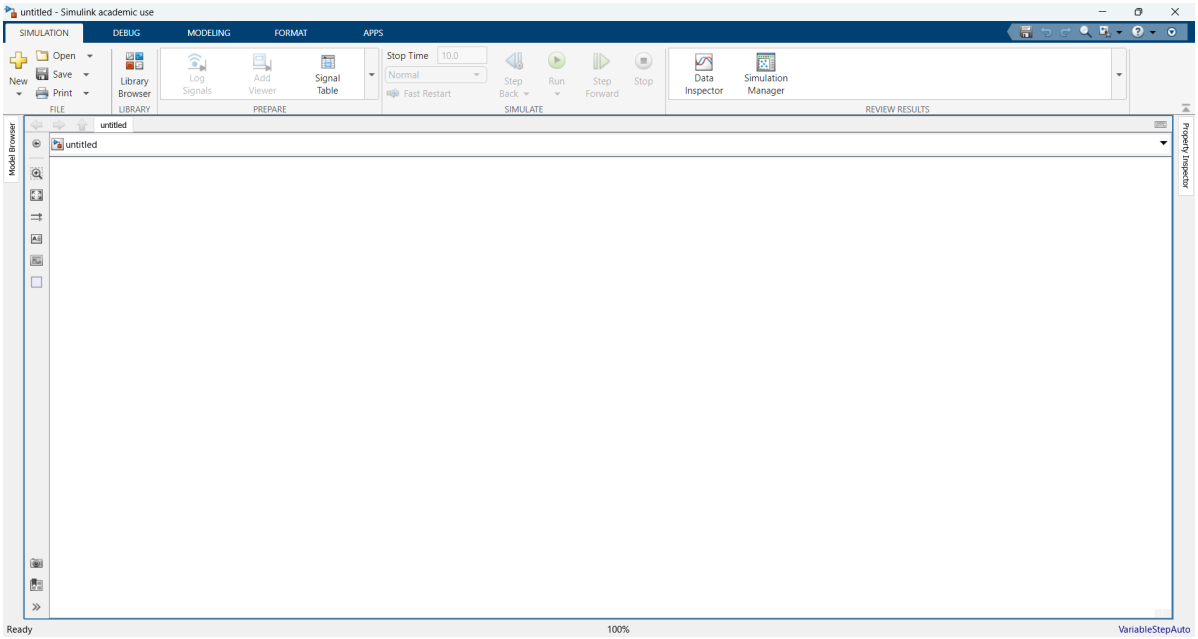


Figura 4.2: Entorno Simulink: modelo `mpc_furuta_test_sin_pendulo.slx`.

4.2 Bloques utilizados para la simulación

El modelo `mpc_furuta_sim.slx` utiliza los siguientes bloques principales:

- **Discrete State-Space**: implementa la planta simulada, con las matrices A_d, B_d de la

Ecuación (3.35) y período de muestreo $T_s = 2$ ms. Su salida es el vector de estado completo y exacto de la planta simulada, $[\theta_k,\ \alpha_k,\ \dot{\theta}_k,\ \dot{\alpha}_k]^T$, que un bloque **Demux** separa en sus cuatro componentes individuales.

- **State X** (subsistema): implementa el mismo estimador por derivación filtrada descrito en el Capítulo 2 (ver Figura 4.4). A partir únicamente de θ_k y α_k (las dos primeras salidas del Demux), calcula las velocidades estimadas $\hat{\dot{\theta}}_k$, $\hat{\dot{\alpha}}_k$ mediante un filtro pasa-altos de la forma $H(s) = \dfrac{50s}{s+50}$ aplicado a cada ángulo, y entrega el vector $\hat{x}_k = [\theta_k,\ \alpha_k,\ \hat{\dot{\theta}}_k,\ \hat{\dot{\alpha}}_k]^T$, replicando en simulación el mismo estimador usado en la implementación física (Capítulo 5). Las salidas 3 y 4 del Demux ($\dot{\theta}_k$, $\dot{\alpha}_k$ exactos) permiten tanto alimentar directamente al MPC en la modalidad "sin estimador" como comparar contra $\hat{\dot{\theta}}_k$, $\hat{\dot{\alpha}}_k$ para validar el estimador (ver **Manual Switch** más abajo).

- **Manual Switch**: permite seleccionar, antes de cada ensayo, cuál vector de estado recibe el bloque MPC: la salida completa y exacta del Discrete State-Space (posición 1, sin pasar por el estimador) o el vector estimado $\hat{x}_k$ que entrega **State X** (posición 2). Esta selección manual es la que distingue las pruebas "sin estimador" (Sección 4.4) de las pruebas "con estimación de estados" (Sección 4.5): son dos ensayos del mismo modelo, cada uno con el switch en una posición distinta.

- **MATLAB Function (`mpc_block`)**: resuelve el problema QP en cada paso de muestreo mediante `quadprog`, tomando como estado la salida del Manual Switch y a partir de las matrices calculadas por `build_mpc_matrices.m` (ver Figura 4.5).

- **Abs** y bloque relacional (*relop*): calculan $|\alpha|$ y lo comparan con el umbral de 30° para la lógica de habilitación de la actuación del MPC (ver **MultiPortSwitch** a continuación).

- **MultiPortSwitch**: selecciona entre la señal de control del MPC y una señal nula ($u = 0$ V) según el estado de la lógica de habilitación.

- **Goto / From**: propagan señales (como el estado $\hat{x}_k$) entre distintas partes del modelo sin necesidad de conectarlas físicamente con líneas.

- **Scope**: registra y permite exportar al *workspace* las señales de interés: ángulo del brazo (*Rotary Arm*), ángulo del péndulo (*Pendulum*), voltaje de actuación (*Motor Voltage*), el vector de estado estimado (*State X*) y las velocidades exactas de la planta simulada, registradas aparte para la validación del estimador.

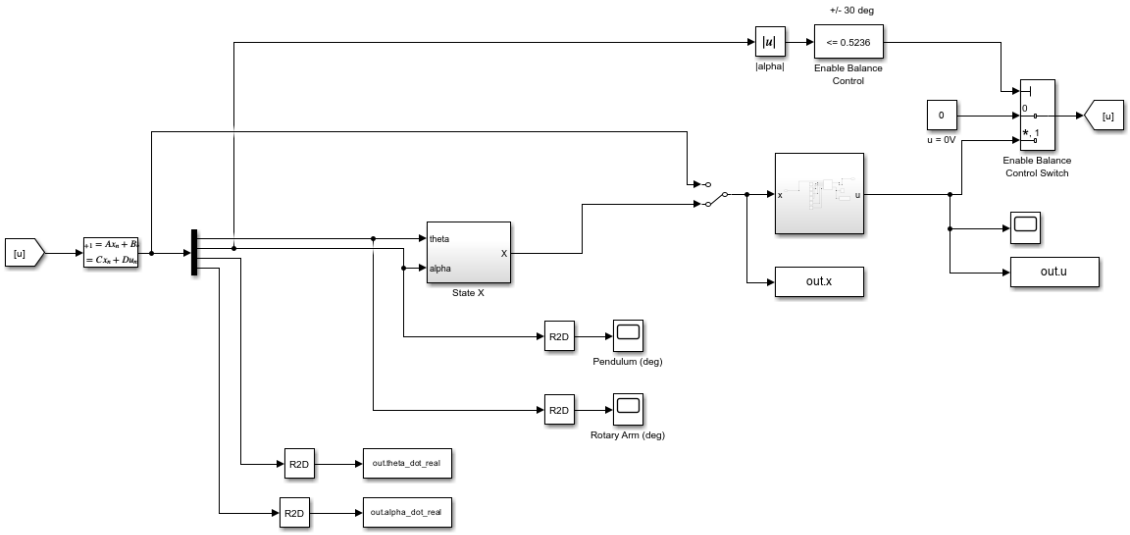


Figura 4.3: Bloques utilizados en Simulink.

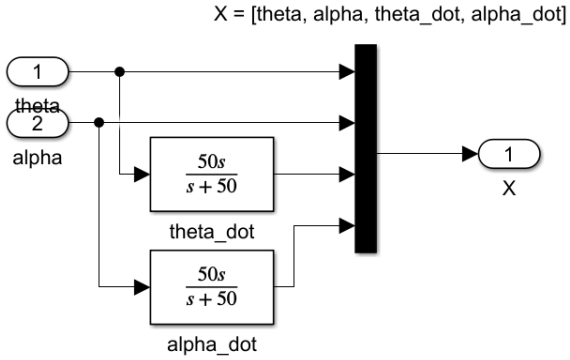


Figura 4.4: Subsistema **State X**: estimador de velocidades angulares por derivación filtrada.

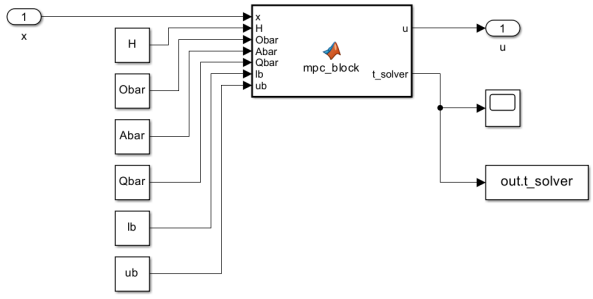


Figura 4.5: Bloque `mpc_block`: resolución del problema QP mediante `quadprog`.

4.3 Preparación de la simulación

Los parámetros del controlador MPC utilizados en la simulación son los mismos definidos y justificados en el Capítulo 3: horizonte $N = 3$, matrices de ponderación $Q = \mathrm{diag}(5, 1, 1, 20)$ y $R = 1$, con el costo terminal Q_N obtenido de la ecuación de Riccati discreta asociada a este mismo par (Q, R) (Sección 3.6.1). Las matrices del problema QP denso (H, $\bar{A}$, $\bar{O}$, $\bar{Q}$, $\bar{R}$) se calculan fuera de línea mediante `build_mpc_matrices.m` y se cargan en el *workspace* de MATLAB antes de ejecutar la simulación en Simulink (script `mpc_furuta.m`). La restricción de actuación se fija en $-10\ \mathrm{V} \leq u \leq 10\ \mathrm{V}$, igual que en el sistema físico, y el paso fijo del solver de Simulink coincide con el período de muestreo del controlador, $T_s = 2$ ms.
Todas las pruebas de simulación de este capítulo parten de una condición inicial $x_0 = [\theta_0,\ \alpha_0,\ \dot{\theta}_0,\ \dot{\alpha}_0]^\top = [0,\ 0.5236,\ 0,\ 0]^\top$, configurada como condición inicial del bloque `Discrete State-Space` del modelo de Simulink, es decir, con el péndulo desplazado 0.5236 rad ($\alpha_0 = 30°$) respecto de la posición vertical invertida, y con el brazo y ambas velocidades angulares inicialmente en reposo. Esta condición inicial de $30°$ y una duración de simulación de 5 s replican directamente las condiciones de los ensayos en el sistema físico del Capítulo 5, en los que el péndulo se libera manualmente desde $\alpha_0 \approx 30°$ **sin maniobra de swing-up**, a diferencia de las simulaciones preliminares de versiones anteriores de este trabajo, que partían de un desplazamiento pequeño ($\alpha_0 \approx 5.7°$). La referencia del controlador corresponde al origen, de modo que el objetivo es estabilizar el péndulo en $\alpha = 0$ manteniendo el brazo en $\theta = 0$.

4.4 Pruebas de simulación sin estimador

En esta sección se evalúa el desempeño del controlador MPC con el **Manual Switch** del modelo en la posición que conecta directamente la salida exacta del bloque Discrete State-Space al MPC, sin pasar por el estimador **State X**. Esto permite aislar el comportamiento del MPC en sí mismo, estableciendo una referencia de desempeño "ideal" antes de evaluar, en la Sección 4.5, el efecto de reemplazar $\dot{\theta}$ y $\dot{\alpha}$ exactos por su estimación filtrada. Para facilitar la comparación directa entre ambas modalidades, las Figuras 4.6, 4.7, 4.8 y 4.9 de esta sección presentan, en un mismo gráfico, la respuesta "sin estimador" junto con la respuesta "con estimador" de la Sección 4.5.

4.4.1 Respuesta al seguimiento de referencia

La Figura 4.6 muestra la evolución de θ_k y α_k partiendo de la condición inicial $\alpha_0 = 30°$ descrita en la Sección 4.3. En la modalidad sin estimador, el ángulo del péndulo α_k presenta un primer transitorio pronunciado, con un mínimo de aproximadamente $-8.5\,°$ en torno a $t \approx 0.34$ s; este primer cruce por el origen constituye el sobreimpulso principal de la

respuesta, medido en $28.48\,\%$ respecto de la condición inicial. Tras este mínimo, la señal converge de forma suave hacia el origen, alcanzando un tiempo de establecimiento (banda del $2\,\%$ de la condición inicial) de $t_s = 1.586$ s. El ángulo del brazo θ_k, cuya dinámica no está acotada a un rango angular limitado, exhibe una excursión transitoria considerablemente mayor (mínimo cercano a -104° en torno a $t \approx 0.43$ s), reflejo del recorrido angular que el MPC utiliza para contrarrestar el momento angular del péndulo al partir de un desplazamiento inicial grande y sin maniobra de swing-up; esta variable también converge monótonamente al origen. No se observa error estacionario apreciable en ninguno de los dos ángulos: al no existir discrepancia entre el modelo interno del MPC y la planta simulada, ni perturbaciones externas, ambos se estabilizan exactamente en el origen hacia el final de los 5 s de simulación.

La Figura 4.7 muestra que, en esta modalidad, la señal de control satura brevemente en $u = -10$ V durante los primeros ~ 8 ms de simulación —respuesta esperable dada la magnitud del desplazamiento inicial de $30°$—, pero no vuelve a acercarse a los límites de actuación en ningún instante posterior: su valor máximo tras la saturación inicial es de apenas 0.92 V (en torno a $t \approx 0.30$ s), lo que evidencia una respuesta bien amortiguada y de bajo esfuerzo de control una vez superado el instante inicial, cuando el MPC dispone del estado exacto de la planta.

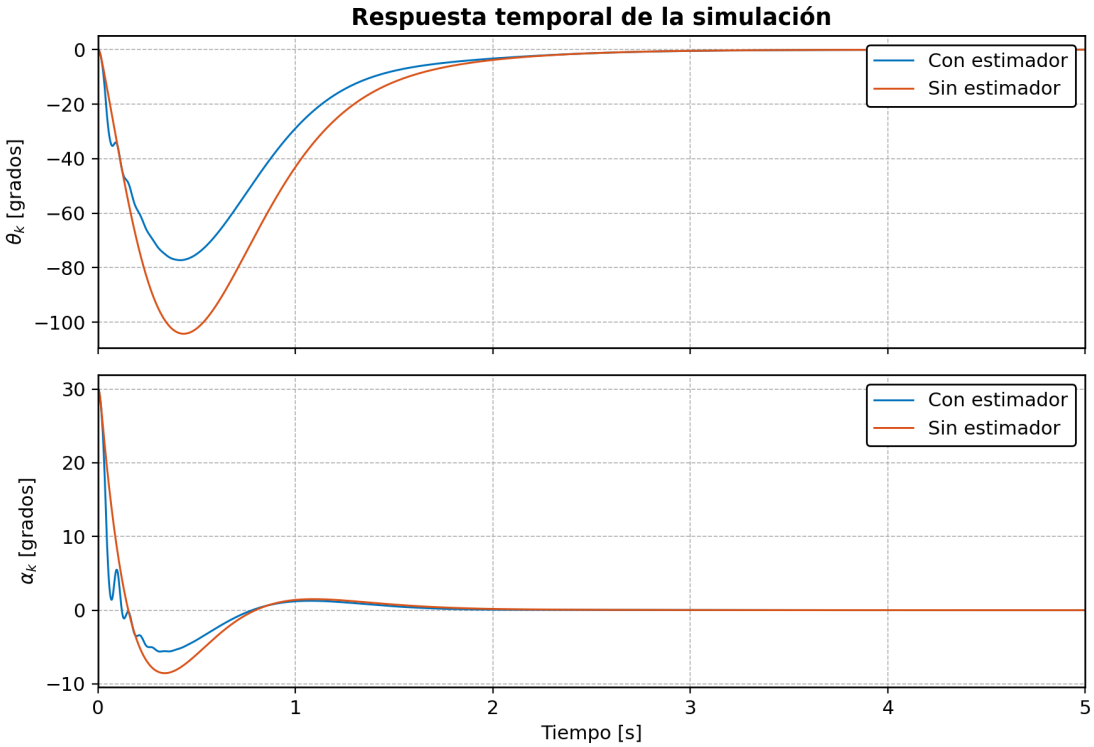


Figura 4.6: Respuesta temporal de θ_k y α_k ante la referencia, con y sin estimador, partiendo de $\alpha_0 = 30°$.

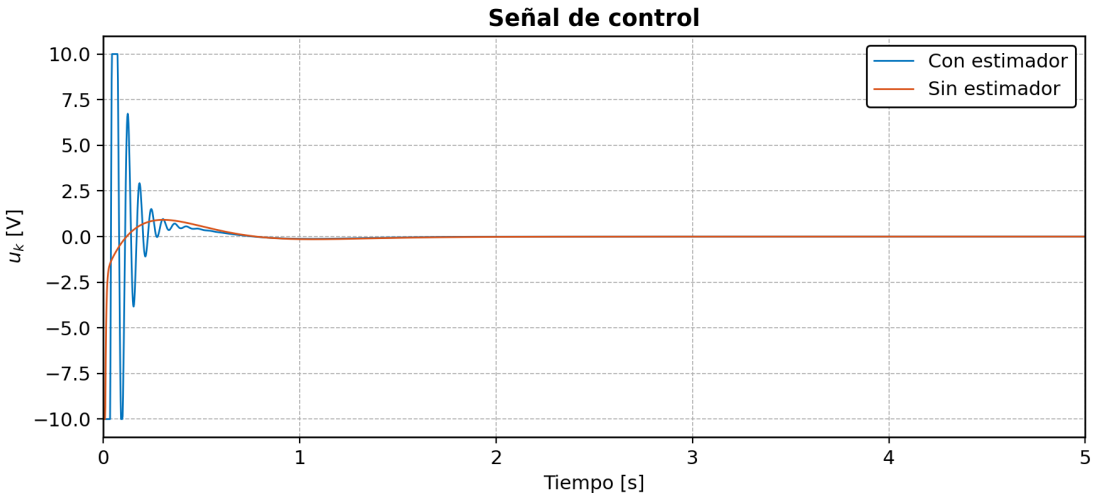


Figura 4.7: Señal de control u_k, con y sin estimador.

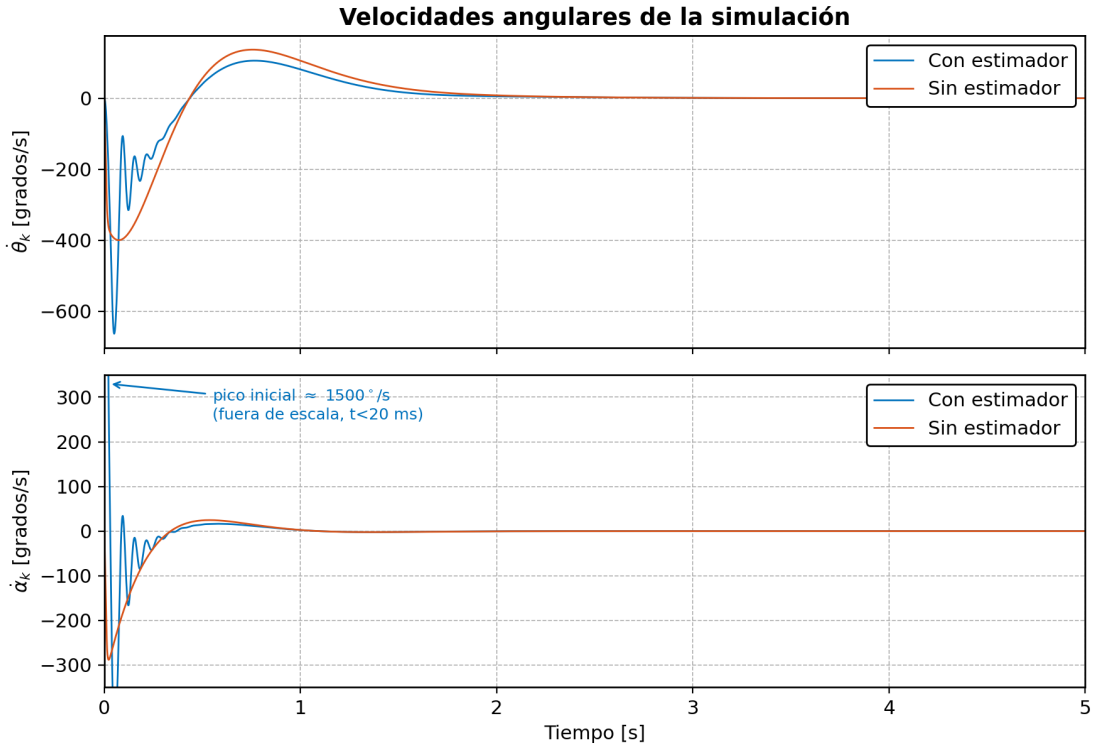


Figura 4.8: Velocidades angulares $\dot{\theta}_k$ y $\dot{\alpha}_k$ de la planta simulada, con y sin estimador.

4.4.2 Tiempos de ejecución del solver quadprog

La Figura 4.9 muestra el tiempo de ejecución de `quadprog` en cada paso de muestreo, para ambas modalidades. En la modalidad sin estimador, la gran mayoría de las resoluciones del problema QP toman entre 1 y 2 μs, con un pico máximo de 27.6 μs (en $t \approx 2.51$ s) y un valor inicial de 25.9 μs; el tiempo medio de ejecución durante toda la simulación es de 1.49 μs. A diferencia de resultados preliminares con simulaciones más cortas, no se observa aquí un pico de *warm-up* significativamente mayor al resto: todos los tiempos de ejecución se mantienen del mismo orden de magnitud a lo largo de los 5 s simulados. En cualquier caso, incluso el mayor tiempo registrado representa apenas un $1.4\,\%$ del período de muestreo $T_s = 2$ ms, por lo que el controlador cumple con amplísimo margen el requisito de factibilidad en tiempo real.

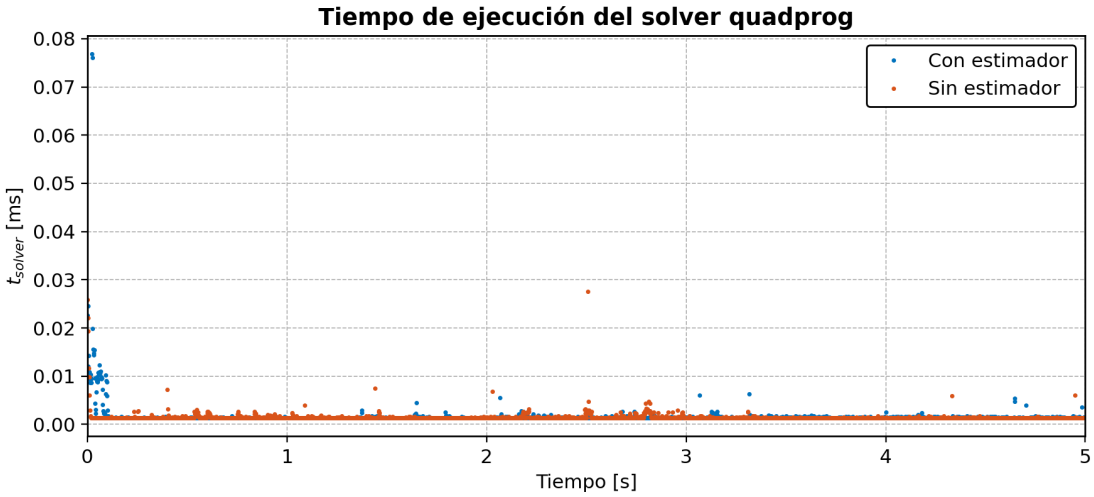


Figura 4.9: Tiempos de ejecución del solver `quadprog` por paso de muestreo, con y sin estimador.

4.5 Pruebas de simulación con estimación de estados

En esta sección se repiten las pruebas de la Sección 4.4, ahora con el **Manual Switch** en la posición que conecta el vector estimado $\hat{x}_k$ del subsistema **State X** (Sección 4.2) a la entrada del MPC, tal como opera el sistema en la implementación física (Capítulo 5). Esto permite cuantificar el efecto del retardo introducido por el filtro $H(s) = 50s/(s+50)$ sobre el desempeño del controlador, comparando estas pruebas contra la referencia "ideal" de la Sección 4.4. Para esta modalidad se registraron, además de $\hat{\dot{\theta}}_k$ y $\hat{\dot{\alpha}}_k$, las velocidades angulares reales de la planta simulada ($\dot{\theta}_k$, $\dot{\alpha}_k$) en el mismo ensayo, lo que permite comparar directamente el estado estimado contra el estado real dentro de un único lazo cerrado.

4.5.1 Validación del estimador (estados reales vs. estimados)

Las Figuras 4.10 y 4.11 comparan, dentro del mismo ensayo con estimador, las velocidades angulares reales $\dot{\theta}_k$ y $\dot{\alpha}_k$ (obtenidas directamente del modelo de la planta) contra sus estimaciones $\hat{\dot{\theta}}_k$ y $\hat{\dot{\alpha}}_k$ entregadas por el filtro pasa-altos $H(s) = 50s/(s+50)$. Al partir de una condición inicial de $\alpha_0 = 30°$ sin maniobra de swing-up, el estimador de $\dot{\alpha}_k$ exhibe un artefacto de inicialización pronunciado: como el filtro pasa-altos parte de un estado interno nulo mientras la planta presenta un salto abrupto de $30°$ en $t = 0$, $\hat{\dot{\alpha}}_k$ arranca en un pico espurio de $1500\,°/\text{s}$ —valor que corresponde exactamente a la ganancia del filtro (50) aplicada al escalón inicial ($50 \times 30° = 1500\,°/\text{s}$)— y decae con la constante de tiempo del filtro ($\tau = 1/50 = 20$ ms), muy por encima del valor real de $\dot{\alpha}_k$ en ese instante (pico real de $804.9\,°/\text{s}$). Este artefacto no tiene contraparte en $\hat{\dot{\theta}}_k$, ya que $\theta_0 = 0$ no presenta un escalón inicial. Considerando todo el ensayo (incluyendo este transitorio de inicialización), el error

RMS es de $40.9\,°/s$ para $\dot{\theta}_k$ y $88.8\,°/s$ para $\dot{\alpha}_k$; si se excluye la ventana de inicialización del filtro ($t \leq 0.1$ s, equivalente a 5τ), el error RMS post-transitorio baja a $15.4\,°/s$ y $14.6\,°/s$ respectivamente, con errores máximos puntuales de $214.8\,°/s$ y $207.4\,°/s$ durante los ciclos de oscilación activa de la respuesta. Esto confirma que el grueso del error de estimación se concentra en el arranque del filtro y en los instantes de mayor actividad de control, mientras que el retardo de fase en régimen más suave es comparativamente moderado.

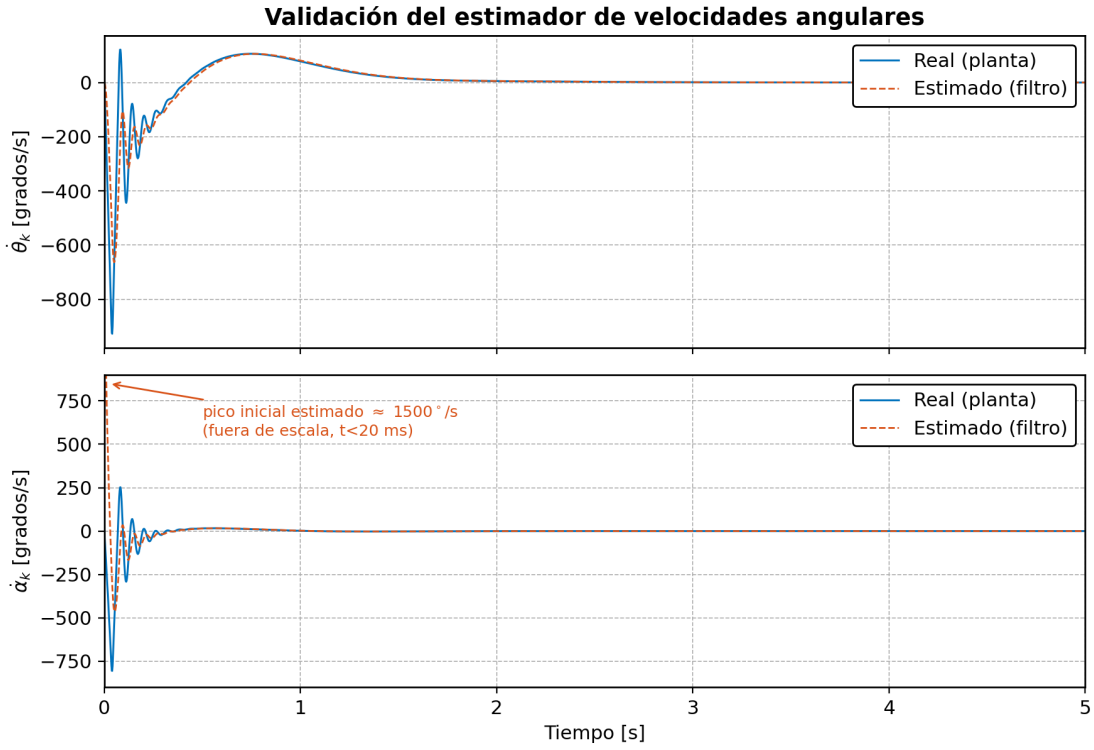


Figura 4.10: Comparación de las velocidades angulares reales y estimadas, ensayo con estimador.

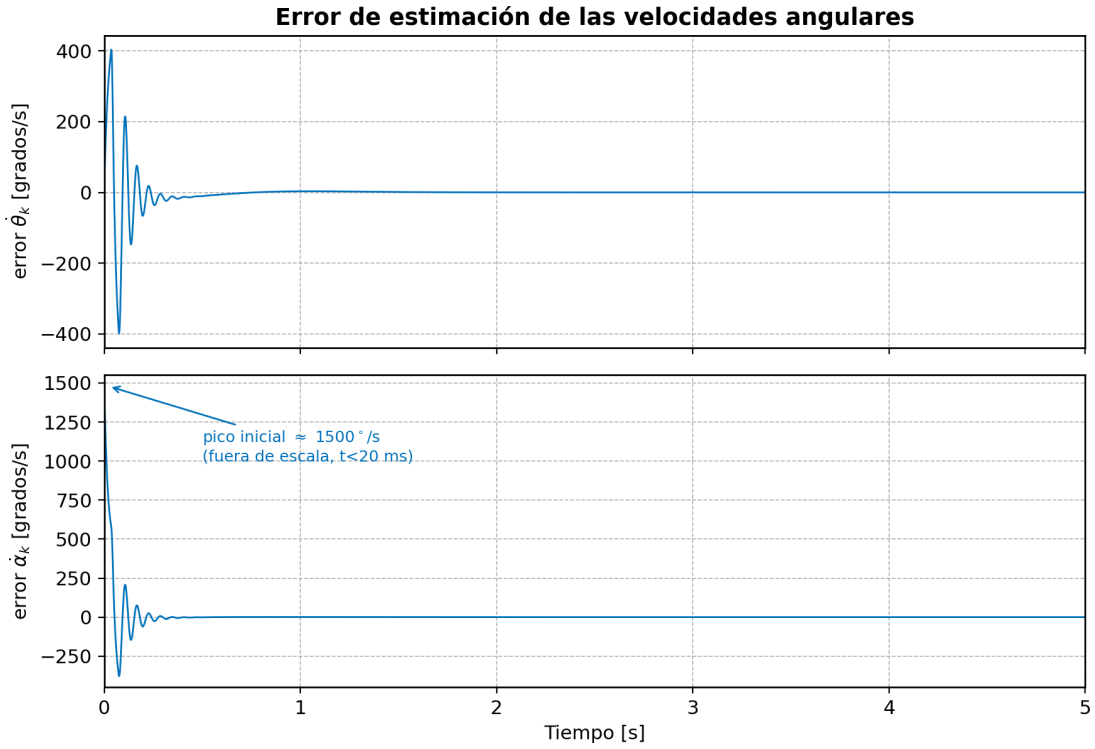


Figura 4.11: Error de estimación de las velocidades angulares (estimado − real).

4.5.2 Desempeño del MPC con estados estimados

Considerando nuevamente las Figuras 4.6, 4.7 y 4.9 —ahora enfocadas en la curva "con estimador"—, el MPC operando sobre el vector de estado estimado $\hat{x}_k$ presenta, sorprendentemente, una respuesta más rápida y con menor sobreimpulso que la modalidad sin estimador al partir de $\alpha_0 = 30°$: α_k alcanza un primer mínimo de $-5.6\,°$ en torno a $t \approx 0.31$ s (sobreimpulso de $18.71\,\%$, notablemente menor que el $28.48\,\%$ sin estimador) y se establece dentro de la banda del $2\,\%$ en $t_s = 1.464$ s, levemente más rápido que sin estimador ($t_s = 1.586$ s). El ángulo del brazo θ_k exhibe una excursión inicial menor en magnitud que en el caso sin estimador (mínimo cercano a $-77.2\,°$ frente a $-104.2\,°$), aunque con una dinámica algo más oscilatoria antes de converger monótonamente al origen. La señal de control u_k (Figura 4.7) satura tanto en -10 V como en $+10$ V durante los primeros ~ 96 ms —a diferencia del caso sin estimador, que solo satura en -10 V durante los primeros ~ 8 ms—, lo que es consistente con el retardo de fase y el artefacto de inicialización que introduce el filtro pasa-altos sobre $\hat{\dot{\alpha}}_k$: al recibir una estimación con retardo y con un pico inicial espurio, el MPC reacciona de forma más enérgica en los primeros instantes. Pese a esta actuación inicial más agresiva, ambos ángulos convergen al origen sin error estacionario apreciable, y en este ensayo particular la modalidad con

estimador logra un desempeño global levemente superior al ideal sin estimador, tanto en tiempo de establecimiento como en sobreimpulso. En cuanto a los tiempos de ejecución del solver (Figura 4.9), el tiempo medio es de $1.63\ \mu$s y el máximo de $76.9\ \mu$s (en $t \approx 0.022$ s), coincidiendo con el intervalo de mayor actividad de control; incluso este máximo representa apenas un $3.8\,\%$ del período de muestreo $T_s = 2$ ms, por lo que el controlador es igualmente factible en tiempo real en esta modalidad.

4.6 Análisis comparativo

Las pruebas de simulación permiten aislar el efecto del estimador por derivación filtrada sobre el desempeño del MPC, manteniendo fijos el modelo de la planta, el horizonte N, las ponderaciones (Q, R), los límites de actuación y la condición inicial $\alpha_0 = 30°$ sin maniobra de swing-up. A diferencia de lo observado en ensayos preliminares con desplazamientos iniciales pequeños, al partir de una condición inicial grande la modalidad con estimador resulta levemente superior a la modalidad sin estimador tanto en tiempo de establecimiento ($t_s = 1.464$ s frente a 1.586 s) como en sobreimpulso ($18.71\,\%$ frente a $28.48\,\%$). Esta tendencia se confirma con las métricas integrales de error: la **Integral del Error Cuadrático** (*Integral Square Error*, ISE) y la **Integral del Error Absoluto** (*Integral Absolute Error*, IAE), calculadas sobre α_k a lo largo de los 5 s de simulación, resultan ambas menores con estimador (ISE$= 0.0112$, IAE$= 0.0744$) que sin estimador (ISE$= 0.0189$, IAE$= 0.1092$). El costo de este mejor desempeño es una actuación inicial más agresiva: con estimador el control satura brevemente tanto en $+10$ V como en -10 V durante los primeros ~ 96 ms, mientras que sin estimador solo satura en -10 V durante los primeros ~ 8 ms. En ambos casos el sistema converge al origen sin error estacionario apreciable, y el tiempo de ejecución del solver `quadprog` se mantiene muy por debajo del período de muestreo $T_s = 2$ ms en todo momento (máximo de $76.9\ \mu$s con estimador y $27.6\ \mu$s sin estimador, ambos bajo el $4\,\%$ de T_s), confirmando que el controlador es factible en tiempo real en ambas modalidades. La Tabla 4.1 resume estas métricas, y la Tabla 4.2 resume, para la modalidad con estimador, el error entre las velocidades angulares estimadas y reales.

Table 4.1: Comparación de métricas de desempeño: MPC sin y con estimación de estados ($\alpha_0 = 30°$).

Métrica	Sin estimador	Con estimador
ISE	0.0189	0.0112
IAE	0.1092	0.0744
t_s [s]	1.586	1.464
Sobreimpulso [%]	28.48	18.71
t_{solver} medio [μs]	1.49	1.63
t_{solver} máx. [μs]	27.60	76.90
Factible en tiempo real	Sí	Sí

Table 4.2: Validación del estimador: error entre velocidades angulares estimadas y reales (ensayo con estimador).

Métrica	$\dot{\theta}_k$	$\dot{\alpha}_k$
RMS error, ensayo completo [°/s]	40.9	88.8
RMS error, post-transitorio ($t > 0.1$ s) [°/s]	15.4	14.6
Error máx. post-transitorio ($t > 0.1$ s) [°/s]	214.8	207.4

CAPÍTULO

5 Implementación física del sistema de control

5.1 Hardware utilizado

5.1.1 Péndulo de Furuta (plataforma Quanser)

El sistema físico utilizado corresponde a la plataforma QUBE-Servo 2 de Quanser (Quanser Inc., 2014), configurada en su variante de péndulo rotatorio (péndulo de Furuta). El equipo integra en un mismo módulo el motor de corriente continua que acciona el brazo horizontal, la caja reductora, el amplificador de potencia interno y dos encoders ópticos incrementales en cuadratura: uno solidario al eje del motor, que mide el ángulo del brazo θ, y otro montado en la articulación del péndulo, que mide el ángulo relativo α respecto del brazo. Los parámetros físicos de la planta (longitudes, masas y momentos de inercia) fueron caracterizados en la Sección 3.1 y se resumen en la Tabla 3.1. La Figura 5.1 muestra el montaje utilizado en el laboratorio, con el péndulo en su posición de reposo antes de cada ensayo.



Figura 5.1: Sistema de péndulo de Furuta y configuración de hardware.

5.1.2 Interfaz de adquisición de datos (USB / QUARC)

A diferencia de una configuración con tarjeta de adquisición de datos (DAQ) independiente, el QUBE-Servo 2 se conecta al equipo host mediante un enlace USB directo. El controlador QUARC de Quanser expone el hardware a Simulink a través del bloque `HIL Initialize` (tarjeta `QUBE-Servo_USB`), y los bloques `HIL Read` / `HIL Write` correspondientes leen las posiciones angulares de los encoders y escriben el voltaje de actuación en cada paso de muestreo, dentro del mismo lazo de Simulink.

5.2 Software: Quanser y Simulink en tiempo real

El modelo de Simulink se ejecuta en modo de tiempo real (*External mode*) sobre el mismo esquema de control descrito en el Capítulo 4, reemplazando el bloque de planta simulada por los bloques `HIL Initialize`, `HIL Read Encoder` y `HIL Write Analog` de la librería QUARC. Estos bloques leen los ángulos θ_k, α_k y escriben el voltaje de actuación u_k directamente sobre el hardware en cada paso de muestreo $T_s = 2$ ms, dentro del mismo diagrama de bloques que implementa el controlador MPC.

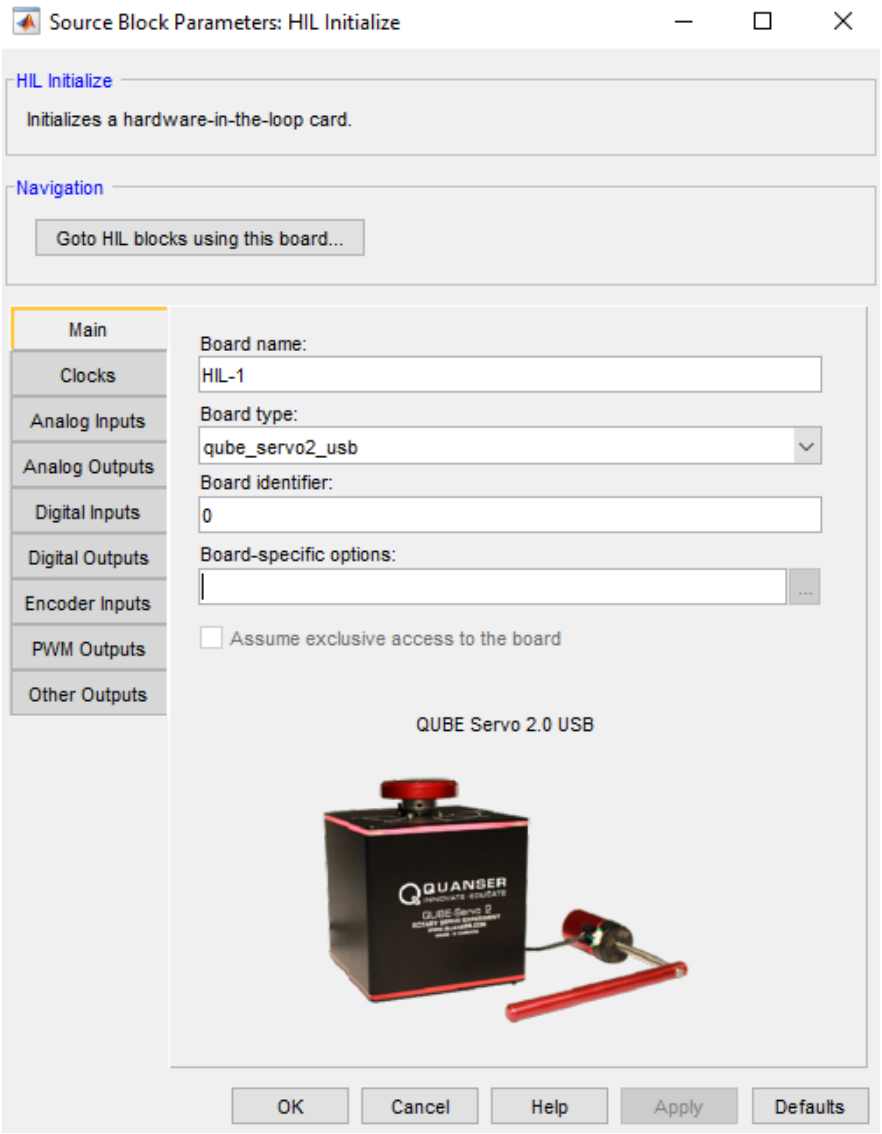


Figura 5.2: Configuración del bloque HIL Initialize para el QUBE-Servo 2.

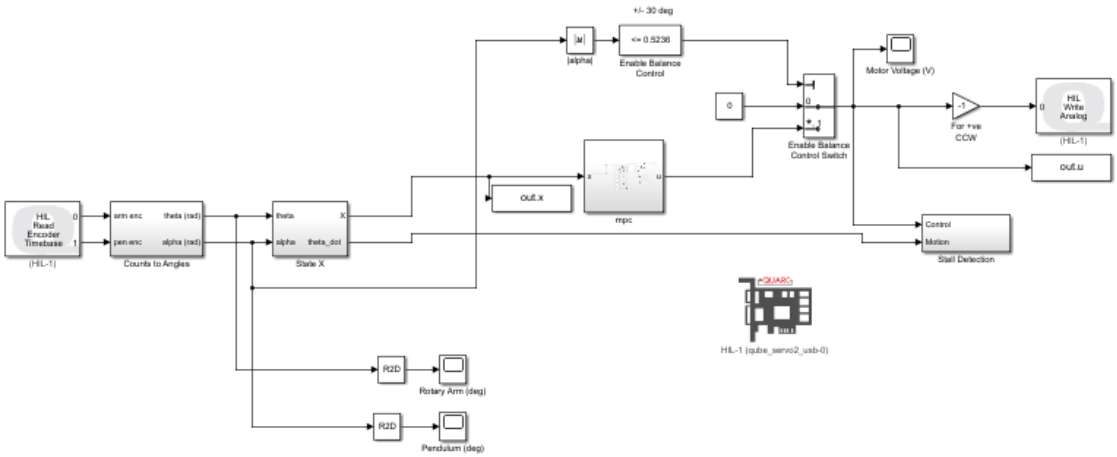


Figura 5.3: Diagrama completo de Simulink para la implementación en tiempo real (configuración con liberación manual, sin swing-up automático).

5.3 Configuración del MPC

La configuración del controlador MPC utilizada en la implementación física reutiliza exactamente los mismos parámetros validados en simulación (Sección 4.3): horizonte de predicción $N = 3$, matrices de ponderación $Q = \mathrm{diag}(5, 1, 1, 20)$ y $R = 1$, con el costo terminal Q_N obtenido de la ecuación de Riccati discreta asociada al mismo par (Q, R), y restricción de actuación $-10\ \text{V} \leq u \leq 10\ \text{V}$. Las matrices densas del problema QP (H, $\bar{A}$, $\bar{O}$, $\bar{Q}$, $\bar{R}$) se calculan fuera de línea con el mismo script `build_mpc_matrices.m` (ver Código 6.5) empleado en el Capítulo 4 y se cargan en el *workspace* de MATLAB antes de iniciar la ejecución en tiempo real. El bloque `MATLAB Function` que resuelve el problema QP mediante `quadprog` es idéntico al utilizado en simulación, de modo que la única diferencia estructural respecto al modelo de la Sección 4.2 es el reemplazo de los bloques de planta simulada por los bloques HIL (Sección 5.1.2), que conectan el controlador directamente con el hardware del QUBE-Servo 2. El paso fijo del solver de Simulink se configura en $T_s = 2$ ms, igual que en las pruebas de simulación, de modo que los resultados de este capítulo sean directamente comparables con los del Capítulo 4.

5.4 Metodología de inicio: liberación manual en el ángulo de trabajo

Versiones preliminares de este trabajo utilizaron una rutina de *swing-up* automática para llevar el péndulo desde el reposo hasta la vecindad de la posición invertida antes de activar el MPC de balance. Sin embargo, el comportamiento de esa rutina resultó inconsistente

entre ensayos sucesivos: la condición angular y de velocidad con la que el péndulo llegaba a la zona de habilitación del balance variaba de una corrida a otra, lo que impedía comparar resultados entre configuraciones bajo una condición inicial verdaderamente controlada. Por este motivo, los experimentos de caracterización de la Sección 5.5 se realizaron mediante **liberación manual**, con la única excepción de la demostración de swing-up (Sección 5.9), que emplea deliberadamente la rutina de swing-up automática: el operador eleva el péndulo con la mano hasta el ángulo de trabajo correspondiente a cada ensayo (por ejemplo, $\alpha_0 = 30^\circ$ en la configuración nominal) y lo sostiene brevemente en esa posición, con el controlador inactivo ($u = 0$). La comparación con el controlador DLQR (Sección 5.7) y la comparación frente a perturbaciones (Sección 5.8) también parten de una liberación manual, bajo la misma banda de habilitación de $\pm 30^\circ$ empleada en el resto de este capítulo. La conmutación se implementa en Simulink mediante un bloque de comparación sobre α_k y un `Multiport Switch`, de forma que

$$u = \begin{cases} u_{\text{MPC}} & \text{si } \alpha_k \text{ ingresa al ángulo de trabajo del ensayo,} \\ 0 & \text{en otro caso,} \end{cases} \tag{5.1}$$

de modo que, al soltar el péndulo, el controlador se activa automáticamente en el instante en que α_k cruza el ángulo objetivo, sin intervención adicional del operador. Este método reemplaza la maniobra automática de swing-up para efectos de la caracterización cuantitativa de este capítulo, y permite iniciar cada ensayo desde una condición inicial consistente y repetible, análoga a la utilizada en las pruebas de simulación del Capítulo 4. La rutina de swing-up automática originalmente implementada no se descarta: se documenta y demuestra, a modo de cierre de este capítulo, en la Sección 5.9, donde se ilustra su funcionamiento junto con el MPC en su configuración estándar de forma completamente autónoma.

5.5 Resultados experimentales

Esta sección presenta los resultados de la implementación física del controlador MPC sobre el QUBE-Servo 2, utilizando la configuración de software descrita en la Sección 5.3: horizonte $N = 3$, ponderaciones $Q = \mathrm{diag}(5, 1, 1, 20)$ y $R = 1$, período de muestreo $T_s = 2$ ms, restricción de actuación $-10\ \mathrm{V} \leq u_k \leq 10\ \mathrm{V}$ y estimador de velocidad angular por derivación filtrada $H(s) = \dfrac{50s}{s + 50}$. Para caracterizar la sensibilidad del controlador a sus principales parámetros de diseño, cada una de las siguientes subsecciones modifica un único parámetro respecto de esta configuración nominal, manteniendo el resto fijo, y compara la respuesta temporal de θ_k y α_k, la señal de actuación u_k y, cuando corresponde, el tiempo de ejecución del solver `quadprog` en cada paso de muestreo. Todas las pruebas parten mediante liberación

manual en el ángulo de trabajo correspondiente (Sección 5.4), replicando la condición inicial utilizada en las pruebas de simulación del Capítulo 4; el tiempo de establecimiento y el sobreimpulso se miden desde el instante en que el MPC se activa, no desde el inicio del registro. Además de los tres parámetros de diseño ya explorados en versiones anteriores de este trabajo (horizonte N, período T_s y restricción de actuación), se agrega en esta versión un cuarto barrido sobre el ángulo inicial α_0, que caracteriza el límite práctico de la región de atracción del controlador sobre el sistema real.

5.5.1 Efecto del horizonte de predicción

El horizonte de predicción N determina cuántos pasos futuros considera el MPC al resolver, en cada muestra, el problema de optimización cuadrática que define su ley de control. Manteniendo fijos $T_s = 2$ ms, Q, R y los límites de actuación, se repite el experimento de estabilización descrito arriba para distintos valores de N en torno al valor nominal utilizado en el resto de este trabajo ($N = 3$). Desde la teoría de control predictivo, se espera que horizontes más largos permitan al controlador anticipar con mayor precisión la dinámica inestable del péndulo y aproximarse al desempeño del controlador de horizonte infinito asociado al mismo par (Q, R), mejorando potencialmente el amortiguamiento y reduciendo el sobreimpulso, a costa de un problema QP de mayor dimensión —el número de variables de decisión del problema denso crece con N— y, por lo tanto, de un mayor tiempo de resolución de `quadprog` en cada paso. Horizontes más cortos, en cambio, resuelven más rápido pero disponen de una ventana de predicción más limitada frente a la dinámica rápida e inestable del péndulo, lo que puede traducirse en una respuesta más agresiva, mayor sobreimpulso o, en el límite, pérdida de estabilidad si N es demasiado pequeño para el ancho de banda del sistema. El principal riesgo práctico de aumentar N en este trabajo es de naturaleza temporal: dado que el controlador opera en tiempo real con $T_s = 2$ ms, el tiempo de resolución de `quadprog` debe mantenerse muy por debajo de este período en todo momento; un horizonte excesivamente largo podría acercar o superar este límite, comprometiendo la factibilidad en tiempo real del lazo de control. Los resultados de esta subsección, obtenidos con el hardware real para $N \in \{2, 3, 5, 7, 15, 30\}$, cuantifican ese punto: como se aprecia en la Tabla 5.1, el tiempo de ejecución del solver crece de forma moderada hasta $N = 15$, pero se dispara en $N = 30$ (tiempo máximo de 289.4 μs, un orden de magnitud por sobre los demás casos), evidenciando dónde comienza a reducirse de forma significativa el margen respecto al período de muestreo $T_s = 2$ ms, aunque sin comprometer la factibilidad en ninguno de los casos evaluados. Las Figuras 5.4 y 5.5 muestran, respectivamente, la respuesta temporal de θ_k/α_k y la señal de actuación u_k para cada valor de N evaluado, mientras que la Figura 5.6 compara los tiempos de ejecución de `quadprog` obtenidos en cada caso.

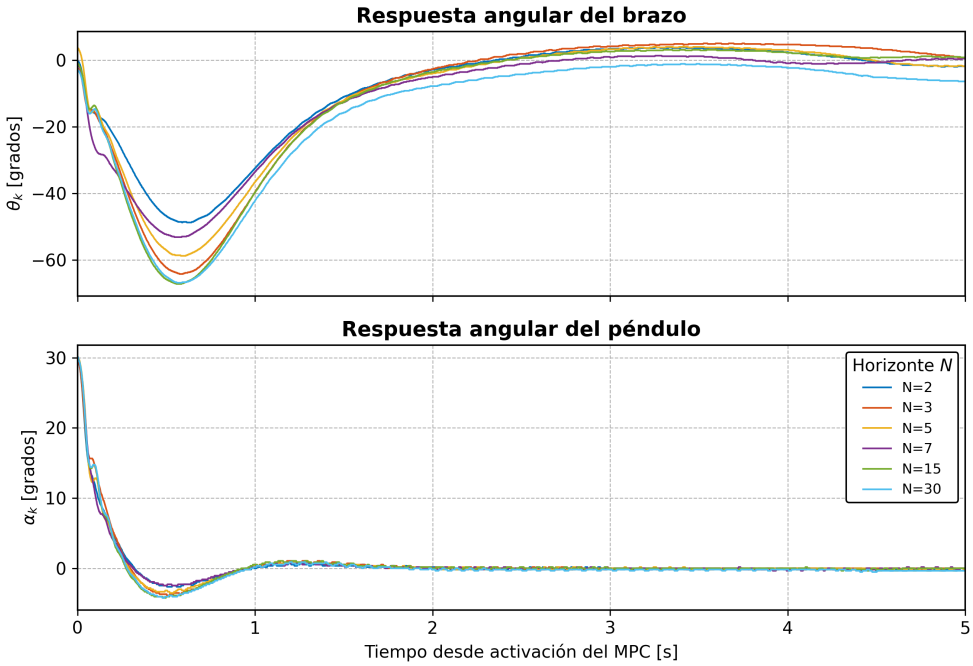


Figura 5.4: Respuesta temporal (θ_k, α_k) para distintos valores de N.

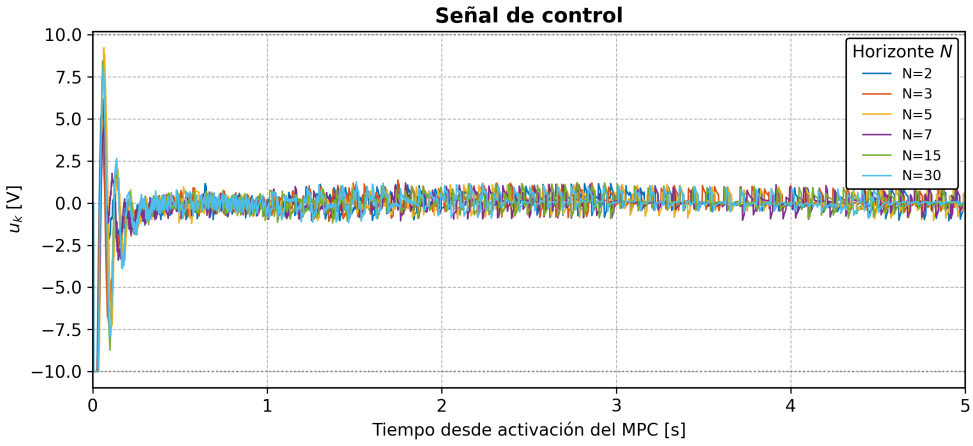


Figura 5.5: Señal de control u_k para distintos valores de N.

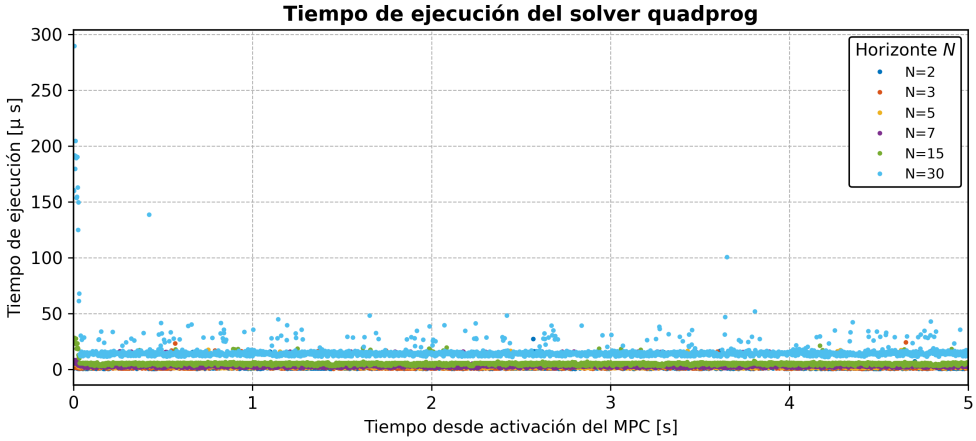


Figura 5.6: Tiempos de ejecución del solver para distintos valores de N.

5.5.2 Efecto del período de muestreo

El período de muestreo T_s afecta tanto a la discretización del modelo de la planta (A_d, B_d, obtenidos mediante discretización ZOH a partir de las matrices continuas A_c, B_c del Capítulo 3) como a la frecuencia con la que el MPC resuelve el problema de optimización y actualiza la señal de control. Manteniendo fijos $N = 3$, Q, R y los límites de actuación, se repite el experimento de estabilización para distintos valores de $T_s \in \{1, 2, 4, 6\}$ ms en torno al valor nominal de este trabajo ($T_s = 2$ ms), recalculando A_d, B_d y las matrices densas del problema QP para cada período evaluado mediante `build_mpc_matrices.m`. Períodos de muestreo más cortos permiten una aproximación más fiel de la dinámica continua e inestable del péndulo y una corrección más frecuente de la señal de control, lo que en general mejora el desempeño en lazo cerrado, pero a costa de una mayor carga computacional por unidad de tiempo —el solver debe resolverse más veces por segundo— y de una mayor sensibilidad al ruido de medición, al derivarse las velocidades angulares a partir de diferencias entre muestras más cercanas entre sí. Períodos más largos relajan la carga computacional, pero introducen mayor error de discretización y reducen la capacidad del controlador de reaccionar a la dinámica rápida del péndulo, lo que puede degradar el margen de estabilidad o producir oscilaciones más pronunciadas. Se eligió $T_s = 2$ ms como valor nominal en este trabajo por coincidir con el período utilizado por el controlador de referencia de Quanser para este mismo hardware (Sección 5.3); esta subsección caracteriza, con datos reales del sistema físico, el rango de períodos de muestreo en que el controlador mantiene un desempeño adecuado. Las Figuras 5.7 y 5.8 muestran la respuesta temporal y la señal de actuación para cada período evaluado, y la Figura 5.9 compara los tiempos de ejecución del solver.

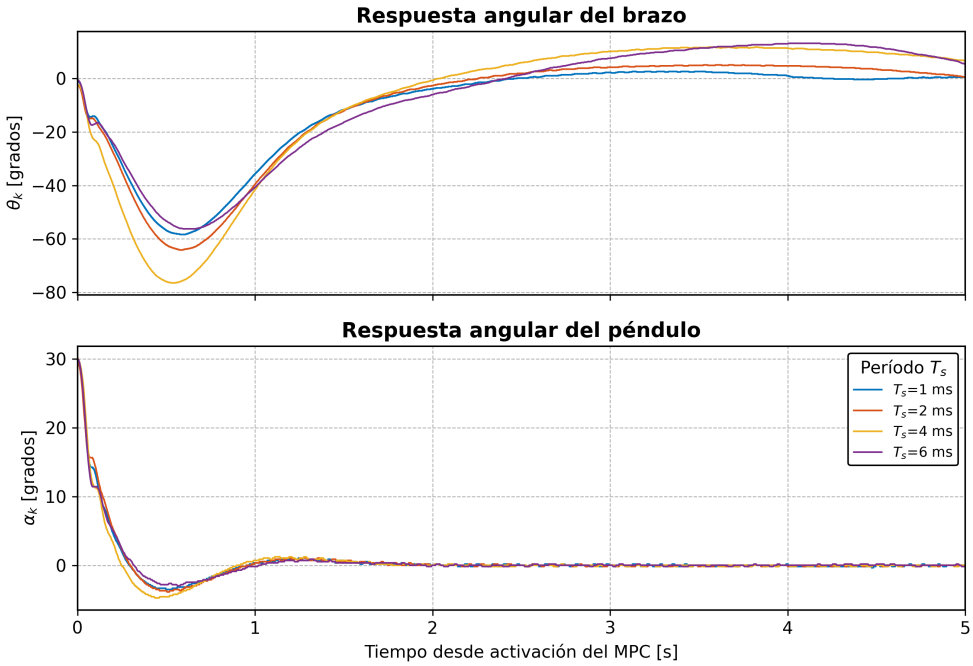


Figura 5.7: Respuesta temporal (θ_k, α_k) para distintos valores de T_s.

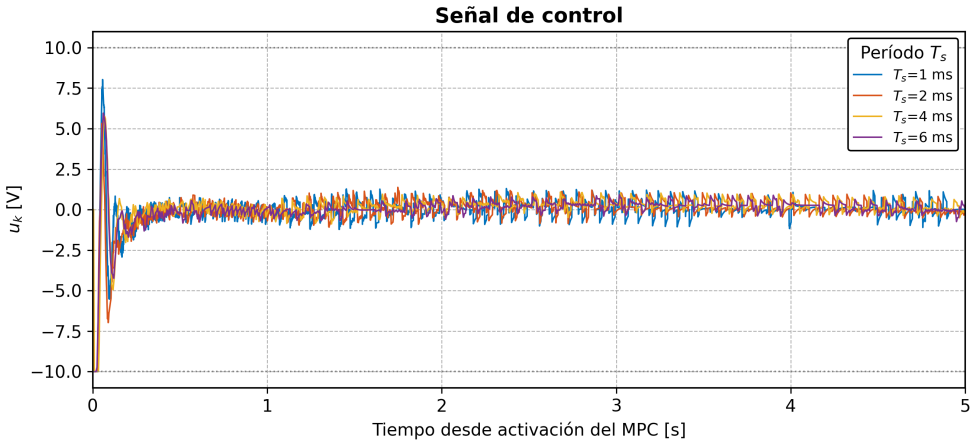


Figura 5.8: Señal de control u_k para distintos valores de T_s.

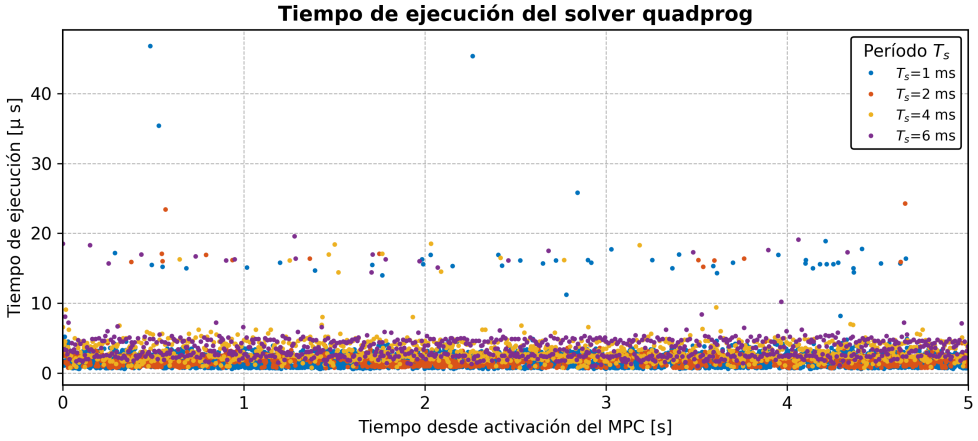


Figura 5.9: Tiempos de ejecución del solver para distintos valores de T_s.

5.5.3 Efecto de las restricciones de actuación

La restricción de actuación $-10\ \text{V} \leq u_k \leq 10\ \text{V}$, incorporada explícitamente como restricción del problema QP (Capítulo 2), refleja el límite de voltaje admisible por el amplificador del QUBE-Servo 2. Para evaluar su efecto sobre el desempeño del controlador, se repite el experimento de estabilización imponiendo un límite de actuación más restrictivo ($u_{\max} = 6$ V) que el nominal ($u_{\max} = 10$ V), manteniendo fijos $N = 3$, Q, R y $T_s = 2$ ms. Desde la teoría de control predictivo con restricciones, se espera que una restricción de actuación más estrecha limite la capacidad del controlador de aplicar el esfuerzo de control necesario durante el transitorio inicial —donde la desviación respecto del equilibrio invertido es mayor y, por lo tanto, la demanda de esfuerzo de control también—, provocando saturación de la señal de control durante un intervalo más prolongado que en el caso nominal. Esta saturación sostenida puede traducirse en una respuesta más lenta (mayor tiempo de establecimiento) o un mayor sobreimpulso. Los resultados de la Tabla 5.3 confirman esta predicción: bajo $u_{\max} = 6$ V el actuador satura durante 142 ms (frente a 24 ms en el caso nominal) y el sobreimpulso aumenta de $12.94\,\%$ a $19.41\,\%$, sin comprometer la estabilidad ni la factibilidad en tiempo real. La Figura 5.10 muestra la respuesta temporal de θ_k y α_k bajo ambos límites, la Figura 5.11 muestra la señal de control u_k evidenciando el intervalo de saturación más prolongado bajo $u_{\max} = 6$ V, y la Figura 5.12 compara los tiempos de ejecución del solver en ambos casos.

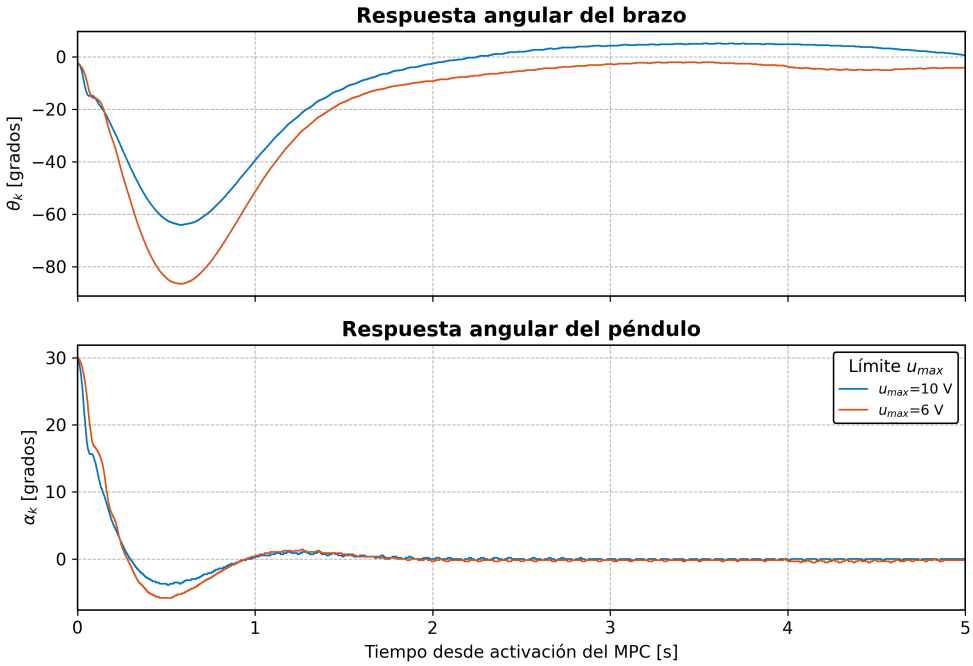


Figura 5.10: Respuesta temporal (θ_k, α_k) para distintas restricciones de actuación $u_{\max}$.

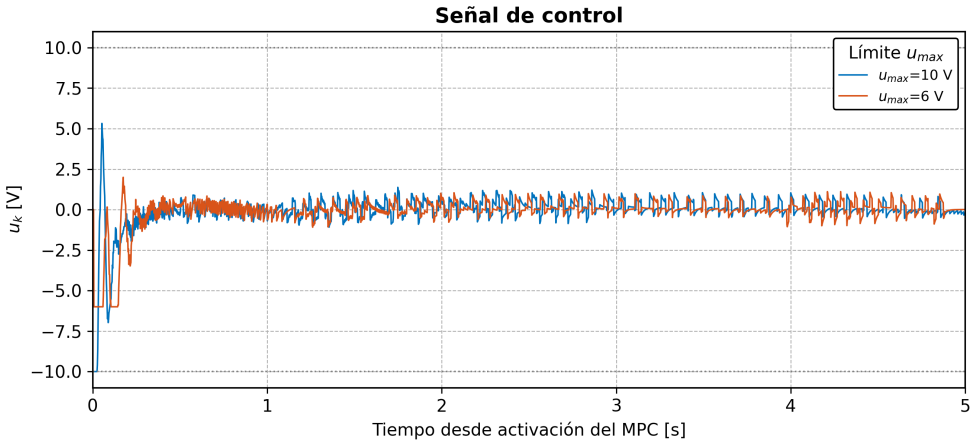


Figura 5.11: Señal de control u_k para distintas restricciones de actuación, mostrando la saturación activa bajo $u_{\max} = 6$ V.

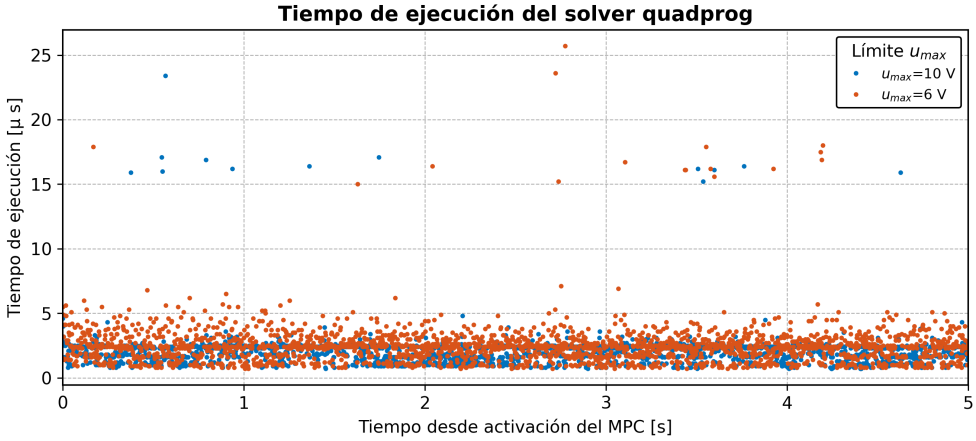


Figura 5.12: Tiempos de ejecución del solver para distintas restricciones de actuación.

5.5.4 Efecto del ángulo inicial

Las subsecciones anteriores caracterizan la sensibilidad del MPC a sus parámetros de diseño manteniendo fija la condición inicial ($\alpha_0 = 30°$). Esta subsección invierte el enfoque: mantiene fija la configuración nominal ($N = 3$, $T_s = 2$ ms, $u_{\max} = 10$ V) y varía el ángulo de trabajo α_0 desde el cual se libera manualmente el péndulo, con el objetivo de caracterizar el límite práctico de la región de atracción del controlador sobre el sistema real. Se evaluaron ángulos iniciales de $10°$, $20°$, $30°$, $40°$, $45°$ y $50°$. El caso de $10°$ se excluye del análisis cuantitativo: al ser la condición inicial pequeña, la banda de asentamiento relativa (2 % de α_0) resulta comparable a la resolución angular del encoder, por lo que el tiempo de establecimiento calculado no es representativo del desempeño real del controlador. Para $\alpha_0 = 50°$, en cambio, el péndulo no logra ser recapturado por el MPC: la desviación inicial supera el rango en que la ley de control linealizada en torno al equilibrio sigue siendo válida, el péndulo sobrepasa el ángulo de habilitación del balance y queda oscilando libremente (con $u = 0$) sin volver a estabilizarse dentro de la ventana de registro. Este resultado es en sí mismo relevante: indica que, para la configuración nominal de este trabajo, el límite práctico de la región de atracción del MPC sobre el hardware real se ubica entre $45°$ y $50°$. La Tabla 5.4 resume los resultados para los cuatro ángulos con asentamiento válido ($20°$, $30°$, $40°$ y $45°$); las Figuras 5.13 y 5.14 muestran la respuesta temporal y la señal de control para estos cuatro casos.

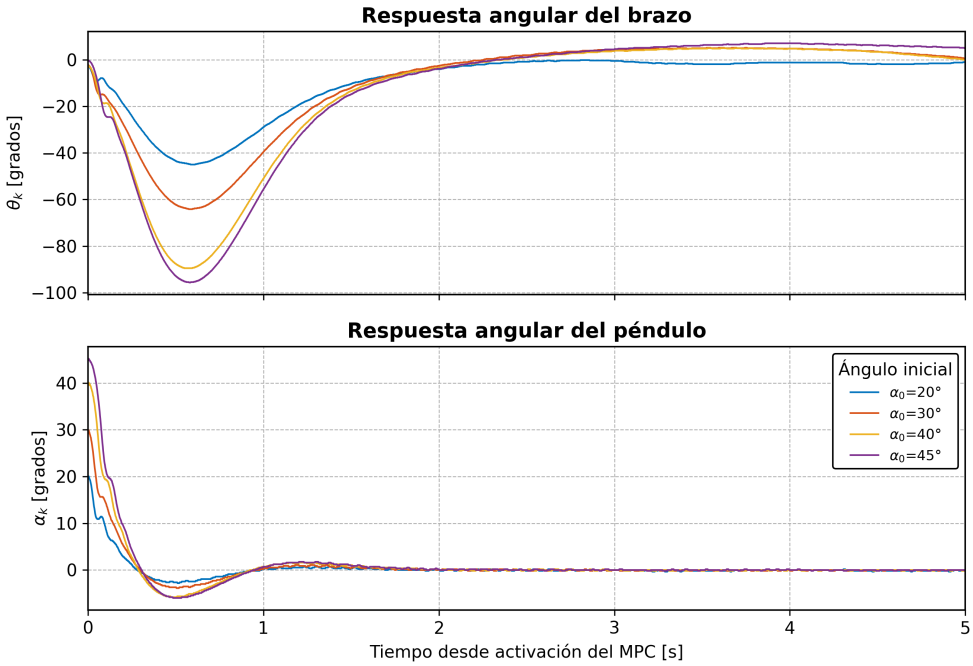


Figura 5.13: Respuesta temporal (θ_k, α_k) para distintos ángulos iniciales α_0.

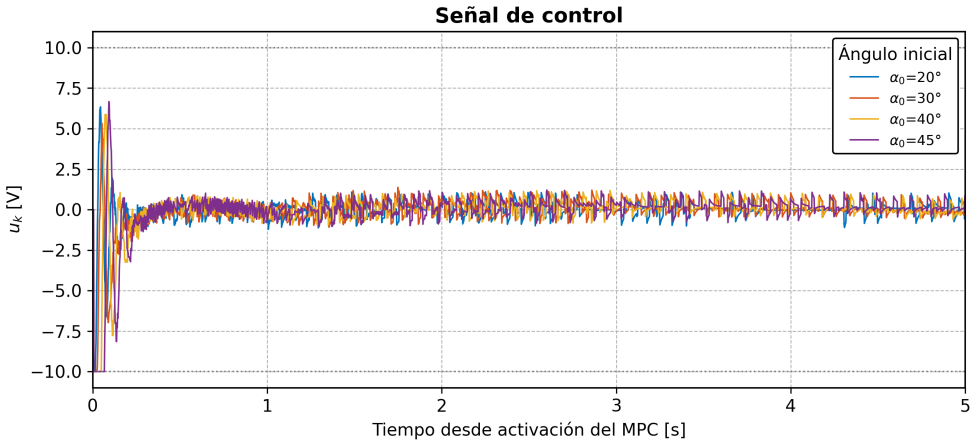


Figura 5.14: Señal de control u_k para distintos ángulos iniciales α_0.

5.6 Tablas de resultados

A modo de resumen cuantitativo de las subsecciones anteriores, las Tablas 5.1 a 5.4 reportan, para cada configuración evaluada, el tiempo de establecimiento t_s (banda del $2\,\%$, medido desde la activación del MPC), el sobreimpulso porcentual, el tiempo de ejecución del solver `quadprog` y la factibilidad en tiempo real del lazo de control.

Table 5.1: Efecto de variar el horizonte de predicción N (con $T_s = 2$ [ms] y restricciones $u \in [-10\ ,\ +10]$ [V]).

N	t_s [s]	Sobreimpulso [%]	t_{solver} medio [μs]	t_{solver} máx [μs]	Factible
2	1.434	8.82	1.68	27.20	Sí
3	1.544	12.94	2.02	24.30	Sí
5	1.474	11.76	2.49	17.50	Sí
7	1.476	8.24	3.23	17.10	Sí
15	1.558	14.12	5.20	28.10	Sí
30	1.484	14.12	14.89	289.40	Sí

Table 5.2: Efecto de variar el período de muestreo T_s (con $N = 3$ y restricciones $u \in [-10\ ,\ +10]$ [V]).

T_s [ms]	t_s [s]	Sobreimpulso [%]	t_{solver} medio [μs]	t_{solver} máx [μs]	Factible
1	1.500	11.76	2.03	46.80	Sí
2	1.544	12.94	2.02	24.30	Sí
4	1.520	15.88	3.03	18.50	Sí
6	1.434	10.00	3.71	19.60	Sí

Table 5.3: Efecto de variar la restricción de actuación $u_{\max}$ (con $N = 3$, $T_s = 2$ ms).

$u_{\max}$ [V]	t_s [s]	Sobreimpulso [%]	t_{solver} medio [μs]	t_{solver} máx [μs]	Factible
10	1.544	12.94	2.02	24.30	Sí
6	1.558	19.41	2.41	25.70	Sí

Table 5.4: Efecto de variar el ángulo inicial α_0 (con $N = 3$, $T_s = 2$ ms, $u_{\max} = 10$ V). Se excluyen 10° (artefacto de cuantización) y 50° (pérdida de control; ver Sección 5.5.4).

α_0 [°]	t_s [s]	Sobreimpulso [%]	t_{solver} medio [μs]	Factible
20	1.530	14.16	2.35	Sí
30	1.544	12.94	2.02	Sí
40	1.574	14.54	2.38	Sí
45	1.552	13.28	2.39	Sí

5.7 Comparación con controlador LQR

Para evaluar el aporte concreto del MPC frente a una alternativa más simple, se compara su desempeño con el controlador DLQR de balance descrito en la Sección 3.6.1, implementado de forma independiente (sin MPC) sobre el mismo hardware y el mismo par (A_d, B_d). A

diferencia del MPC, el DLQR es una ganancia de realimentación de estado fija, calculada una única vez fuera de línea, y no incorpora restricciones de actuación en su formulación: cualquier límite de voltaje debe aplicarse como una saturación posterior a la señal de control, sin que el controlador anticipe dicha restricción. Ambos controladores se evaluaron bajo la misma metodología de liberación manual descrita en la Sección 5.4: el péndulo se lleva con la mano hasta $\alpha_0 \approx 30^\circ$ y se libera, cayendo bajo gravedad hasta cruzar la banda de habilitación de $\pm 30^\circ$ (Sección 3.6.1), momento en el cual el controlador correspondiente se activa automáticamente. Ambos ensayos se realizaron bajo la misma configuración nominal de actuación ($-10\ \mathrm{V} \leq u_k \leq 10\ \mathrm{V}$ como límite de diseño), en corridas independientes para el MPC y el DLQR. La respuesta de cada controlador se presenta por separado, con el tiempo de cada figura reiniciado en su propio instante de activación, siguiendo la misma convención empleada en el resto de este capítulo (Sección 5.5). El tiempo de establecimiento se calcula con la misma banda relativa del $2\,\%$ utilizada en el resto de este capítulo (Sección 5.5.4); dado que $\alpha_0 \approx 30^\circ$ coincide con la condición inicial nominal de dicha sección, esto equivale a una banda absoluta de $\pm 0.6^\circ$ en torno al equilibrio.

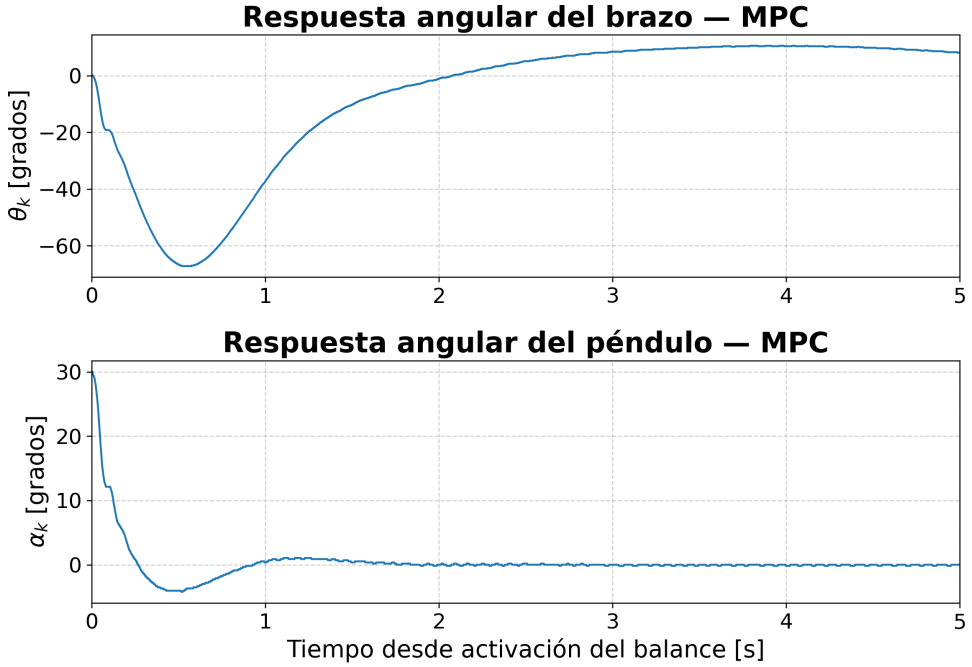


Figura 5.15: Respuesta angular (θ_k, α_k) del MPC en su configuración estándar, medida desde la activación del balance.

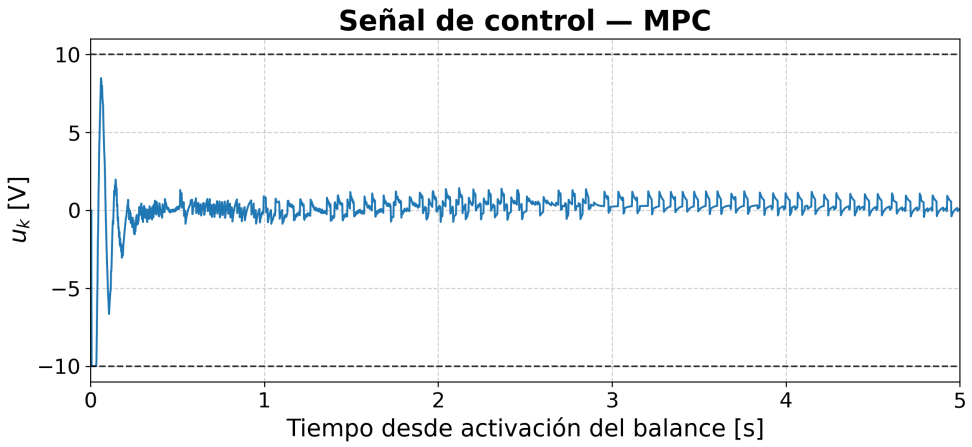


Figura 5.16: Señal de control del MPC en su configuración estándar, saturada en -10 V (línea discontinua) durante el transitorio inicial sin sobrepasar el límite.

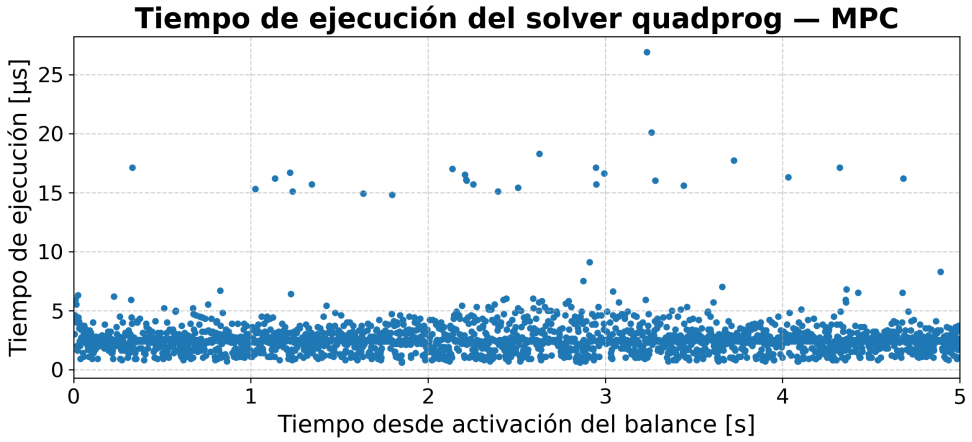


Figura 5.17: Tiempo de ejecución del solver `quadprog` durante la fase de balance del MPC.

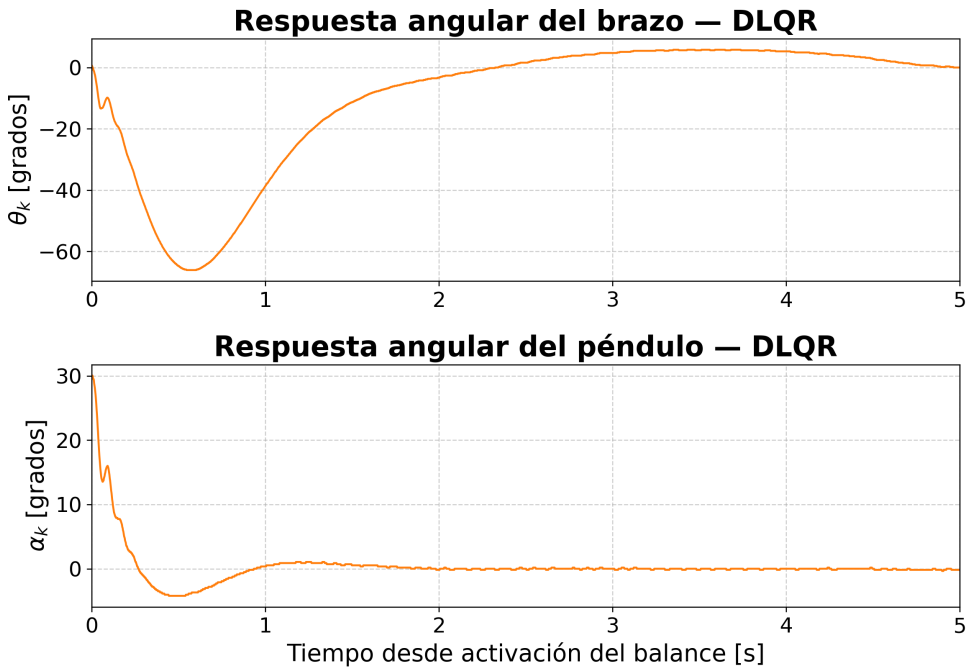


Figura 5.18: Respuesta angular (θ_k, α_k) del DLQR, medida desde la activación del balance.

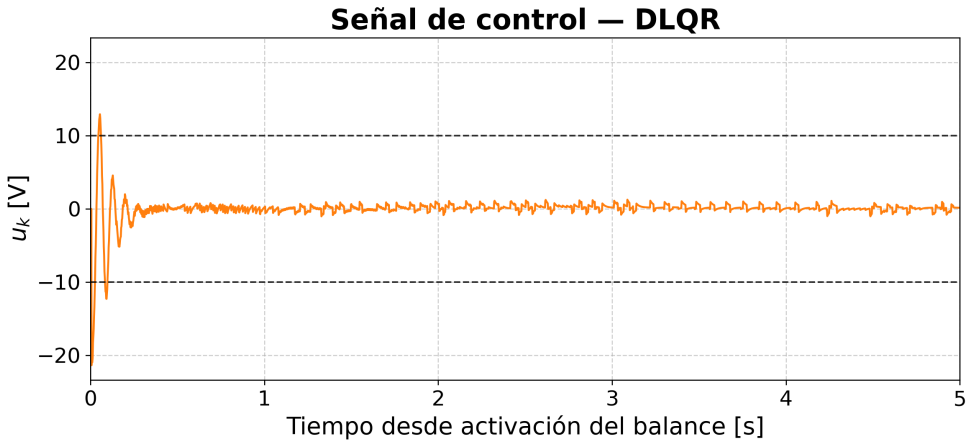


Figura 5.19: Señal de control del DLQR, que excede ampliamente el límite de ± 10 V (línea discontinua) durante el transitorio inicial.

La Tabla 5.5 resume las métricas cuantitativas. El tiempo de ejecución de `quadprog` no aplica al DLQR, ya que su ganancia se calcula una única vez fuera de línea y su evaluación en línea es una multiplicación matriz-vector de costo computacional despreciable frente al problema QP que resuelve el MPC en cada paso de muestreo.

Table 5.5: Comparación de desempeño entre MPC y DLQR en balance nominal, ambos liberados manualmente desde $\alpha_0 \approx 30^\circ$ (banda relativa del $2\,\%$, equivalente a $\pm 0.6^\circ$).

Controlador	t_s [s]	Sobreimpulso [%]	$u_{\max}$ observado [V]	Factible
MPC	1.478	14.12	10.00	Sí
DLQR	1.498	14.12	21.35	—

A diferencia de la versión anterior de este ensayo (basada en swing-up automático, con condiciones de entrada al balance dispares entre ambos controladores), esta vez ambos controladores parten de una condición inicial común y controlada ($\alpha_0 \approx 30^\circ$, liberación manual), lo que permite una comparación directamente controlada. Los tiempos de establecimiento (1.478 s para el MPC, 1.498 s para el DLQR) y los sobreimpulsos (14.12 % en ambos casos) resultan prácticamente idénticos, lo cual es consistente con que el costo terminal del MPC proviene precisamente de este mismo DLQR (Capítulo 2): en ausencia de restricciones activas, ambos controladores se comportan de forma equivalente. La diferencia estructural relevante emerge en la señal de control: el MPC alcanza exactamente -10 V durante el transitorio inicial sin sobrepasarlo, respetando por construcción el límite de actuación, mientras que el DLQR, al no incorporar restricciones en su diseño, exige hasta -21.35 V —más del doble del límite físico del QUBE-Servo 2— para lograr una respuesta de magnitud equivalente. Este resultado, obtenido ahora bajo una condición inicial común para ambos controladores, confirma directamente en esta sección lo que la Sección 5.8 ilustra frente a perturbaciones: el MPC logra el mismo desempeño de balance que el DLQR respetando siempre las restricciones físicas del actuador, mientras que el DLQR solo logra un desempeño comparable solicitando un esfuerzo de control que el hardware real no puede entregar.

5.8 Respuesta frente a perturbaciones

Las Secciones 5.5 y 5.7 caracterizan el desempeño del MPC bajo condiciones nominales y variaciones de diseño (horizonte, período de muestreo, restricción de actuación), y comparan su desempeño con el DLQR en un escenario nominal sin perturbaciones. Esta sección evalúa la respuesta de ambos controladores frente a perturbaciones manuales, usando la configuración nominal $N = 3$, $T_s = 2$ ms, actuación ± 10 V. A diferencia de la demostración de swing-up (Sección 5.9), aquí no se utilizó la rutina de swing-up automática: el péndulo se llevó manualmente cerca de la posición invertida, bajo la misma banda de habilitación de $\pm 30^\circ$ empleada en el resto de este capítulo, sin modificar la ganancia de balance de ninguno de los dos controladores. Una vez en fase de balance, se aplicaron sobre el péndulo varias perturbaciones manuales sucesivas (golpes/empujes) de intensidad creciente dentro de cada ensayo, registrando la traza completa (θ_k, α_k, u_k) para su posterior

análisis. Se realizaron cinco ensayos independientes con cada controlador (MPC y DLQR), bajo la misma configuración nominal.

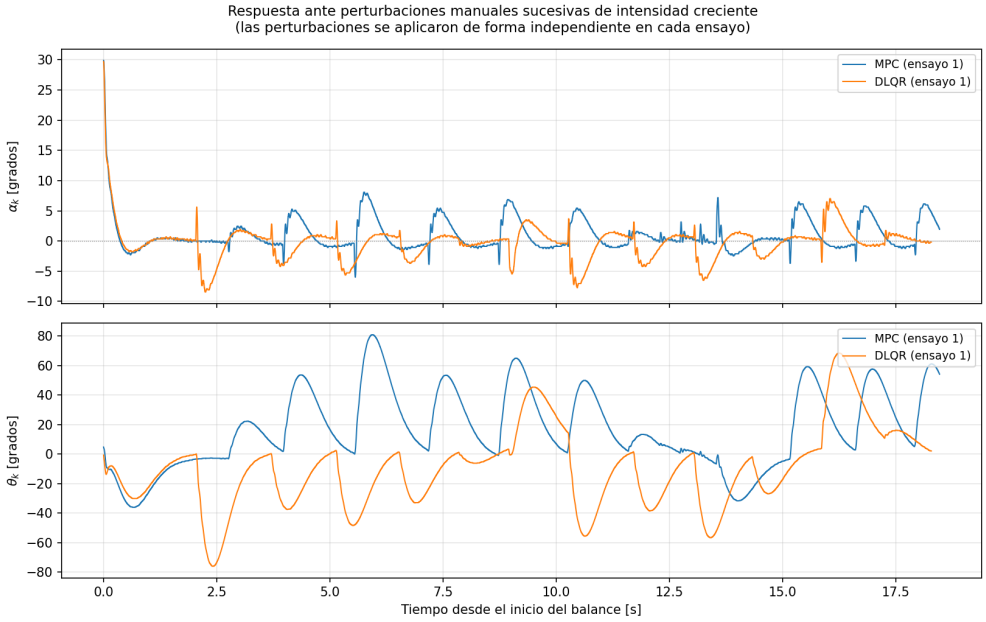


Figura 5.20: Respuesta temporal ante perturbaciones manuales, MPC vs DLQR.

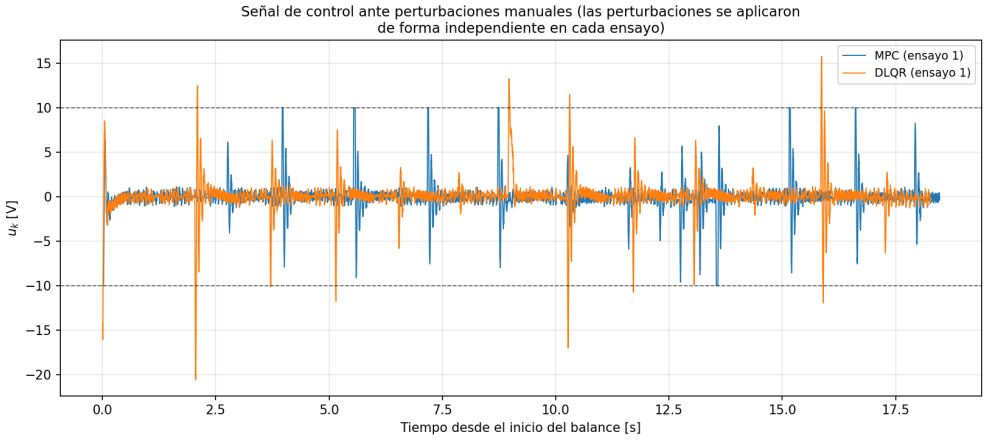


Figura 5.21: Señal de control u_k ante perturbaciones manuales, MPC vs DLQR.

Para cada ensayo se midió la desviación máxima de α tras la perturbación, el tiempo requerido para reingresar a la banda del 2% en torno al punto de equilibrio, y el valor máximo de la señal de control durante la recuperación. La Tabla 5.6 resume los resultados para ambos controladores.

Table 5.6: Respuesta frente a perturbaciones manuales: comparación MPC vs DLQR (desviación máxima, tiempo de recuperación y actuación máxima).

Ensayo	Controlador	Desviación máx. $\lvert\alpha\rvert$ [°]	$t_{\text{recuperación}}$ [s]	$u_{\max}$ [V]	Factible
1	MPC	7.91	0.378	10.0	Sí
1	DLQR	8.61	0.384	20.56	—
2	MPC	3.87	0.246	10.0	Sí
2	DLQR	6.50	0.352	19.28	—
3	MPC	6.15	0.346	10.0	Sí
3	DLQR	8.44	0.364	20.90	—
4	MPC	7.03	0.066	10.0	Sí
4	DLQR	8.61	0.312	22.73	—
5	MPC	7.91	0.386	10.0	Sí
5	DLQR	10.20	0.390	23.79	—

Los resultados confirman la tendencia observada en la Sección 5.7: ambos controladores logran desviaciones máximas de α y tiempos de recuperación comparables entre sí, sin una diferencia sistemática clara en estas dos métricas frente a perturbaciones de intensidad creciente. La diferencia relevante aparece en la magnitud de la señal de control: el MPC se mantiene siempre en su límite de ± 10 V (por diseño, dado que la restricción de actuación forma parte del problema de optimización), mientras que el DLQR –carente de mecanismo de restricción– exige entre 19 y 24 V en los cinco ensayos, más del doble de la tensión físicamente disponible en el hardware. En una implementación real donde el amplificador satura a ± 10 V, esta demanda del DLQR se traduciría en una señal de control recortada de forma no controlada, degradando el desempeño de forma impredecible frente a lo que logra el MPC al incorporar la restricción explícitamente en el diseño. Este resultado refuerza, con datos experimentales bajo perturbaciones reales, la ventaja estructural del MPC frente a controladores lineales sin manejo de restricciones.

5.9 Demostración de swing-up con MPC estándar

La caracterización cuantitativa de las secciones anteriores utiliza liberación manual como condición inicial (Sección 5.4), precisamente porque la rutina de *swing-up* automática mostró ser inconsistente entre ensayos. Sin perjuicio de ello, dicha rutina sigue implementada y disponible, y esta sección la demuestra operando junto con el MPC en su configuración estándar ($N = 3$, $T_s = 2$ ms, $u \in [-10, 10]$ V), para ilustrar la transición autónoma completa desde el péndulo en reposo hasta el balance invertido, sin intervención manual, en línea con el enfoque de swing-up y estabilización mediante MPC reportado en la literatura (Seman et al., 2013).

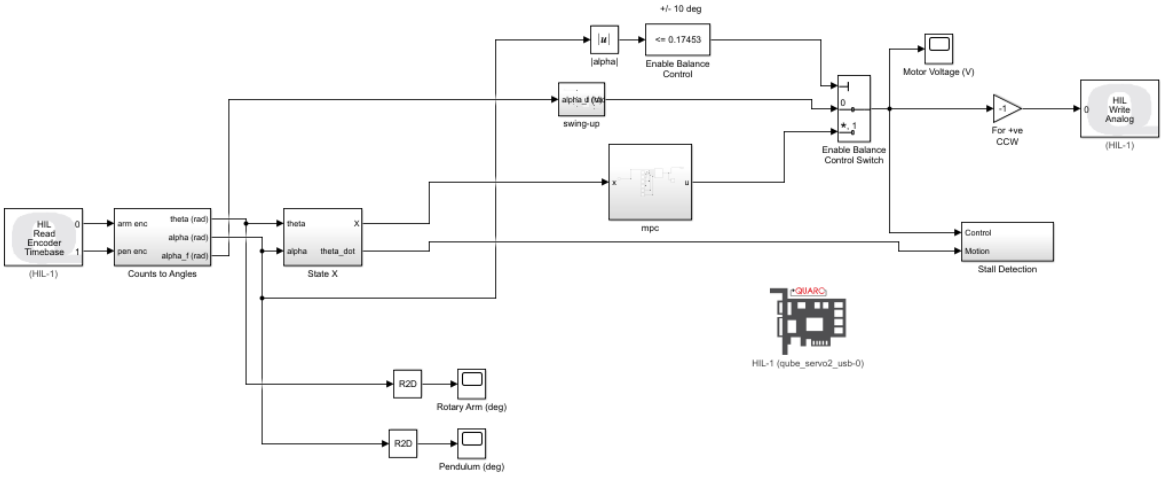


Figura 5.22: Diagrama completo de Simulink para la implementación en tiempo real, incluyendo el subsistema de swing-up automático.

El controlador de swing-up utilizado corresponde al método de control por energía provisto en el material de laboratorio de Quanser para el QUBE-Servo 2 (*Rotary Pendulum: Swing-Up Control*); no es una contribución original de este trabajo. Regula la energía mecánica del péndulo,

$$E = \frac{1}{2} J_p \dot{\alpha}^2 + \frac{1}{2} M_p g L_p (1 - \cos\alpha), \tag{5.2}$$

hacia una referencia E_r igual a la energía potencial en la posición invertida, mediante la ley de control no lineal

$$u_{\text{su}} = \text{sat}_{u_{\max}}\Big(\mu\,(E_r - E)\,\text{sign}(\dot{\alpha}\cos\alpha)\Big), \tag{5.3}$$

donde μ es una ganancia proporcional y $\text{sat}_{u_{\max}}$ satura la señal a la aceleración máxima admisible del pivote. Se usaron los valores sugeridos por Quanser: $\mu = 38\ \text{m/s}^2/\text{J}$, $E_r = 30\ \text{mJ}$ y $u_{\max} = 6\ \text{m/s}^2$. La conmutación entre swing-up y balance se implementa mediante un bloque `Compare To Constant` sobre $|\alpha|$ y un `Multiport Switch`:

$$u = \begin{cases} u_{\text{MPC}} & \text{si } |\alpha| \le 10^\circ, \\ u_{\text{su}} & \text{en otro caso.} \end{cases} \tag{5.4}$$

La Figura 5.23 muestra la respuesta angular completa de un ensayo representativo, partiendo del péndulo colgando en reposo ($\alpha = 180^\circ$): el swing-up bombea energía mediante oscilaciones de amplitud creciente hasta que, en $t \approx 10.16$ s, α cruza por primera vez de forma sostenida el umbral de 10° y el MPC toma el control, estabilizando el péndulo en la posición invertida. La Figura 5.24 muestra la señal de control correspondiente, evidenciando el cambio de carácter de la actuación en el instante de la conmutación: de la

ley no lineal de energía, con picos saturados a $\pm 6\ \mathrm{m/s^2}$ durante el bombeo, a la señal suave y acotada del MPC una vez en balance. Este resultado cierra el capítulo demostrando el sistema en su forma más autónoma: desde el péndulo colgando en reposo hasta el balance invertido, sin ninguna intervención manual del operador, integrando en una sola maniobra la ley de swing-up por control de energía y el controlador MPC caracterizado a lo largo de este capítulo. Cabe notar que la duración de esta maniobra no es estrictamente determinística de un ensayo a otro —comportamiento consistente con la variabilidad que motivó adoptar la liberación manual (Sección 5.4) como metodología para la caracterización cuantitativa de las secciones anteriores—, por lo que esta demostración se presenta como una prueba de factibilidad y no como parte de dicha caracterización.

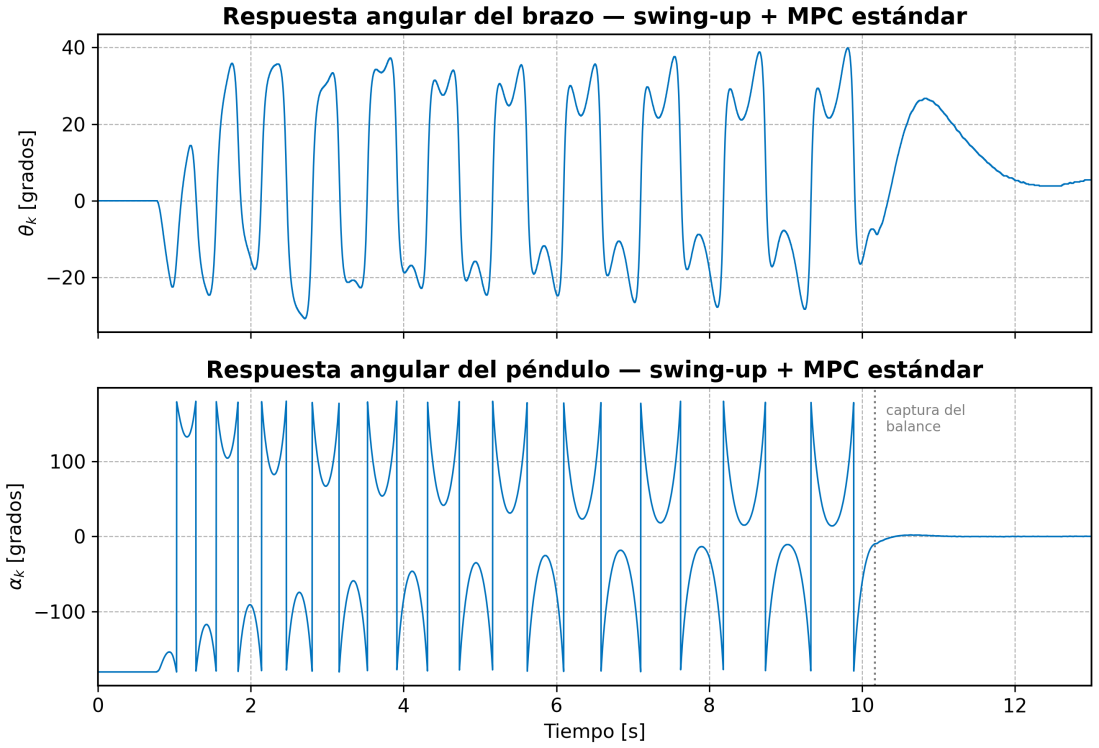


Figura 5.23: Respuesta angular durante la maniobra de swing-up y la posterior estabilización con MPC estándar.

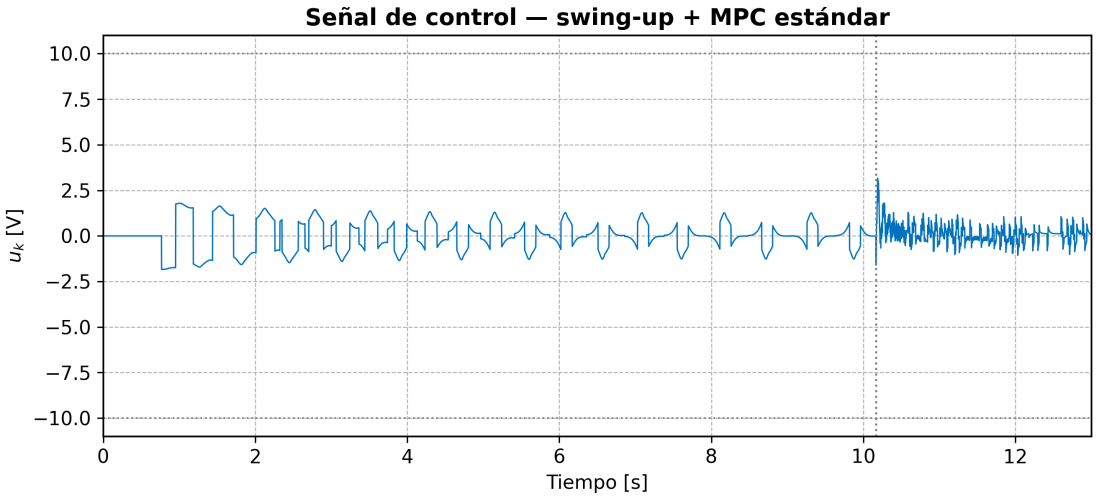


Figura 5.24: Señal de control durante swing-up y balance con MPC estándar.

CAPÍTULO

6 Conclusiones y trabajos futuros

6.1 Conclusiones

El modelo matemático del péndulo de Furuta, obtenido mediante el formalismo de Euler-Lagrange a partir de la documentación oficial del fabricante, y su posterior linealización en torno al punto de equilibrio inestable, resultaron adecuados para el diseño del controlador MPC. La discretización ZOH del modelo linealizado, verificada de forma independiente mediante un cálculo simbólico directo de las matrices Φ y Γ y contrastada con el resultado obtenido mediante el comando `c2d` de MATLAB, arrojó una coincidencia exacta entre ambos métodos, lo que respalda la validez del modelo discreto utilizado tanto en el diseño del MPC como en el DLQR de referencia. El buen desempeño obtenido tanto en simulación como en la plataforma física QUBE-Servo 2 confirma que la aproximación lineal del sistema en torno al equilibrio invertido fue suficientemente representativa de la dinámica real del péndulo para los fines de este trabajo. Respecto al controlador predictivo y su resolución mediante el algoritmo Active-Set del solver `quadprog`, los resultados de la Sección 5.6 mostraron que los tiempos de ejecución del solver se mantuvieron, para todos los horizontes de predicción evaluados ($N = 2, 3, 5, 7, 15, 30$) y para la restricción de actuación más estricta evaluada ($u_{\max} = 6$ V, t_{solver} máximo de $25.7\ \mu$s), muy por debajo del período de muestreo nominal $T_s = 2$ ms. Solo para el horizonte más grande evaluado ($N = 30$) se observó un aumento notable del tiempo de cómputo (máximo de $289.4\ \mu$s, un orden de magnitud por sobre los demás casos), sin llegar en ningún momento a comprometer la factibilidad en tiempo real del lazo de control. En cuanto al período de muestreo, se adoptó $T_s = 2$ ms como valor nominal, coincidente con el utilizado por el controlador de referencia de Quanser para este mismo hardware; el barrido experimental de la Sección 5.5.2 confirmó además que el controlador mantiene un desempeño adecuado en un rango razonable en torno a este valor ($T_s \in \{1, 2, 4, 6\}$ ms), sin observarse pérdida de estabilidad ni de factibilidad en ninguno de los casos evaluados. En conjunto, estos resultados indican que el par ($N = 3$, $T_s = 2$ ms) constituye un punto de operación robusto, con margen computacional suficiente para

su implementación en tiempo real. La estimación de las velocidades angulares $\dot{\theta}$ y $\dot{\alpha}$, no medidas directamente por los encoders del QUBE-Servo 2, se resolvió mediante un estimador de derivación filtrada ($H(s) = 50s/(s + 50)$) en lugar de un observador de estados más complejo. El desempeño satisfactorio del controlador MPC obtenido tanto en simulación como en las pruebas experimentales del Capítulo 5 sugiere que esta aproximación, más simple que un observador tipo Luenberger o un filtro de Kalman, fue suficiente para proporcionar una estimación de velocidades angulares con la calidad necesaria para no degradar de forma apreciable el desempeño del MPC respecto al caso con estados medidos directamente, al menos dentro del rango de operación evaluado. La implementación física del controlador en la plataforma QUBE-Servo 2 permitió validar experimentalmente su desempeño y compararlo con un controlador DLQR diseñado sobre el mismo modelo linealizado. Los resultados de la Sección 5.7, obtenidos con ambos controladores liberados manualmente desde una condición inicial común y controlada ($\alpha_0 \approx 30°$), mostraron que en un ensayo nominal sin perturbaciones ambos logran un desempeño de balance prácticamente idéntico en términos de tiempo de establecimiento y sobreimpulso (1.478 s y 14.12 % para el MPC, 1.498 s y 14.12 % para el DLQR), consistente con que el costo terminal del MPC proviene precisamente de este mismo DLQR. Sin embargo, la diferencia estructural relevante entre ambos ya se evidencia en este mismo ensayo nominal: el MPC alcanza exactamente -10 V sin sobrepasarlo, respetando por construcción el límite de actuación, mientras que el DLQR, al no incorporar restricciones en su diseño, exige hasta -21.35 V —más del doble del límite físico del amplificador—. La Sección 5.8 confirma esta misma tendencia bajo perturbaciones: la ganancia no restringida del DLQR requirió voltajes de entre 19 y 24 V frente a perturbaciones manuales —muy por encima del límite físico del amplificador—, lo que en una implementación real se traduciría en saturación de la señal de control y una consecuente degradación impredecible del desempeño. Esta capacidad de incorporar explícitamente las restricciones de actuación en el problema de optimización constituye la principal ventaja práctica del MPC sobre el DLQR para este sistema. Un hallazgo adicional de la caracterización experimental fue la identificación del límite práctico de la región de atracción del MPC sobre el sistema físico: el barrido del ángulo inicial de la Sección 5.5.4 mostró que el controlador estabiliza el péndulo de forma consistente para ángulos de hasta $45°$, pero pierde el control para $\alpha_0 = 50°$, delimitando así el rango de condiciones iniciales para el cual la aproximación lineal utilizada en el diseño del MPC sigue siendo válida en la práctica. Asimismo, las pruebas con la rutina de swing-up automática (Sección 5.9) revelaron una variabilidad significativa en la condición angular y de velocidad con la que el péndulo llega a la zona de habilitación del balance —de hasta el doble en duración entre ensayos nominalmente idénticos—, lo que motivó adoptar la liberación manual como metodología para la caracterización cuantitativa de este trabajo (Sección 5.4) y sugiere una oportunidad de mejora concreta para trabajos futuros.

En conjunto, los resultados obtenidos permiten concluir que se cumplió el objetivo general planteado en la Sección 1.3: diseñar e implementar un controlador predictivo basado en modelo capaz de estabilizar el péndulo de Furuta en su punto de equilibrio invertido, respetando las restricciones de voltaje del actuador. De los objetivos específicos, se completó la obtención y linealización del modelo matemático, la verificación de sus propiedades de controlabilidad y observabilidad, el diseño del regulador DLQR utilizado para determinar las matrices de ponderación del MPC, la implementación de la estimación de velocidades angulares, la formulación y resolución del problema de programación cuadrática mediante `quadprog`, y la validación del controlador tanto en simulación como en la plataforma física QUBE-Servo 2, evaluando su desempeño frente a variaciones de N, T_s, las restricciones de actuación y el ángulo inicial de liberación.

6.2 Trabajos futuros

A partir de los resultados obtenidos en este trabajo, se identifican las siguientes líneas de investigación y mejora como trabajos futuros:

- **MPC no lineal (NMPC):** Aplicar MPC directamente sobre el modelo no lineal del péndulo de Furuta para mejorar el desempeño fuera del punto de operación linealizado.
- **Robustecer la rutina de swing-up:** En este trabajo se implementó y demostró una rutina de swing-up por control de energía operando junto al MPC (Sección 5.9), pero su comportamiento resultó inconsistente entre ensayos sucesivos. Una línea de trabajo futuro es rediseñarla o resintonizarla para lograr una entrega repetible al MPC de balance, permitiendo prescindir de la liberación manual utilizada en la caracterización cuantitativa de este trabajo.
- **Ampliar la región de atracción:** Explorar rediseños de las ponderaciones (Q, R), del horizonte N, o esquemas de MPC no lineal, que permitan estabilizar el péndulo desde ángulos iniciales mayores a los $45°$ evaluados en este trabajo (Sección 5.5.4).
- **Estimador adaptativo:** Explorar estimadores que ajusten sus parámetros en línea ante variaciones en los parámetros del sistema o perturbaciones no modeladas.
- **MPC robusto:** Considerar incertidumbres en los parámetros del modelo para garantizar estabilidad y desempeño bajo variaciones.
- **Implementación en hardware embebido:** Migrar el controlador a un microcontrolador o FPGA para reducir la dependencia del PC.

BIBLIOGRAFÍA

Luenberger, D. G. (1971). An introduction to observers. *IEEE Transactions on Automatic Control*, 16(6):596–602.

MathWorks (2025). What is model predictive control? https://la.mathworks.com/help/mpc/gs/what-is-mpc.html. Accedido: 18 de agosto de 2025.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, 2nd edition.

Quanser Inc. (2014). *QUBE-Servo Inverted Pendulum Modeling*, v1.0 edition.

Richalet, J., Rault, A., Testud, J. L., and Papon, J. (1978). Model predictive heuristic control: Applications to industrial processes. *Automatica*, 14(5):413–428.

Seman, P., Rohaľ-Ilkiv, B., Juhás, M., and Salaj, M. (2013). Swinging up the furuta pendulum and its stabilization via model predictive control. *Journal of Electrical Engineering*, 64(3):152–158.

ANEXO — CÓDIGOS MATLAB

Este anexo presenta los scripts de MATLAB desarrollados para la caracterización del modelo del péndulo de Furuta y el diseño de los controladores (DLQR y MPC) utilizados en este trabajo, tanto en las simulaciones del Capítulo 4 como en la implementación física del Capítulo 5.

Parámetros físicos del sistema (`qube_rotpen_param.m`)

Define los parámetros físicos y eléctricos del péndulo de Furuta QUBE-Servo 2 (masas, longitudes, momentos de inercia, resistencia y constante del motor) utilizados en el modelo desarrollado en el Capítulo 3.

Código 6.1: qube_rotpen_param.m

```
%% Motor
% Resistencia
Rm = 8.4;
% Constante de par-corriente (N-m/A)
kt = 0.042;
% Constante de fuerza contraelectromotriz (V-s/rad)
km = 0.042;
%
%% Brazo Rotatorio
% Masa (kg)
Mr = 0.095;
% Longitud total (m)
Lr = 0.085;
% Momento de inercia respecto al pivote (kg-m^2)
Jr = Mr*Lr^2/12;
% Coeficiente de amortiguamiento viscoso equivalente (N-m-s/rad)
Dr = 0;
%
%% Eslabón del Péndulo
```

```
% Masa (kg)
Mp = 0.024;
% Longitud total (m)
Lp = 0.129;
% Momento de inercia respecto al pivote (kg-m^2)
Jp = Mp*Lp^2/12;
% Coeficiente de amortiguamiento viscoso equivalente (N-m-s/rad)
Dp = 0;
% Constante de gravedad
g = 9.81;
```

Modelo en espacio de estados y discretización (`QUBE_ROTPEN_ABCD_eqns.m`)

Construye las matrices (A, B, C, D) del modelo linealizado en tiempo continuo, incorpora la dinámica eléctrica del motor y obtiene las matrices discretas (A_d, B_d, C_d, D_d) mediante retenedor de orden cero con $T_s = 2$ ms, tal como se describe en el Capítulo 3.

Código 6.2: QUBE_ROTPEN_ABCD_eqns.m

```
% Representación en espacio de estados
Jt = Jr*Jp + Mp*(Lp/2)^2*Jr + Jp*Mp*Lr^2;
A = [0 0 1 0;
     0 0 0 1;
     0 Mp^2*(Lp/2)^2*Lr*g/Jt -Dr*(Jp+Mp*(Lp/2)^2)/Jt -Mp*(Lp/2)*Lr*Dp/Jt;
     0 Mp*g*(Lp/2)*(Jr+Mp*Lr^2)/Jt -Mp*(Lp/2)*Lr*Dr/Jt -Dp*(Jr+Mp*Lr^2)/Jt
         ↪ ];

B = [0; 0; (Jp+Mp*(Lp/2)^2)/Jt; Mp*(Lp/2)*Lr/Jt];
C = eye(4,4);
D = zeros(4,1);

% Añadir dinámica del actuador
B = kt * B / Rm;
A(3,3) = A(3,3) - kt*kt/Rm*B(3);
A(4,3) = A(4,3) - kt*kt/Rm*B(4);

% Cargar en el sistema en espacio de estados
rp_sys = ss(A,B,C,D);

% Discretizar el sistema en espacio de estados
Ts=2e-3;
rp_sys_d=c2d(rp_sys,Ts);
Ad=rp_sys_d.A;
Bd=rp_sys_d.B;
```

```
25 Cd=rp_sys_d.C;
26 Dd=rp_sys_d.D;
```

Script de configuración del modelo (`setup_qube_rotpen.m`)

Carga los parámetros físicos, construye el modelo en espacio de estados y ejecuta el diseño del regulador DLQR de balance.

Código 6.3: setup_qube_rotpen.m

```
1 %% Cargar Modelo
2 qube_rotpen_param;
3 QUBE_ROTPEN_ABCD_eqns;
4 % Mostrar matrices
5 A
6 B
7 find_dlqr;
```

Diseño del regulador DLQR (`find_dlqr.m`)

Calcula la ganancia del regulador cuadrático lineal discreto K_d sobre el par (A_d, B_d) con las matrices de ponderación $Q = \mathrm{diag}(5, 1, 1, 20)$ y $R = 1$, reutilizadas en el costo terminal del MPC (Capítulo 3).

Código 6.4: find_dlqr.m

```
1 %% Control de Balance
2 % Matrices de Ponderación del LQR
3 Q = diag([5 1 1 20]);
4 R = 1;
5 Kd = dlqr(Ad,Bd,Q,R)
```

Construcción de las matrices densas del QP (`build_mpc_matrices.m`)

Implementa la formulación densa del problema de control predictivo (Sección 2.1.3): construye las matrices $\bar{A}$, $\bar{O}$, $\bar{Q}$, $\bar{R}$ y la Hessiana H del problema QP a partir del modelo discreto y las matrices de ponderación.

Código 6.5: build_mpc_matrices.m

```
1 function [H, Abar, Obar, Qbar, Rbar] = build_mpc_matrices(Ad, Bd, Q, R, QN
      ↪ , N)
2 % BUILD_MPC_MATRICES  Construye las matrices del QP denso para MPC
3 %
4 % Formulación densa: ~x = Abar*x0 + Obar*~u
```

```
%
% ENTRADAS:
%   Ad   - Matriz de estado discreta      (nx x nx)
%   Bd   - Matriz de entrada discreta     (nx x nu)
%   Q    - Peso de estado en el costo     (nx x nx), semidefinida positiva
%   R    - Peso de entrada en el costo    (nu x nu), definida positiva
%   QN   - Peso terminal (S_inf de DARE) (nx x nx), definida positiva
%   N    - Horizonte de predicción (escalar)
%
% SALIDAS:
%   H      - Hessiana del QP: min (1/2)*~u'*H*~u + f'*~u   (N*nu x N*nu)
%   Abar   - Matriz de predicción de estados libres          (N*nx x nx)
%   Obar   - Matriz de Markov (resp. al impulso de entrada)  (N*nx x N*nu)
%   Qbar   - Peso de estados apilado con peso terminal       (N*nx x N*nx)
%   Rbar   - Peso de entradas apilado                        (N*nu x N*nu)

    nx = size(Ad, 1);   % número de estados
    nu = size(Bd, 2);   % número de entradas

    % Matrices de predicción
    Abar = zeros(N*nx, nx);
    Obar = zeros(N*nx, N*nu);

    for i = 1:N
        Abar((i-1)*nx+1 : i*nx, :) = Ad^i;
        for j = 1:i
            Obar((i-1)*nx+1 : i*nx, (j-1)*nu+1 : j*nu) = Ad^(i-j) * Bd;
        end
    end

    % Matrices de costo apiladas
    % Qbar: (N-1) bloques de Q + bloque terminal QN
    % Caso especial N=1: kron(eye(0),Q) es vacío, Qbar = QN directamente
    Qbar = blkdiag(kron(eye(N-1), Q), QN);   % (N*nx x N*nx)
    Rbar = kron(eye(N), R);                   % (N*nu x N*nu)

    % Hessiana del QP
    % H = 2*(Rbar + Obar'*Qbar*Obar)
    H = 2 * (Obar' * Qbar * Obar + Rbar);
    H = (H + H') / 2;   % simetría numérica
end
```

Configuración completa del controlador MPC (`mpc_furuta.m`)

Define el horizonte de predicción $N = 3$, las matrices de ponderación, el costo terminal (vía DARE) y las restricciones de actuación ± 10 V, y construye fuera de línea las matrices del QP mediante `build_mpc_matrices.m`, tal como se emplea en el bloque de control MPC en Simulink.

Código 6.6: mpc_furuta.m

```
% ============================================================
%  Configuración MPC para péndulo Furuta
% ============================================================

% Modelo del sistema
setup_qube_rotpen;

% Dimensiones
nx = size(Ad, 1);   % número de estados  [theta, alpha, dtheta, dalpha]
nu = size(Bd, 2);   % número de entradas [voltaje]

% Parámetros del MPC
N = 3;

Q  = diag([5, 1, 1, 20]);   % peso de estados
R  = 1;                     % peso de entrada

% Peso terminal: solución de la DARE (garantiza estabilidad)
[~, QN, ~] = dlqr(Ad, Bd, Q, R);

% Restricciones de entrada (saturación del amplificador)
lb = -10 * ones(N*nu, 1);   % voltaje mínimo [V]
ub =  10 * ones(N*nu, 1);   % voltaje máximo [V]

% Construcción offline de matrices del QP
[H, Abar, Obar, Qbar, Rbar] = build_mpc_matrices(Ad, Bd, Q, R, QN, N);

% Simulink
open("mpc_furuta_test.slx")
```

Bloque de control MPC en Simulink (`mpc_block.m`)

Función de MATLAB embebida en el bloque **MATLAB Function** del modelo de Simulink (Sección 4.2, Figura 4.5); resuelve el problema QP en cada paso de muestreo mediante `quadprog` con el algoritmo Active-Set, a partir de las matrices H, $\bar{A}$, $\bar{O}$, $\bar{Q}$ calculadas fuera

de línea por `build_mpc_matrices.m`, e incorpora alertas de divergencia y saturación, además de un mecanismo de respaldo (pseudo-inversa) ante fallas de convergencia del solver.

Código 6.7: mpc_block.m

```
function [u,t_solver]   = mpc_block(x, H, Obar, Abar, Qbar, lb, ub)
    %#codegen
% Declaramos funciones extrínsecas para poder usarlas en Simulink sin errores de codegen
    coder.extrinsic('warning', 'fprintf', 'num2str');
% --- Dimensiones y Consistencia ---
    N_nu = size(H, 1);
    u0 = zeros(N_nu, 1);
% --- Definición de Referencia ---
    x_ref = zeros(size(x));
    error_x = x - x_ref;
% --- Vector Lineal del QP ---
    f_vec = 2 * (Obar' * Qbar * Abar) * error_x;
% --- Configuración de quadprog ---
    options = optimoptions('quadprog', ...
'Display', 'off', ...
'Algorithm', 'active-set', ...
'ConstraintTolerance', 1e-6);
% --- Resolver QP ---
    t0=tic;
    [U_opt, ~, exitflag] = quadprog(H, f_vec, [], [], [], [], lb, ub, u0, options);
    t_solver=toc(t0);
% --- Monitoreo y Alertas (Debugging en Simulink) ---
    norm_x = norm(x);
if norm_x > 100
        warning(['[MPC WARNING] El estado del sistema está divergiendo. Norma actual de x: ', num2str(norm_x)]);
end
if exitflag == 1
        u = U_opt(1);
% Alerta si la solución óptima ya está tocando los extremos (Saturación activa)
% Tolerancia pequeña (1e-4) para capturar el límite numérico
if abs(u - lb(1)) < 1e-4 || abs(u - ub(1)) < 1e-4
            warning(['[MPC SATURACIÓN] La entrada óptima se ha pegado a los límites. u calculado: ', num2str(u)]);
end
else
% Si quadprog no converge (exitflag ~= 1)
```

```
36          warning(['[MPC CRÍTICO] quadprog falló con exitflag = ', num2str(
                ↪ exitflag), '. Aplicando fallback seguro.']);
37  % Fallback con pseudoinversa (U_unconstrained = -H \ f_vec)
38          u_unsat = -(H \ f_vec);
39          u = min(max(u_unsat(1), lb(1)), ub(1));
40  end
41  end
```

Ganancias del controlador de swing-up (`swing_up_gains.m`)

Calcula la energía de referencia y la aceleración máxima disponible en el pivote, parámetros utilizados en la ley de control por energía del swing-up descrita en el Capítulo 5.

Código 6.8: swing_up_gains.m

```
1  %% Control de Swing-Up
2  % Energía de referencia (J)
3  Er = Mp*g*Lp;
4  % Torque máximo para 10 V
5  tau_max = kt*10/Rm;
6  % Aceleración máxima del pivote (m/s^2)
7  a_max = tau_max / (Mr*Lr);
8  %
```