

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



“Control predictivo de un sistema Ball & Beam”

Vicente Ignacio Catalán Palma

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
EN ELECTRÓNICA**

PROFESOR GUIA: JUAN CARLOS AGÜERO VASQUEZ

PROFESOR CORREFERENTE: ÁNGEL LEONEL CEDEÑO NIETO

ABRIL 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: Control predictivo de un sistema Ball & Beam

Nombre del candidato(a): Vicente Ignacio Catalán Palma

Carrera / Grado: Ingeniería Civil Electrónica.

Campus: Casa Central Valparaíso. Departamento: Electrónica.

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Juan Carlos Agüero Vásquez, 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: 04/05/2026 Firma: 

Estudiante o Candidato(a):

Fecha: 04/05/2026 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

Primero que todo, quiero agradecer a mi hermano, mamá y papá de todo corazón. Especialmente a mi padre: gracias por darme todo en la vida y por velar que nunca me faltara nada en ningún ámbito. Gracias por estar ahí, apoyándome cuando más lo necesité, y por todas esas horas enseñándome las materias del colegio; valoro enormemente tu sacrificio. Gracias por enseñarme a nunca rendirme, a perseverar y por ser un pilar fundamental en mi crecimiento, tanto académico como personal. Gracias por ser un ejemplo para mí; no pude haber pedido a un mejor papá. Te amo.

Agradecer también a mi mamá por cuidarme y ser también un pilar fundamental en todo mi desarrollo, Te amo mucho.

Agradezco también a mis amigos y a las personas con las que he compartido a lo largo de mi vida, incluso a aquellos que ya no están presentes o con quienes ya no tengo contacto, puesto que cada uno ha aportado, de una u otra forma, a lo que soy hoy en día.

La vida no sería la misma sin las amistades que forjé antes y durante la universidad. Por eso, quiero agradecer personalmente a cada uno de ellos, a quienes llevo en el corazón en todo momento: Diego, Negro, Javi, Arturo, Lillo, Helian, Cata, Nano, Boris, Shio, Kiara, Flippy, Arce, Mariel, Nacha, Cristian, Tuto y Potter. Si bien algunos llegaron antes que otros a mi vida, todos se han ganado un espacio en mi ser.

Por último agradecer al Centro Avanzado de Ingeniería Eléctrica y Electrónica ANID AC3E CIA250006, al proyecto Fondecyt ANID 1261290 del profesor Juan C. Agüero, al proyecto Fondecyt ANID 3240181 del profesor Ángel Cedeño y al Proyecto Interno USM 2025 *PI_LIR_25_12*.

“Sólo habla desde el corazón, y no olvides sonreír”

Resumen

Este trabajo presenta el diseño, implementación y validación en un entorno de simulación de una estrategia de Control Predictivo basado en Modelo (MPC) aplicada a un sistema Ball & Beam de Quanser. El objetivo principal es desarrollar una arquitectura de control robusta que garantice la estabilidad, precisión en el seguimiento de trayectorias y el cumplimiento estricto de restricciones operativas. Debido a que la planta presenta una estabilidad marginal, se diseñó una etapa de pre-estabilización mediante un Regulador Lineal Cuadrático Discreto (DLQR), permitiendo que el optimizador MPC opere sobre una planta aumentada con una dinámica controlada.

La metodología se centra en la gestión de las limitaciones dinámicas del sistema dentro del simulador. Para asegurar un comportamiento suave del modelo, se incorporaron restricciones de Slew Rate en el algoritmo de optimización, mitigando cambios bruscos en el actuador. Asimismo, se implementó un observador de estado extendido de 9 estados para la reconstrucción de variables internas y la estimación de perturbaciones de salida, permitiendo un rechazo efectivo de disturbios.

Las simulaciones, realizadas en MATLAB/Simulink utilizando el solver Quadprog con el método Active-Set, validaron la eficacia de la propuesta. Los resultados muestran que el sistema alcanza los objetivos de control bajo diversas condiciones, con tiempos de cómputo en el orden de 10^{-4} a 10^{-3} segundos. Se concluye que el controlador propuesto logra un correcto desempeño, cumpliendo con los parámetros de diseño y restricciones físicas para el modelo.

Palabras Clave: MPC, Active-set, Ball & Beam, MATLAB/Simulink, Observador de estados, Quarc.

Abstract

This work presents the design, implementation, and validation in a simulation environment of a Model Predictive Control (MPC) strategy applied to a Quanser Ball & Beam system. The main objective is to develop a robust control scheme that guarantees stability, trajectory tracking precision, and strict compliance with operational constraints. Since the plant presents marginal stability, a pre-stabilization stage using a Discrete Linear Quadratic Regulator (DLQR) was designed, allowing the MPC optimizer to operate on an augmented plant with controlled dynamics.

The methodology focuses on managing the system's dynamic limitations within the simulator. To ensure smooth model behavior, Slew Rate constraints were incorporated into the optimization algorithm, mitigating abrupt changes in the actuator. Likewise, a 9-state extended state observer was implemented for the reconstruction of internal variables and the estimation of output disturbances, allowing for effective disturbance rejection.

The simulations, performed in MATLAB/Simulink using the Quadprog solver with the Active-Set method, validated the effectiveness of the proposal. The results show that the system achieves the control objectives under various conditions, with computation times in the order of 10^{-4} to 10^{-3} seconds. It is concluded that the proposed controller achieves a correct performance, complying with the design parameters and physical constraints for the model.

Keywords: MPC, Active-set, Ball & Beam, MATLAB/Simulink, State observer, Quarc.

Glosario

- MPC:** Control predictivo por modelo. Estrategia que utiliza un modelo dinámico para predecir el comportamiento futuro y optimizar la acción de control sujeta a restricciones.
- LQR:** Regulador lineal cuadrático. Método de control óptimo que minimiza una función de costo cuadrática para sistemas dinámicos lineales.
- DLQR:** Regulador lineal cuadrático discreto. Versión en tiempo discreto del LQR, utilizada para estabilizar la planta previo a la etapa de optimización del MPC.
- PID:** Controlador Proporcional-Integral-Derivativo. Algoritmo de control por retroalimentación que calcula la acción de control basada en el error del sistema.
- Active-Set:** Algoritmo de optimización iterativo para problemas de programación cuadrática que identifica las restricciones activas en el punto óptimo.
- Slew rate:** Restricción sobre la tasa de cambio máxima de la acción de control para evitar variaciones bruscas y proteger los componentes mecánicos.
- QP:** Programación Cuadrática. Problema de optimización matemática con función objetivo cuadrática y restricciones lineales.
- Quadprog:** Solver de optimización de MATLAB especializado en la resolución eficiente de problemas de programación cuadrática.
- KKT:** Condiciones de Karush-Kuhn-Tucker. Condiciones necesarias de primer orden para que una solución en optimización sea óptima bajo restricciones.
- Matlab/Simulink:** Entorno de computación numérica y diseño basado en modelos utilizado para la simulación y análisis de sistemas de control.
- LTI:** Sistema Lineal e Invariante en el Tiempo. Clasificación de sistemas dinámicos que cumplen con las propiedades de linealidad y estacionariedad.

Índice general

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
1.3. Objetivos del trabajo	3
2.. Antecedentes Teóricos	5
2.1. Control Predictivo por Modelo (MPC)	5
2.1.1. Funcionamiento y Formulación Cuadrática	6
2.1.2. Definición del Problema de Control	6
2.2. Formulación del Problema QP	9
2.2.1. Formulación Densa	9
2.2.2. Seguimiento de Referencia	12
2.2.3. Restricción de Desigualdad	14
2.2.4. Planteamiento final de problema QP	14
2.2.5. Resultado Final	15
2.3. Método de optimización Active Set Quadprog	15
2.3.1. Procedimiento Operativo del Algoritmo	16
2.4. Muestreo en sistemas de control	17
2.4.1. Representación en espacios de estado discreto	18
2.5. Estimadores de estado u observadores	20
2.5.1. Estructura general de los observadores de estados	21
2.5.2. Observador extendido para la estimación de perturbaciones	22
3.. Planta de interés	24
3.1. Sistema Ball & Beam	24
3.2. Modelo matemático del sistema Ball & Beam.	26
3.2.1. Convenciones y supuestos generales	26
3.2.2. Subsistema Servomotor.	27

3.2.3. Supuestos y definiciones del servomotor	27
3.2.4. Ecuaciones físicas del servomotor DC	28
3.2.5. Transmisión y reflexión de la carga	28
3.2.6. Reflexión de inercia y fricción	29
3.2.7. Modelo en el dominio de Laplace	30
3.2.8. Modelo simplificado del servomotor	30
3.2.9. Modelamiento de la planta Bola-Viga.	31
3.2.10. Diagrama de cuerpo libre.	31
3.2.11. Relación geométrica entre el ángulo del servomotor y el ángulo de la viga.	32
3.2.12. Ecuaciones de movimiento no lineales.	33
3.2.13. Linealización del subsistema Bola-viga.	33
3.2.14. Aproximación de Taylor de primer orden.	34
3.2.15. Obtención de la función de transferencia del modelo lineal del sistema Ball & Beam.	34
3.3. Obtención del modelo en espacios de estado	35
3.3.1. Modelo en espacio de estados en tiempo continuo	36
3.3.2. Modelo completo en espacio de estados en tiempo continuo	37
3.3.3. Modelo en espacio de estados en tiempo discreto	37
3.3.4. Análisis de Observabilidad y Controlabilidad	39
3.3.5. Controlabilidad	39
3.3.6. Observabilidad	40
3.3.7. Pre-estabilización de la planta	41
3.3.8. Planta aumentada	44
3.3.9. Representación matricial aumentado	44
3.3.10. Observador extendido para estimación de perturbaciones	44
3.3.11. Obtención de la Ganancia Óptima del Observador	45
3.3.12. Matrices del estimador de orden extendido	45
4. Simulación del Sistema de Control	46
4.1. Herramientas de Software	46
4.1.1. Matlab	46
4.1.2. Simulink	47
4.1.3. Preparación simulación	48
4.1.4. Pruebas DLQR	53
4.1.5. Pruebas de simulación MPC sin perturbaciones	55
4.1.6. Pruebas de simulación MPC con perturbaciones	61
4.1.7. Implementación de condiciones slew rate	68
4.1.8. Preparación de la simulación	69

Índice general

4.1.9. Resultados de la implementación	71
4.2. Tablas de resultados	89
5. Conclusiones y trabajos a futuro	93
5.1. Conclusiones	93
5.1.1. Trabajos a futuro	94
Referencias	95
Anexo . Códigos MATLAB	97
1. Calculo de A_d y B_d	97
2. Controlabilidad y Observabilidad Sistema Discreto	98
3. Calculo de A y B caligráficas (A_{bar} y O_{bar})	100
4. PreCompilar Ball And Beam	101
5. Precompilar Ball and beam con Slew rate	104
6. Calculo M y c con slew rate	108

Índice de figuras

3.1. Componentes del sistema Ball & Beam.[17]	25
3.2. Esquema servomotor DC.[5]	27
3.3. Diagrama de cuerpo libre bola y viga.[1]	32
3.4. Diagrama de bloque a lazo abierto.	35
3.5. Diagrama de la ley de control con prealimentación de estados.	41
4.1. Entorno de programación Matlab.	46
4.2. Entorno de programación Simulink.	47
4.3. Modelo Simulink planta aumentada.	51
4.4. Observador extendido de la planta aumentada.	51
4.5. Bloque seguimiento a referencia.	51
4.6. Bloque MPC Simulink.	52
4.7. Esquema de control MPC con seguimiento a referencia.	53
4.8. Estimación de la perturbación realizada por el observador externo.	54
4.9. Posición y ángulo de la bola DLQR con perturbación $d=0.02$ [m].	54
4.10. Posición y ángulo de la bola sin perturbaciones.	55
4.11. Posición y ángulo de la bola sin perturbaciones.	55
4.12. Posición y ángulo de la bola condición inicial -5 [cm] sin perturbaciones.	56
4.13. Valor detallado overshoot condición inicial -5 [cm].	56
4.14. Estados del observador sin perturbaciones.	57
4.15. Estado ángulo viga de la planta y del observador sin perturbaciones.	57
4.16. Estado velocidad angular viga de la planta y del observador sin perturbaciones.	58
4.17. Estados posición de la planta y del observador sin perturbaciones.	58
4.18. Estados velocidad bola de la planta y del observador sin perturbaciones.	59
4.19. Tiempo de ejecución del solver sin perturbaciones	59
4.20. Actuación de la planta	60
4.21. Voltaje de salida de MPC.	60
4.22. Valor ángulo detallada	61
4.23. Perturbación de salida original ingresada al sistema.	62
4.24. Perturbación estimada por el observador externo.	62

4.25. Posición bola y ángulo de la viga con solo perturbaciones.	63
4.26. Posición bola y ángulo viga con perturbaciones detallada.	63
4.27. Posición bola y ángulo viga con condición inicial 0 [cm].	64
4.28. Posición bola y ángulo viga con condición inicial 0 [cm] overshoot detallado después de la primera perturbación.	64
4.29. Posición bola y ángulo viga con condición inicial 5 [cm] detallada después de la segunda perturbación.	65
4.30. Posición bola y ángulo viga detallada.	65
4.31. Tiempo de ejecución del solver Quadprog.	66
4.32. Tiempo de ejecución del solver Quadprog detallado.	66
4.33. Estados originales del sistema y de la planta aumentada.	67
4.34. Actuación de la planta con perturbaciones.	67
4.35. Voltaje de salida MPC con perturbaciones.	68
4.36. Esquema de control MPC con Slew Rate.	69
4.37. Esquema seguimiento a referencias para las pruebas.	70
4.38. Esquema final MPC con slew rate.	70
4.39. Posición bola y ángulo viga condiciones iniciales 0 [cm] con slew rate.	71
4.40. Posición bola y ángulo viga condiciones iniciales 0 [cm] con slew rate detallado.	71
4.41. Posición bola y ángulo viga condiciones iniciales 0 [cm] con slew rate detallado2.	72
4.42. Posición bola y ángulo viga seguimiento a referencia.	72
4.43.	73
4.44. Actuación de la planta con slew rate.	73
4.45. Estados del sistema ampliado con slew rate.	74
4.46. Tiempo de ejecución solver Quadprog con slew rate.	74
4.47. Tiempo de ejecución solver Quadprog con slew rate detallado.	75
4.48. Posición bola y ángulo viga nuevos valores.	76
4.49. Overshoot posición bola y ángulo viga nuevos valores.	76
4.50. Posición bola y ángulo viga valor $1e8$.	77
4.51. Overshoot posición bola y ángulo viga valor $1e8$.	78
4.52. Actuación de la planta con slew rate caso 7mo estado valor $1e8$.	78
4.53. Voltaje de salida MPC con slew rate caso: 7mo estado valor $1e8$.	79
4.54. Posición de la bola con $N_p=1$.	79
4.55. Posición de la bola con $N_p=5$.	80
4.56. Posición de la bola con $N_p=6$.	80
4.57. Overshoot estacionario en la posición de la bola con $N_p=7$.	81
4.58. Referencia alcanzada en la posición de la bola con $N_p=7$.	81
4.59. Overshoot estacionario en la posición de la bola con $N_p=8$.	82
4.60. Referencia alcanzada en la posición de la bola con $N_p=8$.	82

4.61. Overshoot estacionario en la posición de la bola con $N_p=9$.	83
4.62. Referencia alcanzada en la posición de la bola con $N_p=9$.	83
4.63. Overshoot estacionario en la posición de la bola con $N_p=10$.	84
4.64. Referencia alcanzada en la posición de la bola con $N_p=10$.	84
4.65. Overshoot estacionario en la posición de la bola con $N_p=11$.	85
4.66. Referencia alcanzada en la posición de la bola con $N_p=11$.	85
4.67. Overshoot estacionario en la posición de la bola con $N_p=12$.	86
4.68. Referencia alcanzada en la posición de la bola con $N_p=12$.	86
4.69. Actuación de la planta ± 10 .	87
4.70. Actuación de la planta ± 4 .	87
4.71. Valor ángulo actuación ± 4 .	88
4.72. Actuación de la planta ± 3 .	88
4.73. Actuación de la planta ± 3 .	89
4.74. Valor ángulo actuación ± 3 .	89

Índice de cuadros

3.1. Listado de componentes del sistema Ball & Beam. [17]	25
3.2. Especificaciones físicas y geométricas del sistema Ball & Beam.	25
4.1. Efecto de la condición inicial para $\Omega_{7,7} = 5 \times 10^7$ ($\Gamma = 0,3, N_p = 12$).	90
4.2. Efecto de la condición inicial para $\Omega_{7,7} = 5 \times 10^7$ ($\Gamma = 0,3, N_p = 12,$ $\Delta u_{max} = 5[V]$) con Slew rate.	90
4.3. Efecto de la condición inicial para $\Omega_{7,7} = 0,7 \times 10^8$ ($\Gamma = 0,3, N_p = 12,$ $\Delta u_{max} = 5[V]$) con Slew rate.	90
4.4. Efecto de la condición inicial para $\Omega_{7,7} = 1 \times 10^8$ ($\Gamma = 0,3, N_p = 12,$ $\Delta u_{max} = 5[V]$) con Slew rate.	90
4.5. Análisis de control y observaciones para diferentes valores de $N_p, (\Omega_{7,7} =$ $1,0 \times 10^8, \Gamma = 0,3, T_s = 0,1, \Delta u_{max} = 5[V])$ con Slew rate.	91
4.6. Análisis del desempeño y tiempos de cómputo para variaciones de $\Omega_{7,7}, (\Gamma =$ $0,3, N_p = 12, T_s = 0,1[s], \Delta u_{max} = 5[V])$ con Slew rate.	91
4.7. Efecto del rango de actuación con Slew rate en el control del sistema ($\Omega_{7,7} =$ $1 \times 10^8, \Delta u_{max} = 5[V]$).	92

1 Introducción

El control de sistemas dinámicos complejos ha sido uno de los principales desafíos en la ingeniería de control. En este contexto, el control predictivo por modelo (MPC, por sus siglas en inglés) ha emergido como una técnica avanzada que ha transformado el diseño e implementación de sistemas de control [19], [20].

El MPC se fundamenta en predecir el comportamiento futuro del sistema mediante un modelo matemático y en optimizar las acciones de control para minimizar una función de costo basada en dichas predicciones. Este enfoque es altamente versátil, ya que permite incorporar restricciones tanto en las variables de entrada como en las salidas del sistema, lo que lo convierte en una solución adecuada para sistemas reales en entornos complejos [11].

El presente trabajo tiene por objetivo diseñar, implementar y validar un controlador predictivo por modelo aplicado a un sistema Ball & Beam de Quanser mediante simulación. Para ello, se estudian las propiedades de controlabilidad y observabilidad del modelo linealizado en torno a un punto de operación; se implementa una etapa de preestabilización para asegurar una dinámica conveniente para la utilización de MPC en donde se formula el problema de control en términos de programación cuadrática (QP) y se selecciona un método de optimización adecuado para su resolución en tiempo real (“Quadprog”, método Active-Set). La validación se lleva a cabo en un entorno de simulación mediante Matlab/Simulink.

1.1 Motivaciones

El presente trabajo busca aplicar la técnica de Control Predictivo basado en Modelo a un sistema Ball and Beam (bola y viga), con el objetivo fundamental de controlar la posición de una bola que se desplaza libremente sobre una viga metálica. El sistema utiliza un servomotor de alta precisión para manipular el ángulo de inclinación de la viga, lo cual permite generar una componente de la fuerza de gravedad que actúa sobre la bola para desplazarla hacia una referencia específica y compensar sus inercias inherentes [1], [18]. De esta manera, se busca lograr una regulación robusta y precisa mediante la implementación del algoritmo MPC.

El Control Predictivo basado en Modelo es una estrategia de control avanzado que, a través de la optimización de una función de costo cuadrática, genera una secuencia de acciones de control óptimas en un horizonte de tiempo finito. Una de las principales motivaciones para utilizar esta técnica en el sistema Ball and Beam es su capacidad intrínseca para gestionar restricciones operativas de manera precisa. Entre estas destacan los límites de voltaje del actuador y las restricciones de estado, tales como la limitación del ángulo de inclinación de la viga, con el propósito de proteger la integridad de los engranajes del servomotor y mantener la bola dentro de los límites físicos de la viga.

Dado el creciente interés y estudio sobre el MPC en los últimos años aplicado a diversos sistemas clásicos, surge la motivación de explorar una alternativa de control más potente y flexible que los esquemas convencionales (como el PID o el LQR) [2], [14]. A diferencia de estos, el MPC permite anticipar la dinámica futura del sistema, lo que resulta ideal para manejar la inercia de la bola y garantizar un seguimiento de referencia con un sobreimpulso mínimo y un estricto cumplimiento de las normas de seguridad del hardware.

1.2 Estado del arte

1.2.1 Control Predictivo Basado en Modelo

El Control Predictivo Basado en Modelo (MPC) es una técnica de control avanzado que se apoya en la predicción del comportamiento futuro de un sistema mediante un modelo matemático. A diferencia de los controladores tradicionales, que reaccionan principalmente a perturbaciones o al error actual, MPC anticipa la evolución del sistema y decide la acción de control a partir de dichas predicciones. En su estructura, MPC incorpora un bloque optimizador que cumple el papel del controlador en las arquitecturas convencionales este está definida por una función de costos la cual se busca minimizar mediante un algoritmo de optimización sujeto a las restricciones del problema en un horizonte de tiempo determinado.

Esta formulación convierte a MPC en una técnica especialmente potente y flexible para sistemas con múltiples restricciones y dinámicas complejas, tanto lineales como no lineales. Las

implementaciones para sistemas lineales el problema de MPC se formula habitualmente como una Programación Cuadrática (QP), es decir, con función de costo cuadrática y restricciones lineales, lo que permite resolverlo de manera eficiente con solucionadores numéricos [19]. No obstante, una limitación importante de MPC es su carga computacional, ya que en cada ciclo de control debe resolverse el problema QP, esto puede ser crítico en sistemas de dinámica rápida, a diferencia de esquemas tradicionales cuya carga computacional es reducida.

1.2.2 Algoritmo de optimización Active-Set

El método *Active-Set* es una de las estrategias más empleadas para resolver problemas de Programación Cuadrática (QP) en el contexto del control predictivo. En MATLAB, este enfoque se encuentra implementado en el solver `Quadprog`, que ofrece distintas opciones algorítmicas según la naturaleza del problema.

El principio de funcionamiento del *Active-Set* consiste en mantener un conjunto de restricciones consideradas activas en el punto óptimo y resolver, en cada iteración, un subproblema de igualdad asociado a dicho conjunto. Este conjunto se actualiza dinámicamente, agregando o eliminando restricciones hasta satisfacer las condiciones de optimalidad.

En aplicaciones de *Model Predictive Control* (MPC), este método resulta especialmente apropiado, ya que entre pasos de muestreo las restricciones tienden a modificarse mínimamente. Esto permite reutilizar información de la solución anterior (*warm start*), reduciendo el número de iteraciones necesarias y, por tanto, el tiempo de cálculo. Además, el *Active-Set* destaca por su buen desempeño en problemas de tamaño pequeño y mediano, y por ofrecer una interpretación clara del estado de las restricciones durante la optimización.

A pesar de estas ventajas, su eficiencia puede verse afectada en problemas de gran escala o con cambios frecuentes en el conjunto activo, donde los métodos de punto interior suelen ser más competitivos. Sin embargo, en aplicaciones experimentales de MPC y en sistemas con horizontes moderados, el método *Active-Set* proporciona una relación equilibrada entre velocidad de convergencia, precisión y requerimientos computacionales.

1.3 Objetivos del trabajo

Este documento se centra en la documentación y aplicación de la técnica de control predictivo basado en modelo (MPC) al sistema Ball and Beam mediante simulación usando Matlab/Simulink.

Para lograr esto, en primera instancia se obtendrá el modelo no lineal que describe el comportamiento del sistema Ball & Beam a partir de las ecuaciones diferenciales que representan al sistema de estudio. para luego realizar una linealización con respecto a un punto de equilibrio con el propósito de obtener una representación en variables de estado, al cual se le aplicará un controlador LQR con el fin de estabilizar la planta, este nuevo sistema será el modelo a

1. INTRODUCCIÓN

optimizar mediante el método MPC. Para ello, es necesario definir el método de optimización, las restricciones del sistema y la posición de la pelota.

Una vez definidas las condiciones de operación del sistema MPC, se buscará simular el sistema de control en Matlab/Simulink.

2 Antecedentes Teóricos

En la actualidad, el Control Predictivo por Modelo (MPC) se ha consolidado como una de las estrategias de control avanzado más potentes y versátiles en la industria, debido principalmente a su capacidad intrínseca para gestionar sistemas complejos con múltiples variables y requisitos de desempeño exigentes. A diferencia de los controladores convencionales como el PID o la realimentación de estados clásica, que operan de manera reactiva ante el error, el MPC posee una naturaleza proactiva: utiliza un modelo matemático del proceso para predecir su comportamiento futuro dentro de un horizonte de tiempo determinado. Esta característica permite que el sistema no solo reaccione al estado presente, sino que anticipe eventos y tome acciones preventivas para optimizar un criterio de desempeño específico.

Además, su mayor valor frente a las técnicas tradicionales radica en el manejo explícito de restricciones. En los sistemas dinámicos modernos, es fundamental respetar los límites físicos de los actuadores y los márgenes de seguridad de los estados para evitar daños o inestabilidad, MPC integra estas limitaciones directamente en su algoritmo de optimización en tiempo real, garantizando que la solución encontrada sea siempre factible y eficiente. Esta combinación de visión a futuro y respeto estricto por los límites operativos lo convierte en la herramienta predilecta para maximizar la eficiencia energética y la precisión en entornos industriales.

A continuación se presentan los fundamentos teóricos de MPC, La formulación se apoya en el trabajo realizado por A.Cedeño, A.Morrison y G.Carvajal [3], en conjunto con las notas de referencia [4].

2.1 Control Predictivo por Modelo (MPC)

El Model Predictive Control (MPC) se define como una técnica de control avanzada que utiliza un modelo matemático del sistema y el valor de los estados en un instante dado para determinar las entradas óptimas. El objetivo es que el sistema siga una referencia a lo largo de una secuencia de muestras futuras, considerando restricciones en las entradas, salidas y estados.

De acuerdo con [3], el MPC opera fundamentalmente bajo cuatro componentes principales:

- Un modelo matemático del sistema a controlar.

2. ANTECEDENTES TEÓRICOS

- Un horizonte de control de N muestras.
- Un conjunto de restricciones sobre estados internos, salidas y entradas.
- Un funcional de costo que penaliza las desviaciones de las entradas y los estados respecto a sus valores deseados.

2.1.1 Funcionamiento y Formulación Cuadrática

En cada instante de muestreo, el MPC predice el comportamiento futuro del sistema a lo largo de un horizonte de predicción y encuentra la secuencia de acciones de control que minimizan el funcional de costo. Una vez calculada la secuencia óptima, se aplica únicamente el primer elemento y el proceso se repite en el siguiente intervalo con un horizonte desplazado, estrategia conocida como *receding horizon*.

Cuando el sistema es lineal, la función de costo es cuadrática y las restricciones son lineales, el problema se puede representar como una formulación canónica de Programación Cuadrática (QP):

$$\text{mín } J(\xi) = \frac{1}{2}\xi^T H\xi + h^T \xi \quad (2.1)$$

sujeto a:

$$A\xi = b \quad (2.2)$$

$$C\xi \leq d \quad (2.3)$$

Donde la matriz H debe ser semidefinida positiva para garantizar que el problema sea convexo y permita encontrar un mínimo global.

2.1.2 Definición del Problema de Control

Se considera un sistema lineal discretizado e invariante en el tiempo (LTI) representado por:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{aligned} \quad (2.4)$$

donde $x_k \in \mathbb{R}^n$ corresponde al vector de estados en el instante de muestreo k , $y_k \in \mathbb{R}^p$ es el vector de salidas y $u_k \in \mathbb{R}^m$ el de entradas de control. Las matrices $A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times m}$ y $C \in \mathbb{R}^{p \times n}$ caracterizan la dinámica: A determina la evolución de los estados, B describe la influencia de la entrada en la transición de estados, y C establece la relación entre estados y salidas. En lo que sigue se asume que los estados son medibles en cada instante de muestreo.

2. ANTECEDENTES TEÓRICOS

El objetivo del controlador es llevar la salida hacia una referencia $r_k \in \mathbb{R}^p$. Para ello se define el error:

$$e_k = r_k - y_k. \quad (2.5)$$

Si se considera $r_k = 0$, el error se reduce a $e_k = -y_k$. De esta manera, penalizar las desviaciones de la salida respecto del origen puede hacerse a través de la función de costo:

$$J_k = y_k^T y_k. \quad (2.6)$$

Al sustituir la relación de salida [2.4](#) en [2.6](#) se obtiene:

$$J_k = x_k^T \underbrace{C^T C}_{\Omega} x_k = x_k^T \Omega x_k \quad (2.7)$$

Donde $\Omega \in \mathbb{R}^{n \times n}$ actúa como una matriz de pesos que refleja la relevancia de cada estado en la salida. Además, para contemplar aspectos prácticos como la protección de actuadores, el consumo energético y restricciones físicas, se añade un término adicional que penaliza variaciones excesivas en las entradas de control u_k mediante una matriz Γ .

$$J_k = x_k^T \Omega x_k + u_k^T \Gamma u_k \quad (2.8)$$

Con $\Gamma \in \mathbb{R}^{m \times m}$ se utiliza como matriz de penalización la cual es definida durante el proceso de diseño. Para el caso de sistemas multivariable es común emplear una forma diagonal $\Gamma = qI$, con I la matriz identidad y q un factor escalar de peso.

Dado que el MPC considera un horizonte de predicción finito para el calculo del costo obtenido para sus futuras entradas, el funcional de costos se expresa como la suma de los costos individuales dentro del horizonte de predicción:

$$J(x_k, u_k) = \sum_{k=0}^{N-1} (x_k^T \Omega x_k + u_k^T \Gamma u_k) + x_N^T \Omega_N x_N \quad (2.9)$$

Donde $\Omega_N \in \mathbb{R}^{n \times n}$ representa una matriz de penalización terminal, la cual se introduce para aproximar el comportamiento de un lazo MPC con horizonte infinito mediante un lazo con horizonte finito.

El para obtenerla acción de control óptima es fundamental determinar la secuencia de entradas óptimas:

$$\bar{u}^* = [u_0^{*T} \quad u_1^{*T} \quad \dots \quad u_{N-1}^{*T}]^T \in \mathbb{R}^{m \cdot N} \quad (2.10)$$

2. ANTECEDENTES TEÓRICOS

Con el fin de minimizar el funcional de costo [2.9](#). Suponiendo que se dispone de una medición o estimación del vector de estados actual x_k , el problema se plantea como un problema de programación cuadrática (QP):

$$\begin{aligned} \bar{u}^* &= \arg \min_{\bar{u}} J(x_k, u_k) \\ \text{sujeto a:} \\ x_0 &= x, \\ x_{k+1} &= Ax_k + Bu_k, \\ u^{\min} &\leq u_k \leq u^{\max}, \\ x^{\min} &\leq x_k \leq x^{\max}, \\ x_N^{\min} &\leq x_N \leq x_N^{\max}. \end{aligned} \tag{2.11}$$

donde x_0 corresponde al estado inicial el cual puede ser estimado por un observador y los términos con superíndice min y max especifican los límites para las entradas, los estados internos y el estado terminal respectivamente.

La formulación anterior plantea un problema en lazo abierto, pues la secuencia de entradas calculadas no consideran perturbaciones que pueden ocurrir en el momento de ejecutar las acciones de control, en presencia de dichas perturbaciones, el desempeño disminuye e incluso generar inestabilidad en el sistema. Es por eso que se emplea la estrategia de *Recedinghorizon*, el cual se basa en resolver de manera iterativa un problema de horizonte finito aplicada solo a la primera acción de control calculada (u_0) y replanificando en el siguiente instante de muestreo con la nueva información disponible, lo que introduce una realimentación indirecta cerrando el lazo de control evitando así las diferencias entre los estados medidos y predichos mejorando la robustez y evitando la propagación de errores de predicción.

Los pasos para aplicar MPC en cada tiempo de muestreo se resumen a continuación:

1. En el instante de muestreo k se mide (o estima) el estado de la planta, x_k .
2. Plantear el problema de programación cuadrática (QP).
3. Calcular el nuevo funcional de costo y resolver para determinar la secuencia óptima de entradas que lo minimizan, \bar{u}^* .
4. Aplicar en la entrada el primer elemento de la secuencia de control óptima, u_0^* .
5. Repetir para el siguiente instante de muestreo, $k + 1$.

2.2 Formulación del Problema QP

Existen dos formas principales de plantear matemáticamente el problema de programación cuadrática (QP) según las variables de decisión elegidas: la **Formulación Densa** y la **Formulación Sparse**, cabe destacar que la elección de utilizar una de las formulaciones anteriores afecta directamente a la carga computacional y los tiempo de resolución.

2.2.1 Formulación Densa

Considerando como variable de decisión a la entrada del sistema, se definen los siguientes vectores:

$$\vec{u} = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \quad (2.12)$$

Donde $\vec{u} \in \mathbb{R}^{(m \cdot N)}$ y $\vec{x} \in \mathbb{R}^{(n \cdot N)}$. Con estas definiciones se reescribe el sistema dinámico como:

$$\vec{x} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\mathcal{A}} x_0 + \underbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}}_{\mathcal{O}} \vec{u}, \quad \vec{x} = \mathcal{A}x_0 + \mathcal{O}\vec{u}, \quad (2.13)$$

Donde $\mathcal{A} \in \mathbb{R}^{(n \cdot N) \times n}$ y $\mathcal{O} \in \mathbb{R}^{(n \cdot N) \times (m \cdot N)}$. También se expande el funcional de costo para dejarlo en términos de x_0 y \vec{u} :

$$J(x_0, \vec{u}) = x_0^T \Omega x_0 + \underbrace{\vec{x}^T}_{\vec{x} + \vec{u}^T} \underbrace{\begin{bmatrix} \Omega & 0 & \dots & 0 \\ 0 & \Omega & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \Omega_N \end{bmatrix}}_Q \underbrace{\begin{bmatrix} \Gamma & 0 & \dots & 0 \\ 0 & \Gamma & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & \dots & \Gamma \end{bmatrix}}_R \vec{u} = x_0^T \Omega x_0 + \vec{x}^T Q \vec{x} + \vec{u}^T R \vec{u} \quad (2.14)$$

donde $Q \in \mathbb{R}^{(n \cdot N) \times (n \cdot N)}$ y $R \in \mathbb{R}^{(m \cdot N) \times (m \cdot N)}$. Luego, al sustituir [2.13](#) después expandiendo y reagrupando los términos, resulta:

2. ANTECEDENTES TEÓRICOS

$$\begin{aligned}
 J(x_0, \vec{u}) &= x_0^T \Omega x_0 + (\mathcal{A}x_0 + \mathcal{O}\vec{u})^T Q (\mathcal{A}x_0 + \mathcal{O}\vec{u}) + \vec{u}^T R \vec{u} \\
 &= x_0^T \Omega x_0 + x_0^T \mathcal{A}^T Q \mathcal{A} x_0 + x_0^T \mathcal{A}^T Q \mathcal{O} \vec{u} + \vec{u}^T \mathcal{O}^T Q \mathcal{A} x_0 + \vec{u}^T \mathcal{O}^T Q \mathcal{O} \vec{u} + \vec{u}^T R \vec{u}
 \end{aligned} \tag{2.15}$$

Donde los términos $x_0^T \mathcal{A}^T Q \mathcal{O} \vec{u}$ y $\vec{u}^T \mathcal{O}^T Q \mathcal{A} x_0$ son iguales ya que uno es el transpuesto del otro y cada uno es de dimensiones 1×1 , por lo que se puede transponer el segundo término y sumarlo al primero, resultando:

$$J(x_0, \vec{u}) = x_0^T (\Omega + \mathcal{A}^T Q \mathcal{A}) x_0 + \vec{u}^T (R + \mathcal{O}^T Q \mathcal{O}) \vec{u} + 2x_0^T \mathcal{A}^T Q \mathcal{O} \vec{u} \tag{2.16}$$

Se define $H \in \mathbb{R}^{(m \cdot N) \times (m \cdot N)}$ y $h^T \in \mathbb{R}^{1 \times (m \cdot N)}$ como:

$$\frac{1}{2} H = R + \mathcal{O}^T Q \mathcal{O} \tag{2.17}$$

$$h^T = 2x_0^T \mathcal{A}^T Q \mathcal{O} \tag{2.18}$$

Reemplazando [2.18](#) y [2.17](#) en [2.16](#) se obtiene el funcional de costo:

$$J(x_0, \vec{u}) = \frac{1}{2} \vec{u}^T H \vec{u} + h^T \vec{u} \tag{2.19}$$

El funcional de costo final, omitiendo términos constantes puesto que no afectan al búsqueda de un \vec{u} que minimice el costo es:

$$J(x_0, \vec{u}) = \frac{1}{2} \vec{u}^T H \vec{u} + h^T \vec{u} \tag{2.20}$$

En cuanto a las restricciones estas pueden escribirse de manera compacta en términos de \vec{u} y \vec{x} .

a) Limitaciones en la señal de estado:

$$\begin{aligned}
 \vec{x}^{min} &\leq \vec{x} \leq \vec{x}^{max} \\
 \vec{x}^{min} &\leq (\mathcal{A}x_0 + \mathcal{O}\vec{u}) \leq \vec{x}^{max}
 \end{aligned} \tag{2.21}$$

$$\begin{aligned}
 (\vec{x}^{min} - \mathcal{A}x_0) &\leq \mathcal{O}\vec{u} \leq (\vec{x}^{max} - \mathcal{A}x_0) \\
 \mathcal{O}\vec{u} &\leq (\vec{x}^{max} - \mathcal{A}x_0) \\
 -\mathcal{O}\vec{u} &\leq -(\vec{x}^{min} - \mathcal{A}x_0)
 \end{aligned} \tag{2.22}$$

donde:

$$\vec{x}^{min} = \begin{bmatrix} x^{min} \\ \vdots \\ x^{min} \end{bmatrix} \quad \vec{x}^{max} = \begin{bmatrix} x^{max} \\ \vdots \\ x^{max} \end{bmatrix} \tag{2.23}$$

2. ANTECEDENTES TEÓRICOS

Tal que, \vec{x}^{min} y $\vec{x}^{max} \in \mathbb{R}^{(n \cdot N)}$.

b) **Región final:**

$$\begin{aligned} x_N^{min} &\leq x_N \leq x_N^{max} \\ x_N^{min} &\leq (\mathcal{A}_N x_0 + \mathcal{O}_N \vec{u}) \leq x_N^{max} \\ (x_N^{min} - \mathcal{A}_N x_0) &\leq \mathcal{O}_N \vec{u} \leq (x_N^{max} - \mathcal{A}_N x_0) \end{aligned} \quad (2.24)$$

$$\begin{aligned} \mathcal{O}_N \vec{u} &\leq (x_N^{max} - \mathcal{A}_N x_0) \\ -\mathcal{O}_N \vec{u} &\leq -(x_N^{min} - \mathcal{A}_N x_0) \end{aligned} \quad (2.25)$$

Donde:

$$\mathcal{O}_N = [A^{N-1}B \quad A^{N-2}B \quad \dots \quad AB \quad B], \quad \mathcal{A}_N = A^N \quad (2.26)$$

tal que $\mathcal{O}_N \in \mathbb{R}^{n \times (m \cdot N)}$ y $\mathcal{A}_N \in \mathbb{R}^{n \times n}$.

c) **Limitaciones en la señal de entrada:**

$$\begin{aligned} u^{min} &\leq u_k \leq u^{max} \\ \vec{u}^{min} &\leq \vec{u} \leq \vec{u}^{max} \end{aligned} \quad (2.27)$$

Donde

$$\vec{u}^{min} = \begin{bmatrix} u^{min} \\ \vdots \\ u^{min} \end{bmatrix} \quad \vec{u}^{max} = \begin{bmatrix} u^{max} \\ \vdots \\ u^{max} \end{bmatrix} \quad (2.28)$$

Tal que \vec{u}^{min} y $\vec{u}^{max} \in \mathbb{R}^{m \cdot N}$.

y en forma matricial, todas las restricciones se pueden escribir como:

$$\underbrace{\begin{bmatrix} \mathcal{O}_N \\ -\mathcal{O}_N \\ \mathcal{O} \\ -\mathcal{O} \end{bmatrix}}_M \vec{u} \leq \underbrace{\begin{bmatrix} \vec{x}_N^{max} - \mathcal{A}_N x_0 \\ -\vec{x}_N^{min} + \mathcal{A}_N x_0 \\ \vec{x}^{max} - \mathcal{A} x_0 \\ -\vec{x}^{min} + \mathcal{A} x_0 \end{bmatrix}}_c, \quad \underbrace{\begin{bmatrix} u^{min} \\ \vdots \\ u^{min} \end{bmatrix}}_a \leq \vec{u} \leq \underbrace{\begin{bmatrix} u^{max} \\ \vdots \\ u^{max} \end{bmatrix}}_b \quad (2.29)$$

Donde $M \in \mathbb{R}^{(2 \cdot n + 2 \cdot n \cdot N) \times (m \cdot N)}$, $c \in \mathbb{R}^{(2 \cdot n + 2 \cdot n \cdot N)}$, $a \in \mathbb{R}^{(m \cdot N)}$ y $b \in \mathbb{R}^{(m \cdot N)}$.

Nótese que las restricciones en (2.8) se transforman en restricciones lineales en \vec{u} , con lo cual se define el problema de optimización QP como:

$$\begin{aligned}
 \vec{u}^* &= \min_{\vec{u}} J(x_0, \vec{u}) \\
 &\text{sujeto a} \\
 &M\vec{u} \leq c \\
 &a \leq \vec{u} \leq b
 \end{aligned} \tag{2.30}$$

2.2.2 Seguimiento de Referencia

Cuando se desea que la salida del sistema siga una referencia distinta de cero ($r \neq 0$), es necesario garantizar que en régimen estacionario se cumpla:

$$y_\infty \rightarrow r,$$

donde y_∞ representa el valor de la salida en equilibrio. Para encontrar los valores de estados y entradas que permiten alcanzar esta condición, se considera el sistema en estado estacionario:

$$\begin{aligned}
 x_\infty &= Ax_\infty + Bu_\infty, \\
 y_\infty &= Cx_\infty.
 \end{aligned} \tag{2.31}$$

Este conjunto de ecuaciones puede organizarse en una forma matricial:

$$\underbrace{\begin{bmatrix} I - A & -B \\ C & 0 \end{bmatrix}}_L \begin{bmatrix} x_\infty \\ u_\infty \end{bmatrix} = \begin{bmatrix} 0 \\ r \end{bmatrix}, \tag{2.32}$$

cuyo sistema se resuelve como:

$$\begin{bmatrix} x_\infty \\ u_\infty \end{bmatrix} = L^{-1} \begin{bmatrix} 0 \\ r \end{bmatrix}, \tag{2.33}$$

donde $L \in \mathbb{R}^{(n+p) \times (n+m)}$. De este modo, (x_∞, u_∞) corresponden al punto de equilibrio asociado a la referencia r .

Para simplificar la formulación del problema se introducen las variables de desviación:

$$\tilde{x} = x - x_\infty, \quad \tilde{u} = u - u_\infty \tag{2.34}$$

de manera que \tilde{x} y \tilde{u} cuantifican la diferencia respecto al equilibrio estacionario. Estas variables cumplen el mismo modelo dinámico que las originales, por lo que es posible reescribir el funcional de costo en términos de ellas.

$$J_N(\tilde{x}_0, \tilde{u}) = \frac{1}{2} \tilde{u}^T H \tilde{u} + \tilde{h}^T \tilde{u} \tag{2.35}$$

2. ANTECEDENTES TEÓRICOS

donde:

$$\begin{aligned}\frac{1}{2}H &= R + \mathcal{O}^T Q \mathcal{O} \\ \tilde{h}^T &= 2\tilde{x}_0^T \mathcal{A}^T Q \mathcal{O}\end{aligned}\quad (2.36)$$

Para encontrar las restricciones de esta nueva función de costo hay que aplicar el cambio de variables [2.34](#) a las restricciones originales [2.29](#). Se llega a:

a) **Limitaciones en las entradas:**

$$\begin{aligned}\bar{u}^{\min} &\leq \vec{u} \leq \bar{u}^{\max} \\ \bar{u}^{\min} &\leq \vec{u} + \vec{u}_\infty \leq \bar{u}^{\max} \\ \bar{u}^{\min} - \vec{u}_\infty &\leq \vec{u} \leq \bar{u}^{\max} - \vec{u}_\infty\end{aligned}\quad (2.37)$$

b) **Limitaciones en los estados:**

$$\begin{aligned}\bar{x}^{\min} &\leq \vec{x} \leq \bar{x}^{\max} \\ \bar{x}^{\min} &\leq \vec{x} + \vec{x}_\infty \leq \bar{x}^{\max} \\ \bar{x}^{\min} - \vec{x}_\infty &\leq \vec{x} \leq \bar{x}^{\max} - \vec{x}_\infty \\ \bar{x}^{\min} - \vec{x}_\infty &\leq \mathcal{A}\tilde{x}_0 + \mathcal{O}\vec{u} \leq \bar{x}^{\max} - \vec{x}_\infty \\ \bar{x}^{\min} - \vec{x}_\infty - \mathcal{A}\tilde{x}_0 &\leq \mathcal{O}\vec{u} \leq \bar{x}^{\max} - \vec{x}_\infty - \mathcal{A}\tilde{x}_0 \\ \mathcal{O}\vec{u} &\leq \bar{x}^{\max} - \vec{x}_\infty - \mathcal{A}\tilde{x}_0 \\ -\mathcal{O}\vec{u} &\leq -(\bar{x}^{\min} - \vec{x}_\infty - \mathcal{A}\tilde{x}_0)\end{aligned}\quad (2.38)$$

$$\begin{aligned}\mathcal{O}\vec{u} &\leq \bar{x}^{\max} - \vec{x}_\infty - \mathcal{A}\tilde{x}_0 \\ -\mathcal{O}\vec{u} &\leq -(\bar{x}^{\min} - \vec{x}_\infty - \mathcal{A}\tilde{x}_0)\end{aligned}\quad (2.39)$$

La extensión a la región final es análoga, y finalmente se pueden representar las restricciones en estado estacionario:

$$\underbrace{\begin{bmatrix} \mathcal{O}_N \\ -\mathcal{O}_N \\ \mathcal{O} \\ -\mathcal{O} \end{bmatrix}}_{\tilde{M}} \vec{u} \leq \underbrace{\begin{bmatrix} \bar{x}_N^{\max} - x_\infty - \mathcal{A}_N \tilde{x}_0 \\ -\bar{x}_N^{\min} + x_\infty + \mathcal{A}_N \tilde{x}_0 \\ \bar{x}^{\max} - x_\infty - \mathcal{A}\tilde{x}_0 \\ -\bar{x}^{\min} + x_\infty + \mathcal{A}\tilde{x}_0 \end{bmatrix}}_{\tilde{c}}, \quad \underbrace{\begin{bmatrix} u^{\min} - u_\infty \\ \vdots \\ u^{\min} - u_\infty \end{bmatrix}}_{\tilde{a}} \leq \vec{u} \leq \underbrace{\begin{bmatrix} u^{\max} - u_\infty \\ \vdots \\ u^{\max} - u_\infty \end{bmatrix}}_{\tilde{b}} \quad (2.40)$$

donde $\tilde{M} \in \mathbb{R}^{(2 \cdot n + 2 \cdot n \cdot N) \times (m \cdot N)}$, $\tilde{c} \in \mathbb{R}^{(2 \cdot n + 2 \cdot n \cdot N)}$, $\tilde{a} \in \mathbb{R}^{(m \cdot N)}$ y $\tilde{b} \in \mathbb{R}^{(m \cdot N)}$.

2.2.3 Restricción de Desigualdad

La restricción de caja en [2.40](#) se pasa a una de desigualdad para que todas queden en una ecuación. Primero se separa en dos ecuaciones de desigualdad:

$$\begin{aligned} \tilde{a} &\leq \vec{u} \leq \tilde{b} \\ \tilde{a} &\leq \vec{u}, \quad \vec{u} \leq \tilde{b} \\ -\vec{u} &\leq -\tilde{a}, \quad \vec{u} \leq \tilde{b} \end{aligned} \tag{37}$$

Luego se juntan con las restricciones de desigualdad de [2.40](#) de la siguiente manera:

$$\begin{bmatrix} I \\ -I \end{bmatrix} \vec{u} \leq \begin{bmatrix} \tilde{b} \\ -\tilde{a} \end{bmatrix} \implies \hat{M} = \begin{bmatrix} \tilde{M} \\ I \\ -I \end{bmatrix}, \quad \hat{c} = \begin{bmatrix} \tilde{c} \\ \tilde{b} \\ -\tilde{a} \end{bmatrix} \tag{2.41}$$

De esta manera \vec{u} queda restringido de la siguiente manera:

$$\hat{M}\vec{u} \leq \hat{c} \tag{2.42}$$

Donde $\hat{M} \in \mathbb{R}^{(2 \cdot n + 2 \cdot n \cdot N + 2 \cdot m \cdot N) \times (m \cdot N)}$ y $\hat{c} \in \mathbb{R}^{(2 \cdot n + 2 \cdot n \cdot N + 2 \cdot m \cdot N)}$.

2.2.4 Planteamiento final de problema QP

Finalmente, el problema QP queda planteado, de forma densa, con seguimiento de referencia r distinta a 0 y con todas las restricciones en forma de desigualdad, como encontrar \vec{u}^* tal que minimice la función de costo $J_N(\tilde{x}_0, \vec{u})$:

$$J_N(\tilde{x}_0, \vec{u}) = \frac{1}{2} \vec{u}^T H \vec{u} + \tilde{h}^T \vec{u} \tag{2.43}$$

Donde:

$$\begin{aligned} \frac{1}{2} H &= R + \mathcal{O}^T Q \mathcal{O} \\ \tilde{h}^T &= 2\tilde{x}_0^T \mathcal{A}^T Q \mathcal{O} \end{aligned} \tag{2.44}$$

Sujeto a:

$$\hat{M}\vec{u} \leq \hat{c}$$

2.2.5 Resultado Final

Una vez encontrado el vector \vec{u}^* es necesario revertir el cambio de variables [2.34](#) que se hizo para seguir la referencia y así obtener las entradas que son apropiadas para aplicar en la planta. Esto se hace de la siguiente manera:

$$\vec{u}^* = \vec{u}^* + \vec{u}_\infty \quad (2.45)$$

El resultado final de MPC es el vector \vec{u}^* que contiene las entradas para N instantes de tiempo que son necesarias para llevar la salida de la planta a la referencia deseada. Solo se aplica la primera entrada u_0^* a la planta y en el siguiente instante de muestreo se repite el proceso.

2.3 Método de optimización Active Set Quadprog

Dentro del esquema de Control Predictivo Basado en Modelo (MPC), la determinación de las acciones de control óptimas requiere la resolución de problemas de optimización numérica. Estos se formulan usualmente bajo la estructura de programación cuadrática, donde se busca minimizar un índice de desempeño de segundo orden condicionado por restricciones lineales de igualdad y desigualdad. El algoritmo Active-Set, integrado en la herramienta quadprog de MATLAB, destaca por su eficiencia en este tipo de aplicaciones.

Matemáticamente, el problema de optimización cuadrática se expresa como:

$$\min_x \Phi(x) = \frac{1}{2}x^T Qx + c^T x \quad (2.46)$$

Sujeto a las siguientes condiciones de contorno:

$$Ax = b \quad (2.47)$$

$$Dx \leq e \quad (2.48)$$

Donde:

- $x \in \mathbb{R}^n$ representa el vector de variables de decisión.
- $Q \in \mathbb{R}^{n \times n}$ es la matriz Hessiana, la cual se asume como simétrica y definida semipositiva para garantizar la convexidad del problema y la existencia de un mínimo global único.
- $c \in \mathbb{R}^n$ es el vector de gradiente que define la componente lineal del costo.

2. ANTECEDENTES TEÓRICOS

- Las ecuaciones (2.47) y (2.48) representan el conjunto de restricciones de igualdad y desigualdad, respectivamente.

El método *Active-Set* basa su estrategia en la gestión dinámica de un conjunto de trabajo (W_k), el cual identifica en cada iteración aquellas restricciones de desigualdad que se comportan como igualdades estrictas en el punto de prueba actual. Mediante esta selección, el algoritmo redefine el espacio de búsqueda en cada subproblema, limitándose a explorar la frontera definida por las restricciones activas hasta satisfacer las condiciones de optimalidad.

2.3.1 Procedimiento Operativo del Algoritmo

El desarrollo del método *Active-Set* se articula a través de las siguientes etapas iterativas:

El proceso comienza con la obtención de un punto inicial x_0 que sea factible, es decir, que cumpla con las condiciones (2.47) y (2.48). En esta etapa se define el primer *Working Set* (W_0), el cual contiene los índices de las restricciones de desigualdad que se cumplen con igualdad estricta en dicho punto, tal que:

$$W_0 = \{i \mid D_i x_0 = e_i\} \quad (2.49)$$

Este conjunto actúa como la base de búsqueda para la primera iteración.

Al momento de la resolución del subproblema cuadrático, en cada iteración k , el algoritmo plantea un subproblema de optimización con el fin de hallar una dirección de búsqueda d . Este subproblema se define como:

$$\min_d \frac{1}{2} d^T Q d + c^T d \quad \text{sujeto a} \quad A_{eq} d = 0 \quad (2.50)$$

Donde A_{eq} representa la unión de las restricciones de igualdad originales y las restricciones de desigualdad que actualmente pertenecen al conjunto activo W_k . El objetivo es encontrar el movimiento óptimo d permaneciendo sobre la frontera de las restricciones activas.

Una vez hallado d_k , el algoritmo evalúa qué tanto puede avanzar en esa dirección antes de violar una restricción que no estaba en el conjunto activo. El tamaño de paso (α_k) se calcula mediante:

$$\alpha_k = \min \left(1, \min_{i \notin W_k, D_i d_k > 0} \frac{f_i - D_i x_k}{D_i d_k} \right) \quad (2.51)$$

Si $\alpha_k = 1$, el punto óptimo del subproblema es factible y el algoritmo se desplaza completamente hacia él. Si $\alpha_k < 1$, el movimiento se detiene prematuramente debido al encuentro con una nueva restricción, la cual se incorpora al conjunto activo para la iteración posterior.

2. ANTECEDENTES TEÓRICOS

Cuando la dirección de búsqueda es nula ($d_k = 0$), se comprueba si el punto actual es el óptimo global validando las condiciones de Karush-Kuhn-Tucker (KKT):

- **Factibilidad primal:** El punto cumple con todas las restricciones del sistema.
- **Estacionariedad:** El gradiente de la función objetivo es compensado por las fuerzas de las restricciones activas, $\nabla f(x^*) = 0$.
- **Factibilidad dual:** Se verifica que los multiplicadores de Lagrange sean no negativos, $\lambda_i \geq 0$.

Este paso representa el núcleo de decisión del algoritmo. Si en el paso anterior se detecta un $\lambda_i < 0$, se interpreta que la restricción i impide la disminución del valor de la función objetivo. Por consiguiente, dicha restricción se elimina del conjunto activo W_k para permitir que el algoritmo explore el interior de la región factible en la siguiente iteración.

Finalmente, se calcula la nueva posición del sistema mediante la relación:

$$x_{k+1} = x_k + \alpha_k d_k \quad (2.52)$$

Este ciclo de adición y eliminación de restricciones en el conjunto activo continúa de forma recursiva hasta satisfacer simultáneamente todas las condiciones de optimalidad descritas con anterioridad.

2.4 Muestreo en sistemas de control

En el panorama actual de la ingeniería, la teoría del muestreo se ha consolidado como el pilar fundamental que sustenta la convergencia entre el entorno físico analógico y la capacidad de procesamiento de los sistemas digitales [16]. Esto no solo constituye un requisito técnico para la adquisición de datos, sino que representa la base matemática que garantiza la integridad de la información al transitar entre dominios temporales. A diferencia de los sistemas de control analógicos tradicionales, limitados por la rigidez de sus componentes de hardware, la susceptibilidad al ruido térmico y la degradación de señales en el tiempo, la implementación de sistemas digitales basados en un muestreo riguroso permite una flexibilidad sin precedentes a través de la reconfiguración mediante software. Hoy en día, esta transición posibilita la ejecución de algoritmos de control digital.

Por esta razón es importante la capacidad de discretizar el sistema eligiendo un tiempo de muestreo óptimo con el propósito de garantizar una reconstrucción fiel de la señal analógica y evitar el fenómeno de aliasing, asegurando que el controlador digital pueda operar sobre el

2. ANTECEDENTES TEÓRICOS

sistema físico.

Esto se lleva a cabo mediante dos etapas fundamentales que actúan como interfaz entre ambos dominios:

- **Convertidor Análogo-Digital (A/D):** Es el dispositivo encargado de transformar una señal física continua en el tiempo en una secuencia de números que el procesador digital puede interpretar.
- **Convertidor Digital-Análogo (D/A):** Realiza la operación inversa, transformando las secuencias numéricas resultantes del algoritmo de control en una señal física continua.

Para aplicar las señales reconstruidas al sistema físico, el método más utilizado es el *Retenedor de Orden Cero* (ZOH, por sus siglas en inglés), el cual mantiene el valor de la señal constante durante los instantes de muestreo, generando una señal en forma de escalera que se aproxima a la señal deseada.

El ZOH se puede representar como un función de transferencia de la forma:

$$G_{zoh}(s) = \frac{1 - e^{-sT_s}}{s} \quad (2.53)$$

Donde T_s corresponde al período de muestreo. En la practica el ZOH es esencial en la implementación de control digital, puesto que permite convertir señales discretas a continuas para interactuar directamente con el sistema físico.

2.4.1 Representación en espacios de estado discreto

En el diseño de sistemas de control modernos, especialmente aquellos basados en optimización como el Control Predictivo por Modelo (MPC) o LQR, la representación en espacio de estados es la herramienta fundamental para modelar la dinámica de sistemas multivariables. Esta metodología permite describir el comportamiento de un sistema físico y su posterior transición al dominio digital, donde se ejecutan las acciones de control.

Un sistema dinámico lineal e invariante en el tiempo (LTI) se describe mediante un conjunto de ecuaciones diferenciales de primer orden que relacionan el estado del sistema, sus entradas y sus salidas:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (2.54)$$

Donde:

$x(t) \in \mathbb{R}^n$: Es el vector de estados, que representa la información mínima necesaria para

2. ANTECEDENTES TEÓRICOS

determinar el comportamiento futuro del sistema.

$u(t) \in \mathbb{R}^m$: Es el vector de entradas o señales de control aplicadas.

A, B, C, D : Son las matrices que definen la dinámica interna, la influencia de la entrada y la relación con las variables medidas.

Para que un controlador digital pueda interactuar con una planta analógica, se requiere un proceso de reconstrucción de señales mediante un convertidor Digital-Analógico. El modelo matemático más utilizado para este fin es el Retenedor de Orden Cero (ZOH). El funcionamiento del ZOH consiste en tomar el valor de una muestra discreta $u[k]$ en el instante kT_s y mantenerlo constante durante todo el periodo de muestreo T_s . Matemáticamente, se define como:

$$u(t) = u[k], \quad \text{para } t \in [kT_s, (k+1)T_s) \quad (2.55)$$

Físicamente, esto genera una señal de control con forma de escalera, asegurando que los actuadores reciban una señal continua y estable entre cada cálculo del algoritmo.

De esta manera es posible discretizar [2.54](#). Al asumir que la entrada $u(t)$ es constante en el intervalo $[kT_s, (k+1)T_s)$, la solución de la ecuación diferencial [2.54](#) conduce al modelo en tiempo discreto:

$$\begin{aligned} x[k+1] &= \Phi x[k] + \Gamma u[k] \\ y[k] &= Cx[k] + Du[k] \end{aligned} \quad (2.56)$$

Las matrices discretas Φ (matriz de transición de estados) y Γ (matriz de entrada) se calculan mediante:

$$\Phi = e^{AT_s}, \quad \Gamma = \left(\int_0^{T_s} e^{A\tau} d\tau \right) B \quad (2.57)$$

La transición del modelo continuo al discreto se basa en la solución de la ecuación diferencial de estado en el intervalo de muestreo $[kT_s, (k+1)T_s)$.

La solución general de la ecuación $\dot{x}(t) = Ax(t) + Bu(t)$ para cualquier tiempo t está dada por:

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^t e^{A(t-\tau)}Bu(\tau)d\tau \quad (2.58)$$

Al evaluar esta solución en el intervalo discreto, con $t_0 = kT_s$ y $t = (k+1)T_s$, obtenemos:

$$x((k+1)T_s) = e^{AT_s}x(kT_s) + \int_{kT_s}^{(k+1)T_s} e^{A((k+1)T_s-\tau)}Bu(\tau)d\tau \quad (2.59)$$

Bajo la hipótesis del Retenedor de Orden Cero (ZOH), la señal de control $u(\tau)$ se mantiene constante e igual a $u[k]$ durante todo el intervalo de integración. Esto permite extraer $u[k]$ de

2. ANTECEDENTES TEÓRICOS

la integral, resultando en:

$$x[k+1] = \underbrace{e^{AT_s}}_{\Phi} x[k] + \underbrace{\left(\int_0^{T_s} e^{A\lambda} d\lambda \right)}_{\Gamma} B u[k] \quad (2.60)$$

Donde se realiza un cambio de variable $\lambda = (k+1)T_s - \tau$ para simplificar los límites de la integral.

Destacar que cuando la matriz del sistema A es no singular, la integral de la exponencial de la matriz que define a Γ se puede resolver directamente como:

$$\int_0^{T_s} e^{A\tau} d\tau = A^{-1}(e^{AT_s} - I) \quad (2.60)$$

Sustituyendo esto en la definición de Γ , y recordando que $\Phi = e^{AT_s}$, la expresión simplificada queda:

$$\Gamma = A^{-1}(\Phi - I)B \quad (2.61)$$

Donde I es la matriz identidad de dimensiones adecuadas ($n \times n$).

2.5 Estimadores de estado u observadores

Dentro del marco del control automático contemporáneo, la medición directa de la totalidad de las variables de estado que componen la dinámica de un proceso suele verse limitada por diversos factores prácticos. Ya sea por el elevado costo económico de la instrumentación requerida, restricciones físicas en el montaje de sensores o la degradación de las señales debido al ruido de medición, es frecuente que el vector de estados sea, en algún grado, inaccesible. Ante este escenario, surge la necesidad de implementar esquemas algorítmicos que, procesando únicamente las entradas conocidas y las salidas disponibles, logren inferir los estados internos omitidos; a estos componentes se les denomina técnicamente observadores de estado [6], [15]. La integración de estos estimadores resulta indispensable en estrategias de control avanzado, tales como el MPC, el cual depende estrictamente del conocimiento preciso del estado actual x_k para proyectar la evolución del sistema a lo largo de un horizonte futuro de predicción. Más allá de suplir la carencia de sensores, un observador bien diseñado robustece el lazo de control al mitigar los efectos de perturbaciones externas y corregir posibles discrepancias entre el modelo matemático utilizado y el comportamiento físico real de la planta.

El presente apartado aborda las bases teóricas de la observación de estados. Este desarrollo permite no solo la reconstrucción de variables cinemáticas complejas de medir directamente,

sino que también establece el punto de partida fidedigno que el optimizador del MPC requiere para garantizar un desempeño eficiente, estable y capaz de compensar las perturbaciones del entorno.

2.5.1 Estructura general de los observadores de estados

Un observador de estados es un sistema que, utilizando un modelo matemático del proceso, genera una estimación $\hat{x}(t)$ del vector de estado real $x(t)$. Su función es actuar como un "sensor virtual", reconstruyendo variables internas a partir de las entradas aplicadas $u(t)$ y las salidas medidas $y(t)$.

Estructura de Luenberger es arquitectura más común, que consiste en una réplica de la dinámica de la planta a la que se le añade un lazo de corrección basado en el error de salida. Considérese el sistema dinámico original:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t)\end{aligned}\tag{2.62}$$

El observador se define mediante la siguiente ecuación diferencial:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t))\tag{2.63}$$

Donde $\hat{y}(t) = C\hat{x}(t)$ es la salida estimada. El término $(y(t) - \hat{y}(t))$ representa la innovación, y la matriz L es la ganancia del observador, encargada de corregir el modelo interno basándose en el error.

El diseño de un observador consiste en determinar la matriz de ganancia L de modo que el error de estimación converja a cero rápidamente. Para ello definimos el error de estimación como $e(t) = x(t) - \hat{x}(t)$. Al derivar esta expresión y sustituir las ecuaciones del sistema y del observador, obtenemos la dinámica del error:

$$\dot{e}(t) = (A - LC)e(t)\tag{2.64}$$

Esta ecuación muestra que el comportamiento del error depende enteramente de los autovalores de la matriz $(A - LC)$ el cual converge a cero si todos los autovalores poseen parte real negativa.

Si bien la ubicación de polos permite definir arbitrariamente la velocidad del observador, en entornos con incertidumbre se prefiere un enfoque de Estimación Lineal Cuadrática (LQE). En este caso, la ganancia L no se elige a mano, sino que se calcula para minimizar la varianza del error de estimación[16]. Este enfoque es la base del Filtro de Kalman en sistemas discre-

tos, donde se busca un equilibrio óptimo entre confiar en el modelo matemático matriz Φ o confiar en las mediciones ruidosas de los sensores matriz C .

2.5.2 Observador extendido para la estimación de perturbaciones

El Observador de Estado Extendido es una evolución del observador convencional diseñada específicamente para enfrentar uno de los mayores retos en control automático la incertidumbre o perturbaciones. Su función principal es tratar las perturbaciones externas y las dinámicas no modeladas como si fueran un estado adicional del sistema para poder estimarlas en tiempo real [4].

Se considera un sistema en tiempo discreto afectado por una perturbación aditiva d_t , la cual se asume con una dinámica constante o de variación lenta $d_{t+1} = d_t$. El modelo se describe de la siguiente forma:

$$\begin{aligned}x_{t+1} &= Ax_t + Bu_t + B_d d_t \\d_{t+1} &= d_t \\y_t &= Cx_t + C_d d_t\end{aligned}\tag{2.65}$$

Para capturar este efecto, se define el estado extendido z_t como la combinación del vector de estados original y la perturbación:

$$z_t = \begin{bmatrix} x_t \\ d_t \end{bmatrix}\tag{2.66}$$

Con la nueva definición del vector z_t , el sistema puede reescribirse mediante matrices aumentadas \bar{A} , \bar{B} y \bar{C} que integran la dinámica de la perturbación:

$$\begin{aligned}z_{t+1} &= \underbrace{\begin{bmatrix} A & B_d \\ 0 & I \end{bmatrix}}_{\bar{A}} z_t + \underbrace{\begin{bmatrix} B \\ 0 \end{bmatrix}}_{\bar{B}} u_t \\y_t &= \underbrace{\begin{bmatrix} C & C_d \end{bmatrix}}_{\bar{C}} z_t\end{aligned}\tag{2.67}$$

A partir de este modelo extendido, se diseña un observador de orden completo en la forma clásica:

$$\hat{z}_{t+1} = \bar{A}\hat{z}_t + \bar{B}u_t + K(y_t - \bar{C}\hat{z}_t)\tag{2.68}$$

Donde $\hat{z}_t = [\hat{x}_t^\top \hat{d}_t^\top]^\top$ representa la estimación conjunta del estado y la perturbación, y L es la ganancia del observador extendido. El diseño de la matriz de ganancia K es crítico para el desempeño del estimador. Se debe asegurar que los autovalores de la matriz de lazo cerrado del observador, $(\bar{A} - K\bar{C})$, se encuentren dentro del círculo unitario en el plano complejo

2. ANTECEDENTES TEÓRICOS

Z. Esto garantiza la estabilidad asintótica del error de estimación, permitiendo que tanto el estado de la planta como la magnitud de la perturbación sean reconstruidos fielmente para su uso en el controlador MPC.

3 Planta de interés

El presente capítulo se centra en la descripción de la planta utilizada en el estudio:

3.1 Sistema Ball & Beam

La planta de estudio corresponde a un sistema Ball & Beam la cual se puede observar en la Figura 3.1 y sus componentes en la Tabla 3.1. El sistema está compuesto por una viga horizontal sobre la cual se desliza una bola metálica que rueda libremente. La viga se encuentra acoplada a un servomotor rotatorio, encargado de generar un ángulo variable de inclinación. Al modificar dicho ángulo, la fuerza de la gravedad provoca el movimiento de la bola hacia uno de los extremos de la viga.

Adicionalmente, el sistema dispone de un sensor de posición de tipo potenciómetro que permite medir la ubicación de la bola a lo largo de la viga. La señal de salida del sensor se traduce en un voltaje de medición $v_{range}(t)$, el cual se utiliza para determinar la posición de la bola en tiempo real.

El control del sistema se realiza mediante la aplicación de un voltaje de entrada al servomotor, lo que permite regular el ángulo de inclinación de la viga y, en consecuencia, la trayectoria de la bola. Considerando como variable de entrada el voltaje aplicado al motor y como variable de salida el voltaje entregado por el sensor de posición, el sistema puede modelarse como un sistema de entrada única y salida única (Single Input Single Output, SISO).

3. PLANTA DE INTERÉS

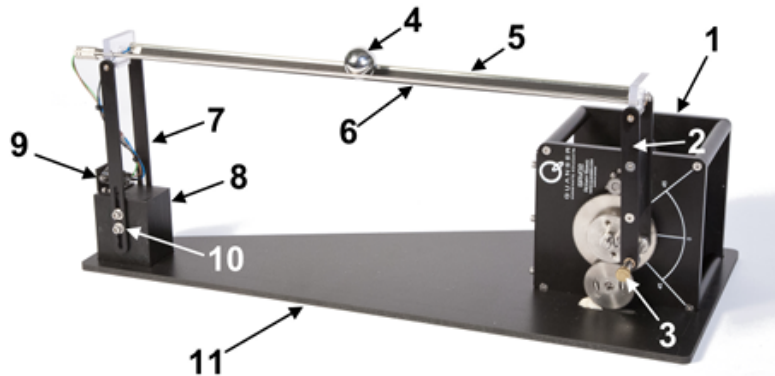


Fig. 3.1: Componentes del sistema Ball & Beam. [17]

Tab. 3.1: Listado de componentes del sistema Ball & Beam. [17]

ID	Componente	ID	Componente
1	Servomotor rotatorio	7	Brazo de soporte
2	Brazo de palanca	8	Base de soporte
3	Tornillo de acoplamiento	9	Conector del sensor analógico de posición de la bola
4	Bola de acero	10	Tornillos del brazo de soporte
5	Sensor potenciómetro Ball & Beam	11	Base de calibración
6	Varilla de acero Ball & Beam		

Tab. 3.2: Especificaciones físicas y geométricas del sistema Ball & Beam.

Símbolo	Descripción	Valor
m_b	Masa del módulo bola–viga	0,65 kg
L_c	Longitud base de calibración	50 cm
d_c	Profundidad base de calibración	22,5 cm
L	Longitud de la viga	42,55 cm
l_a	Longitud del brazo de palanca	12,5 cm
r_{arm}	Distancia entre eje del servo y unión	2,54 cm
r_{servo}	Longitud del eje de salida del servo	1,27 cm
r_b	Radio de la bola	1,27 cm
m_b	Masa de la bola	0,064 kg
K_{bss}	Sensibilidad del sensor de posición	-4,25 cm/V
V_{bias}	Voltaje de polarización del sensor	± 12 V
V_{range}	Rango de medición del sensor	± 5 V
θ_{range}	Rango Ángulo del servomotor	$\pm 56^\circ$

3.2 Modelo matemático del sistema Ball & Beam.

Para obtener el modelo completo del sistema Ball and Beam se tiene que tener que analizar y realizar el modelo matemático de los siguientes dos subsistemas teniendo en cuenta las restricciones descritas en [3.2.1](#).

- **Planta electromecánica (Servomotor):** Convierte la señal eléctrica de entrada en un desplazamiento angular del eje, el cual se transmite a la viga mediante un brazo mecánico.
- **Planta mecánica (Bola-Viga):** Describe el movimiento de la bola sobre la viga inclinada, considerando las fuerzas gravitacionales, las condiciones de rodadura y las restricciones geométricas del mecanismo.

3.2.1 Convenciones y supuestos generales

Antes de proceder con la derivación de las ecuaciones dinámicas del sistema *Ball and Beam*, es necesario establecer un conjunto de convenciones que permitan definir claramente las variables, parámetros y supuestos utilizados en el modelamiento.

Se considera el sistema Bola-Viga accionado por un servomotor con ángulo de eje de carga $\theta_\ell(t)$. La viga se inclina un ángulo $a(t)$ respecto de la horizontal y la bola de masa m_b y radio r_b rueda sin deslizamiento sobre la viga. Se adoptan las siguientes convenciones y supuestos:

- **Entrada del modelo de la Planta electromecánica:** Voltaje de entrada al servomotor $V_m(t)$
- **Salida del modelo Planta electromecánica:** Ángulo del servomotor en el eje de carga, $\theta_{out}(t)=\theta_\ell(t)$.
- **Entrada del modelo de la Planta mecánica:** Ángulo del servomotor en el eje de carga, $\theta_\ell(t)$.
- **Salida del modelo Planta mecánica:** Posición de la bola sobre la viga, $x(t)$, medida desde el punto medio.
- **Parámetros geométricos:** Longitud de la viga L ; radio del brazo del servomotor r_{arm} .
- **Parámetros de la bola:** Masa m_b , radio r_b , momento de inercia J_b .
- **Supuestos:** Rodadura sin deslizamiento; fricción despreciable salvo la estática que garantiza rodadura; ángulos pequeños para linealización posterior.

3.2.2 Subsistema Servomotor.

El servomotor constituye el actuador principal del sistema Ball and Beam. Su función es transformar una señal eléctrica de entrada en un movimiento mecánico de salida, específicamente el ángulo de inclinación de la viga.

El subsistema eléctrico del servomotor está compuesto por un bobinado que, al recibir un voltaje de entrada $V_m(t)$, genera una corriente $i(t)$. Esta corriente produce un campo electromagnético que se traduce en un par motor (torque) como se puede observar en la Figura 3.2.

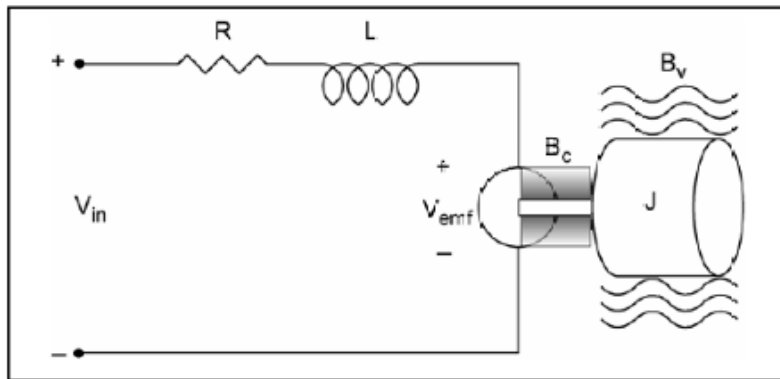


Fig. 3.2: Esquema servomotor DC. [5]

3.2.3 Supuestos y definiciones del servomotor

Se considera un servomotor DC con armadura excitada, acoplado a un tren de engranajes de relación N que mueve el eje de salida. Se usan los siguientes supuestos y definiciones:

- **Eléctrico:** Armadura con resistencia R e inductancia L . FEM proporcional a la velocidad: $e(t) = K_e \omega_m(t)$.
- **Mecánico:** Rotor con inercia J_m y fricción viscosa B_m . Torque electromagnético: $\tau_m(t) = K_t i(t)$.
- **Transmisión:** Reducción N (motor rápido \rightarrow salida lenta). Salida: $\omega_{out}(t) = \omega_m(t)/N$, $\theta_{out}(t) = \int \omega_{out}(t) dt$.
- **Carga:** Inercia J_{out} y fricción B_{out} en el eje de salida, con torque externo $\tau_{out}(t)$ (por la planta mecánica).

3.2.4 Ecuaciones físicas del servomotor DC

El comportamiento eléctrico del servomotor DC se describe a partir de la ley de Kirchhoff aplicada al circuito de armadura. En este circuito se consideran la resistencia y la inductancia de la bobina, así como la fuerza contra electromotriz inducida por la rotación del rotor. Esta última aparece debido a la ley de Faraday, que establece que una variación de flujo magnético genera una tensión proporcional a la velocidad angular del motor.

Analizando el circuito eléctrico de armadura del motor y aplicando la ley de mallas:

$$V_m(t) = L \dot{i}(t) + R i(t) + e(t), \quad (3.1)$$

Donde la fuerza contra electromotriz la cual es la tensión que se opone a la voltaje de entrada aplicado al girar el rotor, el cual tiene la siguiente forma:

$$e(t) = K_e \omega_m(t). \quad (3.2)$$

El torque electromecánico surge de la interacción entre el campo magnético del estator y la corriente que circula por la armadura. La relación entre la corriente $i(t)$ y el torque es lineal y está determinada por la constante de torque del motor K_t . Este principio proviene directamente de la ley de Lorentz, que describe la fuerza ejercida sobre un conductor cuando circula corriente en presencia de un campo magnético, el cual tiene la siguiente ecuación:

$$\tau_m(t) = K_t i(t). \quad (3.3)$$

Por otra parte la dinámica mecánica del rotor se obtiene aplicando la segunda ley de Newton para la rotación. El torque neto que actúa sobre el eje del motor se reparte entre la aceleración angular del rotor que es proporcional a su inercia, y la fricción viscosa que se opone al movimiento. Además, se incluye el efecto del torque de carga reflejado al eje del motor, que representa la resistencia mecánica externa conectada a la salida.

Donde la ecuación de movimiento con fricción viscosa corresponde a:

$$J_m \dot{\omega}_m(t) + B_m \omega_m(t) = \tau_m(t) - \tau_{load,m}(t), \quad (3.4)$$

Donde $\tau_{load,m}(t)$ es el torque de carga reflejado al eje del motor.

3.2.5 Transmisión y reflexión de la carga

La transmisión mecánica que conecta el rotor del motor con el eje de salida introduce una relación cinemática y estática que debe ser considerada en el modelamiento. Mediante el tren de engranajes, la velocidad angular del rotor se reduce y el torque se amplifica en el eje de salida. Para expresar toda la dinámica en el eje del motor, se realiza la reflexión de los parámetros de la carga: la inercia y la fricción presentes en el eje de salida se trasladan al

eje del motor multiplicadas por el cuadrado de la relación de reducción. De esta manera, se obtiene un modelo equivalente en el cual la transmisión y la carga externa quedan absorbidas en parámetros efectivos de inercia y fricción, simplificando el análisis y permitiendo trabajar con una única ecuación mecánica equivalente.

Relación cinemática y estática del tren de engranajes

El tren de engranajes establece la relación cinemática y estática entre el rotor del motor y el eje de salida. Cinemáticamente, la velocidad angular del rotor se reduce en la proporción de la relación de transmisión N , mientras que estáticamente el torque se amplifica en la misma proporción.

$$\omega_{\text{out}}(t) = \frac{\omega_m(t)}{N}, \quad \theta_{\text{out}}(t) = \int \omega_{\text{out}}(t) dt = \frac{1}{N} \int \omega_m(t) dt. \quad (3.5)$$

Donde si:

- $N > 1$: El motor gira más rápido que el eje de salida.
- $N < 1$: El motor gira más lento que el eje de salida.

3.2.6 Reflexión de inercia y fricción

La inercia y la fricción presentes en el eje de salida se reflejan al eje del motor multiplicadas por el cuadrado de la relación de transmisión. De este modo, se definen parámetros equivalentes J_{eq} y B_{eq} que agrupan tanto las características propias del rotor como las de la carga externa.

La inercia y fricción de la salida reflejadas al eje del motor son:

$$J_{eq} = J_m + N^2 J_{\text{out}}, \quad B_{eq} = B_m + N^2 B_{\text{out}}. \quad (3.6)$$

Reflexión del torque de carga

El torque externo aplicado en el eje de salida también se refleja al eje del motor. En este caso, la relación es inversa: el torque de carga se reduce en la proporción de la transmisión, expresándose como $\tau_{\text{load},m} = \tau_{\text{out}}/N$. Esta transformación asegura que el efecto de la carga externa quede correctamente representado en el modelo equivalente del motor.

El torque externo en la salida se refleja como:

$$\tau_{\text{load},m}(t) = \frac{\tau_{\text{out}}(t)}{N}. \quad (3.7)$$

Sustituyendo (3.6) y (3.7) en (3.4) se forma la ecuación mecánica equivalente:

$$J_{eq} \dot{\omega}_m(t) + B_{eq} \omega_m(t) = K_t i(t) - \frac{\tau_{out}(t)}{N}. \quad (3.8)$$

3.2.7 Modelo en el dominio de Laplace

Aplicando transformada de Laplace a (3.1), (3.2) y (3.8) (condiciones iniciales nulas):

$$V_m(s) = L s I(s) + R I(s) + K_e \omega_m(s). \quad (3.9)$$

$$J_{eq} s \omega_m(s) + B_{eq} \omega_m(s) = K_t I(s) - \frac{\tau_{out}(s)}{N}. \quad (3.10)$$

Despejando $I(s)$ de (3.9):

$$I(s) = \frac{V_m(s) - K_e \omega_m(s)}{L s + R}. \quad (3.11)$$

Sustituyendo (3.11) en (3.10):

$$(J_{eq} s + B_{eq}) \omega_m(s) = K_t \frac{V_m(s) - K_e \omega_m(s)}{L s + R} - \frac{\tau_{out}(s)}{N}. \quad (3.12)$$

Multiplicando por $(L s + R)$ y reordenando:

$$\left[(J_{eq} s + B_{eq})(L s + R) + K_t K_e \right] \omega_m(s) = K_t V_m(s) - \frac{(L s + R)}{N} \tau_{out}(s). \quad (3.13)$$

Obteniendo la función de transferencia en relación a la velocidad y el voltaje, suponiendo la carga despreciable en identificación ($\tau_{out}(s) \approx 0$):

$$\frac{\omega_m(s)}{V_m(s)} = \frac{K_t}{(J_{eq} s + B_{eq})(L s + R) + K_t K_e}. \quad (3.14)$$

Por otra parte calculando la función de transferencia en relación a la posición y el voltaje de entrada, usando (3.5) y $\Theta_{out}(s) = \frac{1}{N s} \omega_m(s)$:

$$\frac{\Theta_{out}(s)}{V_m(s)} = \frac{1}{N} \cdot \frac{K_t}{s \left[(J_{eq} s + B_{eq})(L s + R) + K_t K_e \right]}. \quad (3.15)$$

3.2.8 Modelo simplificado del servomotor

Para aplicaciones de laboratorio el modelo anterior se puede simplificar a un sistema de primer orden, teniendo en cuenta las siguientes simplificaciones y supuestos:

3. PLANTA DE INTERÉS

- **Inductancia despreciable:** $L \approx 0$ (armadura eléctrica rápida).
- **Carga ligera en identificación:** $\tau_{\text{out}}(s) \approx 0$.
- **Fricción efectiva viscosa:** B_{eq} lumpada (incluye transmisión).
- **Relación de reducción conocida:** N fijo.

Despreciando la inductancia L es decir $L \approx 0$, la (3.14) se reduce a:

$$\frac{\omega_m(s)}{V(s)} = \frac{K_t}{R J_{eq} s + (R B_{eq} + K_t K_e)}. \quad (3.16)$$

La posición de salida queda:

$$\frac{\Theta_{\text{out}}(s)}{V(s)} = \frac{1}{N} \cdot \frac{K_t}{s(R J_{eq} s + (R B_{eq} + K_t K_e))}. \quad (3.17)$$

De esta forma agrupando los efectos eléctricos y mecánicos en una constante de tiempo τ_m y una ganancia K_m de la forma:

$$K_m \triangleq \frac{K_t}{N (R B_{eq} + K_t K_e)}, \quad \tau_m \triangleq \frac{R J_{eq}}{R B_{eq} + K_t K_e}, \quad (3.18)$$

se obtiene la forma:

$$\frac{\Theta_{\text{out}}(s)}{V_m(s)} = \frac{K_m}{s(\tau_m s + 1)}. \quad (3.19)$$

De esta forma, el servomotor se aproxima mediante una función de transferencia de primer orden con integrador, el cual se utilizará para obtener la función de transferencia total del sistema Ball & Beam.

3.2.9 Modelamiento de la planta Bola-Viga.

La segunda planta del sistema corresponde al movimiento de la bola sobre la viga inclinada. Este subsistema describe cómo el ángulo de la viga $a(t)$, generado por el servomotor, afecta la posición de la bola $x(t)$.

3.2.10 Diagrama de cuerpo libre.

El diagrama de cuerpo libre de la bola sobre la viga se muestra en la Figura 3.3. Según este diagrama, la dinámica se obtiene aplicando la primera ley de Newton en la dirección tangencial a la viga. Se consideran las siguientes fuerzas:

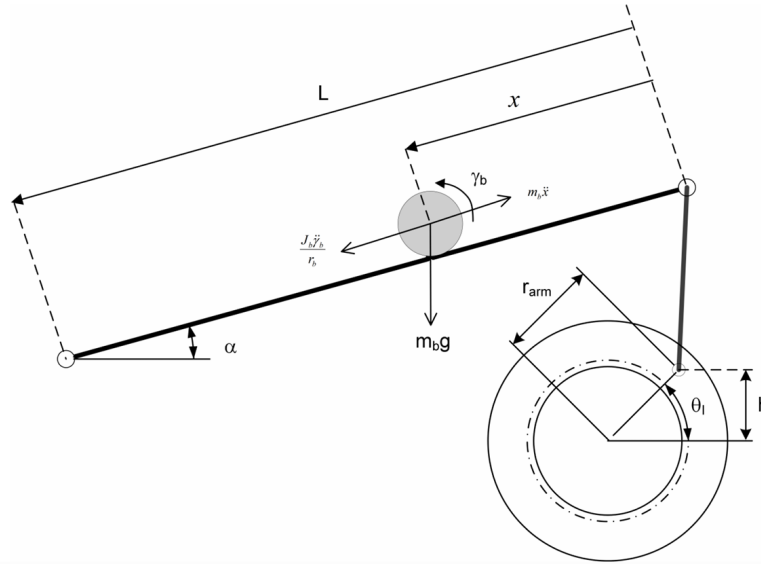


Fig. 3.3: Diagrama de cuerpo libre bola y viga. [1]

- **Componente tangencial del peso:** $F_g(t) = m_b g \sin a(t)$.
- **Fuerza de inercia rotacional:** Generada por la fricción estática y responsable de la rodadura sin deslizamiento, modelada como $F_r(t) = \frac{J_b}{r_b^2} \ddot{x}(t)$.
- **Normal:** equilibra la componente perpendicular del peso, no afecta la dinámica en x .

3.2.11 Relación geométrica entre el ángulo del servomotor y el ángulo de la viga.

Para relacionar el ángulo del servomotor con el ángulo de inclinación de la viga, se considera la geometría del sistema mostrada en la Figura 3.3. Cuando el ángulo del servomotor $\theta_\ell(t)$ es positivo, el extremo derecho de la viga se eleva, generando una inclinación positiva $\alpha(t)$ respecto de la horizontal.

Esta inclinación provoca que la bola se desplace alejándose del eje del servomotor, aumentando su posición $x(t)$ sobre la viga. Por lo tanto, se establece la siguiente convención física:

$$\dot{x}(t) > 0 \quad \Rightarrow \quad \theta_\ell(t) > 0 \quad \Rightarrow \quad \ddot{x}(t) > 0$$

La posición de la bola se define como $x(t) = 0$ cuando está en el eje del servomotor, y el ángulo del servomotor se define como $\theta_\ell(t) = 0$ cuando la viga está paralela al suelo, es decir, $\alpha(t) = 0$.

Desde la geometría del mecanismo, se obtiene la relación entre el ángulo de la viga y el ángulo del servomotor:

$$\sin a(t) = \frac{r_{arm}}{L} \sin \theta_\ell(t) \quad (3.20)$$

3. PLANTA DE INTERÉS

Donde r_{arm} es la longitud del brazo acoplado al servomotor con respecto al eje y L es la longitud efectiva de la viga, revisar valores en Tabla [3.2](#).

Para ángulos pequeños, se aplica la aproximación:

$$a(t) \approx \frac{r_{arm}}{L} \theta_\ell(t). \quad (3.21)$$

3.2.12 Ecuaciones de movimiento no lineales.

Aplicando la segunda ley de Newton en la dirección x , la dinámica de la bola es:

$$m_b \ddot{x}(t) = \sum F = F_r(t) + F_g(t) \quad (3.22)$$

$$\left(m_b + \frac{J_b}{r_b^2} \right) \ddot{x}(t) = m_b g \sin a(t) \quad (3.23)$$

Reordenando:

$$\ddot{x}(t) = \frac{m_b}{m_b + \frac{J_b}{r_b^2}} g \sin a(t) \quad (3.24)$$

Para una esfera sólida, el momento de inercia es:

$$J_b = \frac{2}{5} m_b r_b^2 \quad (3.25)$$

Sustituyendo [\(3.25\)](#) en [\(3.24\)](#):

$$\ddot{x}(t) = \frac{5}{7} g \sin a(t) \quad (3.26)$$

La relación geométrica entre el ángulo de la viga y el del servomotor es:

$$\sin a(t) = \frac{r_{arm}}{L} \sin \theta_\ell(t) \quad (3.27)$$

Sustituyendo [\(3.27\)](#) en [\(3.26\)](#), se obtiene:

$$\ddot{x}(t) = \frac{5}{7} g \frac{r_{arm}}{L} \sin \theta_\ell(t) \quad (3.28)$$

3.2.13 Linealización del subsistema Bola-viga.

Para obtener el modelo lineal que describa el subsistema Bola-Viga aplicamos una aproximación de Taylor de primer orden alrededor de $\theta_\ell(t) = 0$.

3.2.14 Aproximación de Taylor de primer orden.

La ecuación de movimiento no lineal del subsistema Bola-Viga incluye el término

$$\ddot{x}(t) = \frac{5}{7} g \frac{r_{arm}}{L} \sin \theta_\ell(t) \quad (3.29)$$

Para linealizar esta expresión alrededor del punto de equilibrio $\theta_\ell^* = 0$, se utiliza la expansión en serie de Taylor de la función $f(\theta_\ell) = \sin(\theta_\ell)$:

$$f(\theta_\ell) \approx f(\theta_\ell^*) + f'(\theta_\ell^*)(\theta_\ell - \theta_\ell^*) + \frac{f''(\theta_\ell^*)}{2!}(\theta_\ell - \theta_\ell^*)^2 + \dots \quad (3.30)$$

Evaluando en $\theta_\ell^* = 0$:

$$f(0) = \sin(0) = 0, \quad f'(0) = \cos(0) = 1, \quad f''(0) = -\sin(0) = 0$$

Por lo tanto, la aproximación de primer orden es:

$$\sin(\theta_\ell) \approx \theta_\ell \quad (3.31)$$

Sustituyendo esta aproximación en (3.29), se obtiene el modelo linealizado, valido para ángulos pequeños:

$$\ddot{x}(t) \approx \frac{5}{7} g \frac{r_{arm}}{L} \theta_\ell(t) \quad (3.32)$$

Reordenando los parámetros del coeficiente en la ganancia del modelo K_{bb} :

$$\ddot{x}(t) = K_{bb} \theta_\ell(t), \quad K_{bb} = \frac{5}{7} g \frac{r_{arm}}{L} \quad (3.33)$$

Remplazando los valores de las constantes en K_{bb} usando los valores de la Tabla 3.2 y con $g = 9,81 [m/s^2]$, se obtiene:

$$K_{bb} = 0,418 [m/s^2/rad] \quad (3.34)$$

3.2.15 Obtención de la función de transferencia del modelo lineal del sistema Ball & Beam.

A partir de las ecuaciones dinámicas de los subsistemas que conforman la planta ya linealizadas, es posible obtener las ecuaciones de transferencia total del sistema Ball & Beam.

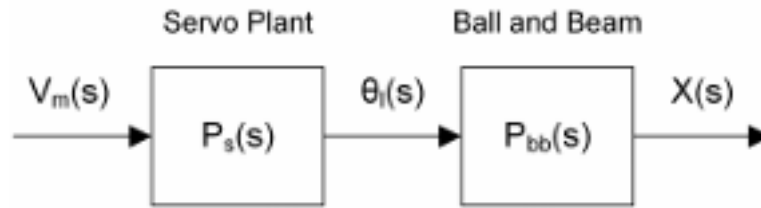


Fig. 3.4: Diagrama de bloque a lazo abierto.

De la figura 3.4 se observa que la función de transferencia completa es :

$$P(s) = P_{bb}(s) P_s(s) \quad (3.35)$$

Donde $P_s(s)$, la función de transferencia del servomotor la cual está dada por :

$$P_s = \frac{\Theta_\ell(s)}{V_m(s)} = \frac{K_m}{s(\tau_m s + 1)} \quad (3.36)$$

y la función de transferencia de la posición y el ángulo de la viga del subsistema Bola-Viga viene dada por aplicar la transformada de Laplace a la Ecuación 3.33 quedando de la siguiente forma:

$$P_{bb} = \frac{X(s)}{\theta_\ell(t)} = \frac{K_{bb}}{s^2} \quad (3.37)$$

Teniendo en cuenta estas dos funciones de transferencia, se puede obtener la función de transferencia del sistema completo definido en 3.35 obteniendo:

$$P(s) = \frac{X(s)}{V_m(s)} = \frac{K_{bb} K_m}{s^3 (\tau_m s + 1)} \quad (3.38)$$

Extrayendo los datos de las constantes para el subsistema del servomotor se tiene que $K_m=1,5$ [rad/s/V], $\tau_m=0,025$ [s] y usando el dato obtenido en la ecuación 3.34 obteniendo:

$$P(s) = \frac{X(s)}{V_m(s)} = \frac{1,23}{s^3 (0,025s + 1)} \quad (3.39)$$

3.3 Obtención del modelo en espacios de estado

En esta sección se desarrolla la representación del sistema en términos de variables de estado a partir de la función de transferencia. La formulación en espacio de estados permite describir la evolución interna del sistema mediante un conjunto de variables, lo que facilita el análisis de su comportamiento temporal, el diseño de estrategias de control avanzadas y la construc-

ción de observadores. Primero se presenta el modelo continuo y, posteriormente, se deriva su versión discreta para su implementación digital.

3.3.1 Modelo en espacio de estados en tiempo continuo

Con el objetivo de representar el sistema Ball & Beam mediante un espacio de estado en tiempo continuo usando la función de transferencia definida en [3.39](#), la cual es corresponde a un sistema lineal invariante en el tiempo (LTI) de 4to orden.

Se define el vector de estados físicos como

$$x(t) = \begin{bmatrix} \theta(t) \\ \dot{\theta}(t) \\ r(t) \\ \dot{r}(t) \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} \quad u(t) = V_m(t) \quad (3.40)$$

Como se observó anteriormente el servomotor se aproxima por un sistema primer orden en velocidad con constante de tiempo τ_m y ganancia K_m como se puede observar en la función de transferencia [3.19](#).

Reordenando los términos para obtener las ecuaciones físicas correspondientes al servomotor:

$$\Theta_{\text{out}}(s) s (\tau_m s + 1) = K_m V_m(s) \quad (3.41)$$

Aplicando la transformada inversa de Laplace a la ecuación [3.41](#), y usando las propiedades:

$$\mathcal{L}^{-1}\{s \Theta_{\text{out}}(s)\} = \dot{\theta}(t), \quad \mathcal{L}^{-1}\{s^2 \Theta_{\text{out}}(s)\} = \ddot{\theta}(t)$$

Se obtiene la ecuación diferencial en el dominio del tiempo:

$$\tau_m \ddot{\theta}_{\text{out}}(t) + \dot{\theta}_{\text{out}}(t) = K_m V_m(t) \quad (3.42)$$

Definiendo $\omega(t) = \dot{\theta}(t)$ equivalente a $\dot{X}_1 = X_2$ en función de los estados, se puede reescribir el sistema como:

$$\dot{\omega}(t) = \frac{-\omega(t)}{\tau_m} + \frac{K_m V_m(t)}{\tau_m} \quad \dot{X}_2 = \frac{-X_2}{\tau_m} + \frac{K_m V_m(t)}{\tau_m} \quad (3.43)$$

Para el subsistema Bola-Viga se tiene la ecuación [3.33](#), la cual se evaluá en términos de estados:

$$\dot{X}_3(t) = X_4(t), \quad \dot{X}_4(t) = K_{bb} X_1(t) \quad (3.44)$$

3.3.2 Modelo completo en espacio de estados en tiempo continuo

Con el propósito de obtener las matrices A , B , C , D del modelo de espacio de estados, se agrupan las ecuaciones en función de los estados.

$$\dot{X}_1 = X_2 \quad (3.45)$$

$$\dot{X}_2 = -\frac{X_2}{\tau_m} + \frac{K_m V_m(t)}{\tau_m} \quad (3.46)$$

$$\dot{X}_3(t) = X_4(t) \quad (3.47)$$

$$\dot{X}_4(t) = K_{bb} X_1(t) \quad (3.48)$$

Obteniendo el siguiente modelo:

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{1}{\tau_m} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ K_{bb} & 0 & 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ \frac{K_m}{\tau_m} \\ 0 \\ 0 \end{bmatrix} u(t)$$

$$y(t) = [0 \ 0 \ 1 \ 0] x(t)$$

Por lo tanto, las matrices resultas de manera numérica ($\tau_m = 0,025$, $K_{bb} = 0,418$, $K_m = 1,5$) del sistema continuo son:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -40 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0,418 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 60 \\ 0 \\ 0 \end{bmatrix} \quad C = [0 \ 60 \ 0 \ 0] \quad D = 0 \quad (3.49)$$

De esta forma se obtiene el modelo que describe la dinámica del sistema usando variables de estado, donde $x(t) \in \mathbb{R}^4$ el cual representa el vector de estados, $u(t)$ es la señal de entrada, y $y(t)$ es la salida del sistema, la cual corresponde a la posición de la bola sobre la viga.

3.3.3 Modelo en espacio de estados en tiempo discreto

Para obtener el modelo en espacio de estados en tiempo discreto, se parte de la representación continua, dado que el sistema Ball and Beam es relativamente lento se define un período de muestreo $T_s=0.1$ para realizar la discretización.

El sistema discreto se expresa como:

3. PLANTA DE INTERÉS

$$\begin{aligned} x[k+1] &= A_d x[k] + B_d u[k] \\ y[k] &= C_d x[k] + D_d u[k] \end{aligned} \quad (3.50)$$

Donde las matrices A_d y B_d se calculan a partir de las matrices continuas.

$$A_d = e^{AT_s} = L^{-1} \{ (sI - A)^{-1} \} (t), \quad B_d = \int_0^{T_s} e^{A\tau} B d\tau \quad (3.51)$$

Mientras que $C_d = C$ y $D_d = D$. Esta transformación garantiza que el modelo discreto preserve la dinámica del sistema original en los instantes de muestreo.

Para el calculo de A_d lo primero que se tiene que realizar es el calculo de $(sI - A)^{-1}$

$$(sI - A)^{-1} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s(s+40)} & 0 & 0 \\ 0 & \frac{1}{s+40} & 0 & 0 \\ \frac{209}{500s^3} & \frac{209}{500s^3(s+40)} & \frac{1}{s} & \frac{1}{s^2} \\ \frac{209}{500s^2} & \frac{209}{500s^2(s+40)} & 0 & \frac{1}{s} \end{bmatrix} \quad (3.52)$$

Realizando la transformada de Laplace inversa de la matriz anterior de manera simbólica usando Matlab se obtiene.

$$L^{-1} \{ (sI - A)^{-1} \} = \begin{bmatrix} 1 & \frac{1}{40} - \frac{e^{-40t}}{40} & 0 & 0 \\ 0 & e^{-40t} & 0 & 0 \\ \frac{209t^2}{1000} & \frac{209t^2}{40000} - \frac{209e^{-40t}}{32000000} - \frac{209t}{800000} + \frac{209}{32000000} & 1 & t \\ \frac{209t}{500} & \frac{209t}{20000} + \frac{209e^{-40t}}{800000} - \frac{209}{800000} & 0 & 1 \end{bmatrix} \quad (3.53)$$

Remplazando t por $T_s = 0, 1[s]$ en [3.53](#) se obtiene la siguiente matriz A_d .

$$A_d = \begin{bmatrix} 1,0000 & 0,0245 & 0 & 0 \\ 0 & 0,0183 & 0 & 0 \\ 0,0021 & 0,0000 & 1,0000 & 0,1000 \\ 0,0418 & 0,0008 & 0 & 1,0000 \end{bmatrix} \quad (3.54)$$

Para obtener la matriz B_d se resuelve de manera simbólica utilizando Matlab la ecuación [3.51](#), para finalmente también evaluar los T por $T_s = 0, 1[s]$ en la ecuación [3.55](#).

$$B_d = \begin{bmatrix} \frac{3T}{2} + \frac{3e^{-40T}}{80} - \frac{3}{80} \\ \frac{3}{2} - \frac{3e^{-40T}}{2} \\ \frac{627T}{1600000} + \frac{627e^{-40T}}{64000000} - \frac{627T^2}{80000} + \frac{209T^3}{2000} - \frac{627}{64000000} \\ \frac{627T^2}{2000} - \frac{627e^{-40T}}{1600000} - \frac{627T}{40000} + \frac{627}{1600000} \end{bmatrix} \quad (3.55)$$

$$B_d = \begin{bmatrix} 0,1132 \\ 1,4725 \\ 0,0001 \\ 0,0020 \end{bmatrix} \quad (3.56)$$

Las matrices finales obtenidas del espacio de estados en tiempo discreto A_d , B_d , C_d y D_d son iguales a las obtenidas utilizando el comando `c2d` de Matlab [12], revisar anexo(PreCompile Ball And Beam)

$$A_d = \begin{bmatrix} 1,0000 & 0,0245 & 0 & 0 \\ 0 & 0,0183 & 0 & 0 \\ 0,0021 & 0,0000 & 1,0000 & 0,1000 \\ 0,0418 & 0,0008 & 0 & 1,0000 \end{bmatrix} \quad B_d = \begin{bmatrix} 0,1132 \\ 1,4725 \\ 0,0001 \\ 0,0020 \end{bmatrix} \quad C_d = [0 \quad 60 \quad 0 \quad 0] \quad D_d = 0 \quad (3.57)$$

3.3.4 Análisis de Observabilidad y Controlabilidad

El análisis de las propiedades estructurales de un sistema dinámico en espacio de estados se centra en dos aspectos fundamentales: la controlabilidad y la observabilidad. Los cuales evalúan la capacidad del modelo para ser controlado y monitoreado en su totalidad, lo cual resulta indispensable en el diseño de estrategias de control por realimentación de estados y la construcción de observadores que permita estimar los valores de las variables no medibles de manera física del sistema.

3.3.5 Controlabilidad

Un sistema discreto descrito por las ecuaciones

$$\begin{aligned} x[k+1] &= A_d x[k] + B_d u[k] \\ y[k] &= C_d x[k] + D_d u[k] \end{aligned} \quad (3.58)$$

Se dice que es completamente controlable si existe una secuencia de entradas $u[k]$ capaz de transferir el estado desde cualquier condición inicial $x[0]$ hacia cualquier estado final x_f en un número finito de pasos.

La condición algebraica se verifica mediante la matriz de controlabilidad:

$$C = [B_d \quad A_d B_d \quad A_d^2 B_d \quad \dots \quad A_d^{n-1} B_d] \quad (3.59)$$

Donde n es la dimensión del vector de estados. El sistema es controlable si:

$$\text{Rank}(\mathcal{C}) = n \quad (3.60)$$

Lo que nos indica que se puede diseñar y aplicar una ley de control por realimentación de estados lo que nos ayuda a ubicar los polos del sistema en lazo cerrado de manera arbitraria dentro del círculo unitario, la cual tiene la siguiente forma:

$$u[k] = -Lx[k] \quad (3.61)$$

3.3.6 Observabilidad

De manera dual, un sistema es completamente observable si, a partir de las salidas $y[k]$ y las entradas $u[k]$, es posible reconstruir el estado inicial $x[0]$ en un número finito de pasos.

La condición algebraica se establece mediante la matriz de observabilidad:

$$\mathcal{O} = \begin{bmatrix} C_d \\ C_d A_d \\ C_d A_d^2 \\ \vdots \\ C_d A_d^{n-1} \end{bmatrix} \quad (3.62)$$

Y el sistema es observable si se cumple que:

$$\text{Rank}(\mathcal{O}) = n \quad (3.63)$$

Esta condición garantiza que todos los estados del sistema pueden ser inferidos a partir de las salidas medidas, lo cual es esencial para el diseño de observadores como el de Luenberger o el filtro de Kalman.

Para realizar, verificar el análisis tanto Controlabilidad y Observabilidad correspondiente al sistema de estudio usando los comandos de Matlab $ctrd(A_d, B_d)$, $obsv(A_d, C_d)$ obteniendo los siguientes resultados:

$$\mathcal{C} = \begin{bmatrix} 0,1132 & 0,1493 & 0,1500 & 0,1500 \\ 1,4725 & 0,0270 & 0,0005 & 0,0000 \\ 0,0001 & 0,0005 & 0,0016 & 0,0034 \\ 0,0020 & 0,0078 & 0,0141 & 0,0204 \end{bmatrix} \quad (3.64)$$

$$\mathcal{O} = \begin{bmatrix} 0 & 0 & 1,0000 & 0 \\ 0,0021 & 0,0000 & 1,0000 & 0,1000 \\ 0,0084 & 0,0002 & 1,0000 & 0,2000 \\ 0,0188 & 0,0004 & 1,0000 & 0,3000 \end{bmatrix} \quad (3.65)$$

3. PLANTA DE INTERÉS

De las matrices se observan que ambas tienen rango 4. Es decir tienen rango completo por lo que el sistema es controlable y observable, lo cual nos permite realizar una pre-estabilización de la planta si ha de ser necesaria. Revisar anexo (Controlabilidad y Observabilidad Sistema Discreto)

3.3.7 Pre-estabilización de la planta

Una vez verificada la controlabilidad y observabilidad del sistema *Ball and Beam* en su representación discreta, se procede a la verificación de estabilidad y al diseño de una ley de control óptima de realimentación de estados. El objetivo es garantizar la estabilidad interna del sistema en lazo cerrado, reubicando los polos dentro del círculo unitario del plano z , y establecer un régimen operativo seguro para la posterior implementación de control predictivo.

Para verificar la estabilidad del sistema se hace un análisis de los polos del sistema, siendo inestable si alguno de los polos están sobre el borde o afuera del círculo unitario. Los polos corresponden a los autovalores de la matriz A_d .

$$\det(\lambda I - A_d) = 0 \quad (3.66)$$

Al remplazar la matriz A_d [3.54](#) en la ecuación [3.66](#) se obtienen los siguientes autovalores:

$$\lambda_1 = 1,0000 \quad \lambda_2 = 1,0000 \quad \lambda_3 = 1,0000 \quad \lambda_4 = 0,0183 \quad (3.67)$$

Puesto que el sistema tiene autovalores con multiplicidad 3 sobre el círculo unitario, se concluye que el sistema es marginalmente estable por lo que se requiere ser pre-estabilizado mediante realimentación de estados observados, la cual tiene la siguiente estructura:

$$u_k = -L\hat{x}_k + r_k \quad (3.68)$$

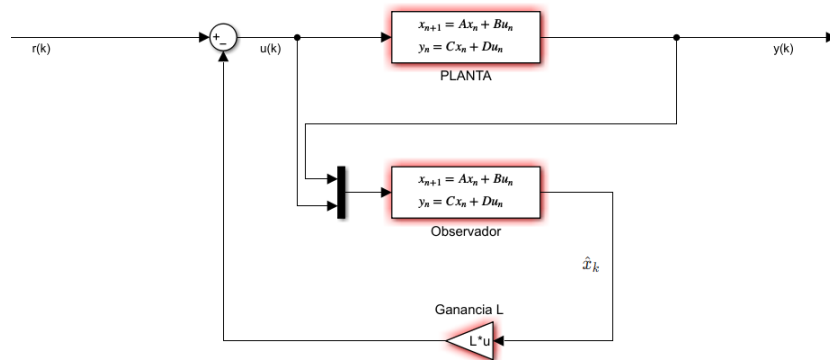


Fig. 3.5: Diagrama de la ley de control con prealimentación de estados.

3. PLANTA DE INTERÉS

Para diseñar la ganancia \mathbf{L} , bajo un enfoque de Control Óptimo Discreto (DLQR), se toma en cuenta solo la parte $u_k = -L\hat{x}_k$ del lazo de control definido en 3.68 y se busca minimizar la función cuadrática de costo para un horizonte infinito:

$$J(\mathbf{u}) = \sum_{k=1}^{\infty} (\mathbf{x}[k]^T \mathbf{Q} \mathbf{x}[k] + \mathbf{u}[k]^T \mathbf{R} \mathbf{u}[k]) \quad (3.69)$$

Donde \mathbf{Q} , \mathbf{R} son las matrices de peso de estado, control respectivamente.

La solución óptima para la ganancia estacionaria \mathbf{L} de la ecuación 3.68, se obtiene mediante la solución de la ecuación Algebraica de Riccati Discreta (DARE) definida como \mathbf{P} .

La matriz de ganancia óptima \mathbf{L} y \mathbf{P} se calcula como:

$$\mathbf{L} = (\mathbf{B}_d^T \mathbf{P} \mathbf{B}_d + \mathbf{R})^{-1} (\mathbf{B}_d^T \mathbf{P} \mathbf{A}_d) \quad (3.70)$$

Donde \mathbf{A}_d y \mathbf{B}_d son las matrices de estado y entrada del sistema discretizado.

$$\mathbf{P} = \mathbf{A}_d^T \mathbf{P} \mathbf{A}_d - (\mathbf{A}_d^T \mathbf{P} \mathbf{B}_d)(\mathbf{R} + \mathbf{B}_d^T \mathbf{P} \mathbf{B}_d)^{-1} (\mathbf{B}_d^T \mathbf{P} \mathbf{A}_d) + \mathbf{Q} \quad (3.71)$$

Al sustituir la ley de control por realimentación de estados 3.68 en el modelo de espacio de estados discretizado 3.50, se obtiene la nueva dinámica del sistema en lazo cerrado.

$$\mathbf{x}[k+1] = \mathbf{A}_d \mathbf{x}[k] + \mathbf{B}_d (-\mathbf{L} \mathbf{x}[k]) \quad (3.72)$$

La nueva representación de estados en lazo cerrado es:

$$\mathbf{x}[k+1] = (\mathbf{A}_d - \mathbf{B}_d \mathbf{L}) \mathbf{x}[k] \quad (3.73)$$

Donde la matriz de dinámica en lazo cerrado (\mathbf{A}_{CL}) ha sido modificada por la ganancia óptima \mathbf{L} :

$$\mathbf{A}_{CL} = \mathbf{A}_d - \mathbf{B}_d \mathbf{L} \quad (3.74)$$

Esta nueva matriz garantiza que los autovalores del sistema a lazo cerrado se reubiquen dentro del círculo unitario, confiriendo la estabilidad asintótica deseada.

$$\lambda_1 = 0,0000 + 0,0000i \quad (3.75)$$

$$\lambda_2 = 0,9082 + 0,0000i \quad (3.76)$$

$$\lambda_{3,4} = 0,9302 \pm 0,0892i \quad (3.77)$$

Otra forma para el calcular la ganancia anterior, se puede utilizar el comando de Matlab *dlqr*,

3. PLANTA DE INTERÉS

el cual tiene la siguiente sintaxis [8]

$$[L, S, P] = dlqr(A_d, B_d, Q, R) \quad (3.78)$$

Dado que la magnitud de todos los autovalores es menor que la unidad ($|\lambda_i| < 1$), se concluye que el sistema ha sido correctamente estabilizado y la representación final del sistema estabilizado, obtenida después de la realimentación de estados, se presenta como:

$$\begin{aligned} \mathbf{x}_{k+1} &= \mathbf{A}_{cl}\mathbf{x}_k + \mathbf{B}_d\bar{\mathbf{u}}_k \\ \mathbf{y}_k &= \mathbf{C}_d\mathbf{x}_k \end{aligned} \quad (3.79)$$

Dado que el esquema de DLQR presenta un estimador de estados (observador) en su estructura [3.5], se diseña el observador sabiendo que su dinámica del se define mediante la siguiente ecuación:

$$\hat{\mathbf{x}}_{k+1} = A_d\hat{\mathbf{x}}_k + B_d u_k + L_o(y_k - C_d\hat{\mathbf{x}}_k) \quad (3.80)$$

Donde:

- $\hat{\mathbf{x}}_k$ es el vector de estados estimado.
- L_o es la ganancia de corrección del observador.
- $(y_k - C_d\hat{\mathbf{x}}_k)$ representa el residuo o error de innovación entre la salida real y la estimada.

Para hallar la ganancia L_o de forma óptima, se resuelve la Ecuación Algebraica de Riccati [3.71], una vez obtenido P, la ganancia óptima del observador se calcula como:

$$L_o = (A_d P C_d^T)(C_d P C_d^T + R)^{-1} \quad (3.81)$$

Por otra parte el valor de P para el observador se puede calcular usando el comando de Matlab *idare*, el cual tiene la siguiente sintaxis [7].

$$P = idare(A'_d, C'_d, Q_o, R_o); \quad (3.82)$$

se puede expresar la dinámica final del observador con la forma:

$$\hat{\mathbf{x}}_{k+1} = (A_d - L_o C_d)\hat{\mathbf{x}}_k + \begin{bmatrix} B_d & L_o \end{bmatrix} \begin{bmatrix} u_k \\ y_k \end{bmatrix} \quad (3.83)$$

A partir de [3.83], se extraen las matrices que definen el comportamiento del estimador ($A_{obs}, B_{obs}, C_{obs}$):

$$\begin{aligned} A_{obs} &= [A_d - L_o C_d] \\ B_{obs} &= [B_d \quad L_o] \\ C_{obs} &= [I_{n \times n}] \end{aligned} \quad (3.84)$$

Con $n = 4$ correspondiente a los estados del sistema.

3.3.8 Planta aumentada

Para la implementación del Control Predictivo por Modelo (MPC), es imperativo contar con una representación que integre tanto la dinámica física de la planta como la dinámica del estimador de estado. Al sustituir la ley de control por realimentación $u_k = -L\hat{x}_k + r_k$ en las ecuaciones de estado y del observador, se obtiene la dinámica de la planta aumentada.

Para obtener la planta aumentada en primer lugar, se sustituye la acción de control en la dinámica original de la planta :

$$x_{k+1} = A_d x_k + B_d(r_k - L\hat{x}_k) \quad (3.85)$$

Posteriormente, se realiza la sustitución en la ecuación del observador [3.83](#) en, considerando la diferencia entre la salida real $y_k = C_d x_k$ y la estimada $C_d \hat{x}_k$, desarrollando paso por paso se obtiene lo obtenido en

$$\begin{aligned} \hat{x}_{k+1} &= A_d \hat{x}_k + B_d(r_k - L\hat{x}_k) + L_o(y_k - C_d \hat{x}_k) \\ &= A_d \hat{x}_k - B_d L \hat{x}_k + B_d r_k + L_o(C_d x_k - C_d \hat{x}_k) \\ &= (A_d - B_d L - L_o C_d) \hat{x}_k + L_o C_d x_k + B_d r_k \end{aligned} \quad (3.86)$$

3.3.9 Representación matricial aumentado

Al definir el vector de estado aumentado como $\mathbf{x}_a = [x_k, \hat{x}_k]^T$, el modelo completo en lazo cerrado queda representado por el par (A_a, B_a) :

$$\begin{bmatrix} x_{k+1} \\ \hat{x}_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_d & -B_d L \\ L_o C_d & A_d - L_o C_d - B_d L \end{bmatrix}}_{A_a} \begin{bmatrix} x_k \\ \hat{x}_k \end{bmatrix} + \underbrace{\begin{bmatrix} B_d \\ B_d \end{bmatrix}}_{B_a} r_k \quad (3.87)$$

3.3.10 Observador extendido para estimación de perturbaciones

Para el diseño de este observador se definen $B_p \in \mathbb{R}^{4 \times 1}$ representa la matriz de perturbación de entrada y $C_p \in \mathbb{R}^{1 \times 1}$ corresponde a la matriz de perturbación de salida, las cuales permiten modelar las incertidumbres y disturbios del sistema físico de forma independiente.

El vector de estado aumentado para este estimador se define como $\mathbf{x}_{extP} = [x, \hat{x}, d]^T \in \mathbb{R}^{9 \times 1}$. Basado en la arquitectura de pre-estabilización y la dinámica de perturbación constante ($d_{k+1} = d_k$), las matrices del sistema extendido se pueden escribir de la siguiente manera:

$$A_{extP} = \begin{bmatrix} A_d & -B_dL & B_p \\ L_oC_d & A_d - L_oC_d - B_dL & B_p \\ \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & 1 \end{bmatrix}, \quad B_{extP} = \begin{bmatrix} B_d \\ B_d \\ 0 \end{bmatrix} \quad (3.88)$$

La matriz de salida se define para que el observador pueda corregir sus estimaciones basándose en la diferencia entre la salida real de la planta y la influencia de la perturbación estimada:

$$C_{extP} = [C_d \quad \mathbf{0}_{1 \times 4} \quad C_p] \quad (3.89)$$

Se ha optado por definir la matriz de perturbación de entrada como nula ($B_p = 0$), asumiendo así la ausencia de disturbios en el actuador. No obstante, el modelo incorpora perturbaciones de salida de tipo constante.

3.3.11 Obtención de la Ganancia Óptima del Observador

La ganancia del observador K_P se obtiene resolviendo el problema dual de control regulador lineal cuadrático mediante la Ecuación Algebraica de Riccati en Tiempo Discreto (DARE). Se definen las matrices de ponderación Q_{oP} y R_{oP} para balancear la confianza entre el modelo dinámico y el ruido de medición:

$$P = \text{idare}(A_{extP}^T, C_{extP}^T, Q_{oP}, R_{oP}) \quad (3.90)$$

Finalmente, la ganancia de corrección K_P que minimiza la varianza del error de estimación se calcula como:

$$K_P = (A_{extP}PC_{extP}^T)(C_{extP}PC_{extP}^T + R_{oP})^{-1} \quad (3.91)$$

3.3.12 Matrices del estimador de orden extendido

Las matrices que definen la dinámica del observador de 9 estados, encargado de reconstruir los estados físicos, los estados del primer estimador y la perturbación de salida, son las siguientes:

$$\begin{aligned} A_{obs} &= [A_{extP} - K_P C_{extP}] \\ B_{obs} &= [B_{extP} \quad K_P] \\ C_{obs} &= [I_{9 \times 9}] \end{aligned} \quad (3.92)$$

4 Simulación del Sistema de Control

En este capítulo se presenta las simulaciones del esquema de control del sistema Ball & Beam con el propósito de validar el correcto funcionamiento del del controlador MPC. La simulación se realiza en el software Matlab/Simulink el cual contiene el modelo de la planta de estudio en conjunto del bloque encargado del control predictivo y el bloque de seguimiento a referencia.

4.1 Herramientas de Software

4.1.1 Matlab

MATLAB es una plataforma de programación y computación numérica de alto nivel diseñada para el análisis de datos, el desarrollo de algoritmos y la creación de modelos matemáticos. [9] Su entorno integra un lenguaje orientado al cálculo matricial, lo que optimiza el procesamiento de operaciones complejas en áreas como la ingeniería, la ciencia y la investigación académica. Gracias a su arquitectura modular, permite la integración con herramientas especializadas (toolboxes) para control automático, procesamiento de señales y aprendizaje profundo.

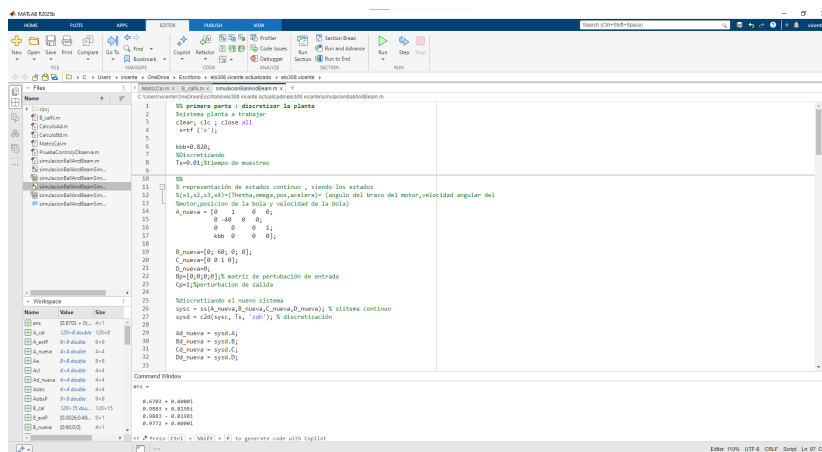


Fig. 4.1: Entorno de programación Matlab.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

Algunos de los componentes principales del entorno de MATLAB que se pueden observar en la figura 4.1 son:

- **Current Folder:** Panel que permite gestionar el sistema de archivos, establecer el directorio de trabajo activo y acceder rápidamente a los scripts y funciones disponibles.
- **Command Window:** Interfaz principal para la ejecución de comandos en tiempo real, permitiendo visualizar resultados inmediatos y depurar líneas de código de forma interactiva e inmediata.
- **Editor:** Entorno de desarrollo integrado (IDE) para la creación, edición y depuración de scripts (.m) y funciones personalizadas de MATLAB.
- **Workspace:** Panel que almacena y muestra las variables creadas durante la sesión actual, detallando sus dimensiones, valores y tipos de datos asociados.

4.1.2 Simulink

Simulink es un entorno de programación gráfica integrado en MATLAB, diseñado para el modelado, simulación y análisis de sistemas dinámicos multidominio. [10] Basado en un esquema de diagramas de bloques, permite representar sistemas complejos de forma visual, facilitando el diseño basado en modelos. Es ampliamente utilizado en la industria para el prototipado rápido de sistemas de control, procesamiento de señales y simulación de hardware en el lazo. Esta plataforma permite que el proceso de desarrollo gire en torno a un modelo virtual, lo que facilita la verificación temprana de diseños y reduce significativamente la dependencia de prototipos físicos.

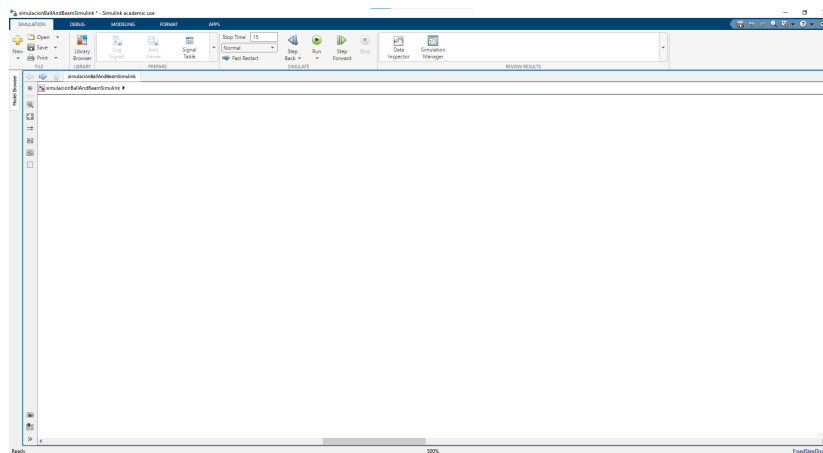


Fig. 4.2: Entorno de programación Simulink.

Los componentes fundamentales del entorno de Simulink incluyen:

4. SIMULACIÓN DEL SISTEMA DE CONTROL

- **Barra de herramientas principal:** Ubicada en la parte superior, contiene pestañas como *Simulation*, *Debug*, *Modeling*, *Format* y *Apps*, que permiten acceder a comandos de ejecución, configuración y análisis.
- **Panel de modelado (Canvas):** Área central donde se colocan y conectan los bloques para crear el diagrama de simulación, soportando múltiples pestañas de modelos abiertos simultáneamente.
- **Library Browser:** Repositorio central que organiza bibliotecas de bloques predefinidos para modelado de algoritmos, funciones matemáticas y sistemas de control.
- **Model Browser:** Panel que permite navegar jerárquicamente por los diversos niveles de subsistemas y organizar los componentes internos del modelo.
- **Property Inspector:** Interfaz utilizada para visualizar y modificar de forma rápida las propiedades y parámetros del bloque o subsistema que se encuentre seleccionado.
- **Controles de simulación:** Incluyen los botones para iniciar, detener o avanzar paso a paso la simulación, además de opciones para configurar el *solver* y el tiempo de ejecución.
- **Herramientas de análisis de resultados:** Ubicado en la pestaña *Review Results*, permiten el uso de herramientas como el *Data Inspector* y el *Simulation Manager* para validar el comportamiento del sistema.

4.1.3 Preparación simulación

Para el desarrollo de las simulaciones se utilizan las matrices en espacios de estados obtenidas en 3.49, discretizando las matrices usando el comando de Matlab `c2d` y utilizando $T_s=0,1[s]$, con el propósito de verificar si el controlador MPC diseñado es capaz de compensar la acción de control de DLQR. Revisar anexo (`PreCompiler Ball And Beam`).

Para realizar las pruebas con el esquema de MPC, se establecen los parámetros de la siguiente manera: $N_p=12$, $\Gamma = 0,3$ y Ω mostrada en la figura 4.1, como matriz de pesos de penalización

4. SIMULACIÓN DEL SISTEMA DE CONTROL

$$\Omega = \begin{bmatrix} 1e6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1e2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2e4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2e6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1e2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1e1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 5e7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1e1 \end{bmatrix} \quad (4.1)$$

Como se busca verificar la compensación de MPC, el vector de estado iniciales se expresa de la forma:

$$x_0 = [0 \ 0 \ 0 \ 0]^T \quad (4.2)$$

Por otra parte se establecen las restricciones para los estado, principalmente la posición de la bola para mantenerla dentro de los limites físicos de la viga, también las restricciones de actuación dentro del rango de voltaje máximo soportado para no dañar el servomotor y el ángulo de la viga por debajo del valor máximo soportado para evitar daños mecánicos.

$$x_{min} = \begin{bmatrix} -\pi/4 \\ -100 \\ -0,21 \\ -100 \\ -\pi/4 \\ -100 \\ -0,21 \\ -100 \end{bmatrix}, \quad x_{max} = \begin{bmatrix} \pi/4 \\ 100 \\ 0,21 \\ 100 \\ \pi/4 \\ 100 \\ 0,21 \\ 100 \end{bmatrix} \quad (4.3)$$

$$u_{min} = -10, \quad u_{max} = 10$$

Sabemos que el estado inicial del sistema original [4.2](#) tiene cuatro estados, el vector de restricciones contiene ocho, puesto que MPC actúa sobre la planta aumentada.

La ganancia de DLQR como las ganancias de los observadores están calculadas en Matlab en el anexo (PreCompiler Ball And Beam) y tienen los siguientes valores.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

$$L = [1,8252 \quad 0,0460 \quad 2,4846 \quad 4,7238] L_o = \begin{bmatrix} 0,0001 \\ 0,0000 \\ 1,0009 \\ 0,0093 \end{bmatrix} KP = \begin{bmatrix} -0,0727 \\ -0,2505 \\ 0,0822 \\ 0,2395 \\ -0,0751 \\ -0,2505 \\ 0,0662 \\ 0,0796 \\ 0,9375 \end{bmatrix} \quad (4.4)$$

Las matrices de peso ocupadas para configurar tanto para DLQR como para los observadores tienen los siguientes valores:

$$Q = \begin{bmatrix} 1e1 & 0 & 0 & 0 \\ 0 & 5e2 & 0 & 0 \\ 0 & 0 & 5e3 & 0 \\ 0 & 0 & 0 & 3e3 \end{bmatrix} \quad R = 0,05 \quad (4.5)$$

$$Q_o = \begin{bmatrix} 1e-2 & 0 & 0 & 0 \\ 0 & 1e0 & 0 & 0 \\ 0 & 0 & 1e6 & 0 \\ 0 & 0 & 0 & 1e4 \end{bmatrix} \quad R_o = 0,1 \quad (4.6)$$

$$Q_{oP} = \begin{bmatrix} 1e-2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1e0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1e2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1e2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1e-2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1e0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1e2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1e2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1e9 \end{bmatrix}, \quad R_{oP} = 1 \quad (4.7)$$

A continuación se muestran las partes del esquema de control implementado en Simulink el cual está compuesto por el modelo de la planta aumentada, el controlador MPC y el bloque de seguimiento a referencia, estos bloques son los que se van a utilizar durante las simulaciones.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

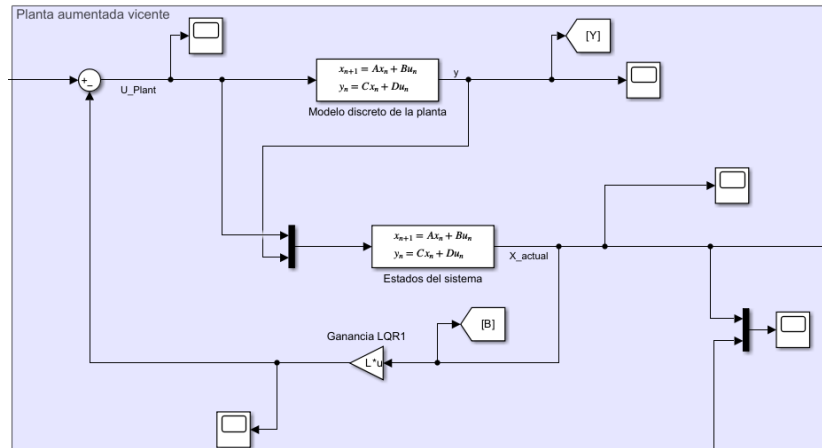


Fig. 4.3: Modelo Simulink planta aumentada.

En la figura 4.3 se muestra el modelo de la planta aumentada implementada en Simulink según el esquema definido en la figura 3.5, en la entrada a la planta se conecta el observador extendido 4.4, el cual se encarga de estimar los estados de la planta aumentada, para el posterior uso en el bloque MPC, el cual opera sobre la planta aumentada con el propósito de compensar la acción de control de DLQR y así cumplir con las restricciones definidas en 4.3.

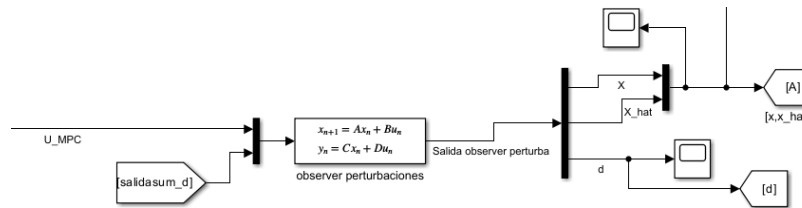


Fig. 4.4: Observador extendido de la planta aumentada.

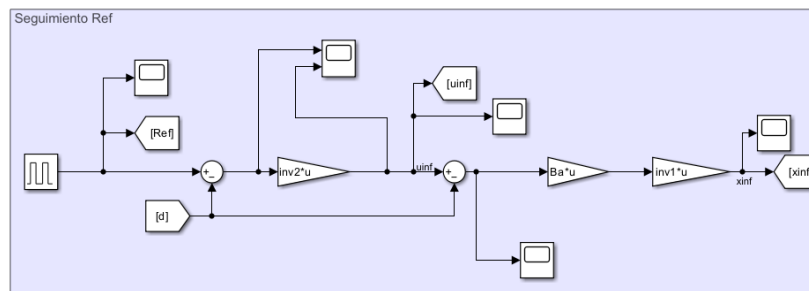


Fig. 4.5: Bloque seguimiento a referencia.

En la Figura 4.5 se presenta la implementación del bloque de seguimiento de referencia. Este sistema incorpora tres ganancias precalculadas en MATLAB, entre las cuales destaca la

4. SIMULACIÓN DEL SISTEMA DE CONTROL

matriz B_a , definida según los ocho estados de la planta aumentada, cabe recalcar que también presenta dos sumadores donde el primero es el único que se va a usar puesto que es el que corresponde para la entrada de perturbaciones de salida el segundo sumador correspondiente a $u_i n.f$ es utilizado para las perturbaciones de entrada que en nuestro caso es una matriz nula. Para un mayor detalle sobre este cálculo, consultar el anexo (Precompilar Ball And Beam).

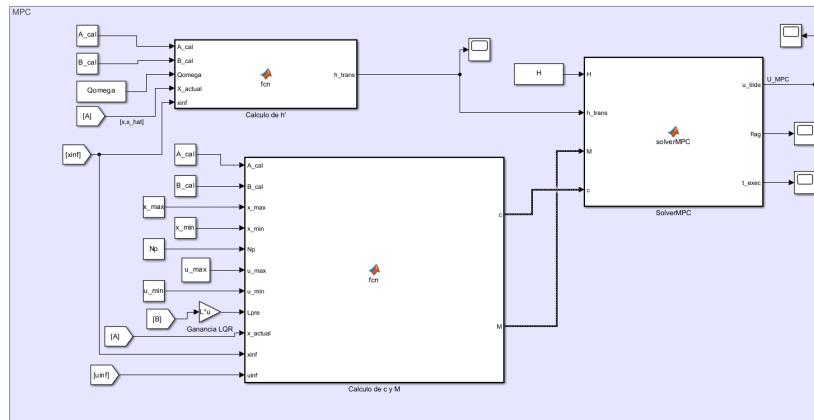


Fig. 4.6: Bloque MPC Simulink.

En la Figura 4.6 se ilustra la arquitectura del controlador MPC, la cual está integrada por tres módulos principales: Cálculo de h' , Cálculo de c y M y SolverMPC. Los dos primeros se encargan de la formulación del problema de optimización. El bloque Cálculo de h' determina el término lineal de la función de costo, mientras que el bloque Cálculo de c y M define las restricciones de desigualdad del sistema. Estos parámetros son procesados por el bloque SolverMPC, que ejecuta el algoritmo *quadprog* de MATLAB configurado con el método *active-set*. Para una descripción detallada de la lógica interna de cada componente, consúltense los anexos (Solver MPC, Cálculo De h' y Cálculo De c y M).

4. SIMULACIÓN DEL SISTEMA DE CONTROL

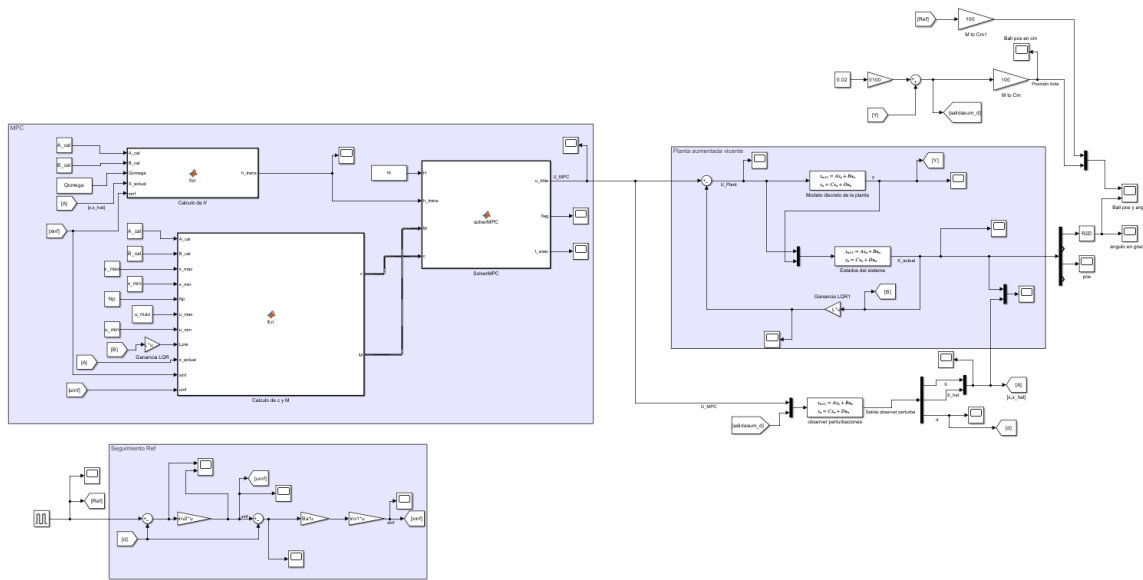


Fig. 4.7: Esquema de control MPC con seguimiento a referencia.

Finalmente, se presenta el esquema de control completo en la figura 4.7 con el cual se lleva acabo las pruebas iniciales.

4.1.4 Pruebas DLQR

Para la primera parte se va a corroborar el correcto funcionamiento de DLQR añadiendo una perturbación de salida en $t=35$ [s] de 0.02 [m], donde el observador externo con acción integral estima la perturbación correspondiente como se observa en la figura 4.8

4. SIMULACIÓN DEL SISTEMA DE CONTROL

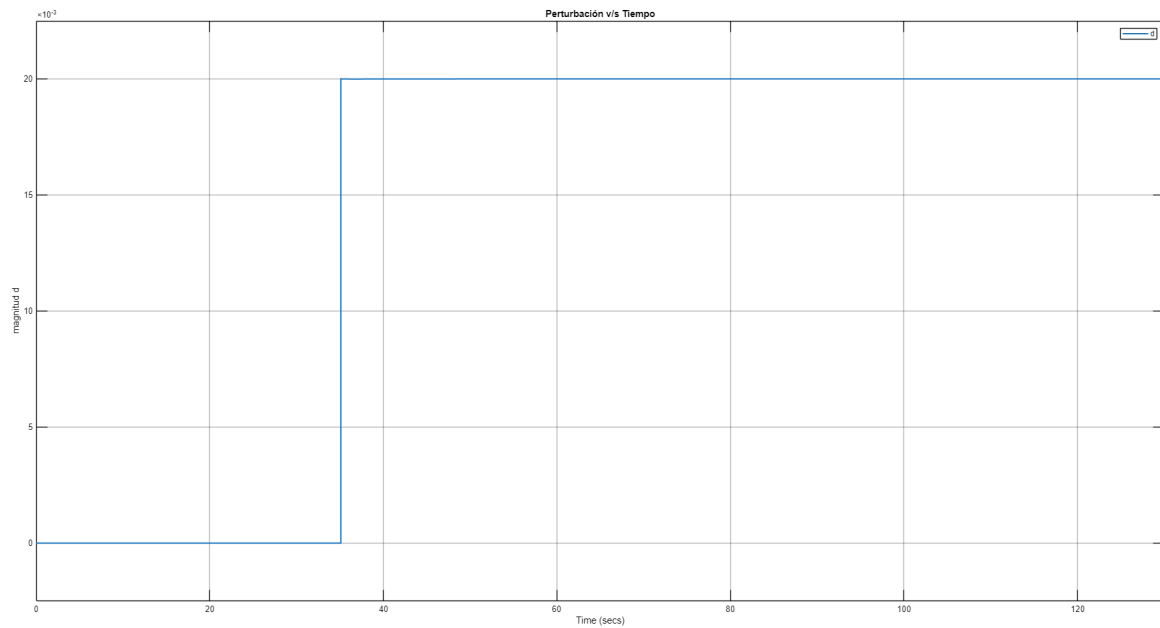


Fig. 4.8: Estimación de la perturbación realizada por el observador externo.

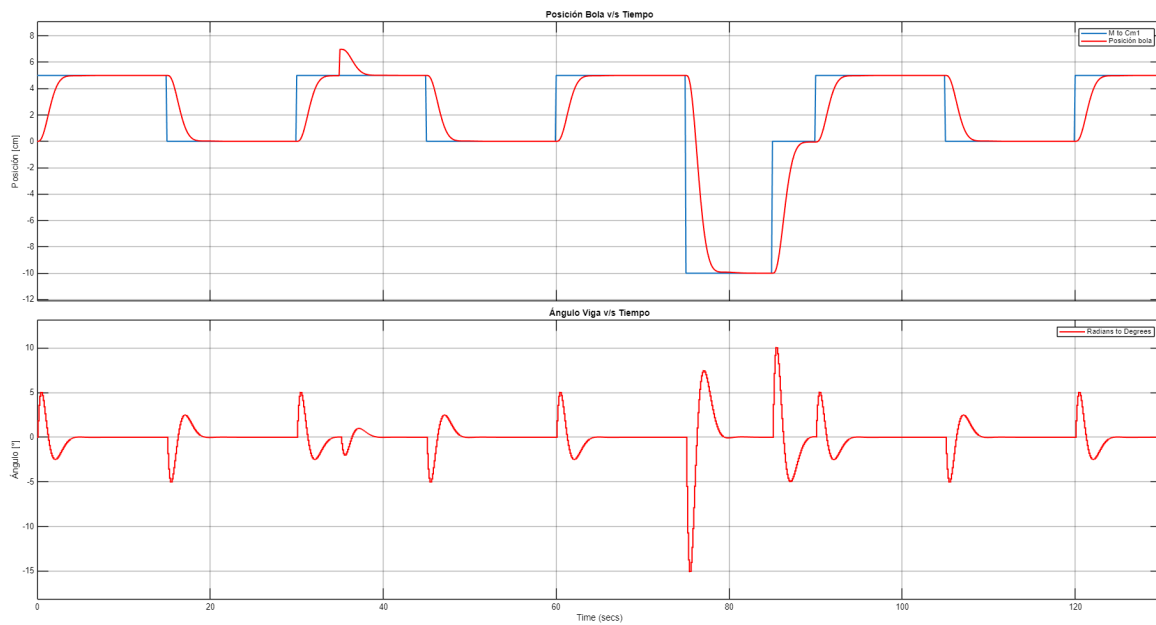


Fig. 4.9: Posición y ángulo de la bola DLQR con perturbación $d=0.02$ [m].

En la figura [4.9](#) se observa que DLQR es capaz de rechazar la perturbaciones y seguir la referencia sin ningún problema.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

4.1.5 Pruebas de simulación MPC sin perturbaciones

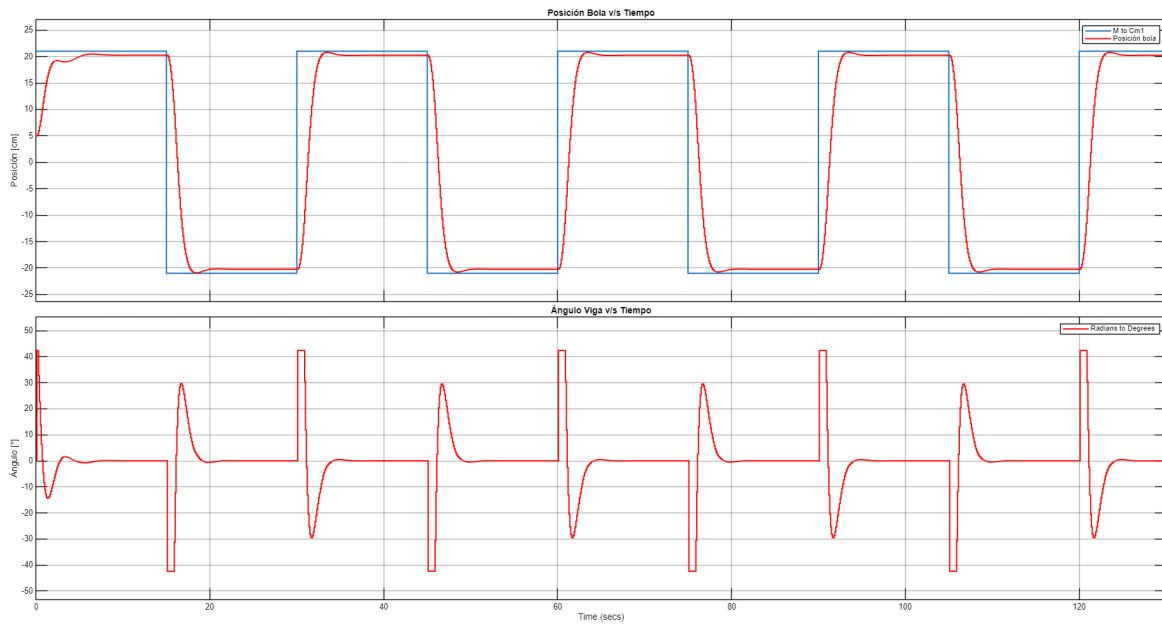


Fig. 4.10: Posición y ángulo de la bola sin perturbaciones.

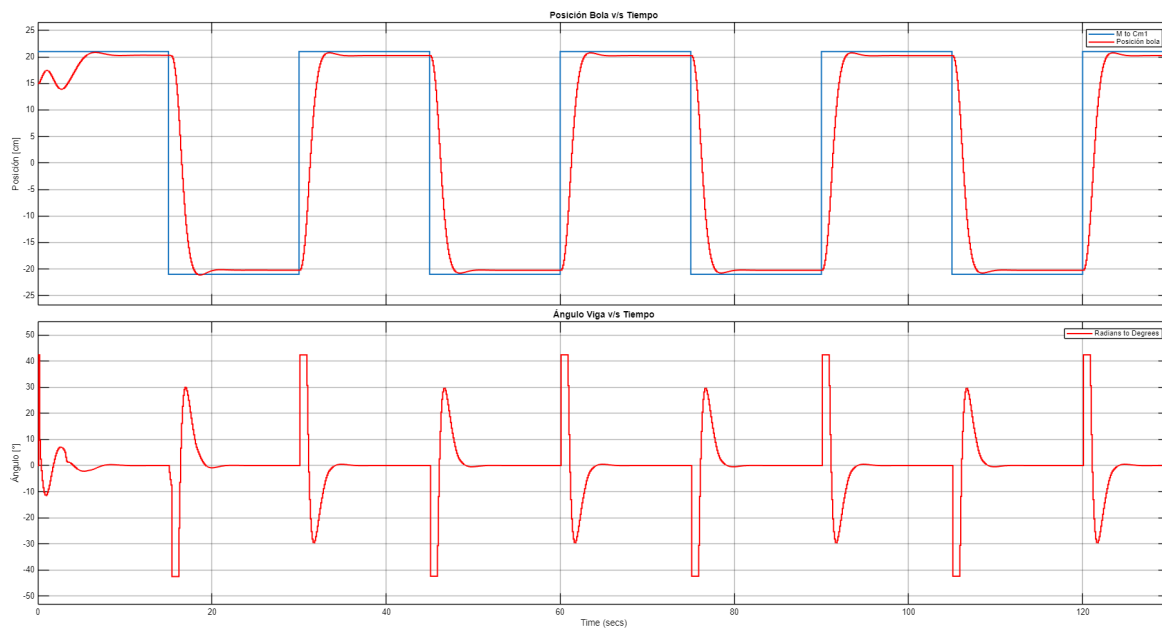


Fig. 4.11: Posición y ángulo de la bola sin perturbaciones.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

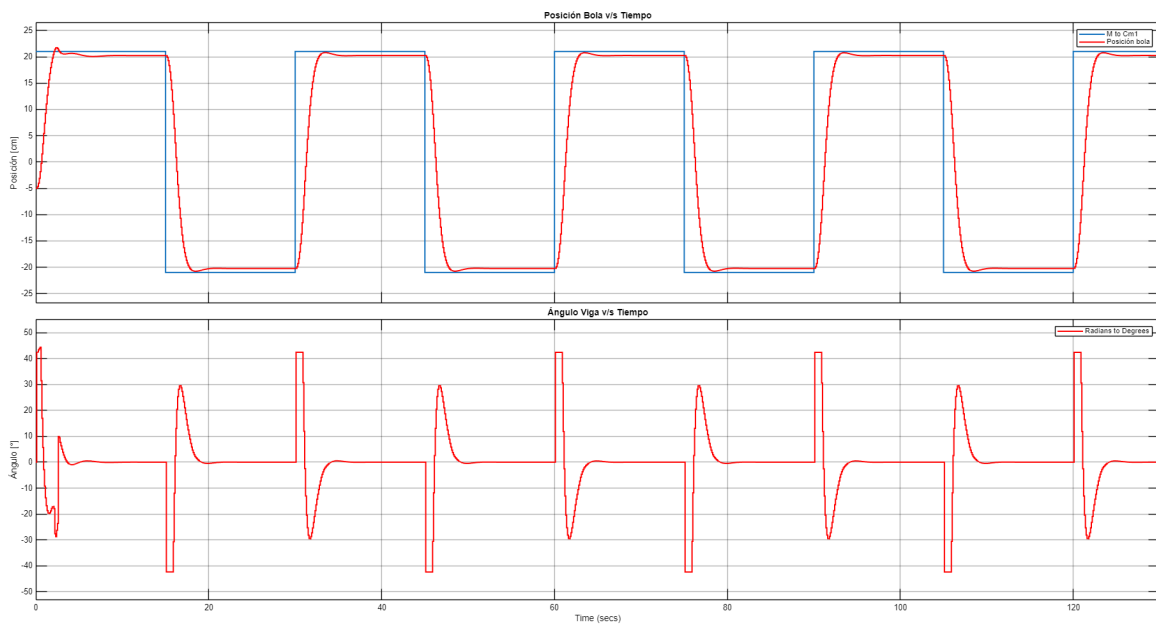


Fig. 4.12: Posición y ángulo de la bola condición inicial -5[cm] sin perturbaciones.

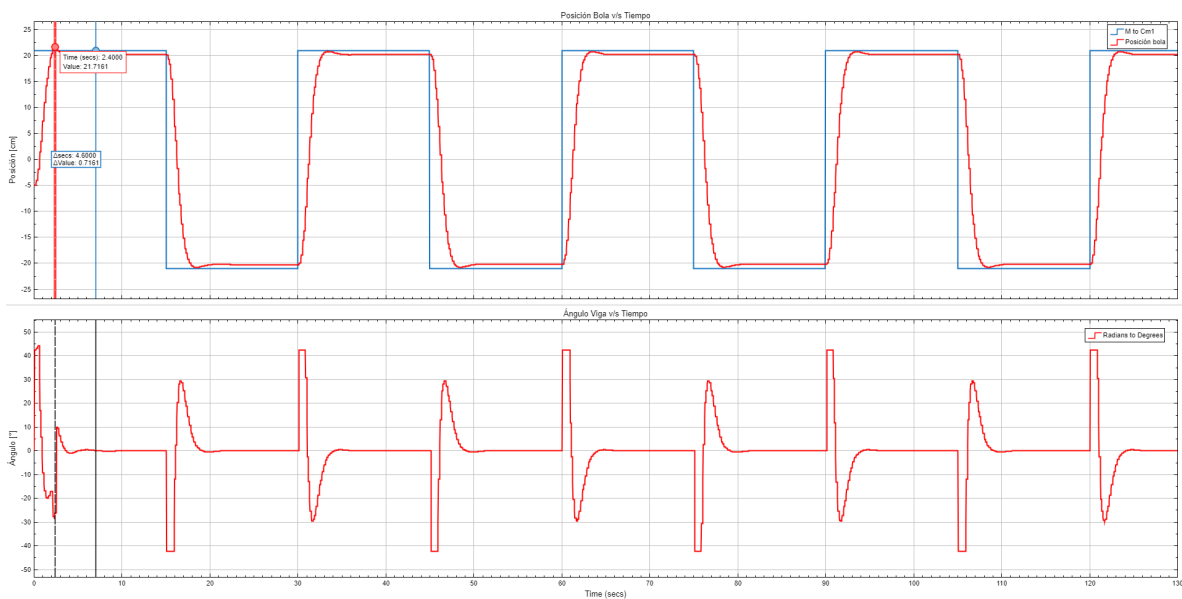


Fig. 4.13: Valor detallado overshoot condición inicial -5 [cm].

De las figuras anteriores se puede concluir que tiene una gran importancia en donde se define la posición inicial de la bola, debido a que existe un overshoot en el transiente producto de la inercia de la bola que el controlador no es capaz de frenar, como se puede observar en [4.11](#) no existe riesgo de overshoot que produzca que la bola salga del límite físico de la viga, también

4. SIMULACIÓN DEL SISTEMA DE CONTROL

se puede observar que en 4.10 y 4.12 a medida que la condición inicial es definida más lejos de la referencia existe más complicaciones al momento de parar el efecto de la inercia, hasta tal punto de que la bola se sale de los límites de la viga, rompiendo también los límites de seguridad de 0.5 [cm] extra en ambos lados de la viga como se puede observar en 4.13.

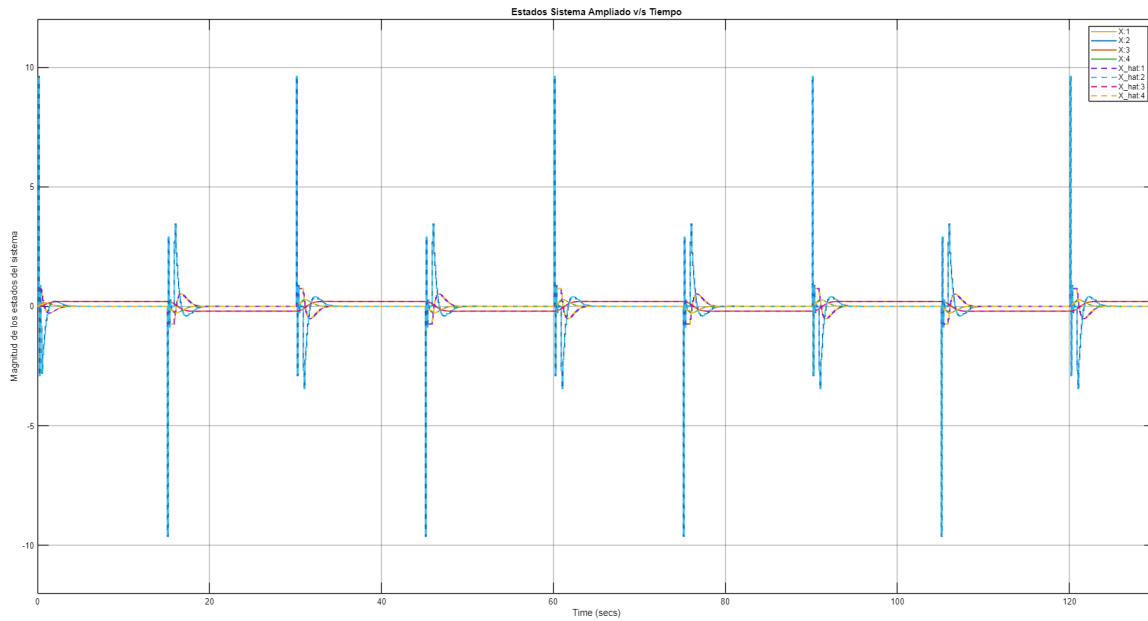


Fig. 4.14: Estados del observador sin perturbaciones.

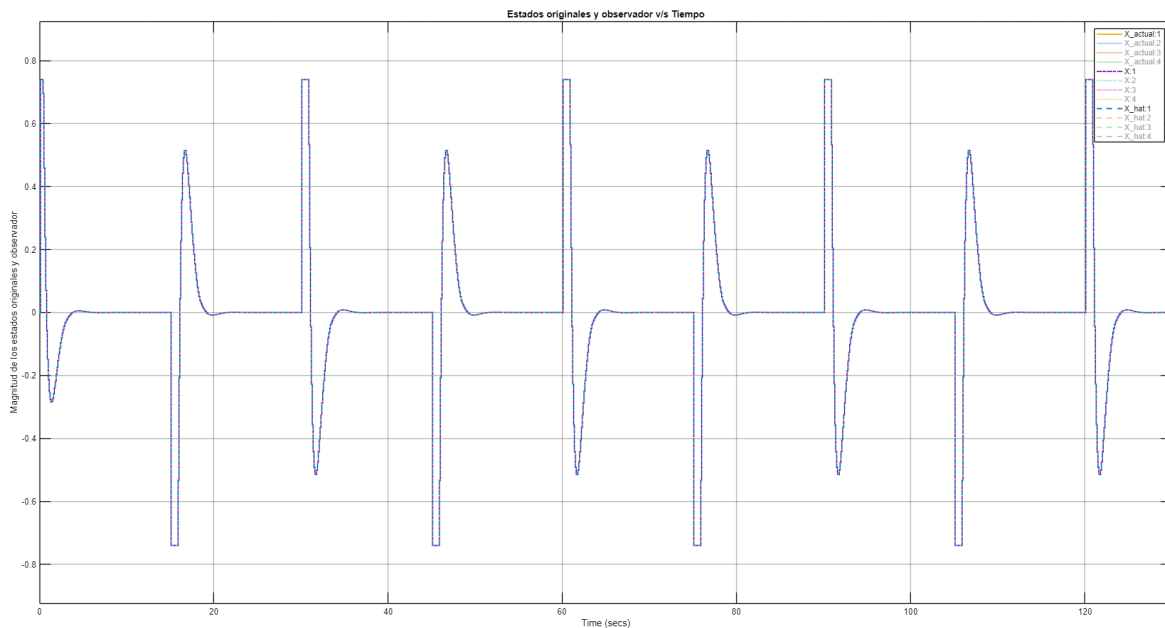


Fig. 4.15: Estado ángulo viga de la planta y del observador sin perturbaciones.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

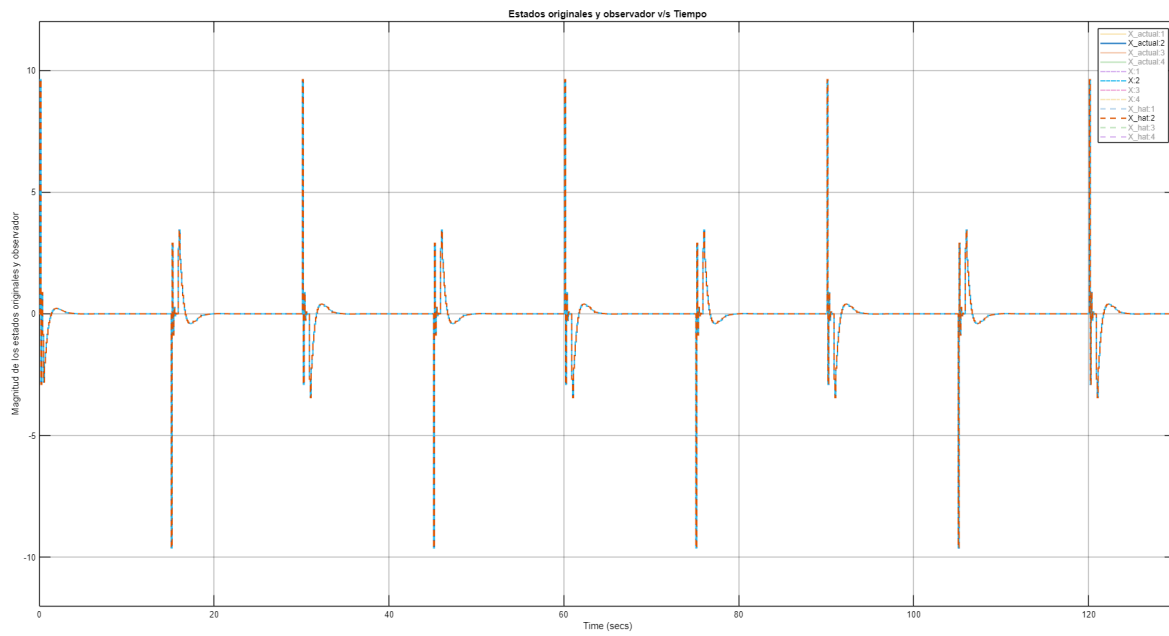


Fig. 4.16: Estado velocidad angular viga de la planta y del observador sin perturbaciones.

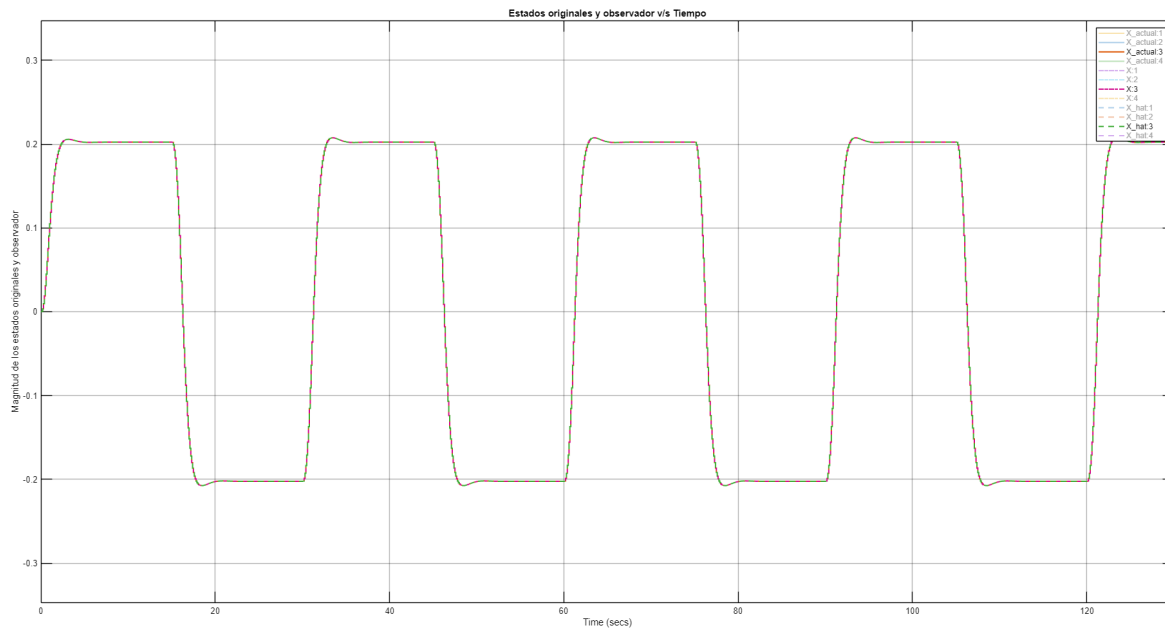


Fig. 4.17: Estados posición de la planta y del observador sin perturbaciones.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

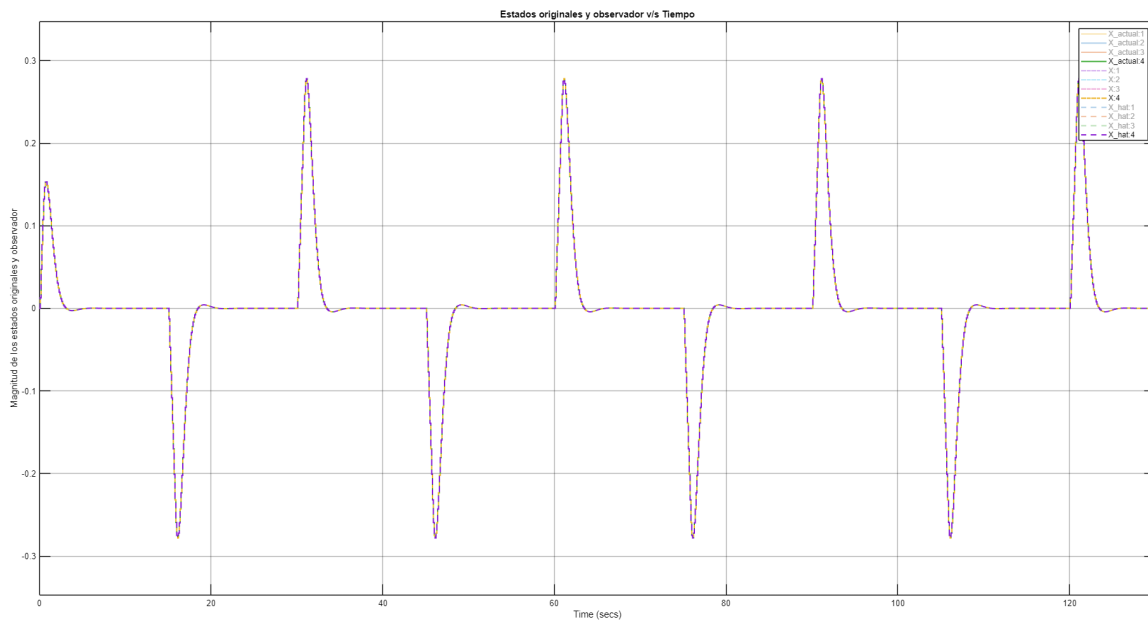


Fig. 4.18: Estados velocidad bola de la planta y del observador sin perturbaciones.

De las figuras 4.14, 4.15, 4.16, 4.17 y 4.18 se puede ver que el observador externo el cual es encargado de estimar tantos los estados como la perturbación de salida es capaz de estimar los estados originales de la planta de manera correcta.

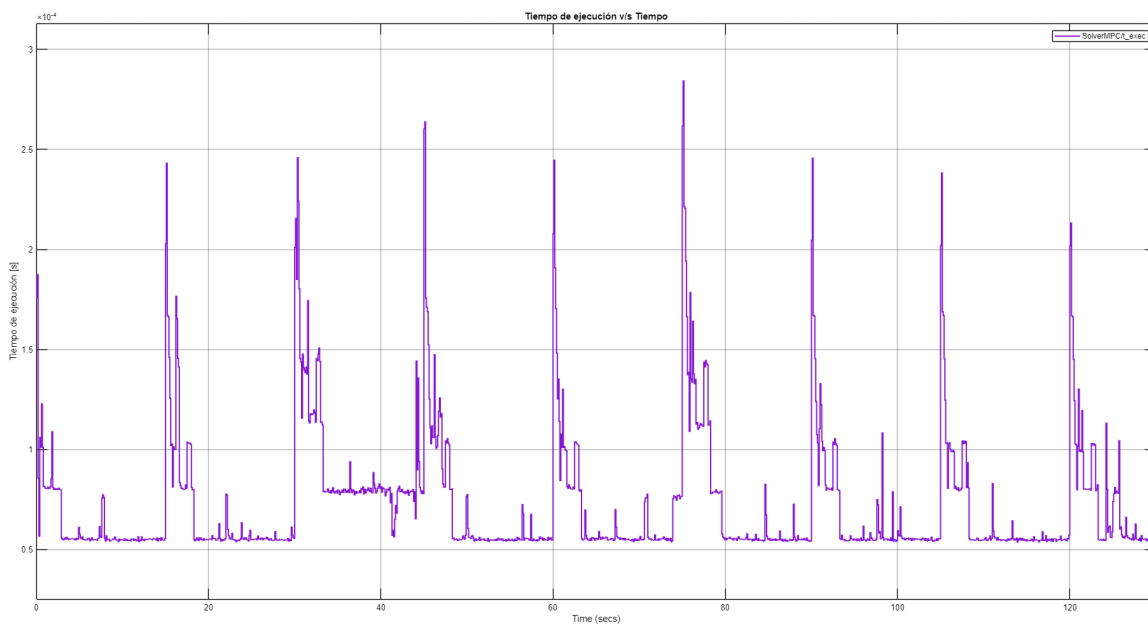


Fig. 4.19: Tiempo de ejecución del solver sin perturbaciones

En la figura 4.19 se puede observar que el tiempo de ejecución del solver Quadprog está en

4. SIMULACIÓN DEL SISTEMA DE CONTROL

el orden de los 10^{-4} , lo que nos indica que el calculo lo realiza con una diferencia de tres ordenes de magnitud con respecto al tiempo de muestreo.

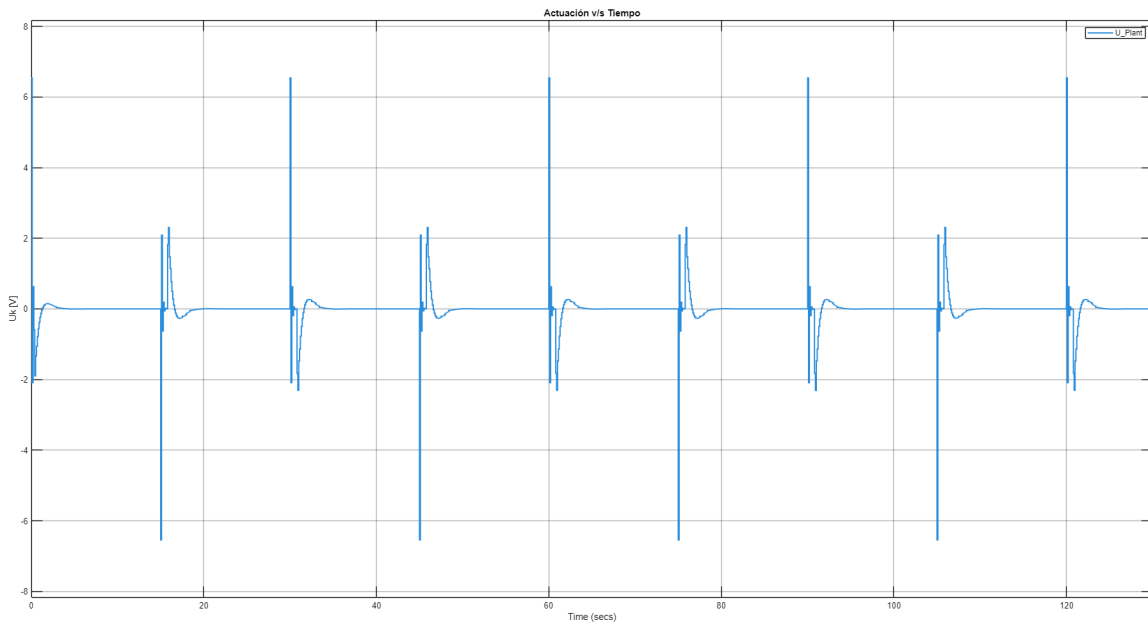


Fig. 4.20: Actuación de la planta

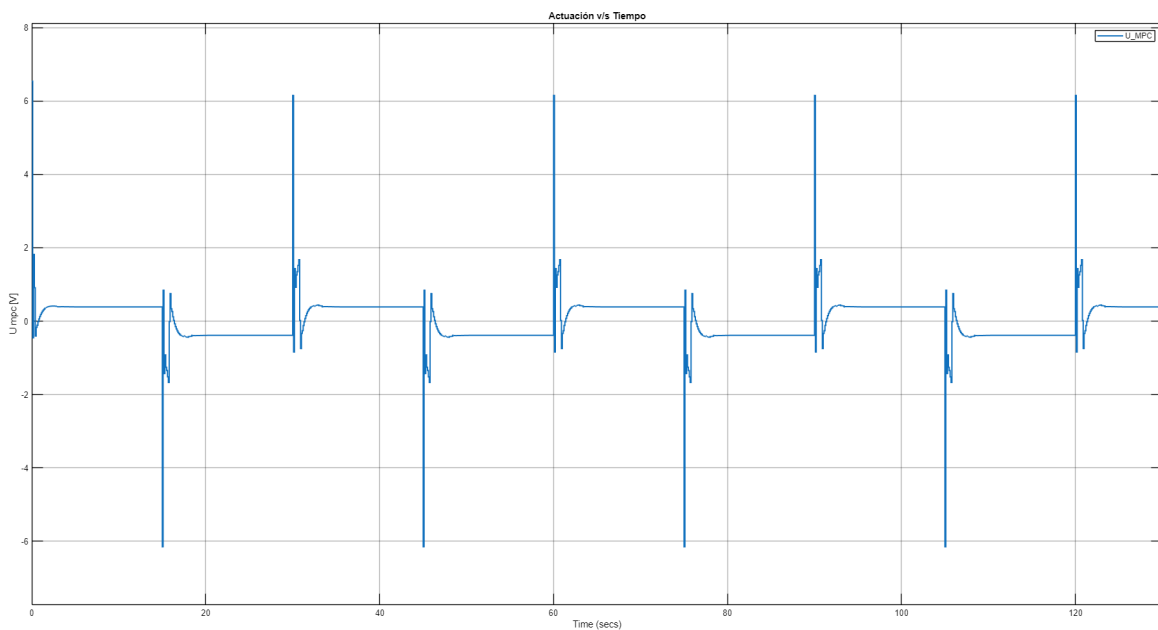


Fig. 4.21: Voltaje de salida de MPC.

de la figura 4.20 se puede observar que en ningún momento el voltaje aplicado a la planta

4. SIMULACIÓN DEL SISTEMA DE CONTROL

sobrepasa los límites definidos en [4.3](#)

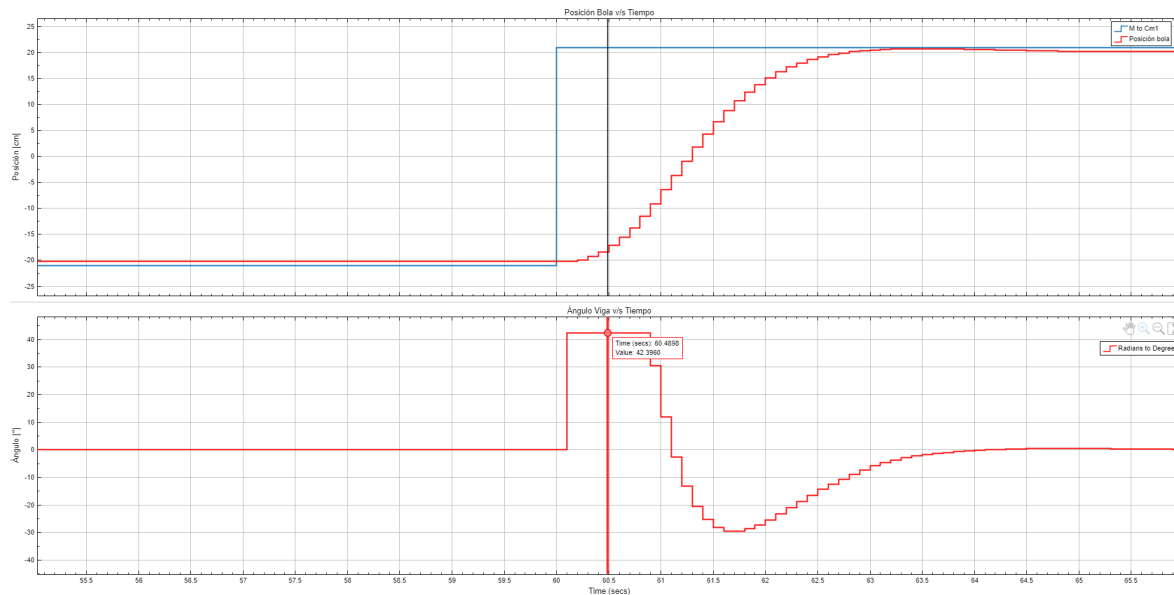


Fig. 4.22: Valor ángulo detallada

En la figura [4.22](#) se observa que el ángulo pasa de 0° a 42° cumpliendo con los requisitos definidos en [4.3](#) en un instante de tiempo muy pequeño, estos cambios bruscos pueden ocasionar daño en el servomotor por lo que a futuro se propone implementar condiciones de actuación del tipo *SlewRate* el cual no permita saltos tan bruscos de ángulos.

4.1.6 Pruebas de simulación MPC con perturbaciones

Para las pruebas realizadas en esta sección se incorpora dos bloques step a la salida del sistema, lo cual simula perturbaciones constantes. Las perturbaciones involucradas son de 1[cm] la cual se activa a los 20 segundos y la otra es de 0.5[cm] la cual se activa a los 50 segundos, también se mantienen los pesos de las matrices de penalización y el horizonte de predicción definidos con anterioridad.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

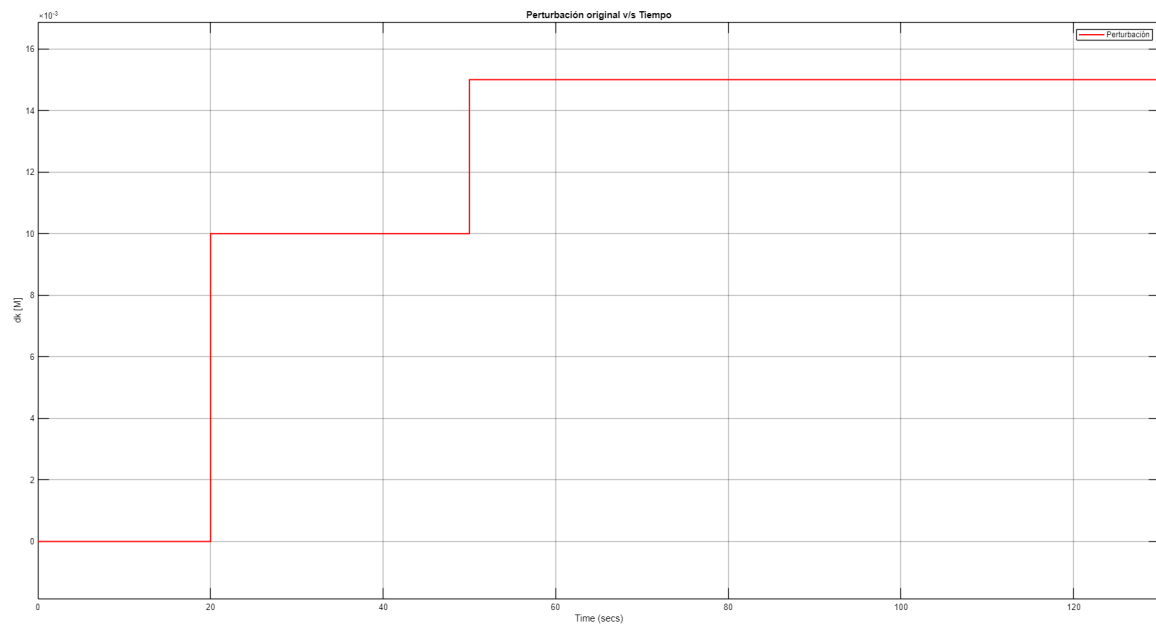


Fig. 4.23: Perturbación de salida original ingresada al sistema.

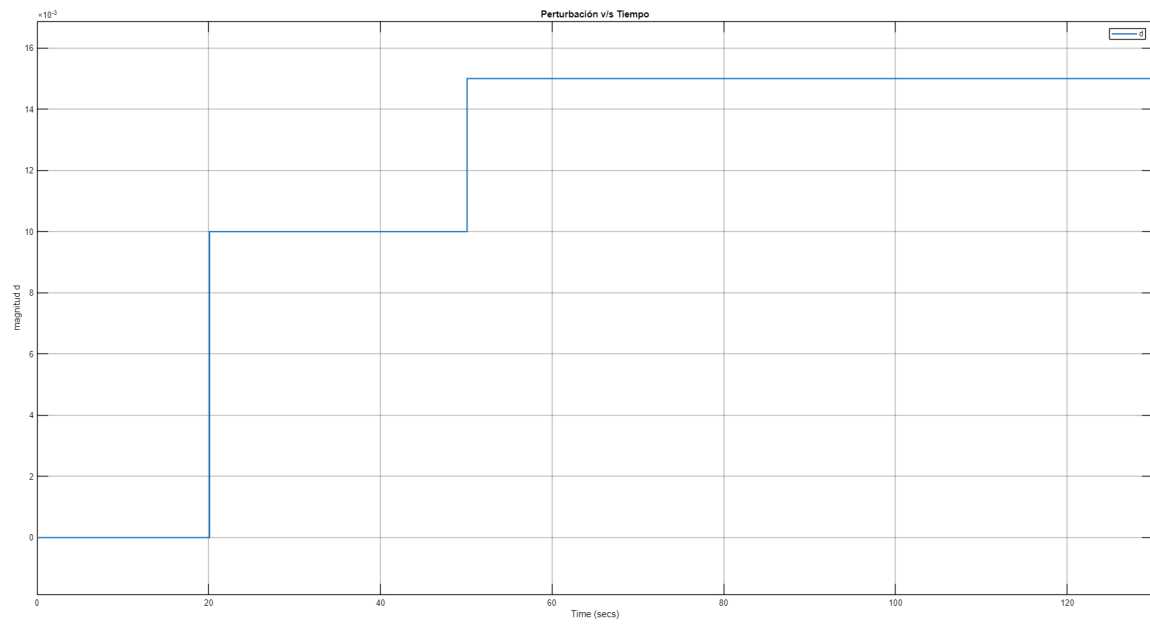


Fig. 4.24: Perturbación estimada por el observador externo.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

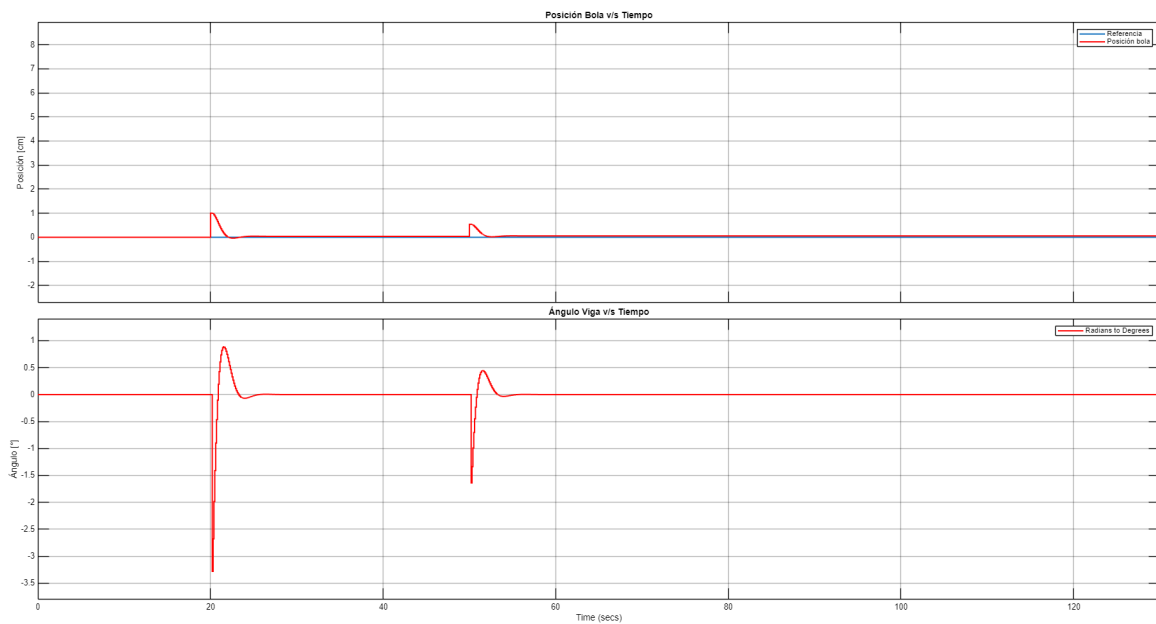


Fig. 4.25: Posición bola y ángulo de la viga con solo perturbaciones.

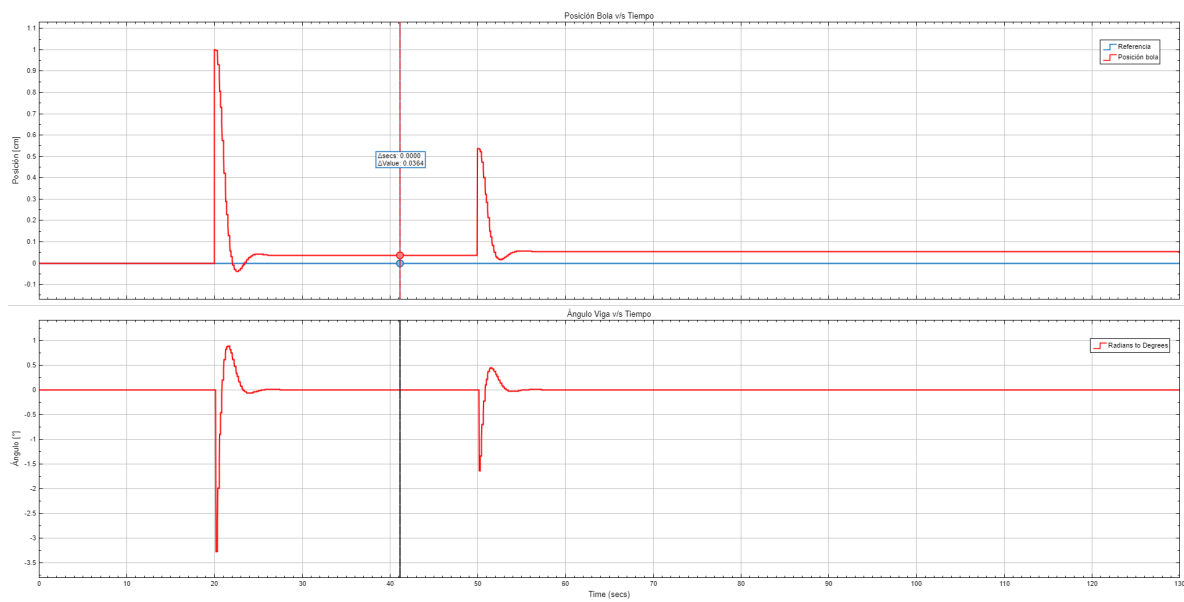


Fig. 4.26: Posición bola y ángulo viga con perturbaciones detallada.

De las figuras [4.25](#) y [4.26](#) se puede observar que con los pesos definidos MPC no es capaz de rechazar al 100 % la perturbación alcanzando un valor final de 0,0364[cm].

4. SIMULACIÓN DEL SISTEMA DE CONTROL

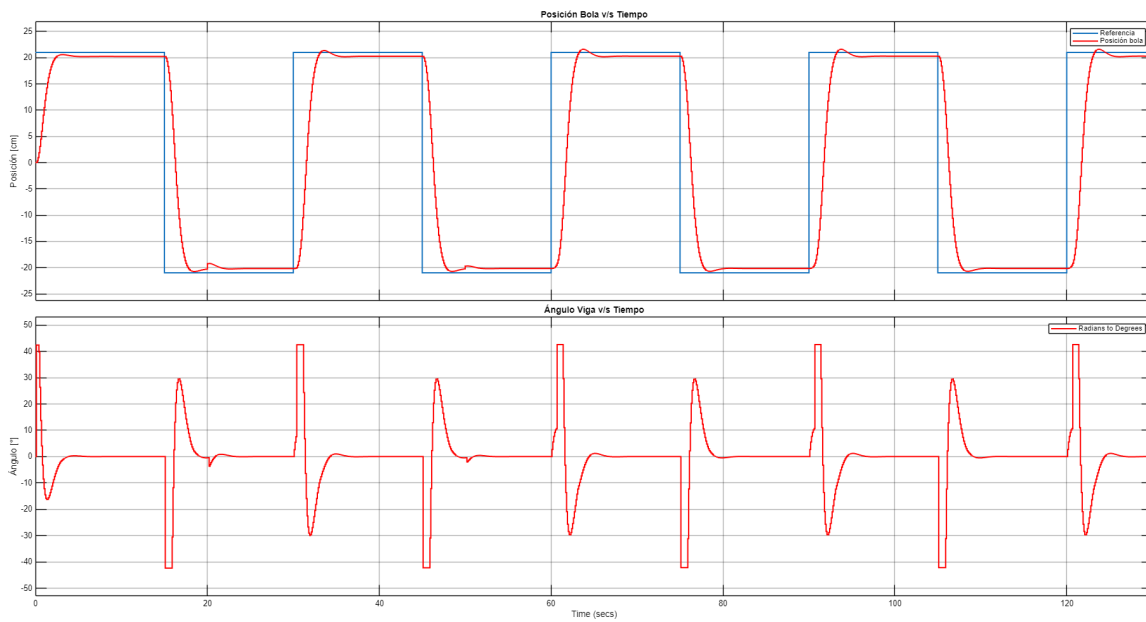


Fig. 4.27: Posición bola y ángulo viga con condición inicial 0 [cm].

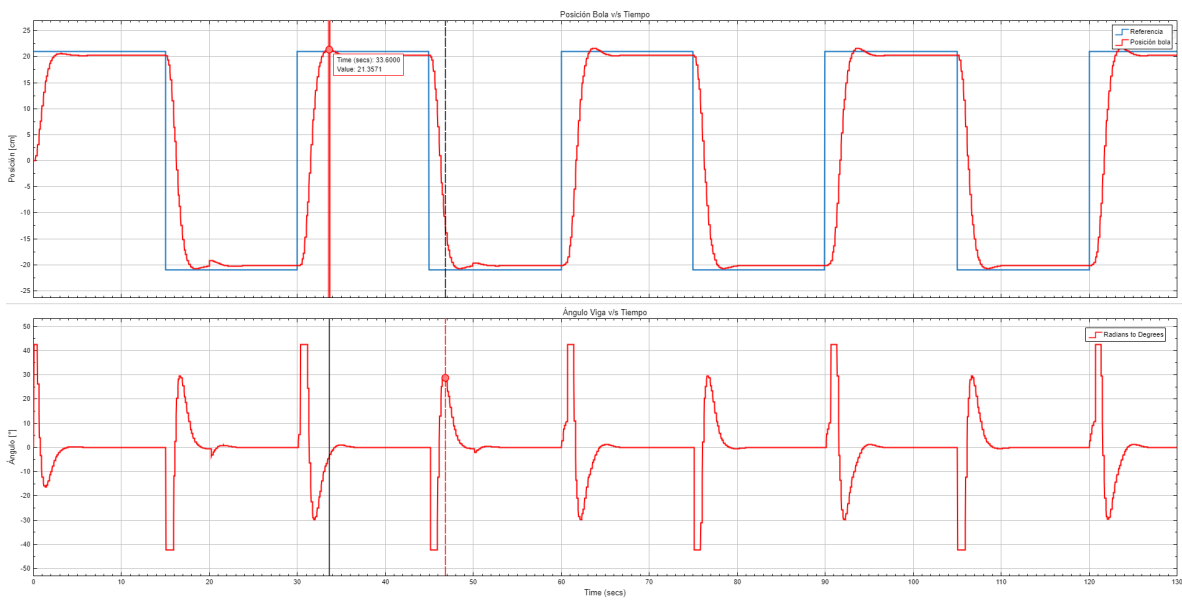


Fig. 4.28: Posición bola y ángulo viga con condición inicial 0 [cm] overshoot detallado después de la primera perturbación.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

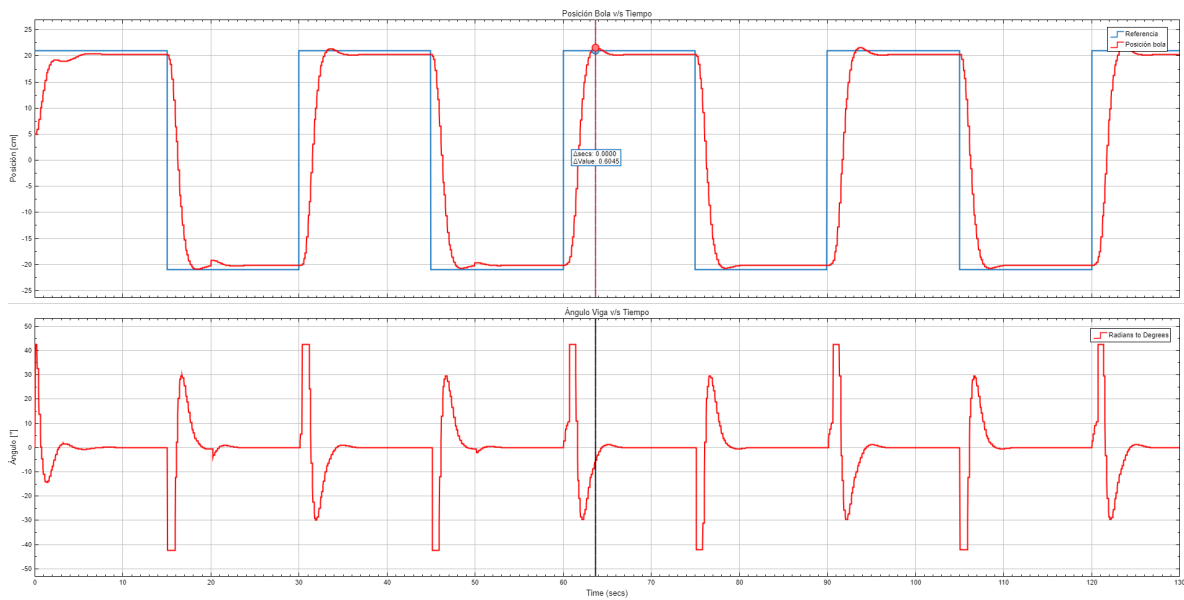


Fig. 4.29: Posición bola y ángulo viga con condición inicial 5 [cm] detallada después de la segunda perturbación.

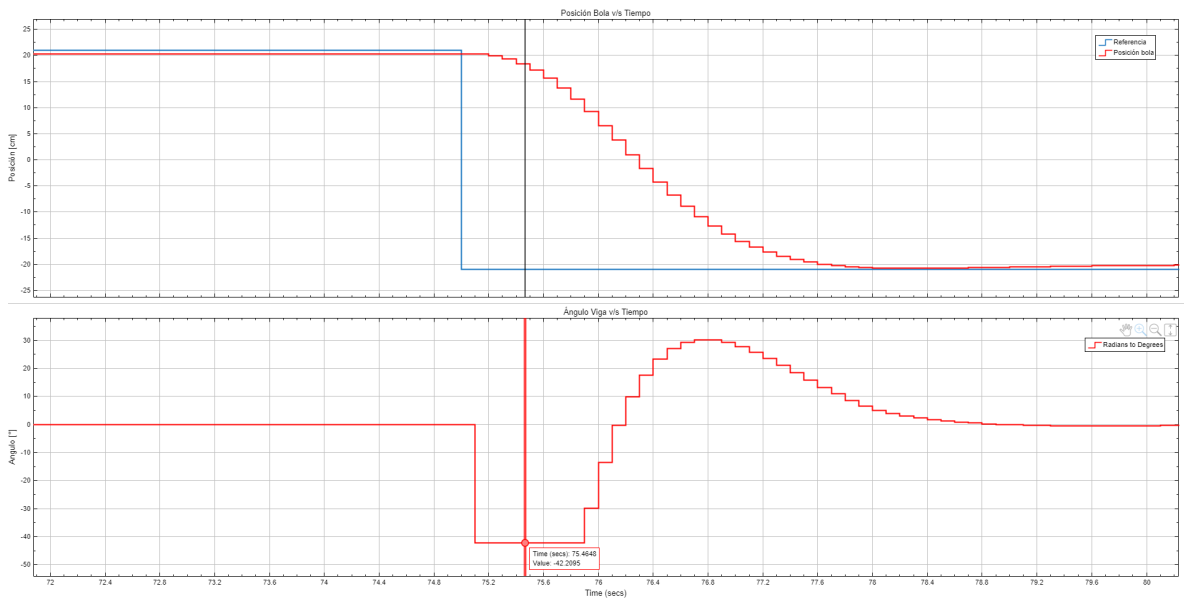


Fig. 4.30: Posición bola y ángulo viga detallada.

Las figuras [4.27](#), [4.28](#) y [4.29](#) muestran que el overshoot después de la primera perturbación cumple con el límite de seguridad de la viga, después del segundo overshoot no se llega a cumplir dicho límite llegando a un valor de 21,6045 [cm], también se puede observar que los ángulos a los que llega la viga están dentro de las condiciones pero presentan el mismo

4. SIMULACIÓN DEL SISTEMA DE CONTROL

problema que anteriormente se mostró, el cual nos da una justificación para implementar condiciones de slew rate para observar si los saltos de los ángulos no son tan agresivos.

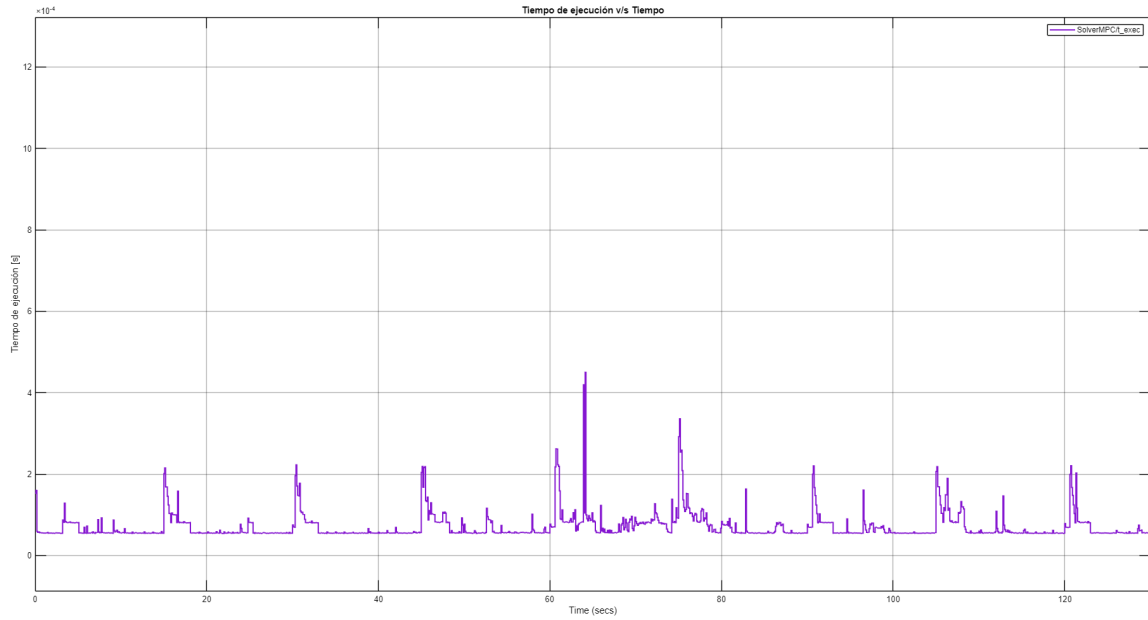


Fig. 4.31: Tiempo de ejecución del solver Quadprog.

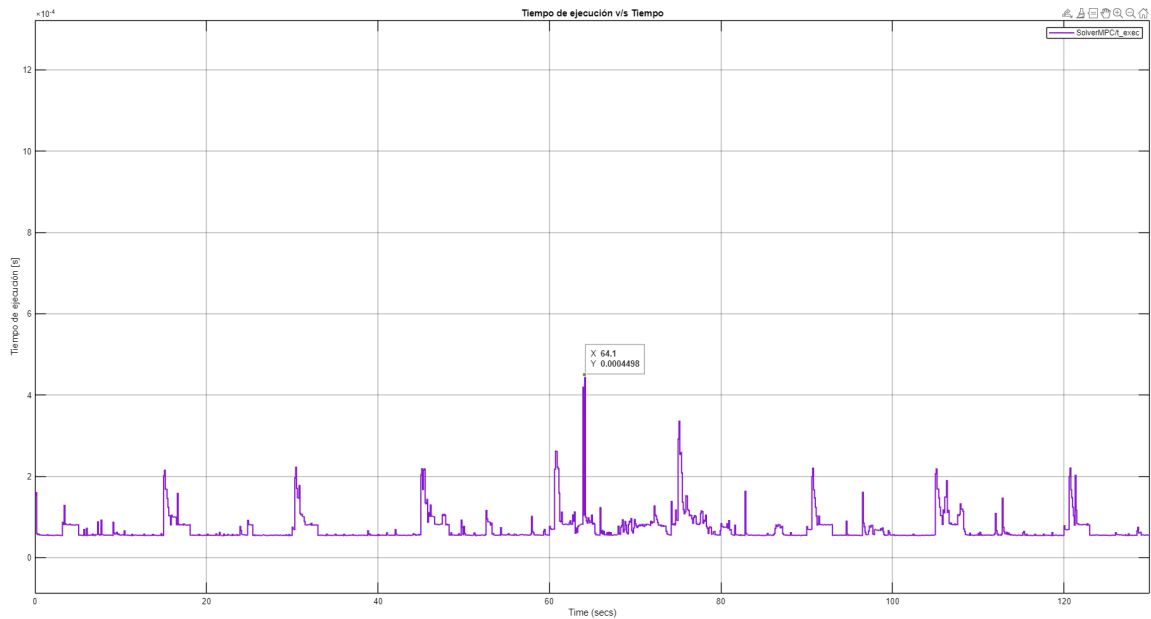


Fig. 4.32: Tiempo de ejecución del solver Quadprog detallado.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

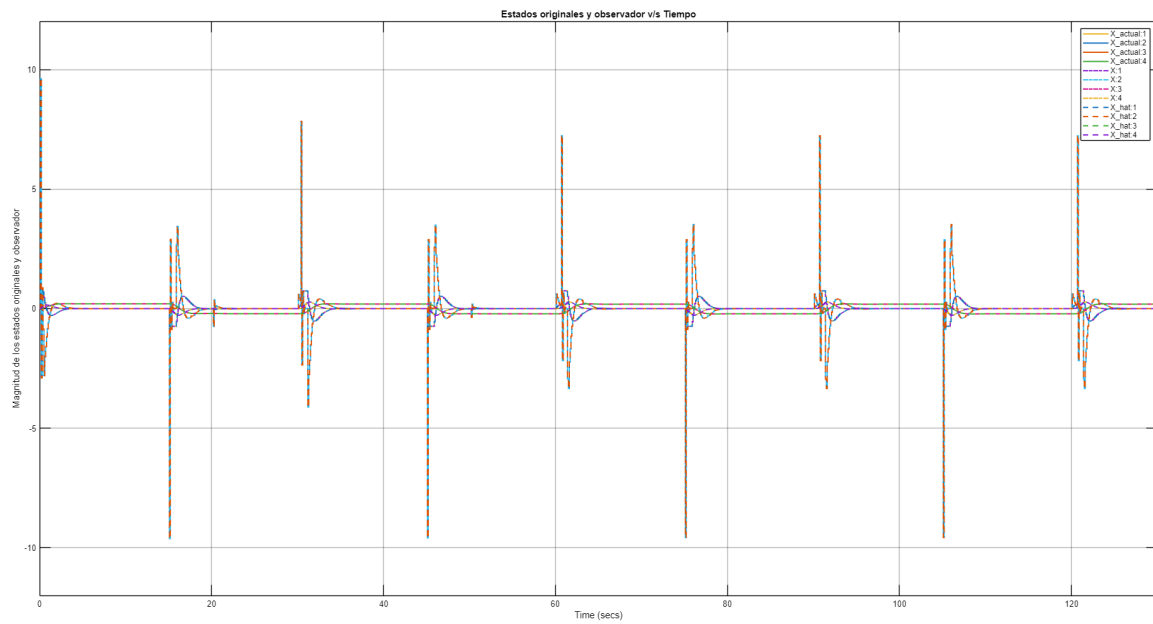


Fig. 4.33: Estados originales del sistema y de la planta aumentada.

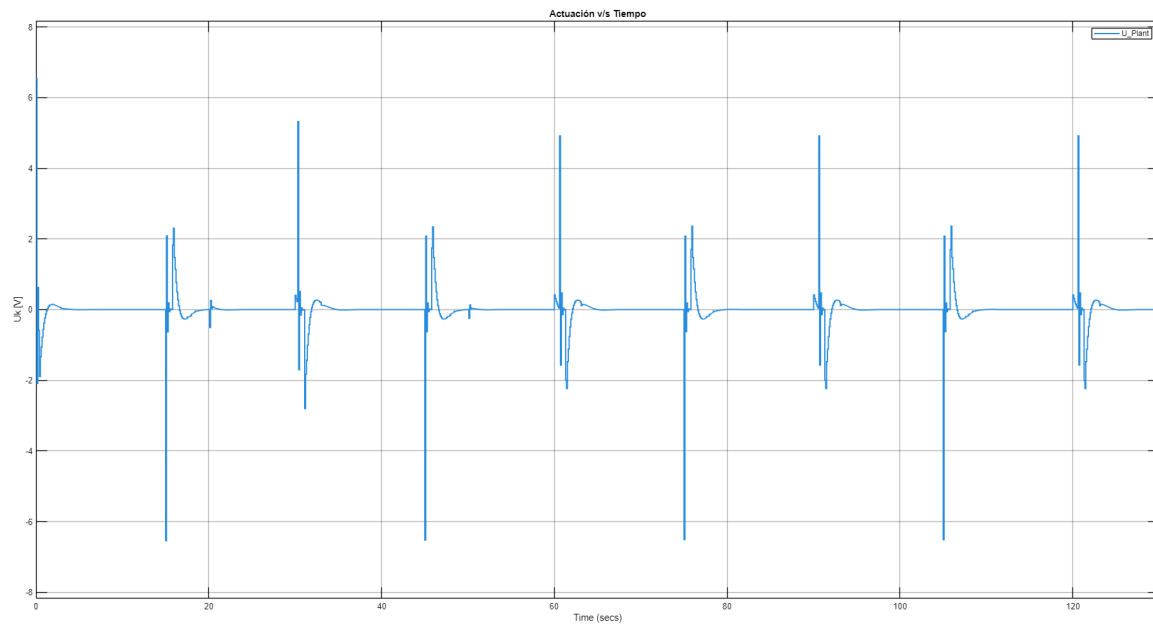


Fig. 4.34: Actuación de la planta con perturbaciones.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

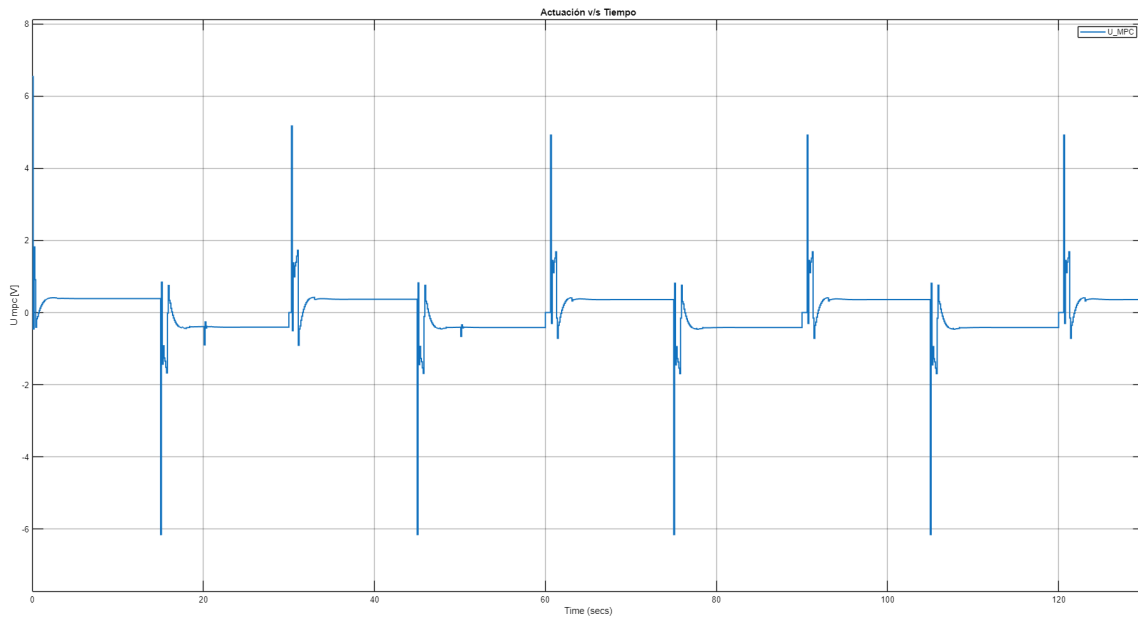


Fig. 4.35: Voltaje de salida MPC con perturbaciones.

4.1.7 Implementación de condiciones slew rate

En sistemas mecatrónicos reales, como la planta Ball and Beam, los actuadores presentan limitaciones físicas inherentes a su construcción y capacidad de torque. El servomotor SRV02 posee una velocidad angular máxima finita, lo que le impide responder ante variaciones instantáneas en la señal de mando.

Sin una restricción adecuada, un controlador MPC puede generar señales de control tipo escalón que exigen aceleraciones teóricamente infinitas. Como se evidenció en las pruebas experimentales previas, este comportamiento deriva en estrés mecánico severo sobre la caja de engranajes, picos de corriente en el motor y la excitación de dinámicas no modeladas que degradan la precisión del seguimiento.

Para garantizar una simulación lo mas cercana a la realidad con el propósito de obtener una operación segura y fluida, se implementa una restricción de Slew Rate. Esta se define como una restricción de desigualdad aplicada sobre la variación temporal de la acción de control entre dos instantes de muestreo consecutivos (k y $k - 1$), expresándose matemáticamente como:

$$|u(k) - u(k - 1)| \leq \Delta u_{max} \quad (4.8)$$

Donde Δu_{max} representa el incremento máximo permitido por el actuador en un periodo de

4. SIMULACIÓN DEL SISTEMA DE CONTROL

muestreo T_s .

Desde la perspectiva de la optimización, esta restricción no modifica la estructura de la función objetivo del controlador; en su lugar, actúa directamente sobre el espacio de búsqueda del problema de Programación Cuadrática QP. Al delimitar el área efectiva de soluciones factibles, se asegura que el voltaje enviado al motor sea siempre compatible con sus capacidades, protegiendo así la vida útil del hardware.

4.1.8 Preparación de la simulación

Para estas pruebas se usan los mismos valores definidos en las pruebas anteriores agregando un valor nuevo que es Δu_{max} , el cual tiene un valor de 5 [V] para las pruebas con slew rate.

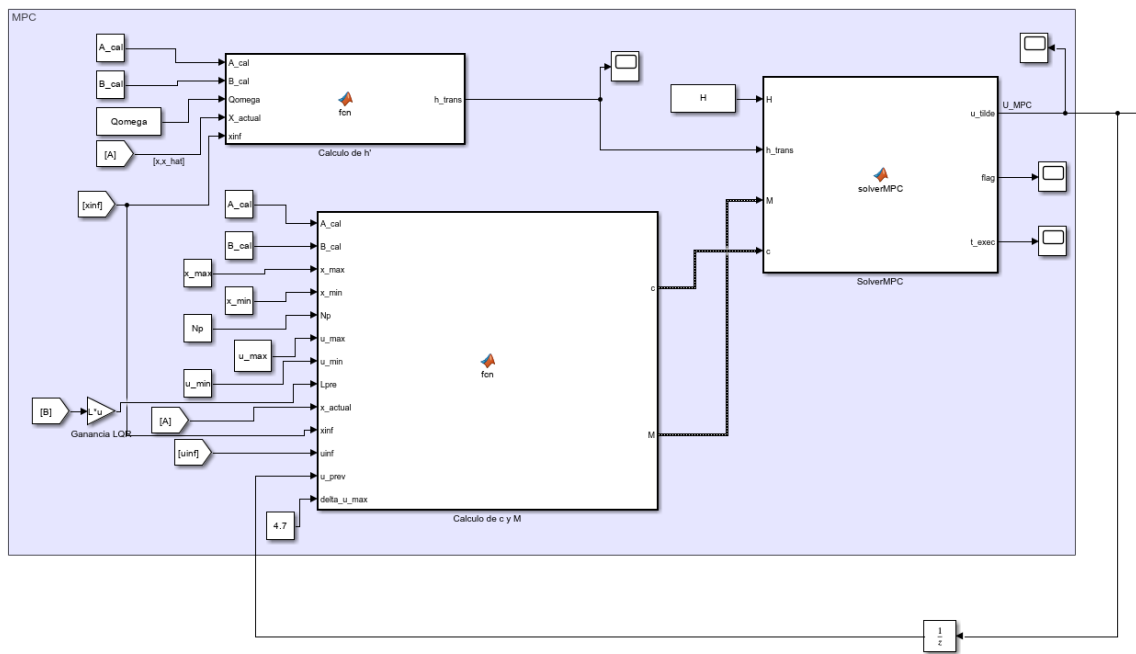


Fig. 4.36: Esquema de control MPC con Slew Rate.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

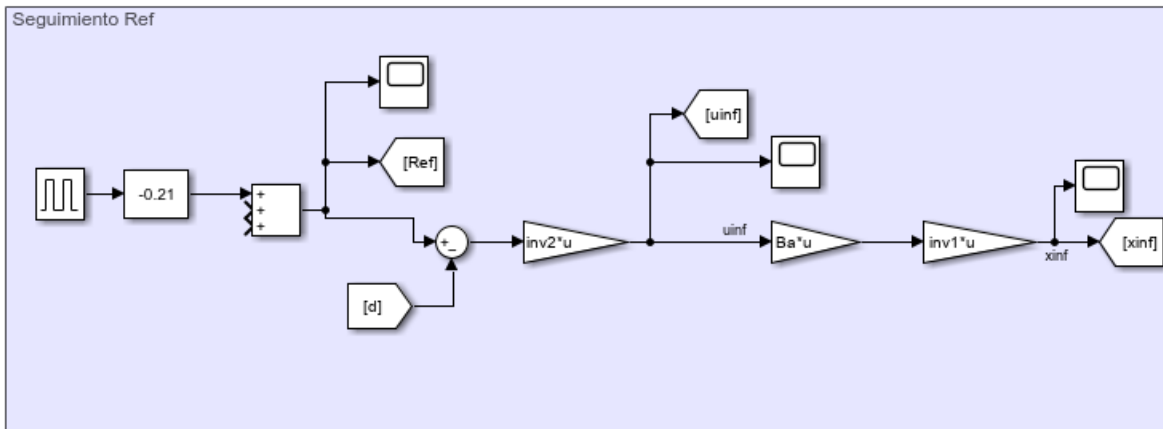


Fig. 4.37: Esquema seguimiento a referencias para las pruebas.

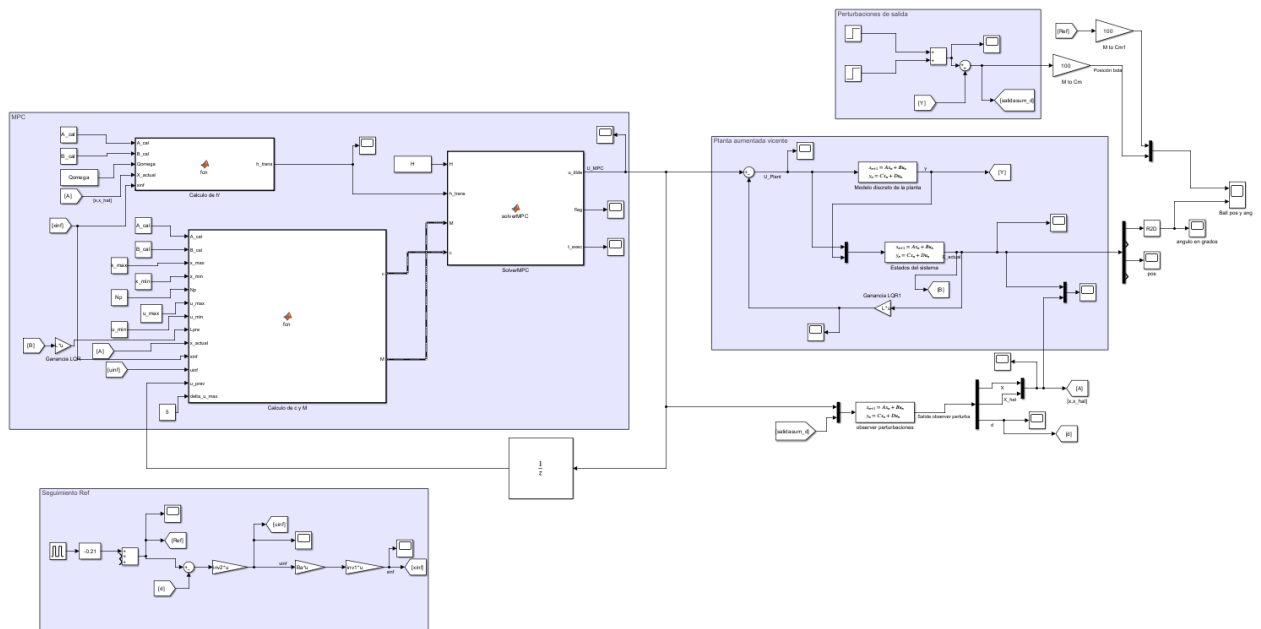


Fig. 4.38: Esquema final MPC con slew rate.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

4.1.9 Resultados de la implementación

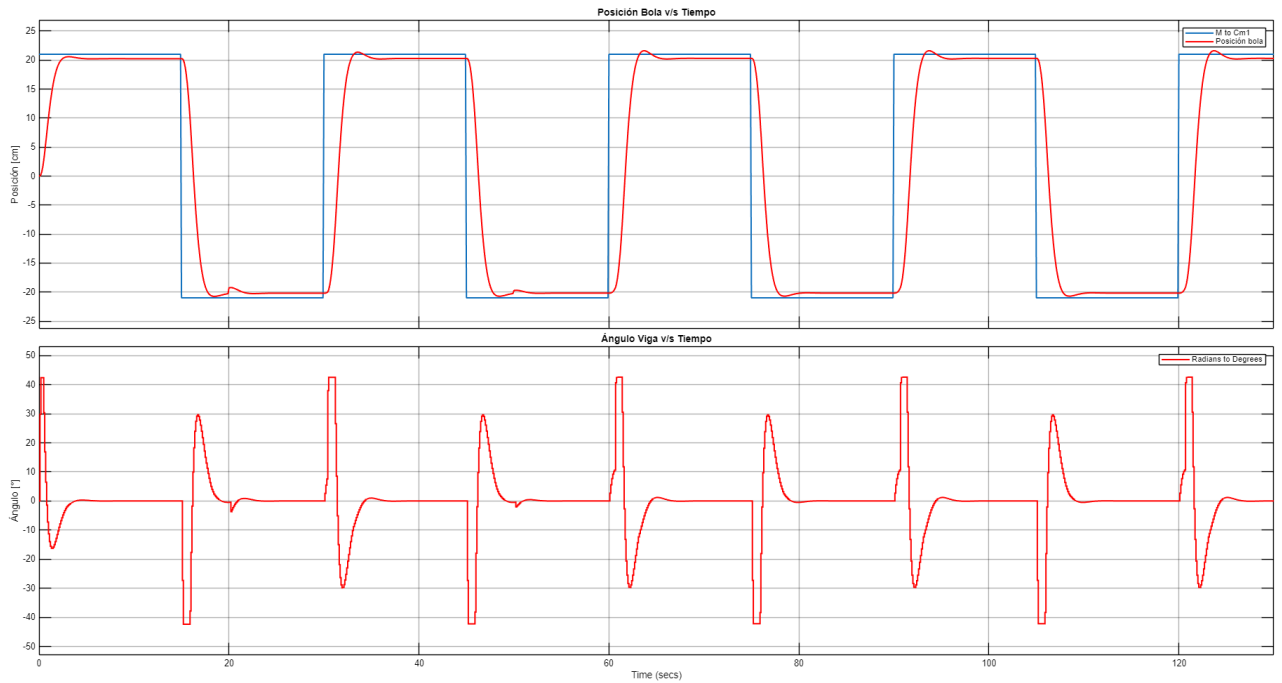


Fig. 4.39: Posición bola y ángulo viga condiciones iniciales 0 [cm] con slew rate.

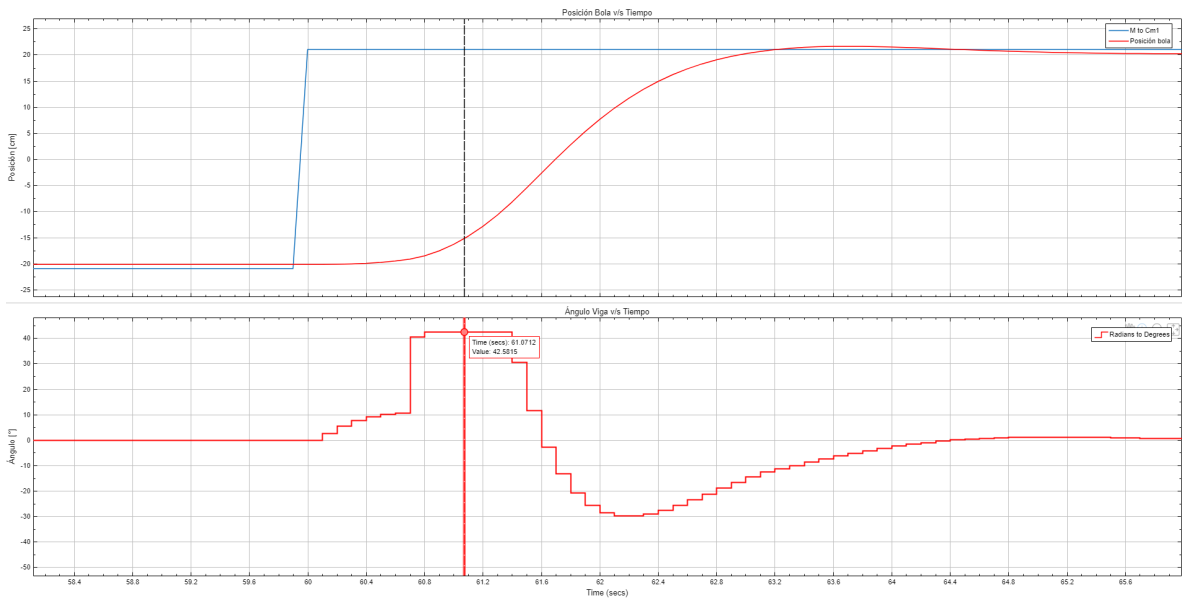


Fig. 4.40: Posición bola y ángulo viga condiciones iniciales 0 [cm] con slew rate detallado.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

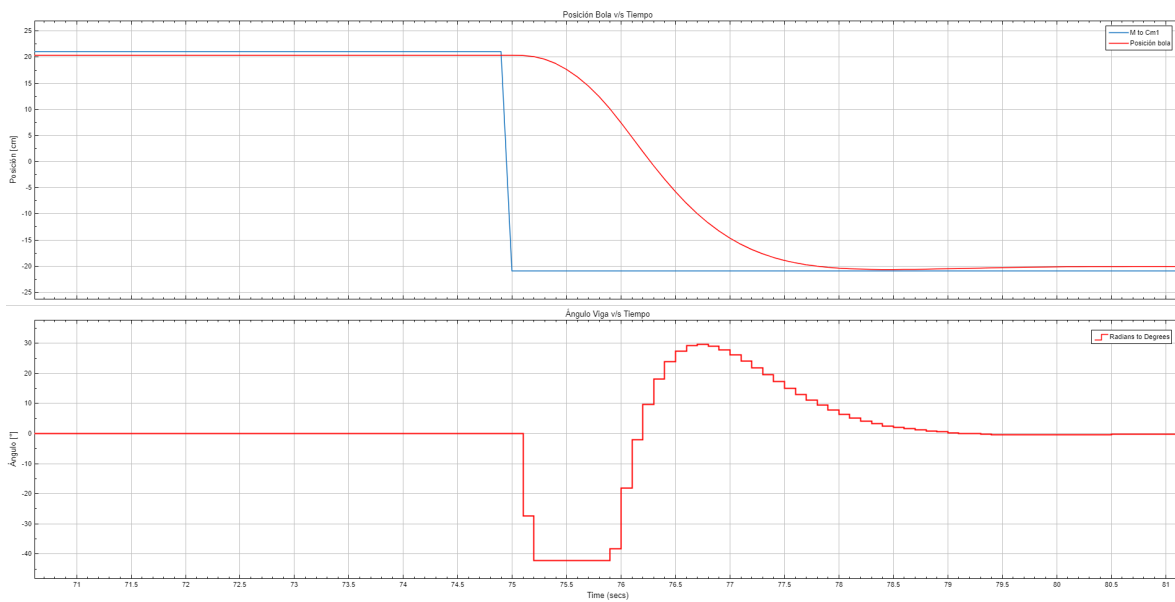


Fig. 4.41: Posición bola y ángulo viga condiciones iniciales 0 [cm] con slew rate detallado2.

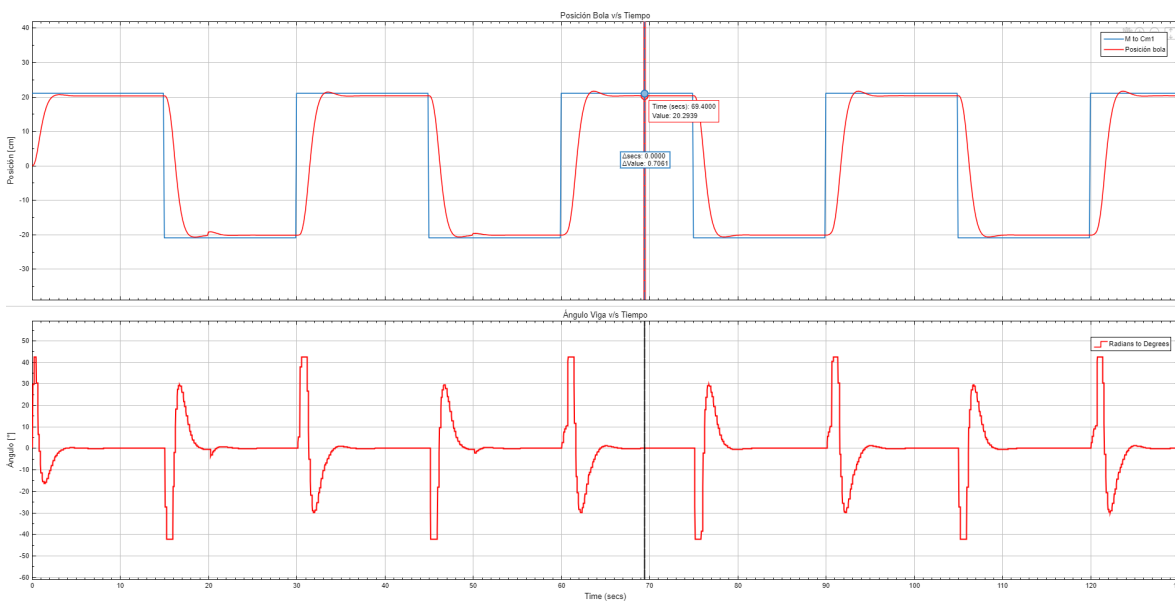


Fig. 4.42: Posición bola y ángulo viga seguimiento a referencia.

Se puede observar de las figuras [4.39](#), [4.40](#), [4.41](#) que los saltos en los ángulos de la viga al compararlos con los resultados anteriores vistos en [4.22](#) y [4.30](#) son de menor agresividad por lo que el servomotor no debería llegar a sufrir daños debido al cambio brusco en los ángulos. también al igual que en las figuras anteriormente comentadas se puede observar que en todo momento se cumple las restricciones con respecto al ángulo de la viga.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

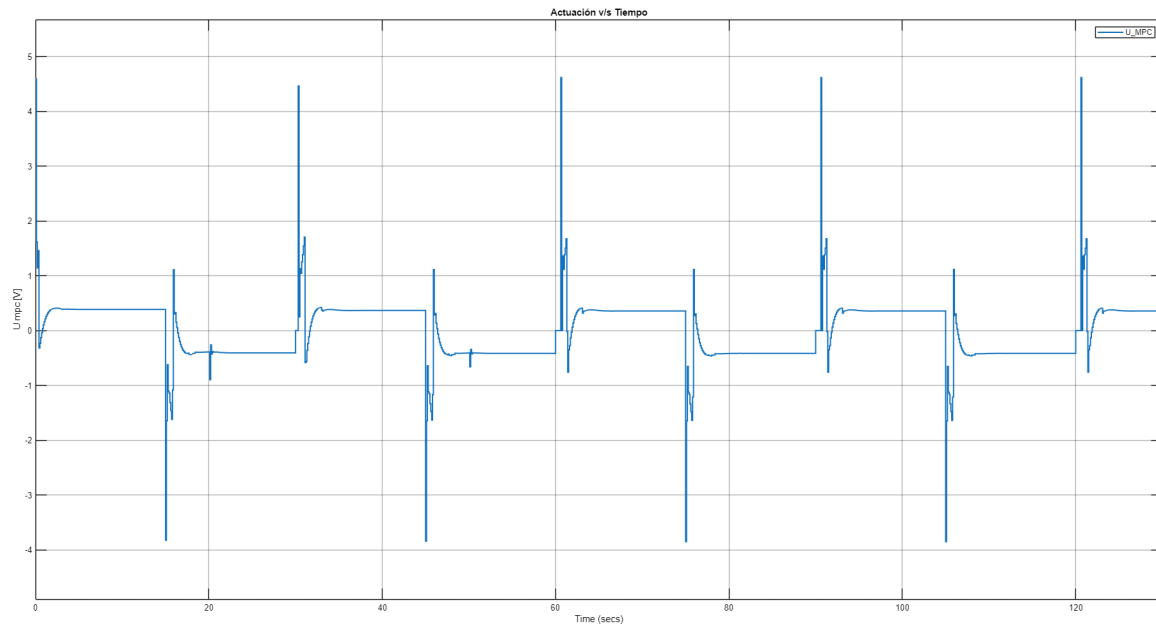


Fig. 4.43:
Voltaje de salida MPC con slew rate.

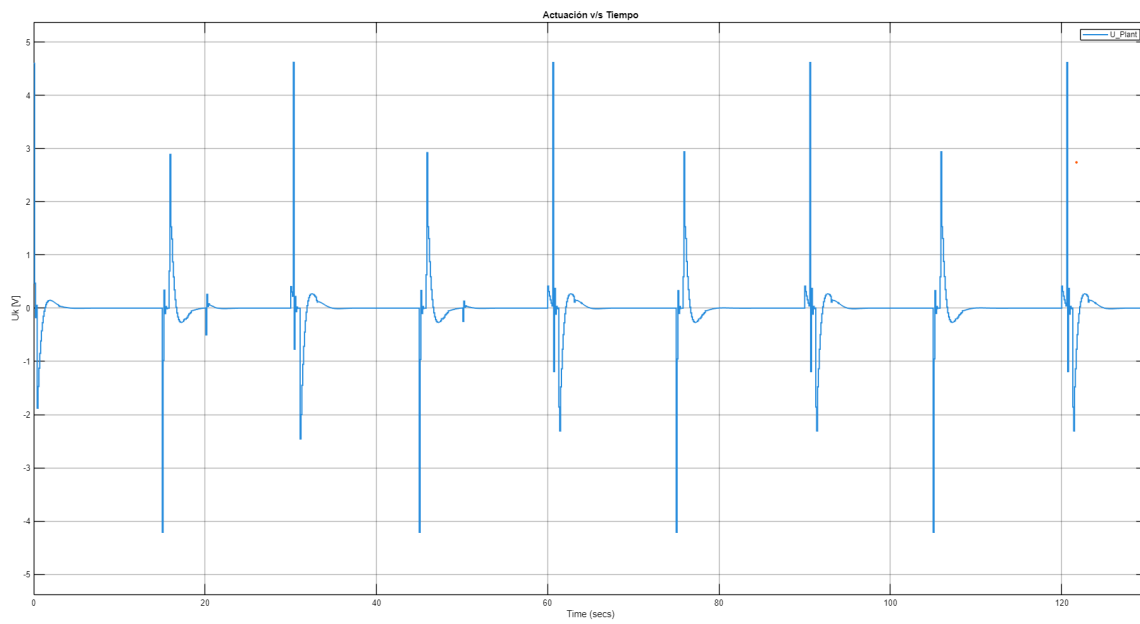


Fig. 4.44: Actuación de la planta con slew rate.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

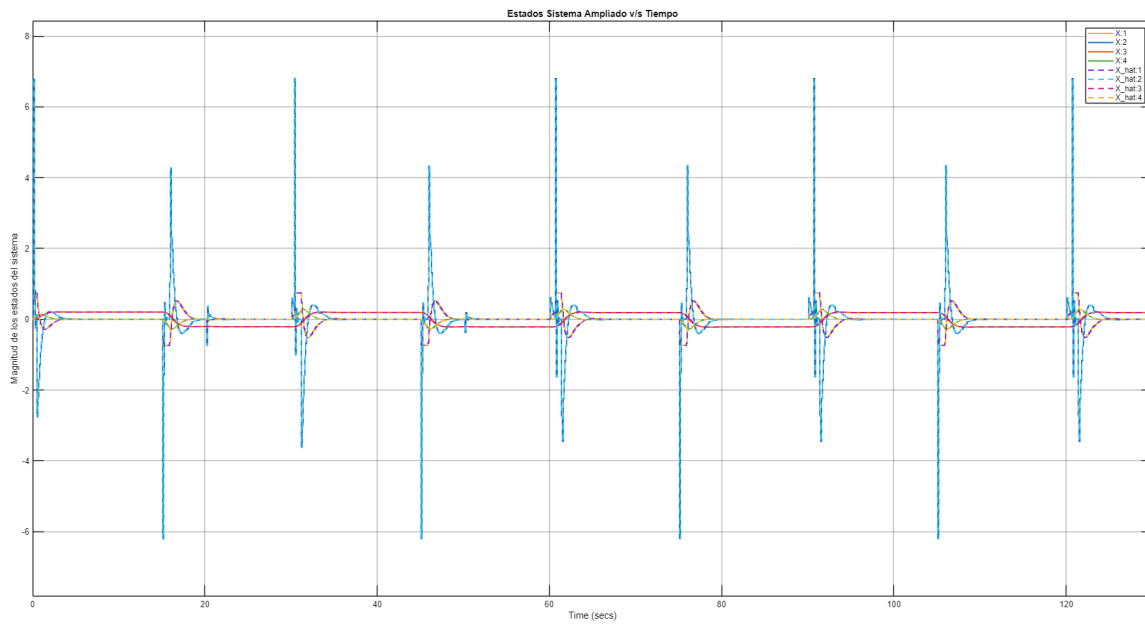


Fig. 4.45: Estados del sistema ampliado con slew rate.

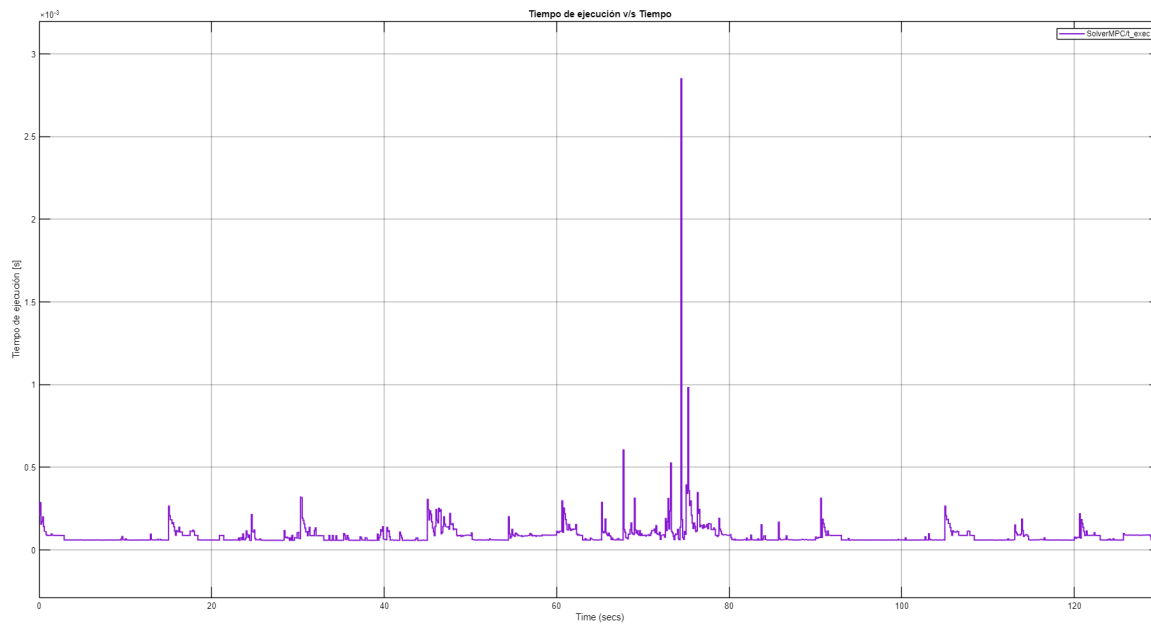


Fig. 4.46: Tiempo de ejecución solver Quadprog con slew rate.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

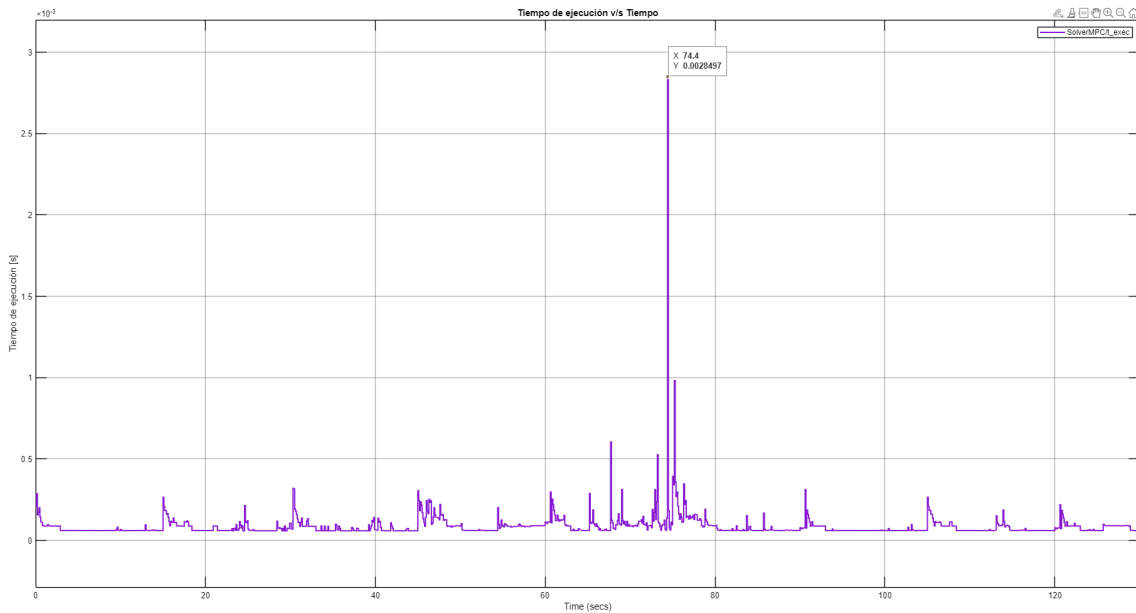


Fig. 4.47: Tiempo de ejecución solver Quadprog con slew rate detallado.

La figura 4.47 presenta que en el peor de los casos el solver de Quadprog se demora $2,8497 \times 10^{-3}$ [s] valor que es mucho menor que al periodo de muestreo, lo que indica la factibilidad del método propuesto utilizado.

Con el propósito de mejorar el seguimiento de la referencia, se buscan nuevos valores para la matriz de penalización del MPC. Se prioriza el ajuste del séptimo estado, el cual representa el error de posición, al incrementar este valor, la diferencia entre la referencia y la posición real se reduce.

Manteniendo los valores de $N_p = 12$, $\Gamma = 0,3$. La nueva matriz Ω queda de la siguiente manera:

$$\Omega = \begin{bmatrix} 1e6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1e2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2e4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2e6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1e2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1e1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0,7e8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1e1 \end{bmatrix} \quad (4.9)$$

Para estas pruebas también se deja solo la perturbación de 1 [cm] la cual se activa a los 20 [s] obteniendo los siguientes resultados:

4. SIMULACIÓN DEL SISTEMA DE CONTROL

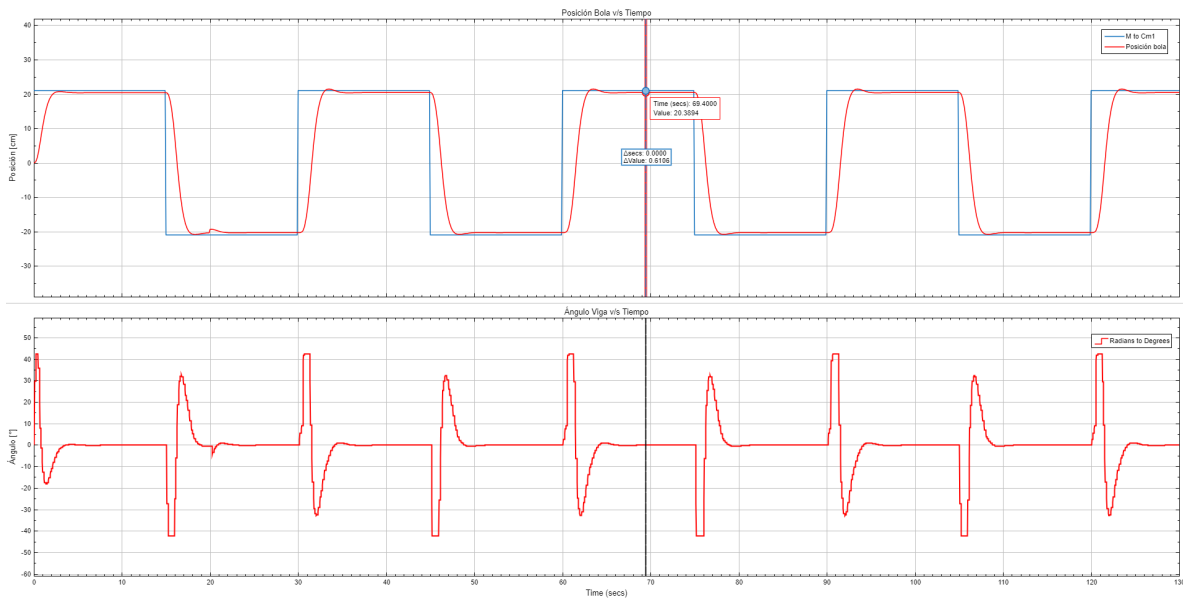


Fig. 4.48: Posición bola y ángulo viga nuevos valores.

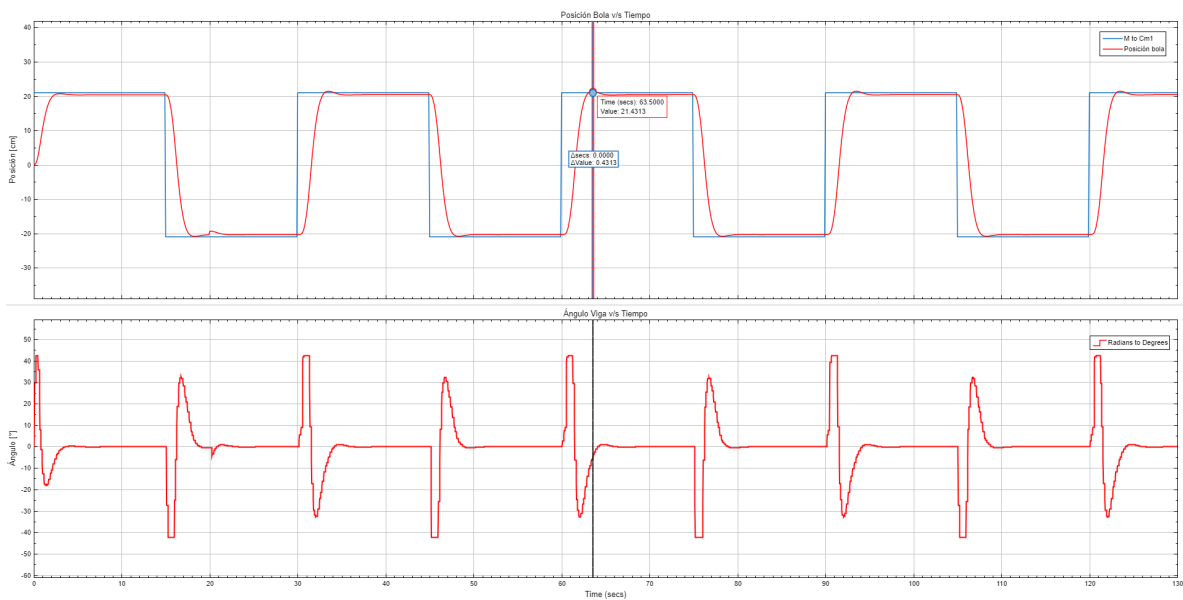


Fig. 4.49: Overshoot posición bola y ángulo viga nuevos valores.

Si comparamos los resultados obtenidos en [4.48](#), [4.49](#) y [4.28](#) se puede observar que existe una relación directa entre el overshoot estacionario y la precisión de la referencia. En donde a mayor precisión mayor es el overshoot presentado.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

Esto se puede confirmar en la siguiente figura donde se sube a $1e8$ el valor del estado siete de la matriz Ω

$$\Omega = \begin{bmatrix} 1e6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1e2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2e4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2e6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1e2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1e1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1e8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1e1 \end{bmatrix} \quad (4.10)$$

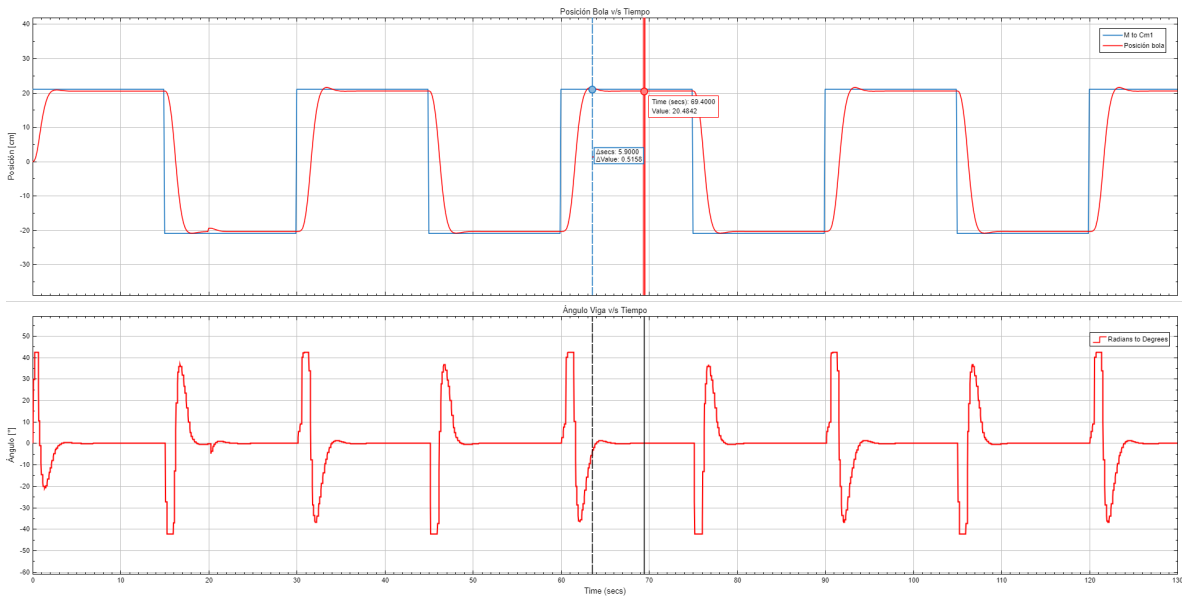


Fig. 4.50: Posición bola y ángulo viga valor $1e8$.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

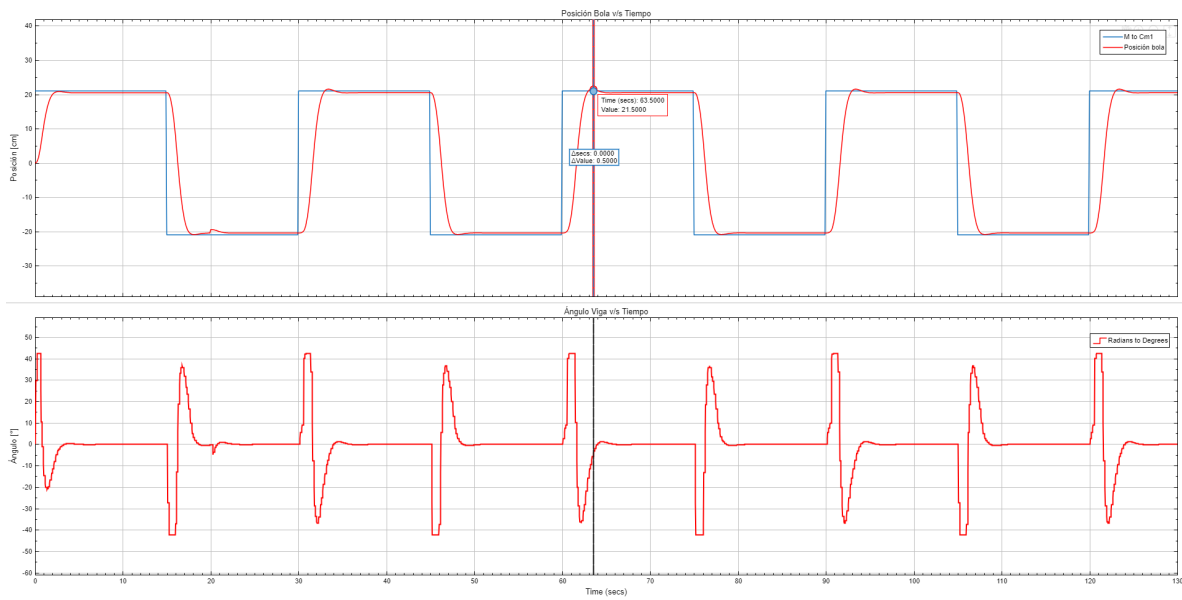


Fig. 4.51: Overshoot posición bola y ángulo viga valor $1e8$.

Donde se puede observar que el cambio de 0.7 a 1 es suficiente para llegar con un overshoot al límite de la zona de seguridad de la viga.

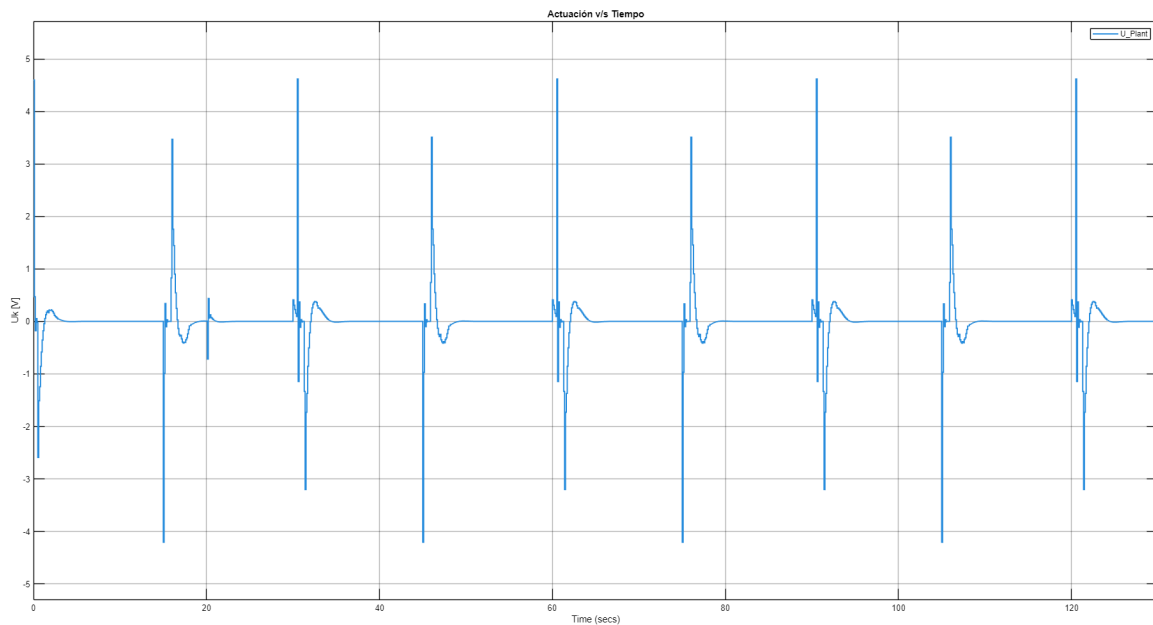


Fig. 4.52: Actuación de la planta con slew rate caso 7mo estado valor $1e8$.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

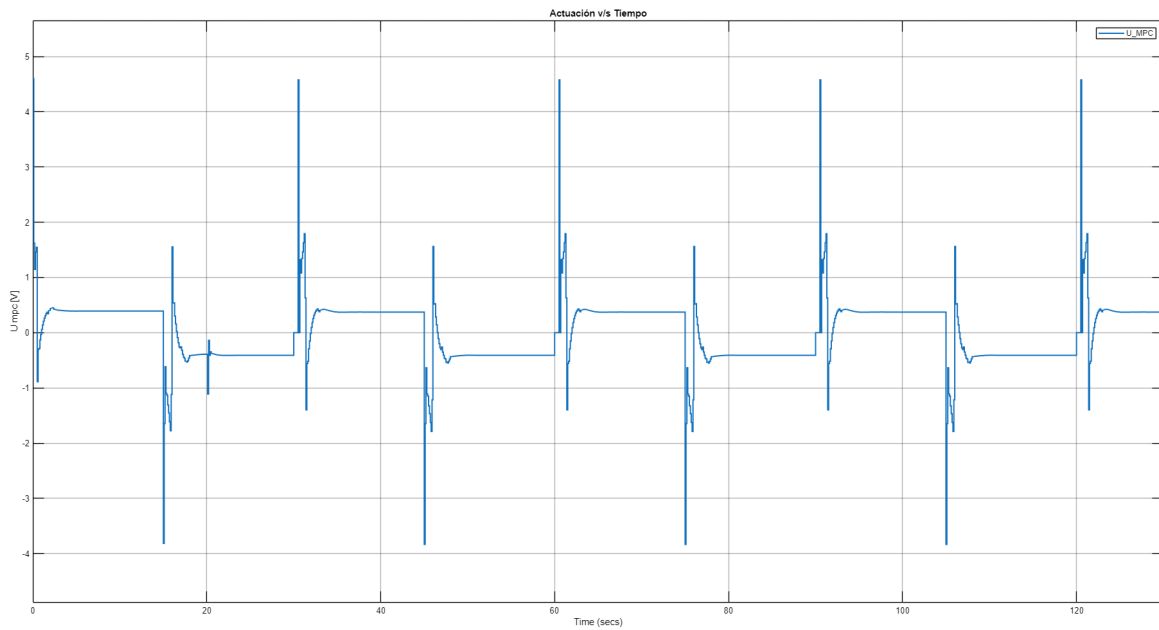


Fig. 4.53: Voltaje de salida MPC con slew rate caso: 7mo estado valor $1e8$.

Con el propósito de ver como se comporta el control se procede a variar el horizonte de predicción y estos son los resultados:

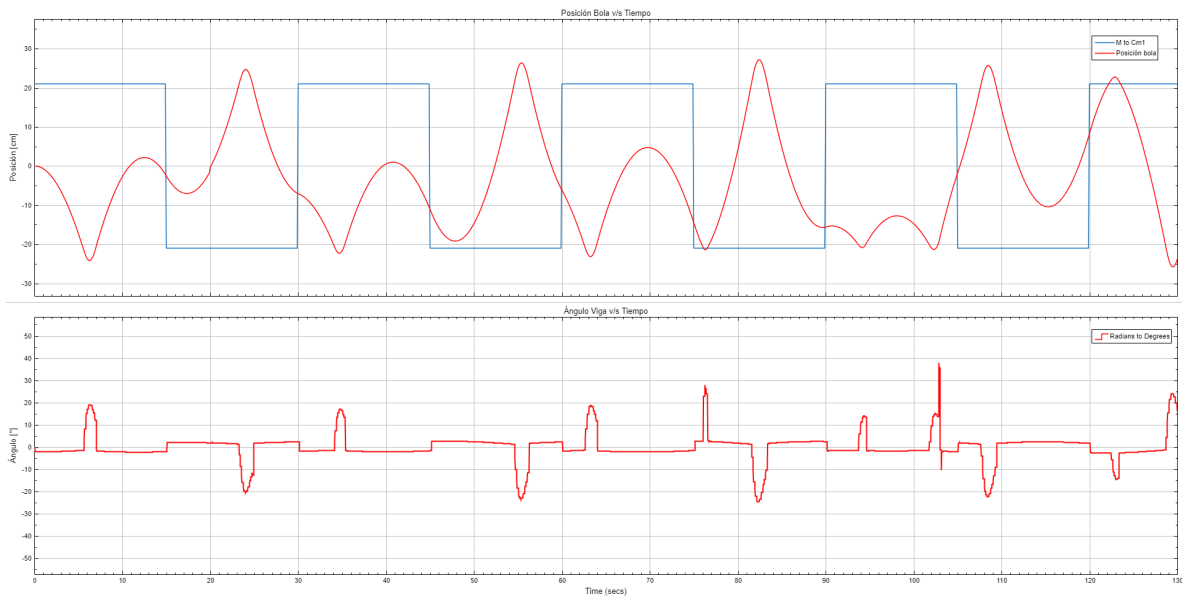


Fig. 4.54: Posición de la bola con $N_p=1$.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

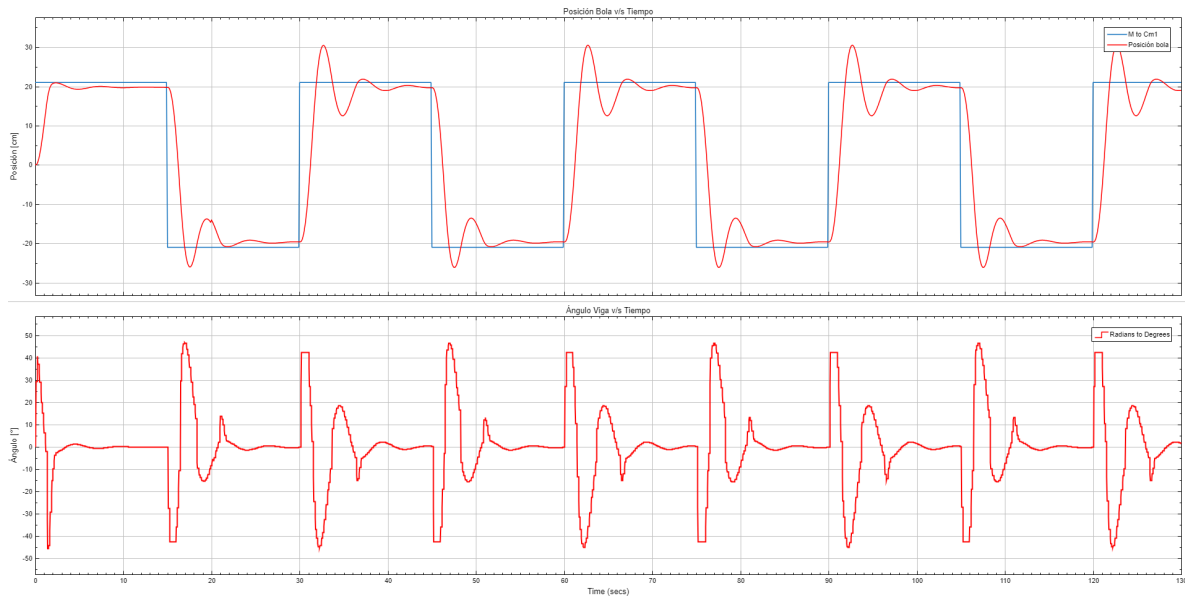


Fig. 4.55: Posición de la bola con $N_p=5$.

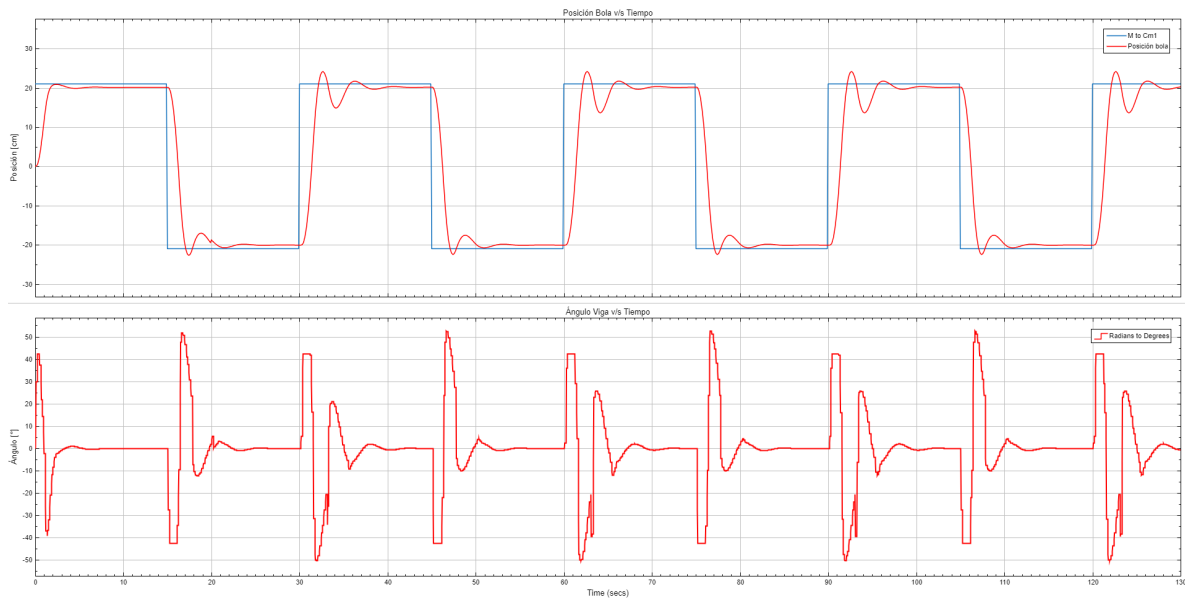


Fig. 4.56: Posición de la bola con $N_p=6$.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

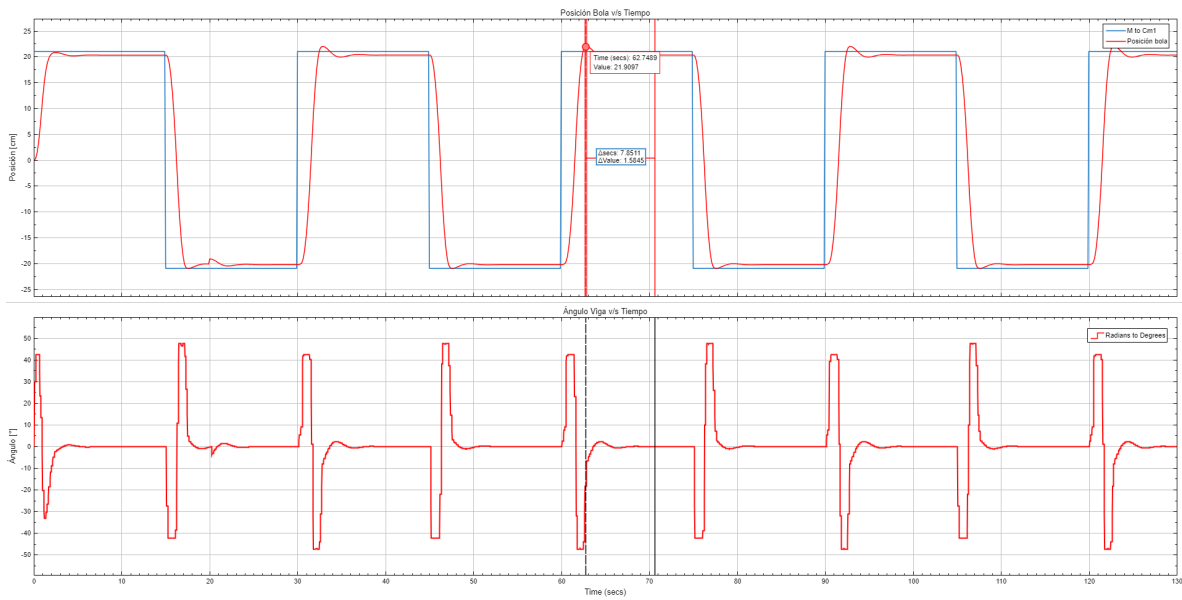


Fig. 4.57: Overshoot estacionario en la posición de la bola con $N_p=7$.

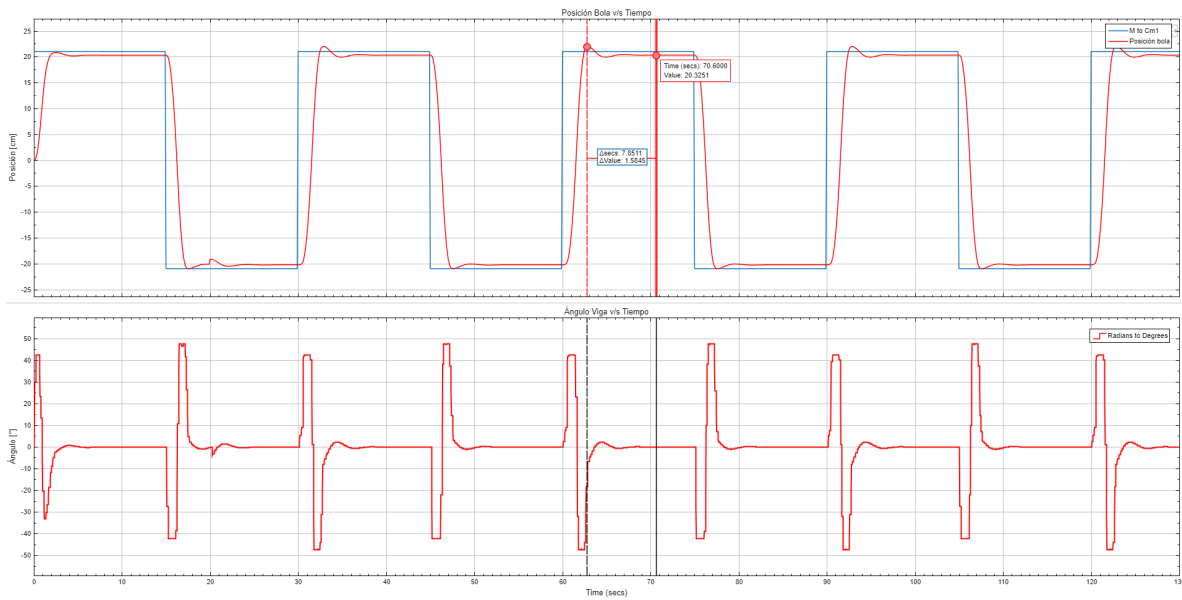


Fig. 4.58: Referencia alcanzada en la posición de la bola con $N_p=7$.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

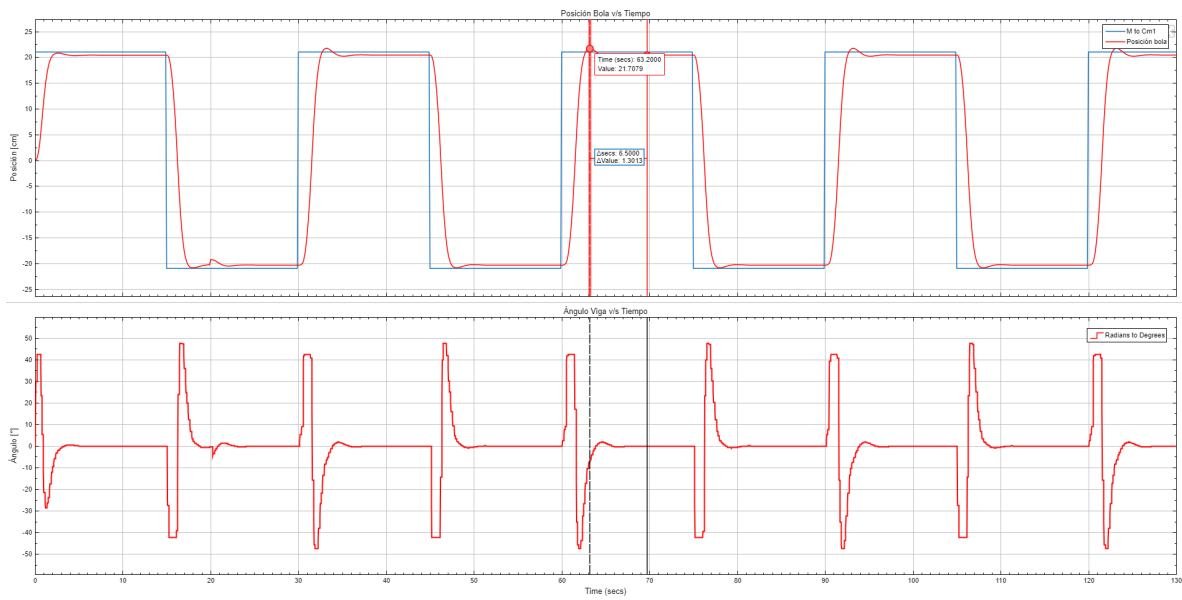


Fig. 4.59: Overshoot estacionario en la posición de la bola con $N_p=8$.

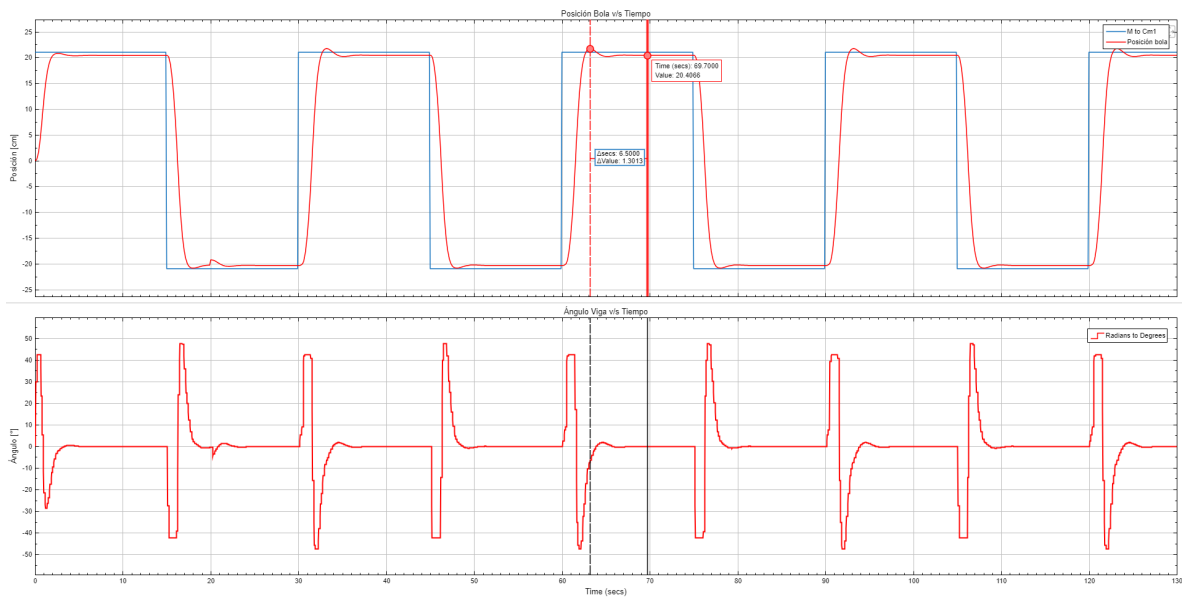


Fig. 4.60: Referencia alcanzada en la posición de la bola con $N_p=8$.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

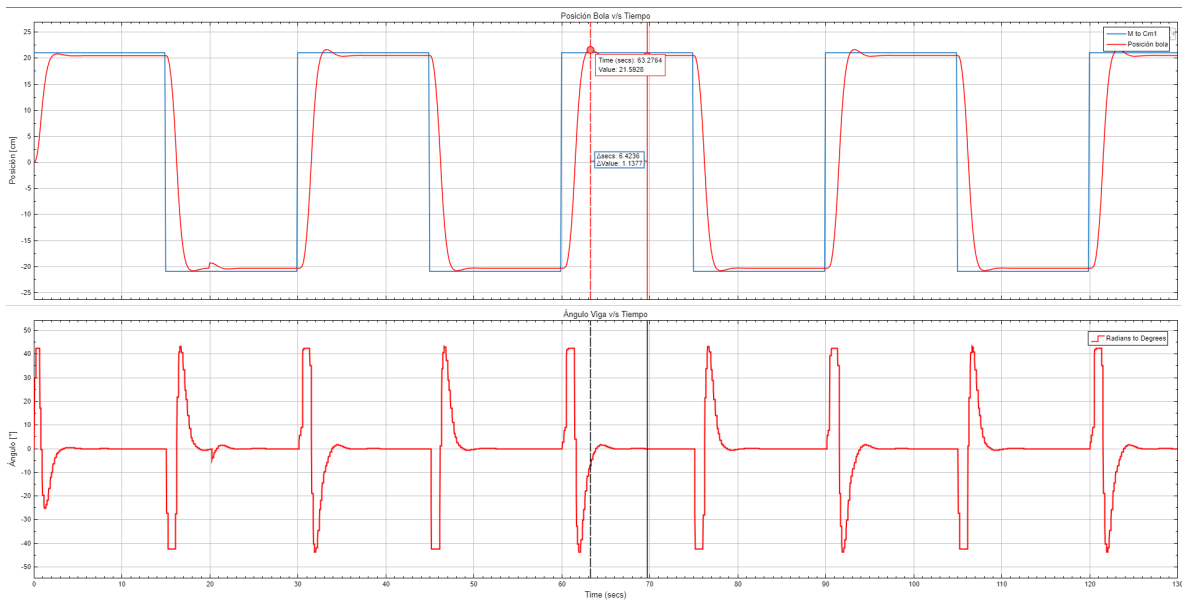


Fig. 4.61: Overshoot estacionario en la posición de la bola con $N_p=9$.

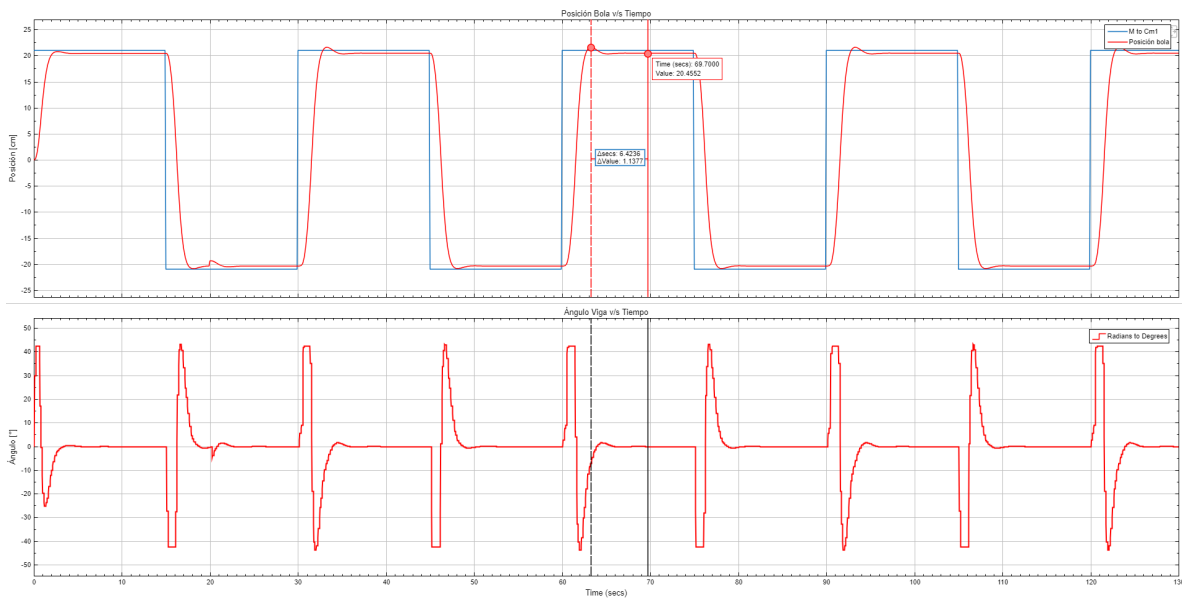


Fig. 4.62: Referencia alcanzada en la posición de la bola con $N_p=9$.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

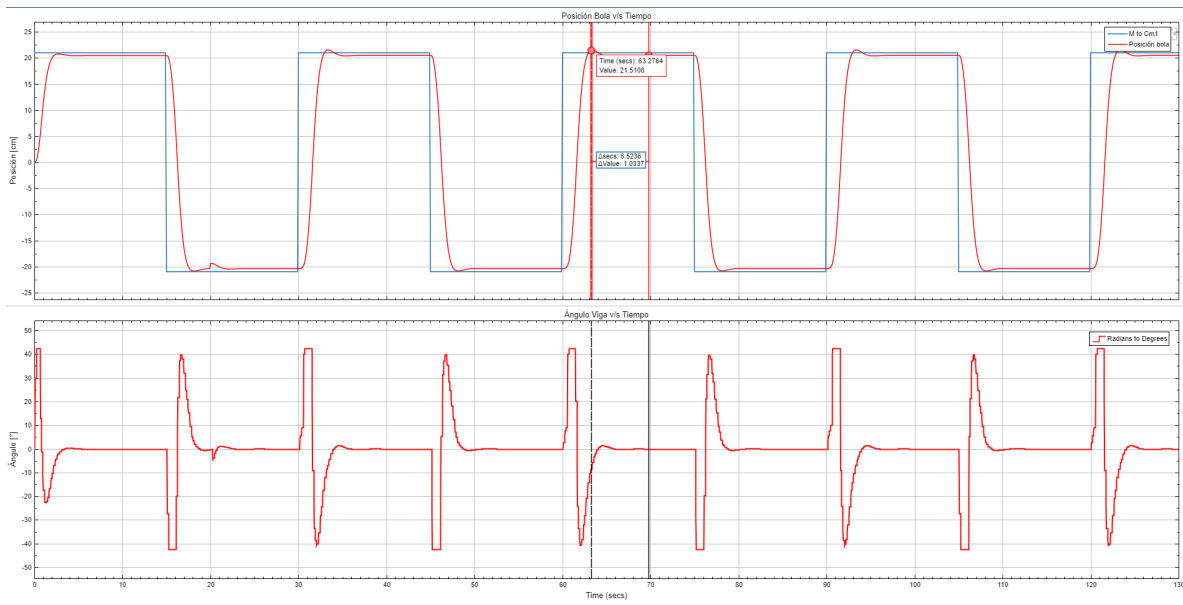


Fig. 4.63: Overshoot estacionario en la posición de la bola con $N_p=10$.

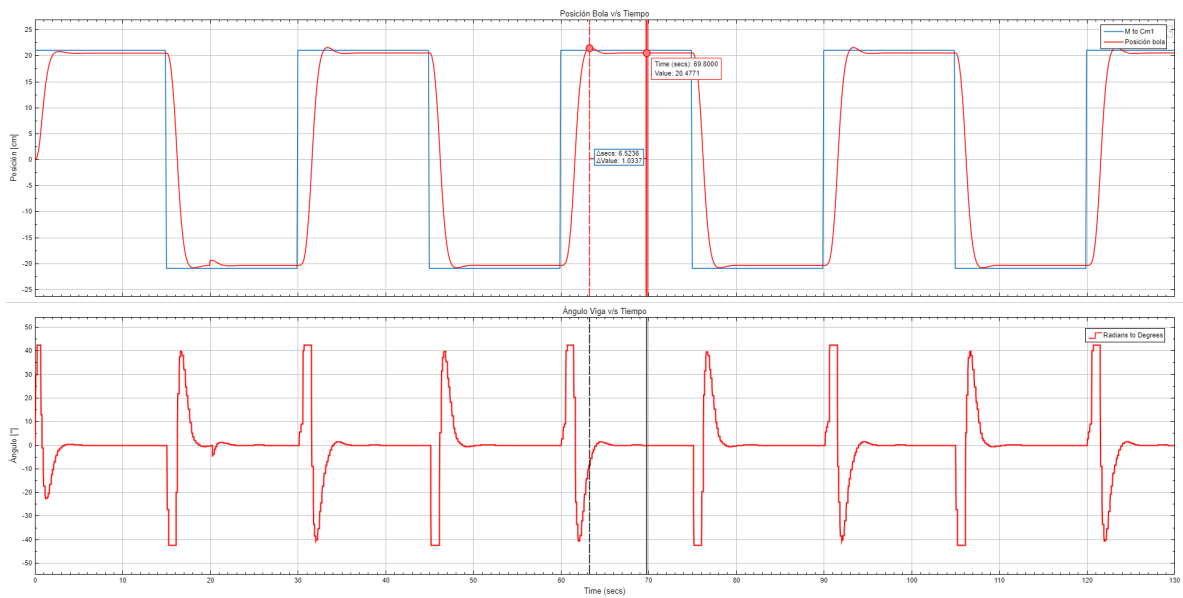


Fig. 4.64: Referencia alcanzada en la posición de la bola con $N_p=10$.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

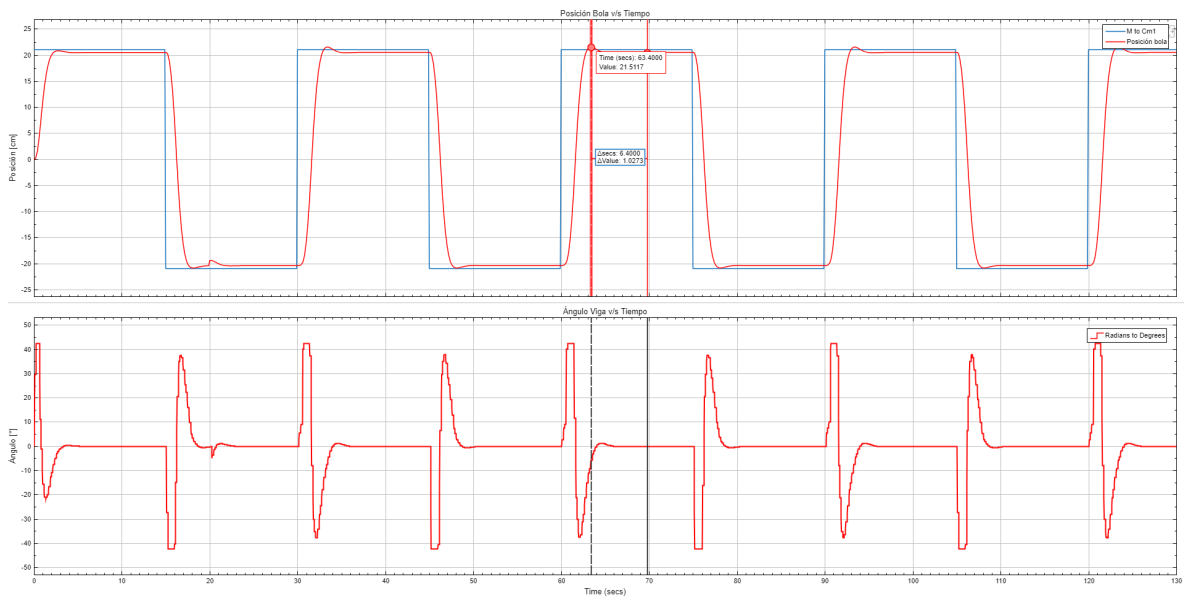


Fig. 4.65: Overshoot estacionario en la posición de la bola con $N_p=11$.

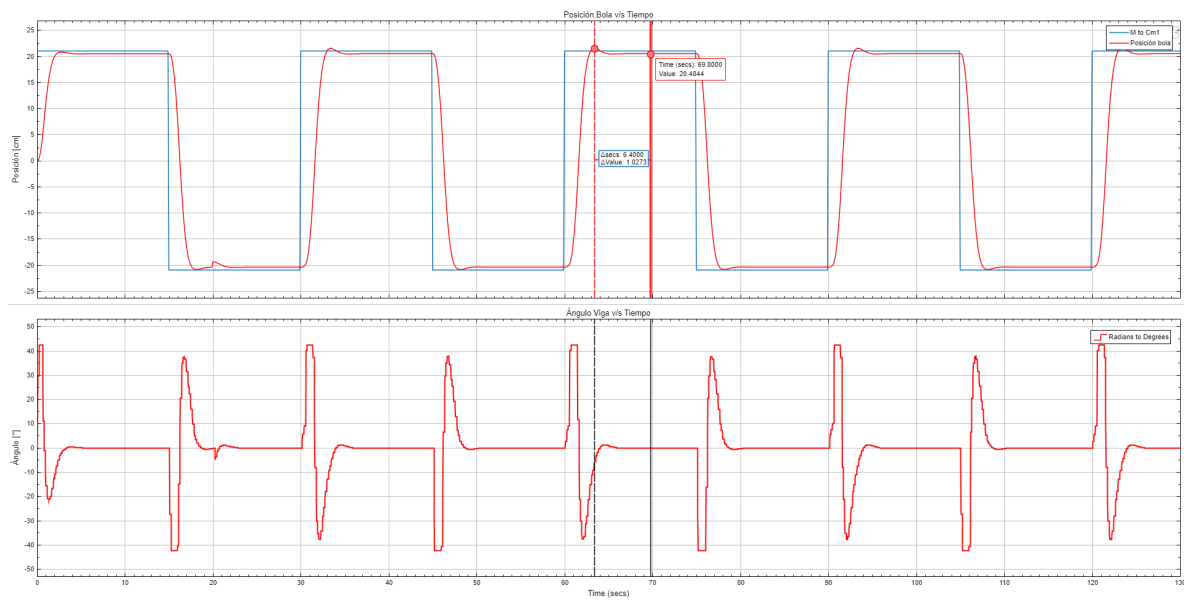


Fig. 4.66: Referencia alcanzada en la posición de la bola con $N_p=11$.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

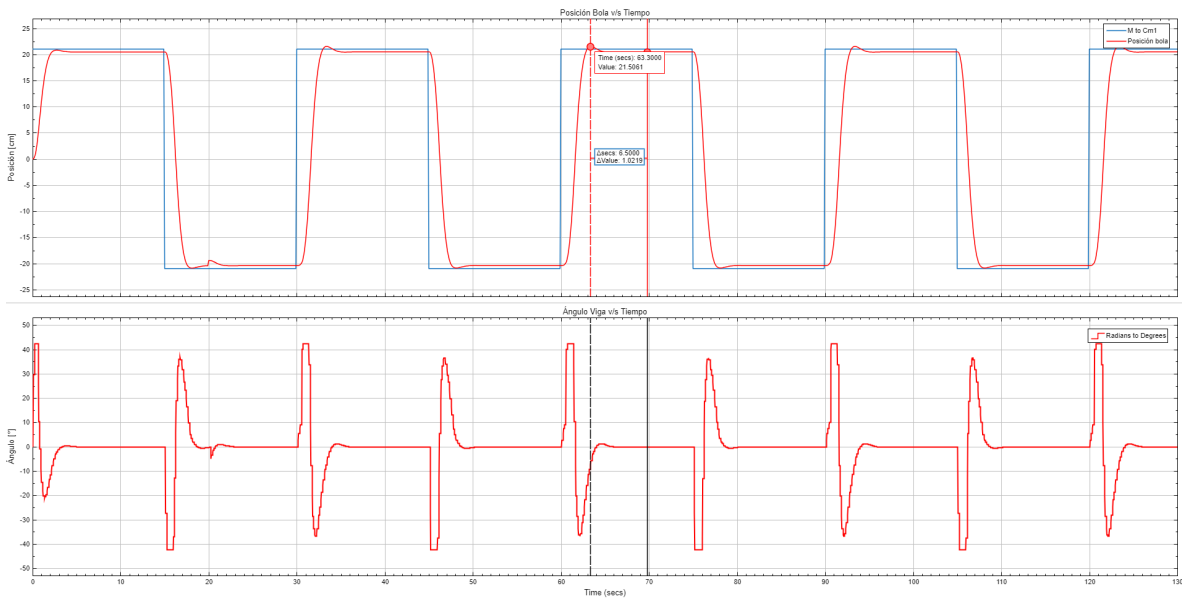


Fig. 4.67: Overshoot estacionario en la posición de la bola con $N_p=12$.

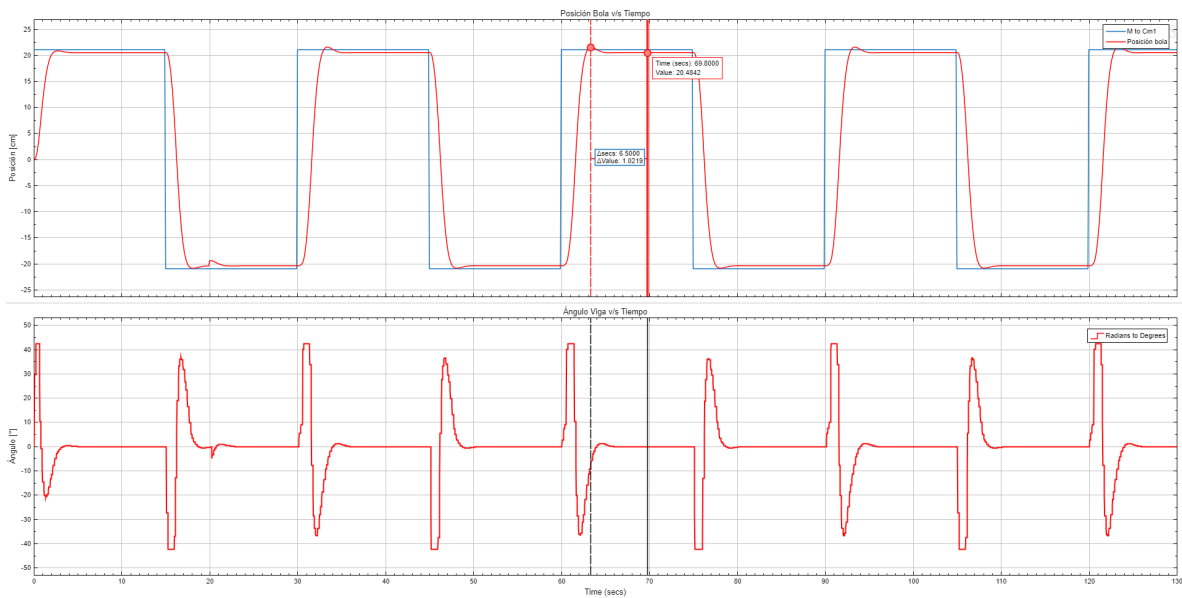


Fig. 4.68: Referencia alcanzada en la posición de la bola con $N_p=12$.

También con el propósito de analizar más en detalle los tiempos de computo del solver se varió el séptimo término de la matriz omega los resultados se pueden observar en la tabla [4.6](#). Por último se varió la restricción de actuación para observar como se comporta tanto la actuación que le entra a la planta como el funcionamiento del slew rate.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

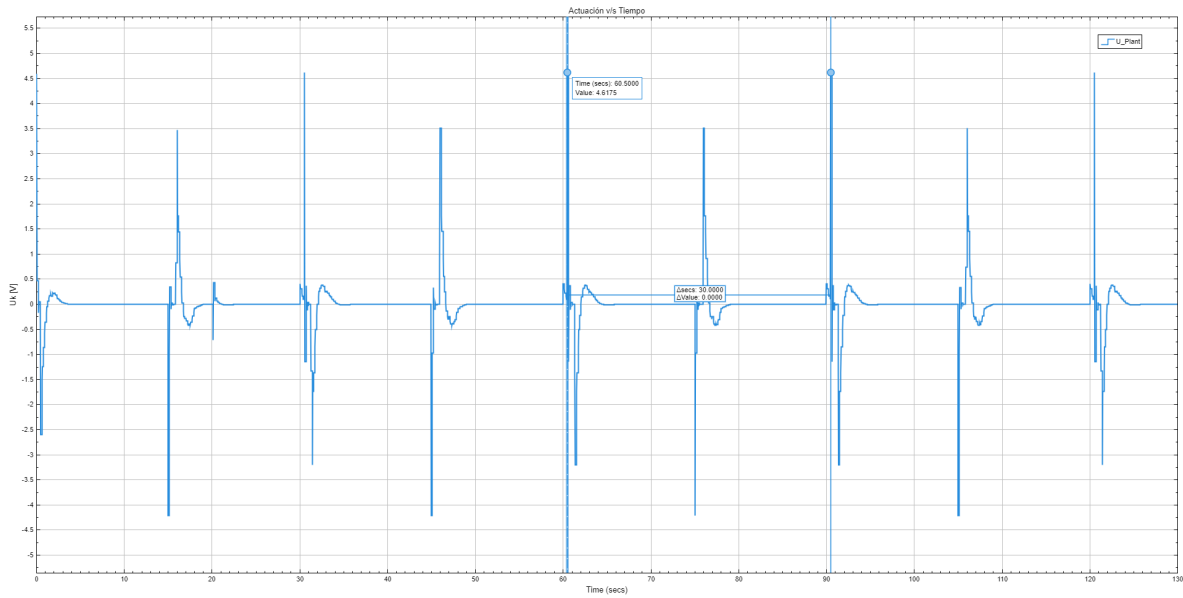


Fig. 4.69: Actuación de la planta ± 10 .

Para los límites de control en el intervalo $u \in [-9, 9]$ hasta $u \in [-5, 5]$, se observa que el sistema se comporta de la misma forma que en [4.69](#).

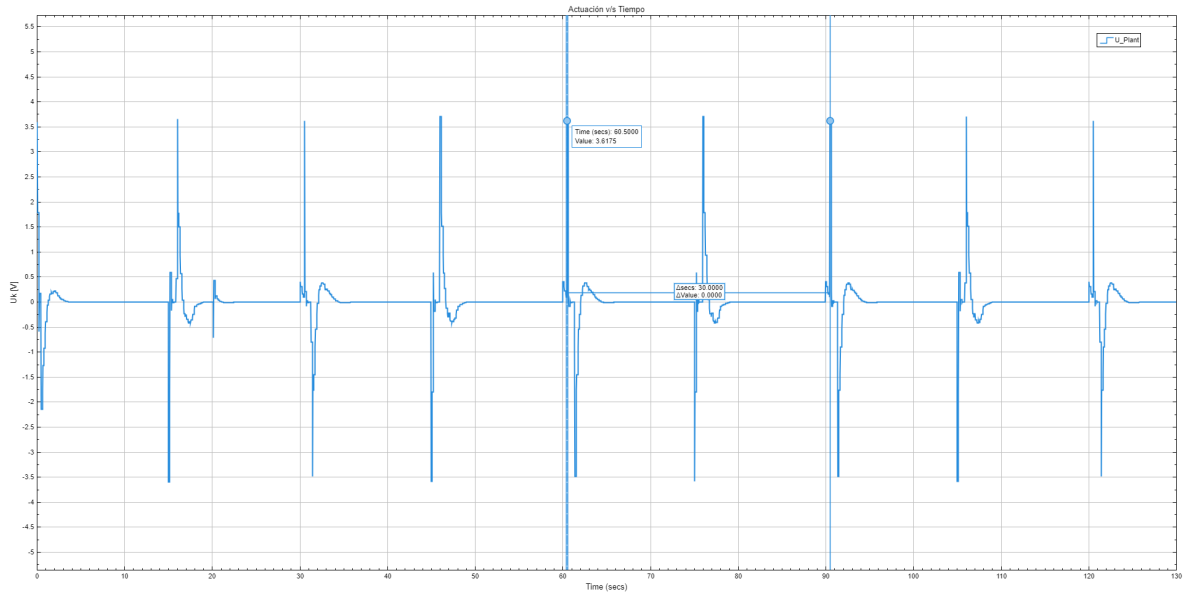


Fig. 4.70: Actuación de la planta ± 4 .

4. SIMULACIÓN DEL SISTEMA DE CONTROL

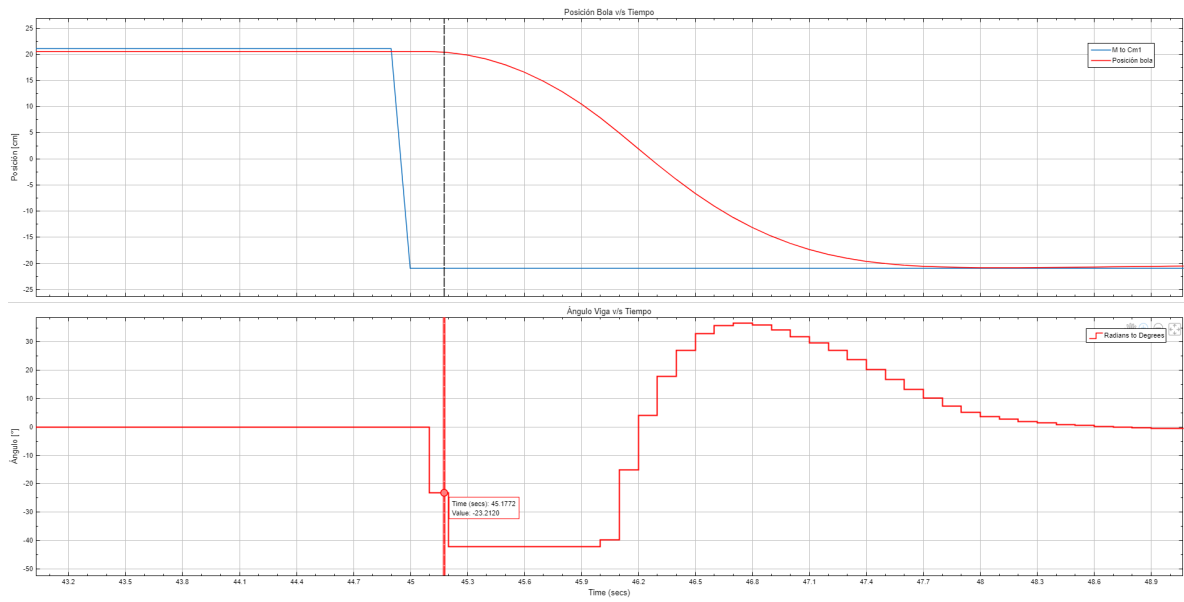


Fig. 4.71: Valor ángulo actuación ± 4 .

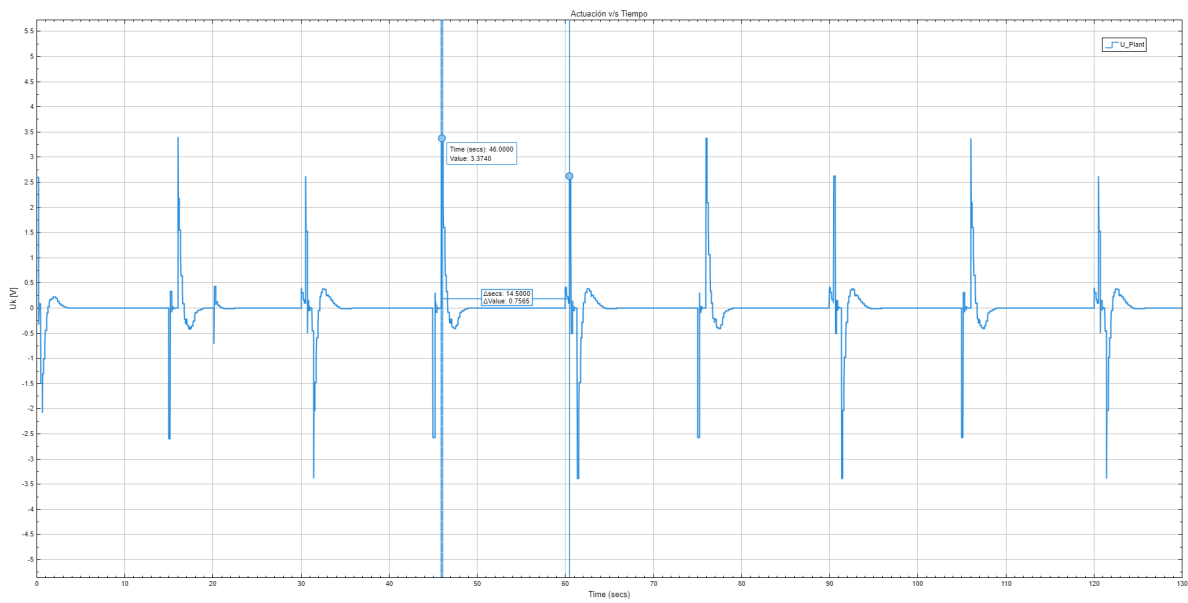


Fig. 4.72: Actuación de la planta ± 3 .

4. SIMULACIÓN DEL SISTEMA DE CONTROL

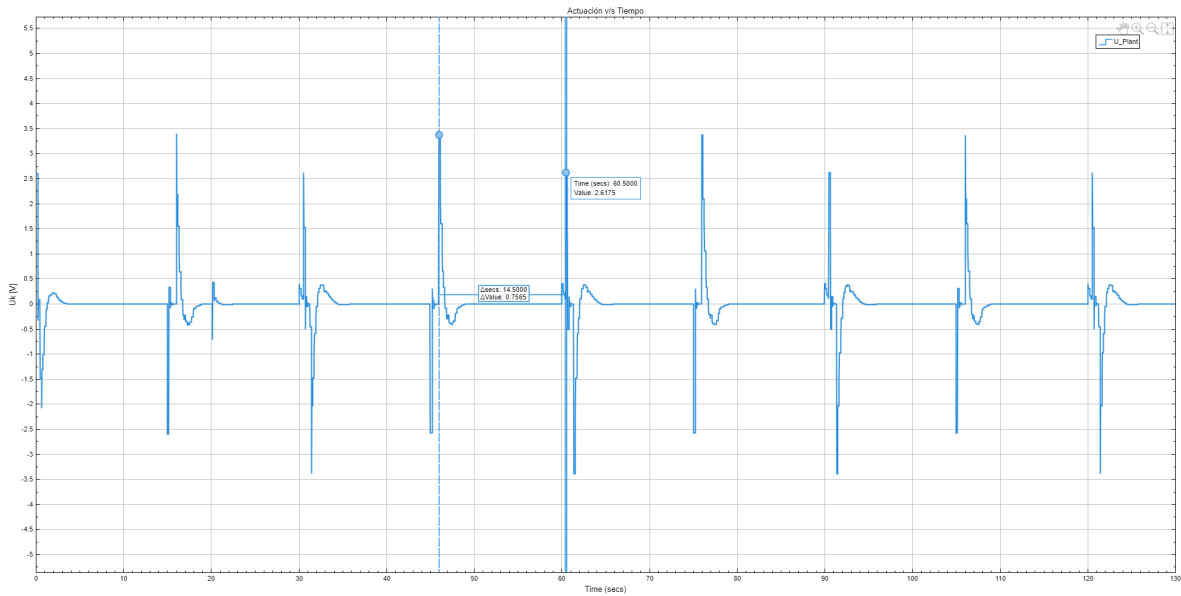


Fig. 4.73: Actuación de la planta ± 3 .

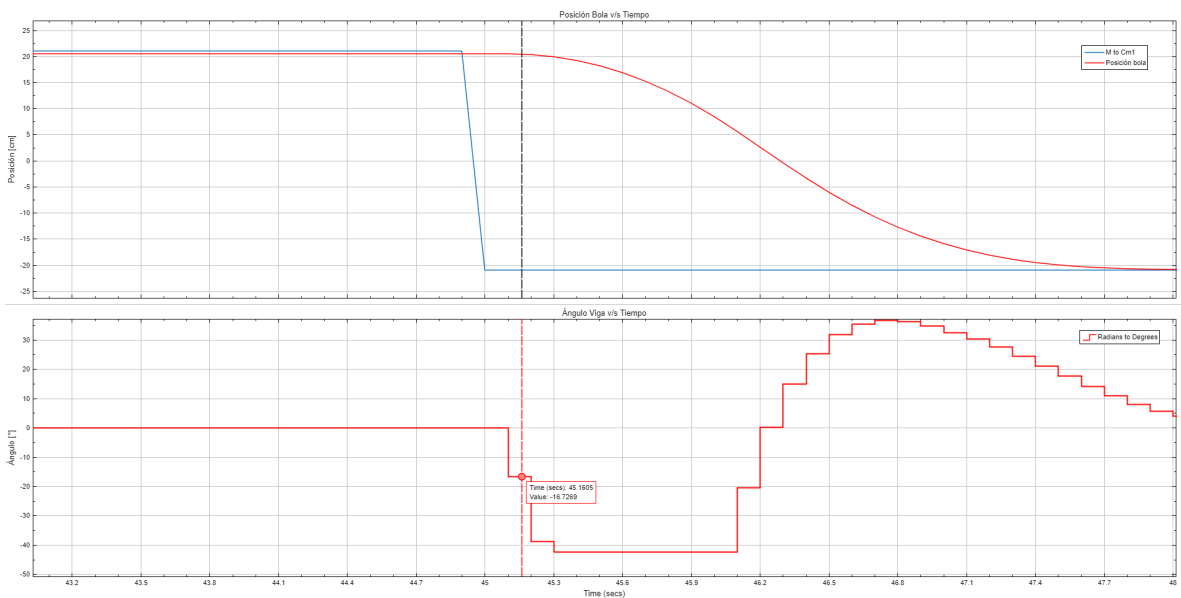


Fig. 4.74: Valor ángulo actuación ± 3 .

4.2 Tablas de resultados

Los resultados anteriores se pueden resumir en las siguientes tablas:

4. SIMULACIÓN DEL SISTEMA DE CONTROL

Tab. 4.1: Efecto de la condición inicial para $\Omega_{7,7} = 5 \times 10^7$ ($\Gamma = 0,3$, $N_p = 12$).

Condición Inicial	Overshoot Transiente	Controla	Observaciones
15	No	Sí	No presenta overshoot en el transiente.
5	No	Sí	No presenta overshoot en el transiente.
-5	Sí	Sí	Presenta overshoot en el transiente de 21,71.

Tab. 4.2: Efecto de la condición inicial para $\Omega_{7,7} = 5 \times 10^7$ ($\Gamma = 0,3$, $N_p = 12$, $\Delta u_{max} = 5[V]$) con Slew rate.

Condición Inicial	Referencia alcanzada	Overshoot Transiente	Overshoot Estacionario	Controla	Observaciones
-5	20,27	21,81	21,35	Sí	Slew rate actúa de manera efectiva.
0	20,27	No	21,35	Sí	Slew rate actúa de manera efectiva.
5	20,27	No	21,35	Sí	Slew rate actúa de manera efectiva.

Tab. 4.3: Efecto de la condición inicial para $\Omega_{7,7} = 0,7 \times 10^8$ ($\Gamma = 0,3$, $N_p = 12$, $\Delta u_{max} = 5[V]$) con Slew rate.

Condición Inicial	Referencia alcanzada	Overshoot Transiente	Overshoot Estacionario	Controla	Observaciones
-5	20,38	21,86	21,43	Sí	Slew rate actúa de manera efectiva.
0	20,38	No	21,43	Sí	Slew rate actúa de manera efectiva.
5	20,38	No	21,43	Sí	Slew rate actúa de manera efectiva.

Tab. 4.4: Efecto de la condición inicial para $\Omega_{7,7} = 1 \times 10^8$ ($\Gamma = 0,3$, $N_p = 12$, $\Delta u_{max} = 5[V]$) con Slew rate.

Condición Inicial	Referencia alcanzada	Overshoot Transiente	Overshoot Estacionario	Controla	Observaciones
-5	20,48	21,71	21,50	Sí	Slew rate actúa de manera efectiva.
0	20,48	No	21,50	Sí	Slew rate actúa de manera efectiva.
5	20,48	No	21,50	Sí	Slew rate actúa de manera efectiva.

4. SIMULACIÓN DEL SISTEMA DE CONTROL

Tab. 4.5: Análisis de control y observaciones para diferentes valores de N_p , ($\Omega_{7,7} = 1,0 \times 10^8$, $\Gamma = 0,3$, $T_s = 0,1$, $\Delta u_{max} = 5[V]$) con Slew rate.

Np	Controla	Observaciones
1	No	Sistema inestable
5	Sí	Se observan respuestas subamortiguadas, no respeta límite físico
6	Sí	Se observan respuestas subamortiguada de menor amplitud, no respeta límite físico
7	Sí	Se observan respuestas subamortiguadas con menor overshoot 21.91, estabiliza en 20.32
8	Sí	Se observa respuesta subamortiguada con menor overshoot 21.70, estabiliza en 20.40
9	Sí	Se observa respuesta subamortiguada overshoot 21.59, estabiliza en 20.45
10	Sí	Respuesta subamortiguada overshoot 21.51, estabiliza en 20.47
11	Sí	Respuesta subamortiguada overshoot 21.51, estabiliza en 20.48
12	Sí	Respuesta subamortiguada overshoot 21.50, estabiliza en 20.48

Tab. 4.6: Análisis del desempeño y tiempos de cómputo para variaciones de $\Omega_{7,7}$, ($\Gamma = 0,3$, $N_p = 12$, $T_s = 0,1[s]$, $\Delta u_{max} = 5[V]$) con Slew rate.

Omega ($\Omega_{7,7}$)	Referencia Alcanzada	Overshoot Estacionario	T_solver max [ms]	T_solver prom [ms]
$0,7 \times 10^8$	20.38	20.43	0.4050	0.0768
$1,0 \times 10^8$	20.48	20.50	0.4630	0.0910
$1,2 \times 10^8$	20.50	21.58	0.3653	0.0925
$1,5 \times 10^8$	20.56	21.63	0.4194	0.0969
$1,7 \times 10^8$	20.59	21.65	0.6344	0.0969
$2,0 \times 10^8$	20.61	21.65	0.4498	0.1008

4. SIMULACIÓN DEL SISTEMA DE CONTROL

Tab. 4.7: Efecto del rango de actuación con Slew rate en el control del sistema ($\Omega_{7,7} = 1 \times 10^8$), $\Delta u_{max} = 5[V]$.

Límite u [V]	$\Omega_{7,7}$	Gamma	Controla	Observaciones
$[-9, 9]$	1×10^8	0.3	Sí	Slew rate es efectivo, $u_k = 4,6$ [V].
$[-8, 8]$	1×10^8	0.3	Sí	Slew rate es efectivo, $u_k = 4,6$ [V], se comporta de la misma forma.
$[-7, 7]$	1×10^8	0.3	Sí	Slew rate es efectivo, $u_k = 4,6$ [V], se comporta de la misma forma.
$[-6, 6]$	1×10^8	0.3	Sí	Slew rate es efectivo, $u_k = 4,6$ [V], se comporta de la misma forma.
$[-5, 5]$	1×10^8	0.3	Sí	Slew rate es efectivo, $u_k = 4,6$ [V], se comporta de la misma forma.
$[-4, 4]$	1×10^8	0.3	Sí	Disminuye en amplitud $u_k = 4,6$ [V] a $3,6$ [V]. Ángulo disminuye de $-27,21^\circ$ a $-23,21^\circ$. Saltos de ángulos más pequeños.
$[-3, 3]$	1×10^8	0.3	Sí	Disminuye en amplitud $u_k = 3,6$ [V] a $2,6$ [V]. Saltos de ángulos más suaves.

5 Conclusiones y trabajos a futuro

5.1 Conclusiones

Este proyecto consistió en el diseño e implementación de un sistema de Control Predictivo por Modelo (MPC) aplicado a una planta Ball & Beam de Quanser mediante un proceso de modelado y simulación. El desarrollo comenzó con la obtención de las ecuaciones físicas de los subsistemas electromecánico y mecánico, las cuales fueron linealizadas y representadas en un espacio de estados. Debido a la estabilidad marginal de la planta, se implementó una estrategia de pre-estabilización mediante DLQR, integrando además un observador de estado extendido para la reconstrucción de variables internas y la estimación de perturbaciones. Finalmente, el sistema fue discretizado con un periodo de muestreo de 0,1 segundos y validado en MATLAB/Simulink, donde el controlador optimizó las acciones de mando sujeto a restricciones de voltaje y seguridad operativa utilizando el algoritmo Active-Set mediante el solver Quadprog.

A partir de los resultados obtenidos, se pueden extraer las siguientes conclusiones:

Las simulaciones demostraron que el controlador MPC rechaza perturbaciones externas de manera efectiva. Se identificó que la magnitud del overshoot transitorio exhibe una dependencia directa con la condición inicial, incrementándose a medida que el estado inicial se aleja del punto de operación objetivo.

Al incrementar la ponderación del séptimo estado en la matriz Ω (error de posición en la planta aumentada), pero genera una respuesta transitoria con mayor overshoot alcanzando un valor referencia máxima de 20,48 cumpliendo con las restricciones físicas de la viga.

Se determinó que un horizonte de predicción $N_p \geq 10$ es necesario para asegurar la estabilidad y el cumplimiento de las restricciones frente a la dinámica del sistema.

La implementación de restricciones de Slew Rate permitió suavizar los cambios bruscos en el ángulo de la viga. Esto es vital para reducir el estrés mecánico en la caja de engranajes del servomotor y evitar picos de corriente que degraden el equipo.

5. CONCLUSIONES Y TRABAJOS A FUTURO

Se verificó que el solver Quadprog con el método Active-Set es viable para aplicaciones en tiempo real. Los tiempos de ejecución entre 10^{-4} y 10^{-3} segundos son significativamente inferiores al periodo de muestreo de 0,1 segundos, garantizando el cálculo de la acción óptima en cada ciclo.

Las pruebas de simulación validan que el controlador diseñado cumple con regular la posición de la bola bajo una estrategia predictiva. La correcta identificación de parámetros y la linealización permitieron un modelo cuya controlabilidad y observabilidad fueron verificadas. La eficacia del DLQR fue crítica para permitir el funcionamiento seguro del MPC. Se garantizó el cumplimiento de restricciones de voltaje, ángulos físicos y límites de slew rate, confirmando que la arquitectura propuesta es capaz de adaptarse a los requisitos de desempeño exigidos para el sistema y obtener una correcta implementación física.

5.1.1 Trabajos a futuro

Para trabajos a futuro se propone:

- **Validación e implementación en hardware real:** Esto permitirá evaluar el desempeño del controlador MPC diseñado ante efectos no modelados, como la fricción no lineal del servomotor, retardos en la comunicación y el ruido real de los sensores.
- **Exploración de técnicas de optimización alternativas:** Se propone estudiar la aplicación de algoritmos de optimización distintos al método de Active-Set. En particular, el uso del Método de Multiplicadores de Dirección Alternante (ADMM) con el propósito de implementar el controlador en hardware que tienen recursos limitados como lo puede ser una FPGA.
- **Implementación de aproximaciones no lineales basado en redes neuronales tipo ReLU:** Se sugiere explorar el uso de funciones de activación de tipo ReLU para acelerar los tiempos de cómputo facilitando la implementación y entrenamiento de una red neuronal que ayude a la implementación de modelos de predicción más complejos y precisos con la intención de aplicarlo en hardware de recursos limitados como una FPGA.

Referencias

- [1] J. Apkarian, P. Karam, M. Lévis y H. Gurocak, *Ball and Beam Experiment for MATLAB®/Simulink® Users, Instructor Workbook*, Standardized for ABET Evaluation Criteria, Quanser Inc., Markham, Ontario, Canada, 2015.
- [2] M. A. Azúa Poblete, “Implementación de un sistema de control predictivo usando el toolbox QUARC,” Tesis de Pregrado, Universidad Tecnica Federico Santa María, Valparaíso, Chile, 2022.
- [3] A. Cedeño, A. Morrison y G. Carvajal, “Control Predictivo Por Modelo,” Lecture notes, 2024.
- [4] A. L. Cedeño, “Lecture Notes on Model Predictive Control,” Apuntes de clase sobre MPC, 2025.
- [5] B. A. Falcon Mendoza, D. R. Nacato Estrella, Á. S. Velásquez Benavidez, F. J. Santacruz Sulca y D. F. Veloz Cherres, “Modelado Matemático, Simulación e Implementación de Pruebas en un Motor DC labvolt,” *Polo del Conocimiento*, vol. 7, n.º 7, págs. 997-1018, jul. de 2022, ISSN: 2550-682X.
- [6] R. J. Mantz, “Observadores de estados,” Universidad Nacional de La Plata, Facultad de Ingeniería, Departamento de Electrotecnia, Cátedra de Control Moderno, La Plata, Argentina, Informe técnico, 2003.
- [7] MathWorks, “idare - Solver implícito de ecuaciones algebraicas de Riccati de tiempo discreto,” Tesis doct., 2024. dirección: <https://es.mathworks.com/help/control/ref/idare.html>
- [8] MathWorks, “dlqr - Regulador lineal cuadrático (LQ) de retroalimentación de estados para un sistema de espacio de estados en tiempo discreto,” Tesis doct., 2025. dirección: <https://es.mathworks.com/help/control/ref/dlqr.html>
- [9] MathWorks. “MATLAB - El lenguaje del cálculo técnico,” MathWorks. dirección: <https://es.mathworks.com/products/matlab.html>
- [10] MathWorks. “Simulación y diseño basado en modelos con Simulink,” MATLAB Documentation. dirección: <https://es.mathworks.com/products/simulink.html>

REFERENCIAS

- [11] MathWorks, “What Is Model Predictive Control?” Tesis doct., 2025. dirección: <https://es.mathworks.com/help/mpc/gs/what-is-model-predictive-control.html>
- [12] MathWorks, “c2d - Conversión de modelos de tiempo continuo a discreto,” Tesis doct., 2026. dirección: <https://es.mathworks.com/help/control/ref/dynamicsys.c2d.html>
- [13] J. Mazurkiewicz, “The basics of motion control - Part 1: Here are easy-to-follow equations for determining drive mechanics - friction and inertia - for any electromechanical positioning application,” *Power Transmission Design*, págs. 43-45, sep. de 1995.
- [14] H. Nak, Ş. Akkaya y E. Yumuk, “Active Set Method Based Model Predictive Control for a Ball and Beam System,” en *Proceedings of the Istanbul Technical University*, Department of Control and Automation Engineering, Istanbul, Turkey: Istanbul Technical University, 2021. dirección: <https://www.researchgate.net/publication/351659850>
- [15] K. Ogata, *Ingeniería de control moderna*, 4.^a ed. Madrid, España: Pearson Educación, 2003, ISBN: 9788420536781. dirección: https://books.google.cl/books?id=QK148EPC_m0C
- [16] K. Ogata, *Modern Control Engineering* (Instrumentation and controls series), 5.^a ed. Upper Saddle River, NJ: Prentice Hall, 2010, ISBN: 9780136156734. dirección: <https://books.google.cl/books?id=Wu5GpNAelzkC>
- [17] Quanser, *Ball and Beam - User Manual*, Documento técnico / Manual de usuario, Quanser Inc., 2024.
- [18] Quanser Consulting Inc., *SRV02-Series Rotary Servo Plant User Manual*, Quanser Innovate Educate, Markham, Ontario, Canada, 2015.
- [19] J. B. Rawlings, D. Q. Mayne y M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd. Madison, Wisconsin, USA: Nob Hill Publishing, LLC, 2017, ISBN: 9780975937730. dirección: <http://www.nobhillpublishing.com>
- [20] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB®* (Advances in Industrial Control). London, UK: Springer-Verlag, 2009, ISBN: 9781848823303. dirección: <https://link.springer.com/book/10.1007/978-1-84882-331-0>
- [21] E. W. Wong, “Active-Set Methods for Quadratic Programming,” Tesis doct., Stanford University, 2011. dirección: <https://web.stanford.edu/group/SOL/papers/ELWthesis.pdf>

Anexo | Códigos MATLAB

1 Cálculo de A_d y B_d

```
%%cálculo de Ad manual
clear; clc ; close all

kbb=0.418;
tau=0.025;

% Definir variables simbólicas
syms s t T tau

%Discretizando
Ts=0.1;          %tiempo de muestreo

%(x1,x2,x3,x4)=(Theta,omega,pos,acelera)
A_nueva = [0    1    0    0;
0 -40    0    0;
0    0    0    1;
kbb 0    0    0];

B_nueva=[0; 60; 0; 0];

% usando Matriz A_nueva

% Identidad
I = eye(size(A_nueva));

% Resolver (sI - A)
sI_A = s*I - A_nueva;
```

```
% Inversa simbólica
inv_sI_A = simplify(inv(sI_A));

% Inicializar matriz para la transformada inversa
invLap = sym(zeros(size(inv_sI_A)));

% Aplicar ilaplace a cada elemento
for i = 1:size(inv_sI_A, 1)
for j = 1:size(inv_sI_A, 2)
invLap(i, j) = ilaplace(inv_sI_A(i, j), s, t);
end
end

A_d2 = subs(invLap, t, Ts);

% Pasar a valores numéricos
A_d2 = double(vpa(A_d2, 12));

disp('Ad calculado con Laplace inversa =');
disp(A_d2)

%usando B_nueva calculo de Bd, manual
% Exponencial de matriz simbolica
expA_tau = expm(A_nueva * tau); % e^{A tau}
% Integral simbolica de 0 a T del producto de e^{A tau} * B
Bd2 = int(expA_tau * B_nueva, tau, 0, T)

Bd_eval = simplify(subs(Bd2, T, Ts));
double(Bd_eval)
```

2 Controlabilidad y Observabilidad Sistema Discreto

```
%% primera parte : discretizar la planta
clear; clc ; close all
s=tf ('s');

kbb=0.418;
```

```
%Discretizando
Ts=0.1;      %tiempo de muestreo

%%
% representación de estados continuo , siendo los estados
%(x1,x2,x3,x4)=(Theta,omega,pos,acelera)
A_nueva = [0    1    0    0;
0 -40    0    0;
0     0    0    1;
kbb  0     0    0];

B_nueva=[0; 60; 0; 0];
C_nueva=[0 0 1 0];
D_nueva=0;

%discretizando el sistema
sysc = ss(A_nueva,B_nueva,C_nueva,D_nueva); % sistema continuo
sysd = c2d(sysc, Ts, 'zoh'); % discretización

Ad_nueva = sysd.A;
Bd_nueva = sysd.B;
Cd_nueva = sysd.C;
Dd_nueva = sysd.D;

%% Calculando los autovalores
% sistema continuo y discreto

eig_continuo=eig(A_nueva); %marginalmente estable tiene parte real =0
eig_discreto=eig(Ad_nueva); %marginalmente estable porque tiene valores
                        %1 pero dentro del circulo

%% Calculando observabilidad y Controlabilidad Sistema Discreto

Contro_d= ctrb(Ad_nueva,Bd_nueva);%prueba controlabilidad
rank_Co_d=rank(Contro_d)
```

```
Ob_d= obsv(Ad_nueva,Cd_nueva);%prueba observabilidad
rank_ob_d=rank(Ob_d)
```

```
%el rango es igual al numero de estados de la
matriz A
% tanto en control como en obs 4 = 4 estados
%es observable y controlable
```

3 **Calculo de A y B caligráficas (A_{bar} y O_{bar})**

```
function [A_cal, B_cal] = MatrizCal(A, B, N)
% Construye las matrices A_bar y O_bar
% Entrada:
%   A - matriz del sistema
%   B - matriz de entrada
%   N - horizonte de predicción
% Salida:
%   A_cal - apilamiento de  $A^i * x_0$ 
%   B_cal - matriz extendida de dinámica de control

[nx, nu] = size(B); % tamaño del sistema

% Matriz A_cal
A_cal = zeros(N * nx, nx);
for i = 1:N
A_cal((i-1)*nx + 1:i*nx, :) = A^i;
end

% Matriz B_cal
B_cal = zeros(N * nx, N * nu);
for i = 1:N
for j = 1:i
B_cal((i-1)*nx + 1:i*nx, (j-1)*nu + 1:j*nu) = A^(i-j) * B;
end
end
end
```

4 PreCompile Ball And Beam

```
clear; clc ; close all
s=tf ('s');

kbb=0.418;
%Discretizando
Ts=0.1;%tiempo de muestreo

%%
% representación de estados continuo , siendo los estados
%(x1,x2,x3,x4)=(Thetha,omega,pos,acelera)= (angulo del brazo del
%motor,velocidad angular de la viga,posicion de
%la bola y velocidad de la bola)

A_nueva = [0    1    0    0;
0 -40    0    0;
0    0    0    1;
kbb 0    0    0];

B_nueva=[0; 60; 0; 0];
C_nueva=[0 0 1 0];
D_nueva=0;
Bp=[0;0;0;0];% matriz de perturbación de entrada
Cp=1;%perturbacion de salida

%discretizando el nuevo sistema
sysc = ss(A_nueva,B_nueva,C_nueva,D_nueva); % sistema continuo
sysd = c2d(sysc, Ts, 'zoh'); % discretización

Ad_nueva = sysd.A;
Bd_nueva = sysd.B;
Cd_nueva = sysd.C;
Dd_nueva = sysd.D;

%%DLQR
Q = diag([1e1, 5e2, 5e3, 3e3]); % matriz de pesos para el control DLQR
```

Anexo. Códigos MATLAB

```
R = 0.05; % Matriz de penalización para el control

L = dlqr(Ad_nueva, Bd_nueva, Q, R); % Ganancia del controlador DLQR
Kr = inv(Cd_nueva*inv(eye(4)-(Ad_nueva-Bd_nueva*L))*Bd_nueva);
disp("autovalores prealimentado")
eig(Ad_nueva-Bd_nueva*L) %ver estabilidad del sistema
%prealimentado por dlqr
[L2,S,P] = dlqr(Ad_nueva, Bd_nueva, Q, R);

%diseñar observador planta usando (ricatti)

%Qo=1;%penalizacion al error del estado estimado
Qo=diag([1e-2 1e0 1e6 1e4]);
Ro=0.1;%penalizacion del ruido en la salida

%solucion optima ricatti
P_optimo_ricatti=idare(Ad_nueva',Cd_nueva',Qo,Ro);

%ganancia del obserdor G en la literatura
Lo=(Ad_nueva*P_optimo_ricatti*Cd_nueva')/
(Cd_nueva*P_optimo_ricatti*Cd_nueva'+Ro);

Aobs=Ad_nueva-Lo*Cd_nueva;
Bobs =[Bd_nueva Lo];
Cobs=eye(size(Ad_nueva,1));
Dobs=zeros(size(Ad_nueva,1),2);

%% Nueva planta lazo cerrado LQR
Acl=Ad_nueva-Bd_nueva*L;

%dinámica estabilizada por LQR (A-BL)\(KC A-KC-BL)
%esta es la nueva planta aumentada

%matriz A aumentada de la nueva planta
Aa = [ Ad_nueva,-Bd_nueva*L ;(Lo*Cd_nueva),
      (Ad_nueva-Lo*Cd_nueva-Bd_nueva*L)];
```

Anexo. Códigos MATLAB

```
Ba = [Bd_nueva;Bd_nueva];
Ca = [Cd_nueva,zeros(1,4)];
Da = Dd_nueva;

%segundo observador con estimacion de perturbaciones (x ;x_estimado;d)
A_extP=[Ad_nueva,-Bd_nueva*L,zeros(4,1);
Lo*Cd_nueva,(Ad_nueva - Lo*Cd_nueva - Bd_nueva*L),zeros(4,1);
zeros(1,4),zeros(1,4),1];
B_extP=[Bd_nueva;Bd_nueva;0];
C_extP=[Cd_nueva,zeros(1,4),1];
D_extP=zeros(9,1);

%QoP=1; %penalizacion al error del estado estimado

%penalizacion al error del estado estimado
QoP=diag([1e-2 1e0 1e2 1e2 1e-2 1e0 1e2 1e2 1e9]);
%penalizacion del ruido de salida
RoP=1;

%ganancia de ricatti para el observador con perturbaciones

P_optimo_ricatti_P=idare(A_extP',C_extP',QoP,RoP);

%ganancia del obserdor K en la notacion formal

KP=(A_extP*P_optimo_ricatti_P*C_extP')/
(C_extP*P_optimo_ricatti_P*C_extP'+RoP);

AobsP=A_extP-KP*C_extP;
BobsP=[B_extP,KP];
CobsP=eye(9);
DobsP=zeros(9,2);

%% MPC : Horizontes y pesos
Np =12;          % horizonte de prediccion

x0 = [0;0;0;0];% estado inicial
```

Anexo. Códigos MATLAB

```
nx = size(Ad_nueva,1); %4 estados
nu = size(Bd_nueva,2); %1 entrada

%matrices de Penalizacion de estados

diag_Q = [1e0, 1e0, 1e0, 1e0, 1e0, 0.5e1, 1e6, 1e0];
Omega = diag(diag_Q);
gam = 0.5;

[A_cal, B_cal] = MatrizCal(Aa, Ba, Np); %A caligrafica, B caligrafica

Qomega=kron(eye(Np),Omega);% matriz Q de omegas
Rmatriz=gam*eye(Np*nu);%matriz R de gammas

H=2*(Rmatriz+B_cal'*Qomega*B_cal);

%matrices de estados
valorgrande=100;
x_min = -valorgrande*ones(4,1);
x_max = valorgrande*ones(4,1);
x_min(1)=-45*pi/180; % -56 grados el max soportado por el servo
x_max(1)= 45*pi/180; %+56 grados max soportado por el servo
x_min(3) = -0.21;%limitaciones de salida
x_max(3) = 0.21;%limitaciones de salida

%restricciones de la actuación: voltaje max del servo +-10 V
u_min=-10; %volts
u_max=10;
%% seguimiento a referencias

inv1 = inv(eye(size(Aa))-Aa);
inv2 = inv(Ca*inv1*Ba);
```

5 Precompilar Ball and beam con Slew rate

```
clear; clc ; close all
s=tf ('s');
```

Anexo. Códigos MATLAB

```
kbb=0.418;
%Discretizando
Ts=0.1;%tiempo de muestreo

%%
% representación de estados continuo , siendo los estados
%(x1,x2,x3,x4)=(Thetha,omega,pos,acelera)= (angulo del brazo del
%motor,velocidad angular de la viga,
%posicion de la bola y velocidad de la bola)
A_nueva = [0    1    0    0;
0 -40    0    0;
0    0    0    1;
kbb 0    0    0];

B_nueva=[0; 60; 0; 0];
C_nueva=[0 0 1 0];
D_nueva=0;
Bp=[0;0;0;0];% matriz de perturbación de entrada
Cp=1;%perturbacion de salida

%discretizando el nuevo sistema
sysc = ss(A_nueva,B_nueva,C_nueva,D_nueva); % sistema continuo
sysd = c2d(sysc, Ts, 'zoh'); % discretización

Ad_nueva = sysd.A;
Bd_nueva = sysd.B;
Cd_nueva = sysd.C;
Dd_nueva = sysd.D;

%%DLQR
Q = diag([1e1, 5e1, 5e3, 3e3]); % matriz de pesos para el control DLQR
R = 0.05; % Matriz de penalización para el control

L = dlqr(Ad_nueva, Bd_nueva, Q, R); % Ganancia del controlador DLQR
Kr = inv(Cd_nueva*inv(eye(4) -(Ad_nueva-Bd_nueva*L))*Bd_nueva);
disp("autovalores prealimentado")

% ver estabilidad del sistema prealimentado por dlqr
```

Anexo. Códigos MATLAB

```
eig(Ad_nueva-Bd_nueva*L)
[L2,S,P] = dlqr(Ad_nueva, Bd_nueva, Q, R);

%diseñar observador planta usando (ricatti)

Qo=diag([1e-2 1e0 1e6 1e4]);
Ro=0.1; %penalizacion del ruido en la salida

%solucion optima a ricatti
P_optimo_ricatti=idare(Ad_nueva',Cd_nueva',Qo,Ro);

%ganancia del obserdor G en la literatura
Lo=(Ad_nueva*P_optimo_ricatti*Cd_nueva')/
    (Cd_nueva*P_optimo_ricatti*Cd_nueva'+Ro);

Aobs=Ad_nueva-Lo*Cd_nueva;
Bobs =[Bd_nueva Lo];
Cobs=eye(size(Ad_nueva,1));
Dobs=zeros(size(Ad_nueva,1),2);

%% Nueva planta lazo cerrado LQR

% dinámica estabilizada por LQR (A-BL)\(KC A-KC-BL)
%esta es la nueva planta aumentada

Acl = Ad_nueva - Bd_nueva*L;

% matriz A aumentada de la nueva planta

Aa = [Ad_nueva, -Bd_nueva*L; (Lo*Cd_nueva),
      (Ad_nueva-Lo*Cd_nueva-Bd_nueva*L)];
Ba = [Bd_nueva; Bd_nueva];
Ca = [Cd_nueva, zeros(1,4)];
Da = Dd_nueva;

%segundo observador con estimacion de perturbaciones (x ;x_estimado;d)
A_extP=[Ad_nueva, -Bd_nueva*L, zeros(4,1)];
```

Anexo. Códigos MATLAB

```
Lo*Cd_nueva, (Ad_nueva-Lo*Cd_nueva-Bd_nueva*L), zeros(4,1);
zeros(1,4), zeros(1,4), 1];
B_extP=[Bd_nueva;Bd_nueva;0];
C_extP=[Cd_nueva, zeros(1,4), 1];
D_extP=zeros(9,1);

%penalizacion al error del estado estimado

QoP=diag([1e-2 1e0 1e2 1e2 1e-2 1e0 1e2 1e2 1e6]);
RoP=1;%penalizacion del ruido de salida

%ganancia de ricatti para el observador con perturbaciones
P_optimo_ricatti_P=idare(A_extP',C_extP',QoP,RoP);

%ganancia del obserdor K en la notacion formal
KP=(A_extP*P_optimo_ricatti_P*C_extP')/
    (C_extP*P_optimo_ricatti_P*C_extP'+RoP);

AobsP=A_extP-KP*C_extP;
BobsP=[B_extP, KP];
CobsP=eye(9);
DobsP=zeros(9,2);

%% MPC : Horizontes y pesos
Np =12; % horizonte de predicción

x0 = [0;0;0;0];% estado inicial
nx = size(Ad_nueva,1); %4 estados
nu = size(Bd_nueva,2); %1 entrada

%matrices de Penalizacion de estados

diag_Q = [1e0, 1e0, 1e0, 1e0, 1.2e3, 0.1e1, 0.9e6, 1e0];
Omega = diag(diag_Q);
gam = 0.3;

[A_cal, B_cal] = MatrizCal(Aa, Ba, Np); %A caligrafica, B caligrafica
```

```
Qomega=kron(eye(Np),Omega);% matriz Q de omegas
Rmatriz=gam*eye(Np*nu);%matriz R de gammas

H=2*(Rmatriz+B_cal'*Qomega*B_cal);

%matrices de estados
valorgrande=100;
x_min = -valorgrande*ones(4,1);
x_max =  valorgrande*ones(4,1);
x_min(1)=-pi/4; % -56 grados el max soportado por el servo
x_max(1)= pi/4; %+56 grados max soportado por el servo
x_min(3) = -0.21;%limitaciones de salida
x_max(3) = 0.21;%limitaciones de salida

%restricciones de la actuación: voltaje max del servo +-10 V
u_min=-10; %volts
u_max=10;
%% seguimiento a referencias

inv1 = inv(eye(size(Aa))-Aa);
inv2 = inv(Ca*inv1*Ba);

%% 2. Calcular el promedio en milisegundos
%promedio_ms = mean(datos) * 1000;

% 3. Calcular el valor máximo (pico) en milisegundos
%pico_ms = max(datos) * 1000;

% 4. Imprimir en consola
%fprintf('Promedio: %.4f ms\n', promedio_ms);
%fprintf('Máximo: %.4f ms\n', pico_ms);
```

6 Cálculo M y c con slew rate

```
function [c, M]= fcn( A_cal,B_cal,x_max,x_min,Np,
                    u_max,u_min,Lpre,x_actual,xinf,
                    uinf, u_prev, delta_u_max)

% armando las desviaciones
```

```
x0_virguli = x_actual - xinf;
%armado vector de estados
x_minrep = repmat(x_min,2,1);
x_maxrep = repmat(x_max,2,1);

%redimensionado de las matrices dependiendo de Np
xmin_stack = kron(ones(Np,1), x_minrep); % (Np*n) x 1
xmax_stack = kron(ones(Np,1), x_maxrep); % (Np*n) x 1
u_minrestric = repmat(u_min,Np,1);
u_maxrestric = repmat(u_max,Np,1);

xinf_sat=xinf;
xinf_sat=min(max(xinf_sat,x_minrep),x_maxrep);

%armado de desviacion estados y entradas
xmin_virguli=xmin_stack-repmat(xinf_sat,Np,1); % limitacion de estados
xmax_virguli=xmax_stack-repmat(xinf_sat,Np,1);

umin_virguli = u_minrestric - repmat(uinf,Np,1);% limitacion de entrada
umax_virguli = u_maxrestric - repmat(uinf,Np,1);

%prediccion por x_original
X_pred_stack = A_cal * x0_virguli;

% 1. Parte del calculo M y c respecto a los estados
M_x = [B_cal; -B_cal];
c_x = [xmax_virguli - X_pred_stack; -xmin_virguli + X_pred_stack];

% Restricciones de ENTRADA APLICADA usando L_pre
nu = numel(u_min);

% entradas por paso
Iu = eye(Np * nu);

% L_pre
Lpre_stack = repmat(Lpre(:), Np, 1);
```

Anexo. Códigos MATLAB

```
% 1. Parte actual: Restricciones de magnitud (u_min <= u <= u_max)
M_u = [ Iu; -Iu ];
c_u = [ umax_virguli + Lpre_stack; -umin_virguli - Lpre_stack ];

%*****Slew Rate*****
% 2. Restricciones de Slew Rate (|u_k - u_{k-1}| <= delta_u_max)
% D * u_tilde = [u_tilde_0; u_tilde_1 - u_tilde_0; ... ]

D = eye(Np*nu) - diag(ones((Np-1)*nu, 1), -nu);

% Límite superior: u_k - u_{k-1} <= delta_u_max
c_slew_up = delta_u_max * ones(Np*nu, 1);

% Ajuste para el primer paso (k=0): u_0 - u_prev <= delta_u_max
% (u_tilde_0 + uinf - Lpre) - u_prev <= delta_u_max

c_slew_up(1:nu) = delta_u_max + u_prev - uinf + Lpre;

% Límite inferior: -(u_k - u_{k-1}) <= delta_u_max
c_slew_lo = delta_u_max * ones(Np*nu, 1);

% Ajuste para el primer paso (k=0): -(u_0 - u_prev) <= delta_u_max
% -u_tilde_0 <= delta_u_max - u_prev + uinf - Lpre

c_slew_lo(1:nu) = delta_u_max - u_prev + uinf - Lpre;

M_slew = [D; -D];
c_slew = [c_slew_up; c_slew_lo];

% 3. Unión de todas las restricciones
M = [ M_x ; M_u ; M_slew ];
c = [ c_x ; c_u ; c_slew ];
end
```