

UNIVERSIDAD SANTO TOMÁS

SIMULACIÓN DE UN CONTROLADOR PARA EL SISTEMA BALL AND BEAM QUANSER BASADO EN ALGORITMOS DE APRENDIZAJE AUTOMÁTICO

Realizado por

Vincenth Adolfo Gaviria Zapata

Proyecto de Grado presentado en cumplimiento del requisito
para optar por el grado de Ingeniería Electrónica



Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)
Facultad de Ingeniería Electrónica
División de Ingenierías

Diciembre de 2022

SIMULACIÓN DE UN CONTROLADOR PARA EL SISTEMA BALL AND BEAM QUANSER BASADO EN ALGORITMOS DE APRENDIZAJE AUTOMÁTICO

Realizado por

Vincenth Adolfo Gaviria Zapata

Proyecto de Grado presentado en cumplimiento del requisito
para optar por el grado de Ingeniería Electrónica

Dirigido por

M.Sc. Armando Mateus Rojas

Co-Dirigido por

M.Sc. Sindy Paola Amaya

Grupo de Investigación GED (Grupo de Estudio y Desarrollo en Robótica)
Facultad de Ingeniería Electrónica
División de Ingenierías

Diciembre de 2022

Declaración de Autoría

Me declaro como el autor que ha contribuido con el contenido intelectual de este trabajo de grado, que el documento no ha sido publicado previamente, no esta en proceso de revisión y que los datos e información contenida bajo influencia relevante han sido citados.

Dedicatoria

A mis padres por darme todo lo necesario para continuar con mi vida y llegar a este punto.

A mi esposa por darme todo su apoyo y compañía en los momentos difíciles.

Agradecimientos

Quiero empezar este agradecimiento con la persona que siempre vela por mi bienestar, seguridad y amor; mi madre, una figura ejemplar, honesta, comprensiva, tolerante y trabajadora, todas estas y mas cualidades que también he adoptado en mi vida, gracias a su constante ejemplo. Continuando con mi padre, ese hombre valiente, perseverante, arriesgado, único y polifacético, una persona que también admiro por su autenticidad. Mis dos pilares y las personas por las cuales me encuentro hoy en día cumpliendo la primera de muchas metas que tendré gracias a ambos por ser los precursores de mi existencia.

También quiero agradecer a mis maestros, las personas que me proporcionaron los conocimientos y las herramientas para lograr este objetivo. Gracias por dedicar sus vidas a compartir el conocimiento a través de la enseñanza, y el saber a hombres y mujeres profesionales.

Agradezco a mis amigos y pareja por hacer parte de este proceso y por apoyarme en los momentos mas difíciles y problemáticos, aquellas personas que están en los buenos momentos y en los malos.

Finalmente agradecer a mi persona por las horas de estudio y algunas veces las lágrimas en los momentos en los que creí que no iba poder pero finalmente siempre hubo la voluntad y el deseo de cumplir este sueño.

Resumen

En este documento se presenta el desarrollo de un controlador basado en algoritmos de aprendizaje automático para el sistema no lineal *ball and beam*, mostrando cómo implementar un sistema de control basado en algoritmos de aprendizaje automático a sistemas no lineales e inestables y que estos se ajusten a los parámetros de diseño establecidos. El algoritmo de control de aprendizaje que se diseña en este trabajo es uno de aprendizaje iterativo, para el cual primero se diseña un controlador clásico basado en el modelo de espacio de estados y se toma como base para implementar el algoritmo de control seleccionado. Se compara el desempeño del controlador basado en el algoritmo de aprendizaje iterativo respecto al controlador clásico, evidenciando que el algoritmo por aprendizaje iterativo genera un error de estado estacionario que si bien es mínimo, le da una ventaja al controlador clásico, pues con este no se tiene error de estado estacionario. Finalmente se evidencia con este trabajo que la estrategia de control seleccionada a pesar de ser simple, cumple con la suficiente robustez para ser implementado en un problema complejo de control no lineal sin presentar mayor diferencia respecto a un controlador clásico.

Abstract

This paper presents the development of a controller based on machine learning algorithms for the nonlinear ball and beam system, showing how to build a control system based on machine learning algorithms to nonlinear and unstable systems and, these conform to the established design parameters. The learning control algorithm designed in this work is an iterative learning one, for which first a classical controller based on the state space model is designed and taken as a basis to implement the selected control algorithm. The performance of the controller based on the iterative learning algorithm is compared with the classical controller, showing that the iterative learning algorithm generates a steady state error that although it is minimal, it gives an advantage to the classical controller, because with this one there is no steady state error. Finally, this work shows that the selected control strategy, despite being simple, is robust enough to be implemented in a complex nonlinear control problem without presenting a major difference with respect to a classical controller.

Índice general

1. Introducción	1
2. Planteamiento del problema	3
3. Justificación	4
4. Antecedentes	5
5. Objetivos	8
5.1. Principal	8
5.2. Específicos	8
6. Marco teórico	9
6.1. Ball and Beam Quanser	9
6.1.1. Ecuación de movimiento no lineal	9
6.1.1.1. Convenciones del modelo	10
6.1.2. Modelo matemático del servo	10
6.1.3. Función de transferencia	11
6.2. Control por variables de estado	12
6.2.1. Los cuatro teoremas importantes	13
6.3. Control por aprendizaje Iterativo	14
6.3.1. Ley de actualización del ILC	15
6.3.2. Convergencia	15
7. Desarrollo metodológico	17
7.1. Implementación del modelo matemático en simulador	17
7.2. Controlador Clásico	18
7.2.1. Diseño en SIMULINK del Controlador Clásico	18
7.2.2. Diseño matemático y en MATLAB del Controlador Clásico	20
7.3. Controlador por aprendizaje iterativo	22
8. Resultados	26
8.1. Controlador Clásico	26
8.2. Controlador aprendizaje iterativo	29
8.3. Comparativa	32

9. Conclusiones	33
10. Trabajos Futuros	35
Bibliografía	36

Capítulo 1

Introducción

En los últimos años los algoritmos de aprendizaje automático han cautivado los ojos de los ingenieros y diseñadores de control, teniendo a lo largo de los años diversas aplicaciones de estos a los sistemas de control ya sea como un ayudante o un controlador principal. El avance de estos algoritmos ha mostrado como se puede llegar a controlar sistemas que en un inicio eran bastante complejos mediante técnicas clásicas, mostrando que no era necesario caracterizar un modelo matemático de algunos sistemas sino que en su mayoría se debía contar con bastantes datos del sistema para así entrenar un comportamiento con dichos algoritmos.

La implementación de algoritmos de aprendizaje automático como estrategia de control ha mostrado bastante ser de interés en trabajo con sistemas inestables, esto debido a la facilidad que se tiene de trabajar con estos sin necesidad de un modelo matemático, lo único que se requiere es una buena cantidad de datos del comportamiento del sistema para así entrenar el algoritmo. El sistema que se enseña en este documento es la planta *ball and beam* Quanser, el cual es un sistema inestable que consta de una barra y una esfera como dice su nombre cuyo objetivo de control es establecer la posición de la esfera en un punto deseado de la barra. En este documento se presentará el resultado de la comparación del desempeño de un controlador creado mediante una estrategia de control clásica con respecto a uno hecho mediante un algoritmo de aprendizaje automático.

En el capítulo 2 se encuentra el planteamiento del problema el cual establece una problemática del desarrollo de controladores mediante aprendizaje automático, además de brindar una descripción de la planta *ball and beam* Quanser que se va a trabajar, en el capítulo 3 se presenta la justificación la cual presenta como este proyecto esta guiada a la comunidad académica generando evidencias de implementación de algoritmos de aprendizaje automático como estrategia

de control a un sistema no lineal; en el capítulo 4 se encuentran los objetivos del proyecto, tanto el principal como los específicos los cuales están orientados con respecto al título del proyecto, en el capítulo 5 se encuentra el marco teórico en el cual se encuentran plasmados los conocimientos principales que se requieren para el desarrollo del proyecto, en el capítulo 6 está plasmado el estado del arte el cual brinda un contexto histórico de el avance que ha tenido los algoritmos de aprendizaje automático aplicado como estrategia de control, además de explicar detalladamente de que consiste la planta *ball and beam*, en el capítulo 7 se encuentra la metodología que se siguió para el desarrollo del proyecto, en el capítulo 8 se muestran los resultados, en el capítulo 9 se dan las conclusiones y finalmente en el capítulo 10 se da un contexto de los trabajos futuros que pueden surgir de este proyecto.

Capítulo 2

Planteamiento del problema

En los últimos años, los algoritmos de *Aprendizaje Automático* (AA, Machine Learning en inglés) han sido empleados como estrategias de control, sin embargo se han presentado diversos problemas como por ejemplo garantizar un buen desempeño en corto tiempo sin afectar el sistema [1] o al usar aprendizaje por refuerzo el problema radica en tener una buena interacción con el proceso para lograr los objetivos [2].

Los métodos de control por AA son usados preferiblemente en sistemas no lineales debido a dificultades en la obtención de los modelos matemáticos de los mismos, uno de los fallos de las aplicaciones a sistemas no lineales es que estos métodos fácilmente convergen a un mínimo local si la función de costo no es convexa con respecto a sus parámetros, fallando así en situaciones complejas de adaptación[3].

Con lo anterior se puede evidenciar que algunos algoritmos de AA pueden llegar a tener problemas al ser puestos a pruebas en sistemas no lineales. Entre los sistemas no lineales clásicos en este tipo de aplicaciones se encuentra el Ball and Beam [4], el cual consta de una barra y una esfera en donde el objetivo de control es estabilizar la esfera en un punto deseado de la barra, otro sistema clásico es el péndulo invertido [5] cuyo objetivo de control consiste en estabilizar y mantener una barra en posición vertical invertida.

Dada la problemática presentada llegamos a la siguiente pregunta de investigación ¿Cómo implementar un sistema de control basado en algoritmos de aprendizaje automático para el problema ball and beam Quanser que se ajuste a los parámetros de diseño?

Capítulo 3

Justificación

Con este proyecto se pretende generar un aporte a la comunidad académica, a partir de las evidencias generadas de la implementación de un algoritmo de aprendizaje automático aplicado a un sistema no lineal, aumentando el impacto social (en cuanto aplicaciones académicas e industriales) del control no lineal, el cual actualmente es de un 21 % según la revisión anual de control[6]. En este proyecto se busca realizar una comparación del desempeño de técnicas de control poco convencionales aplicadas a controladores no lineales con respecto a técnicas clásicas.

Este proyecto pretende incentivar el uso de las plantas Quanser para el aprendizaje de teoría del control, esto debido a que el sistema y control a mostrar en el proyecto serán implementados en esta. Las plantas Quanser tienen una gran variedad de usos, teniendo la posibilidad de poder representar distintos sistemas interesantes en los cuales se pueden aplicar diversas técnicas de control [7], mostrando un resultado tangible de los controladores desarrollados.

Este trabajo pretende ser un aporte al grupo de investigación en robótica GED de la Facultad de Ingeniería Electrónica de la Universidad Santo Tomas, ya que este se relaciona con las temáticas que se manejan en este grupo de investigación como inteligencia artificial y plataformas robóticas, por otro lado pretende aportar al currículo del programa de ingeniería electrónica en el área de control y automatización.

Capítulo 4

Antecedentes

El desarrollar controladores para sistemas altamente no lineales siempre ha sido una tarea de bastante trabajo, requiriendo el cálculo de un controlador complejo. Uno de los principales métodos para controlar estos sistemas es mediante algoritmos de aprendizaje automático, uno de estos algoritmos es el aprendizaje por refuerzo el cual se basa en el aprendizaje autónomo a través de interacciones de prueba y error con un entorno dinámico[8]. A través de la historia se han implementado varios de estos algoritmos en sistemas no lineales. En un principio estos algoritmos se usan de manera híbrida en conjunto con otros controladores, un ejemplo de esto es un controlador de navegación 3D realizado en el 2001 para un vehículo submarino autónomo, para este diseño se utiliza el aprendizaje por refuerzo en conjunto con una red neuronal, mostrando grandes resultados[9, 10].

Poco después el aprendizaje por refuerzo se convierte en el centro de atención como método de aprendizaje para el control robótico, esto se debe a su capacidad de adquirir acciones reflexivas y adaptativas, sin embargo, hay muchos problemas que resolver para poner el método en uso práctico, uno de esos problemas es el manejo de espacios de estados. Para esto, en 2002 se presenta una solución, se decide combinar este algoritmo con otro de actor/crítico que puede manejar el espacio continuo y el espacio de acción (espacio sobre el cual se mueve la planta)[11]. En el mismo año se plantea el método de mínimos cuadrados para solucionar problemas de predicción, se hace un esfuerzo para remediar algunas de las limitaciones del algoritmo y tanto este método como el anterior son puesto a prueba en un péndulo invertido, el cual es un sistema no lineal[12].

Pasa el tiempo y se presenta el algoritmo de aprendizaje por refuerzo utilizando redes neuronales, dando como resultado un aumento en la velocidad de aprendizaje, una ventaja de esta

combinación es que el algoritmo no necesita estimar una recompensa esperada debido a que el controlador recibe una señal de refuerzo real continua cada vez que realiza una acción [13]. Tiempo después se investigan diferentes entornos de simulación para poder hacer pruebas de los distintos controladores que se diseñaban para la época, dando como uno de los resultados que la herramienta MATLAB/SIMULINK es un entorno ideal para investigar y realizar diseños de teoría del control[14].

En el 2011 se implementa el método de aprendizaje supervisado en conjunto con controladores de redes neuronales para controlar una planta no lineal, en este experimento se evidencia la importancia de la recolección de datos tanto para entrenar el algoritmo como para diseñar la red neuronal, esta combinación da como resultado una perfecta estabilidad de la planta, sin embargo, no se obtuvieron mediciones debido a la poca cantidad de datos tomados para el diseño[15, 16]. Este mismo año se hace una prueba mediante condiciones reales de un nuevo algoritmo basado en técnicas de mapeo de células y métodos de aprendizaje por refuerzo para obtener la planificación óptima del movimiento de un vehículo considerando factores como: dinámica, cinemática y obstáculos. El algoritmo muestra una reducción del tiempo invertido en la etapa de aprendizaje, al utilizar una transformación del mapeo de célula a célula[17].

En el 2012 y 2013 se evidencia un avance en los algoritmos de aprendizaje automático implementados a los sistemas no lineales, para esta época se implementan controles de red neuronal adaptativo para controlar sistemas no lineales mostrando que estos controladores por redes neuronales adaptativos garantizan un equilibrio dinámico en presencia de incertidumbres del sistema, además de un rechazo a las perturbaciones externas demostrando su eficacia[3, 18]. Se plantea nuevamente el problema del tiempo de aprendizaje para el controlador de un robot utilizando el algoritmo de aprendizaje por refuerzo, por lo tanto se realizan estudios detallados combinando varias técnicas que incluyen la intervención humana para acelerar el proceso de aprendizaje por refuerzo, en particular, acciones del usuario como la demostración inicial de la tarea a realizar o dando una calificación del desempeño al robot, mejorando significativamente el tiempo de aprendizaje[19].

Para los últimos años se estudia en gran medida el algoritmo de aprendizaje por refuerzo, en especial el algoritmo Q-learning, el cual se puede llegar a optimizar mediante redes neuronales[20] o también se implementa este algoritmo en conjunto con un controlador PD para optimizar el control de sistemas no lineales[2, 21]. Para la actualidad aún se plantean usos del algoritmo de aprendizaje reforzado para optimizar controladores[1], además de que aún se encuentran diversos problemas como la dimensionalidad al aplicar el algoritmo de aprendizaje por refuerzo o el rastreo de funciones de aprendizaje al combinarlos con redes neuronales multicapa[22, 23].

En relación a la planta *ball and beam*, es un sistema no lineal el cual consta de una esfera y una barra por la cual circula la esfera, el objetivo de control es llevar la esfera a una posición deseada de la barra y que esta se estabilice en este punto, para esto se han implementado diversos controladores como: PID clásicos [24] usando diversas técnicas de sintonización y controladores un poco más complejos como en el 2006 que se propone un algoritmo de control adaptativo el cual se basa en una métrica del error que combina la trayectoria de la esfera y la estabilidad de la barra.[25] Para el 2010 se realiza entre muchos un controlador de lógica difusa el cual es comparado con un PID clásico, mostrando este una mejor respuesta en estado transitorio y estable[26]. Poco después se implementan al sistema *ball and beam* dos controladores diferentes, uno es un par de controladores difusos desacoplados en modo deslizante comparado con un algoritmo de optimización de colonia de hormigas (AOCH)[27], otro es un algoritmo de optimización de enjambre de partículas para optimizar un controlador PID y compararlo con un PID clásico[4]. Con lo anterior se puede ver cómo para esta planta se han implementado en diversas ocasiones algoritmos de lógica difusa ya sea para optimizar o para controlar[28].

Capítulo 5

Objetivos

5.1. Principal

Diseñar e implementar una estrategia de control de posición del sistema *ball and beam* Quanser basado en algoritmos de aprendizaje automático para evaluar su desempeño con respecto a un controlador clásico.

5.2. Específicos

- Caracterizar matemáticamente el sistema *ball and beam* Quanser del que se dispone en la Universidad Santo Tomas mediante la definición de su modelo lineal de espacio de estados para diseñar un controlador clásico y desarrollar simulaciones.
- Seleccionar un algoritmo de aprendizaje automático para el control de un sistema *ball and beam* haciendo uso del estado del arte.
- Simular el algoritmo seleccionado para el sistema *ball and beam* Quanser caracterizado, a partir de una interfaz en MATLAB
- Evaluar el desempeño del controlador diseñado respecto a una técnica de control clásico aplicada a la planta *ball and beam* Quanser respecto al error en estado estable y el tiempo de respuesta.

Capítulo 6

Marco teórico

6.1. Ball and Beam Quanser

La planta *Ball and Beam* Quanser es un sistema no lineal el cual consta de un servo que mueve una barra y sobre dicha barra se desplaza una esfera como se muestra en la figura 1.



FIGURA 1: Ball and Beam tomada de Quanser [29]

6.1.1. Ecuación de movimiento no lineal

Para encontrar la ecuación de transferencia de este sistema se debe modelar la ecuación de movimiento no lineal. Lo primero que se debe tener es el diagrama de cuerpo libre tal como se muestra en la Figura 2, con el cual se puede encontrar la ecuación de movimiento relacionando el movimiento de la esfera x y el ángulo de la barra α , esto se hace basándose en la primera ley

de Newton la suma de fuerzas que actúan sobre la esfera a través de la barra es la siguiente:

$$m_b \ddot{x}(t) = \sum F = F_{x,t} - F_{x,r} \quad (6.1)$$

En la anterior ecuación m_b es la masa de la esfera, x es el desplazamiento, $F_{x,r}$ es la fuerza de inercia de la esfera y $F_{x,t}$ es la fuerza de traslación generada por la gravedad. La fricción y la viscosidad son despreciadas.

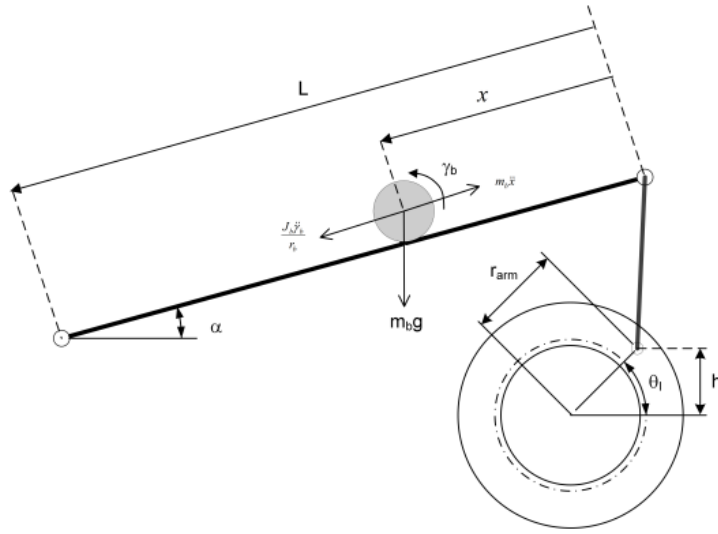


FIGURA 2: Diagrama de cuerpo libre Ball and Beam tomada de Quanser [29]

6.1.1.1. Convenciones del modelo

- Aplicar un voltaje positivo hace que el engranaje del servo se mueva en sentido positivo (en contra de las manecillas del reloj), el movimiento de la barra hacia arriba hace que la esfera se mueva en dirección positiva.
- La posición de la bola es cero cuando se encuentra en el centro de la barra.
- El ángulo del servo es cero cuando la barra se encuentra en paralelo al suelo.

6.1.2. Modelo matemático del servo

Para calcular la posición de la esfera con respecto al ángulo del servo, se requiere encontrar la relación del ángulo de la barra α y el ángulo de carga del engranaje rotatorio del servo θ_l , esto se hace usando el esquemático de la figura 2. Considerando los ángulos de la barra y el servo

para cambiar la altura de la barra h , se tiene la siguiente ecuación tomando los senos del ángulo de la barra:

$$\sin \alpha(t) = \frac{h}{L} \quad (6.2)$$

Y tomando los senos de los ángulos del servo se tiene la siguiente ecuación:

$$\sin \Theta_l(t) = \frac{h}{r_{arm}} \quad (6.3)$$

6.1.3. Función de transferencia

La figura 3 muestra el diagrama de bloques del sistema el cual esta compuesto por dos plantas, el servo y el ball and beam.

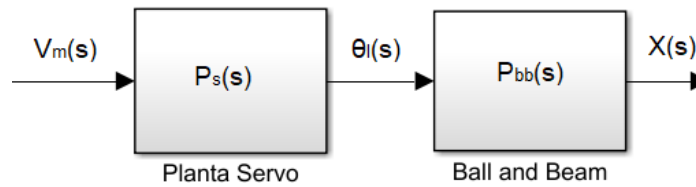


FIGURA 3: Diagrama de bloques ball and beam [29]

La ecuación de transferencia completa es:

$$P(s) = P_s(s)P_{bb}(s) \quad (6.4)$$

La ecuación de transferencia del ball and beam que describe el desplazamiento lineal de la esfera $x(t)$ con respecto al ángulo del engranaje rotatorio del servo $\Theta_l(t)$ es:

$$P_{bb}(s) = \frac{X(s)}{\Theta_l(s)} \quad (6.5)$$

$$P_{bb}(s) = \frac{K_{bb}}{s^2} \quad (6.6)$$

La ecuación de transferencia del servo que describe el ángulo del engranaje rotatorio del servo $\Theta_l(t)$ respecto al voltaje del entrada del servo $V_m(t)$ es la siguiente:

$$P_s(s) = \frac{\Theta_l(s)}{V_m(s)} \quad (6.7)$$

$$P_s(s) = \frac{K}{s(\tau s + 1)} \quad (6.8)$$

6.2. Control por variables de estado

Para poder realizar el control mediante este método se debe realizar una conversión de función de transferencia a variables de estado en donde el sistema pasa a tener una representación en función de las variables de estado, de tal manera:

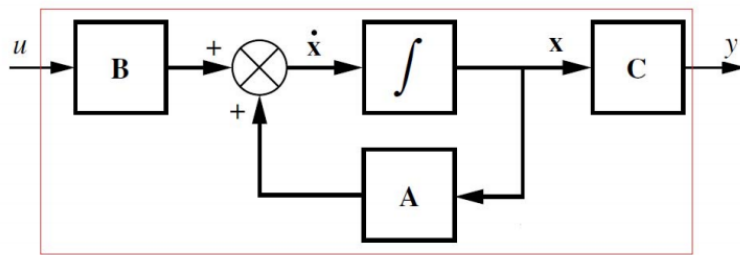


FIGURA 4: Sistema representado en variables de estado

En la figura 4 se puede ver el sistema representado en variables de estado en donde el par A y B deben ser controlables y C es la salida del sistema, además las ecuaciones de $\dot{x}$ y y son de la siguiente manera:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (6.9)$$

$$y(t) = Cx(t) \quad (6.10)$$

Los métodos clásicos de diseño no permiten especificar los polos a lazo cerrado en sistemas de orden superior a dos, esto es porque al realimentar la salida no se dispone del número suficiente de grados de libertad para ubicar de manera independiente todos los polos en lazo cerrado, por lo tanto la solución es realimentar todos los estados de la siguiente manera:

En la figura 5 se puede ver como la realimentación se realiza directamente desde la salida de los estados mediante la variable K , la cual es la variable de control, esta realimentación es negativa y la entrada esta representada en la ecuación 6.11:

$$u(t) = r(t) - Kx(t) \quad (6.11)$$

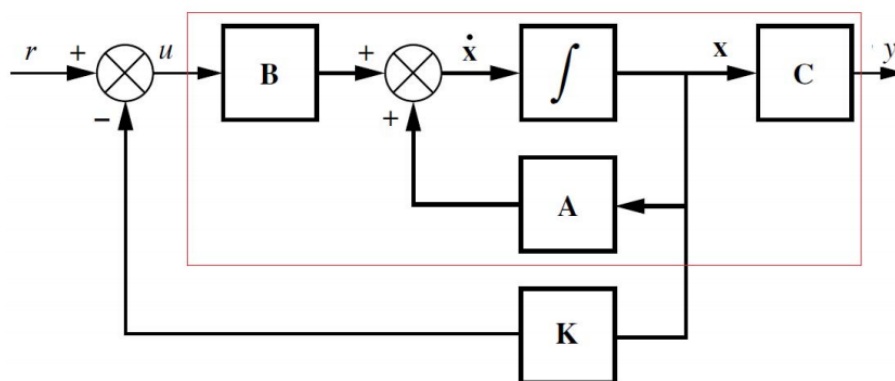


FIGURA 5: Sistema realimentado mediante variable de estados

En la ecuación 6.11 la letra r corresponde a la referencia del sistema, se tiene la variable K multiplicando los estados, además de esto las ecuaciones de $\dot{X}$ y y quedan de la siguiente manera:

$$\dot{x}(t) = (A - BK)x(t) + Br(t) \quad (6.12)$$

$$y(t) = Cx(t) \quad (6.13)$$

Por ahora se tiene que todos los estados se pueden medir para poder realimentarlos. Para que el sistema sea estable, se diseña K de tal manera que todos los autovalores de $A - BK$ tengan parte real positiva.

6.2.1. Los cuatro teoremas importantes

Para poder realizar el control mediante esta técnica es necesario tener en cuenta los siguientes cuatro teoremas:[30]

1. Asignación de autovalores. Si el par (A,B) es controlable, entonces mediante la asignación de estados (ecuación 6.11), donde K es un vector fila real constante, los autovalores de $A-BK$ pueden ser asignados arbitrariamente, siempre y cuando los autovalores complejos conjugados se asignen en pares.
2. La realimentación de estados pueden mover los polos, sin embargo no tienen ningún efecto sobre los ceros.

3. La controlabilidad de una planta es invariante con respecto a la realimentación de los estados. Por lo tanto si el par $[A,B]$ es controlable, para cualquier vector K el par $[(A-BK),B]$ también es controlable.

Si la planta no es controlable, no podrá convertirse en controlable mediante la realimentación de los estados.

4. La observabilidad de un sistema de lazo cerrado puede ser destruida por la realimentación de estados. Esto quiere decir que la observabilidad de un sistema no es invariante con respecto a la realimentación de los estados.

Si con la realimentación se mueven los polos de tal manera que se cancele con algún cero se pierde la observabilidad.

6.3. Control por aprendizaje Iterativo

El control por aprendizaje iterativo (ILC, Iterative Learning Control) es un controlador de tipo enchufe, es decir que puede ser conectado a un ciclo de control existente ya sea en paralelo o en serie. Ambos tipos de conexión son igual de válidas y en este caso se usará el de conexión en paralelo como se muestra en la figura 6.

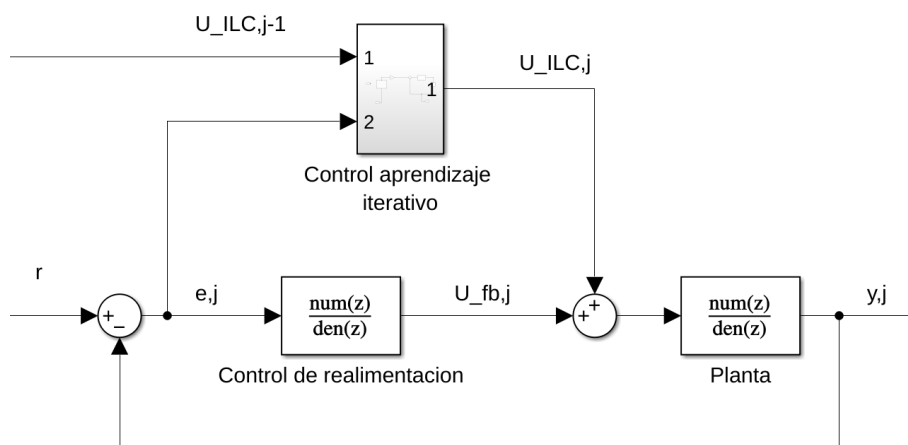


FIGURA 6: Diagrama en paralelo del proceso de control por aprendizaje iterativo

Existen diferentes paradigmas de diseño para el ILC, como los son el diseño de procesos repetitivos lineales, diseño de modelos internos, norma diseño óptimo y diseño en el dominio de la frecuencia, en este caso se trabajara con el diseño en el dominio de la frecuencia, pues una de las ventajas de usar este tipo de diseño es que mucho de los controladores de aprendizaje

de dominio de frecuencia son sintonizables como controladores proporcionales y por ende no requiere un conocimiento profundo del sistema. [31]

6.3.1. Ley de actualización del ILC

La formula de actualización en tiempo continuo para el diseño en el dominio de la frecuencia es la siguiente:

$$U_{ILC,j+1}(s) = Q(s)(U_{ILC,j}(s) + L(s)e_j(s)) \quad (6.14)$$

Donde:

$$e_j(s) = r(s) - y_j(s) \quad (6.15)$$

$$y_j(s) = T(s)r(s) + P_S(s)U_{ILC,j}(s) \quad (6.16)$$

En las ecuaciones 6.14, 6.15 y 6.16, $e_j(s)$ es la señal de seguimiento del error entre la referencia $r(s)$ y la salida $y_j(s)$, $T(s)$ es la función de sensibilidad complementaria, y $P_S(s)$ es la función de sensibilidad de la planta, definida como la relación entre la señal de control actual del ILC $U_{ILC,j}(s)$ y la salida del sistema $y_j(s)$, y j es el índice de iteración.

Los elementos de diseño de la ecuación 6.14, incluye un filtro Q y un filtro L . $Q(s)$ generalmente es diseñado como un filtro pasa bajas con el objetivo de limitar el ancho de banda de aprendizaje y aportar robustez al sistema, mientras que $L(s)$, también conocido como el filtro de aprendizaje, está diseñado como filtro pasa altas para maximizar el ancho de banda de aprendizaje y la tasa de convergencia. [32]

6.3.2. Convergencia

El objetivo de diseñar $Q(s)$ y $L(s)$ es asegurar la convergencia monótona de la señal de control, sustituyendo las ecuaciones 6.15 y 6.16 en 6.14 y reorganizando términos da como resultado la ecuación:

$$U_{ILC,j+1}(s) = Q(s)(1 - L(s)P_S(s))U_{ILC,j}(s) + Q(s)L(s)r(s) \quad (6.17)$$

Una convergencia monótona de $U_{ILC,j+1}(s)$ a $U_{ILC,j}(s)$ se logra asegurando:

$$\|Q(s)(1 - L(s)P_S(s))\|_\infty < 1 \quad (6.18)$$

Capítulo 7

Desarrollo metodológico

7.1. Implementación del modelo matemático en simulador

Para el modelo matemático del Ball and Beam y del servo se usa el previamente visto en el marco teórico, el cual es tomado directamente de Quanser. Se realiza la implementación de dichos modelos en el software SIMULINK de MATLAB teniendo lo siguiente:

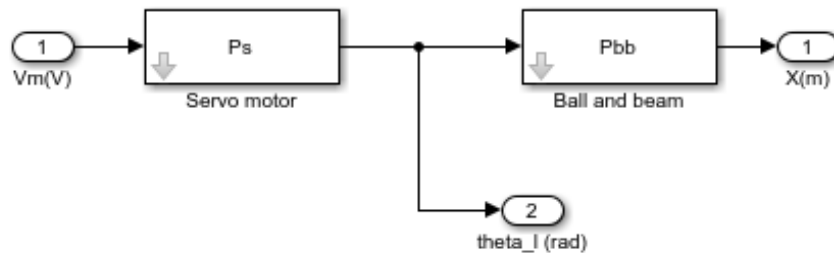


FIGURA 7: Planta ball and beam implementada en SIMULINK

En la figura 7 se muestra el sistema de la planta que se trabajará en el proyecto. V_m hace referencia al voltaje de entrada del servo cuya salida es un ángulo en radianes, el cual entra al Ball and Beam y este tiene como salida la posición de la esfera.

7.2. Controlador Clásico

7.2.1. Diseño en SIMULINK del Controlador Clásico

Para el diseño del controlador clásico se propone un sistema de control como se muestra en la figura 8, el cual consiste en un control interno para el servo motor y otro control externo para el Ball and Beam, en este sistema se tiene como entradas la posición deseada de la esfera y como salida la posición de la esfera.

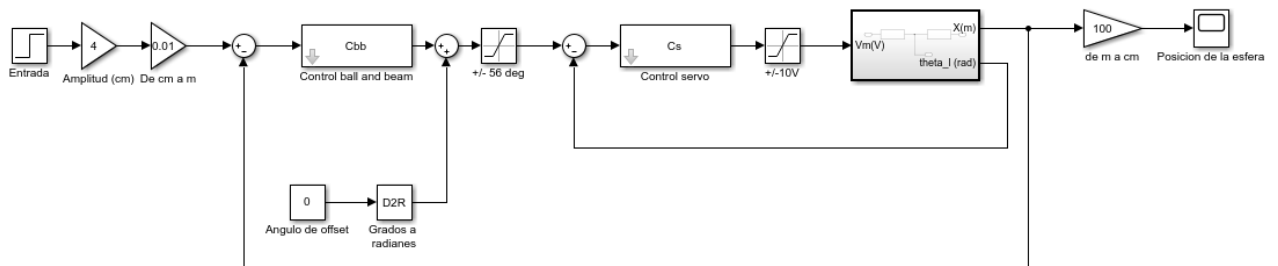


FIGURA 8: Sistema de control implementado en SIMULINK

Tanto para el controlador del servo como el controlador del ball and beam se realiza el diseño de un controlador por variables de estado, tal como se muestra en el marco teórico lo primero que se hace es construir el modelo representado en variables de estado en SIMULINK, se toma como base el sistema mostrado en la imagen 7, dando como resultado el siguiente modelo:

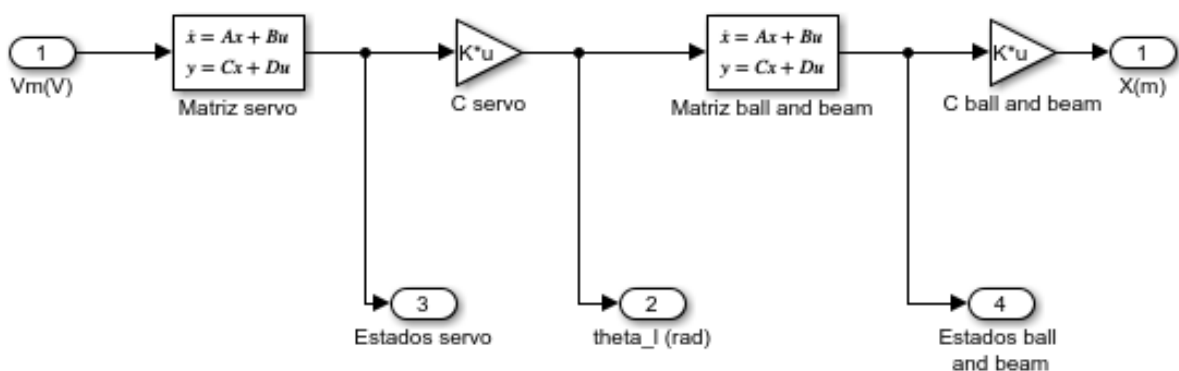


FIGURA 9: Planta ball and beam implementada en SIMULINK mediante variables de estado

Debido que el diseño por variables de estado es un diseño matricial, se debe tener en cuenta como funciona el bloque de matriz en SIMULINK pues se requiere tener la variable C de la

matriz de manera externa como se muestra en la imagen 9, ya de esto depende la medición de la señal de salida hacia el controlador, para ello se debe indicar en la matriz el comando **eye** como se muestra en la imagen 10, y de tal forma poder tener de manera independiente la variable C y así tomar la medición. Adicionalmente como la variable D, son ceros, se usa el comando "zeros" para llenarla.

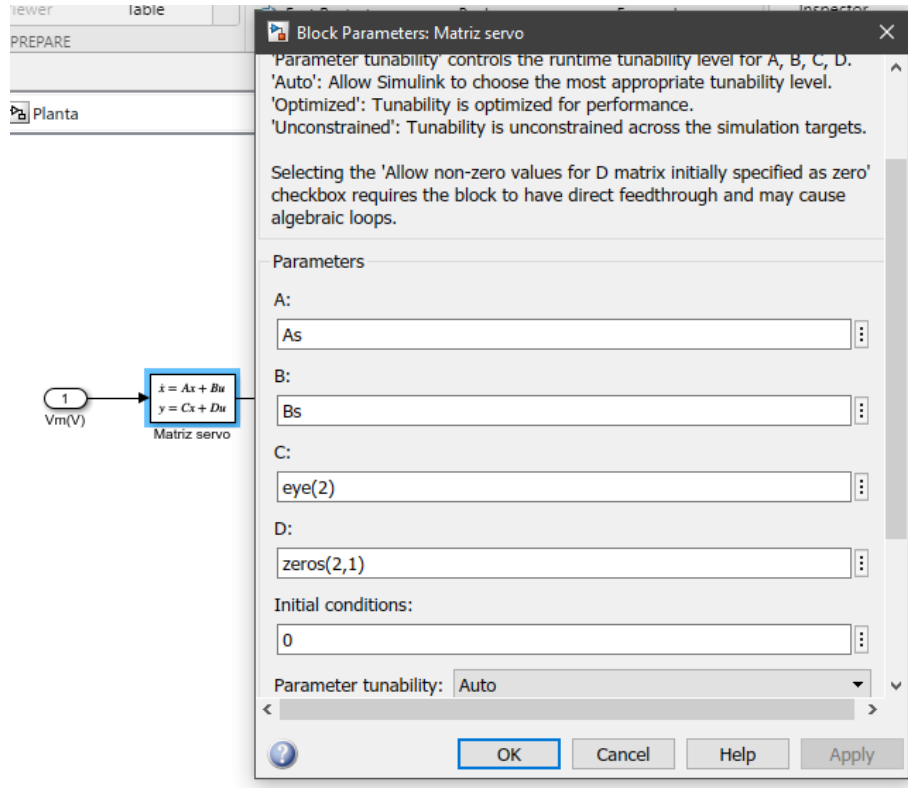


FIGURA 10: Configuración de las matrices de la planta

Para el diseño general del sistema, se toma como referencia el mostrado en la imagen 8, quedando de la siguiente manera:

En la imagen 11, inicialmente se tiene una señal de entrada, dicha señal corresponde a un escalón y representa la referencia o la posición en la que se desea que quede la esfera en la barra. Seguido a la entrada se evidencia un precompensador, este existe tanto para el ball and beam como para el servo, este hace parte de los diseños de realimentación de estados y su objetivo es acondicionar la señal al sistema, dicho valor se calcula de la siguiente manera:

$$Precompensador\ Ballandbeam = \frac{1}{Thao_{0bb}} \quad (7.1)$$

$$Precompensador\ Servo = \frac{1}{Thao_{0s}} \quad (7.2)$$

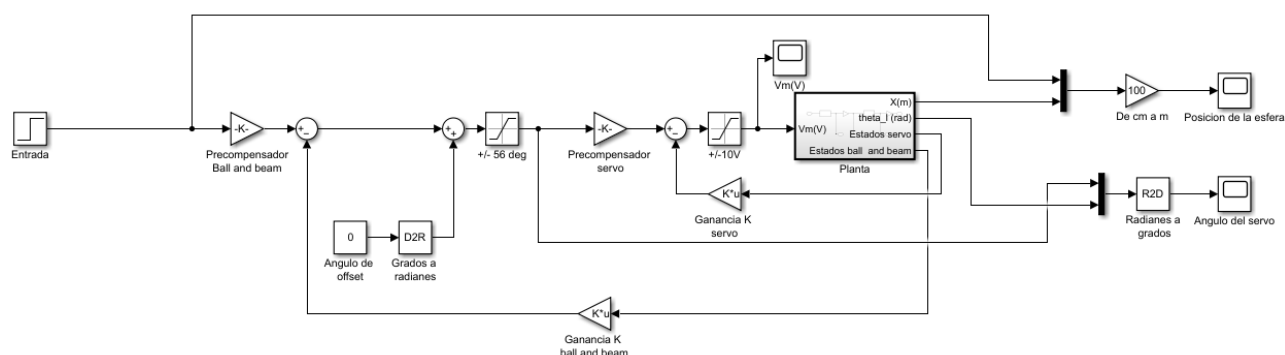


FIGURA 11: Sistema de control implementado en SIMULINK mediante variables de estado

Adicional a los precompensadores se evidencia otro bloque que también nos ayuda a acondicionar la señal y son las saturaciones, existe una para el servo como para el ball and beam, el objetivo de estas es evitar una señal mayor a el valor indicado, pues en el montaje en físico, una señal mayor podría dañar la planta. La primera saturación es una de ángulo, esta va desde -56 a 56 grados y corresponde a el máximo ángulo que se puede alcanzar con el Servo QUANSER, la segunda saturación es una de voltaje, esta va de -10V a 10V y corresponde al máximo voltaje permitido por el motor del Servo QUANSER.

Se evidencia un bloque de planta cuya entrada es el voltaje del servo y cuenta con 4 salidas, las cuales corresponden a las vistas en la imagen 9, Estas señales corresponden a las señales de salida tanto en posición de la bola como ángulo de la barra y a los estados del servo y del ball and beam que posteriormente llegaran a una ganancia K que los ira ajustando en el lazo de realimentación.

Cabe resaltar que la señal de entrada esta expresada en centímetros pues así trabaja el sistema de control, sin embargo en el código se asigna el valor en metros y la salida también se evidencia en metros.

7.2.2. Diseño matemático y en MATLAB del Controlador Clásico

Para el diseño del controlador en el simulador se requiere mediante MATLAB diseñar y calcular las variables y las ecuaciones de transferencia que se usan en SIMULINK, esto se hace de la siguiente manera:

Inicialmente para la señal de entrada se define una variable cuyo valor va a ser la referencia y va a estar multiplicado por un factor de 0.01 con el fin de pasar el valor de metros a cm pues el

modelo trabaja en cm. Seguido a esto se procede a definir la ecuación de transferencia del servo, para esto se utiliza la ecuación 6.8. los valores de K y $thao$ dependen del modelo del servo, para el QUBE SERVO de QUANSER, los valores son los siguientes: $K = 1,5$ y $Thao = 0,025$. Teniendo los datos de K y $Thao$, se procede a calcular la ecuación de transferencia, esto se hace de la siguiente manera:

Se crean dos funciones de transferencia:

$$Ps1 = \frac{K}{Thao * s + 1} \quad (7.3)$$

$$Ps2 = \frac{1}{s} \quad (7.4)$$

Luego estas dos funciones se multiplican y nos da como resultado la función de transferencia del servo vista en la ecuación 6.8

```
Thau = 0.025;
K = 1.5;

Ps1 = tf(K, [Thau 1]);
Ps2 = tf(1, [1 0]);
Ps = Ps1*Ps2;
```

FIGURA 12: Código de la ecuación de transferencia del servo

Para la crear la función de transferencia del ball and beam, se usa la ecuación 6.6, en donde se tiene que $K_{bb} = 1$ y para este caso el código es el siguiente:

```
K_bb = 1;

Pbb = tf(K_bb, [1 0 0]);
```

FIGURA 13: Código de la ecuación de transferencia del ball and beam

Luego se especifican los valores objetivos del control a diseñar, los cuales son un tiempo de establecimiento menor a 3 segundos y porcentaje de sobrepaso menor a 5 %, y se procede a hacer el código para el controlador del ball and beam el cual es similar al del servo únicamente cambiando las funciones de transferencia:

En la primera linea del código lo que se hace es convertir la ecuación de transferencia de la planta en sistema por espacio de estados y seguido a esto se obtienen las matrices A, B, C y D

```

C_bb=ss(Pbb);
[Abb,Bbb,Cbb,Dbb]=ssdata(C_bb);
eig(Abb);
pole(Pbb);
ob_bb=[Cbb;Cbb*Abb];
rank(ob_bb)
co_bb=[Bbb Abb*Bbb];
rank(co_bb)

```

FIGURA 14: Código de diseño del controlador parte 1

del sistema en espacio de estados con el comando `ssdata`. Luego se obtienen los eigenvalores de la matriz A y se obtiene los polos de la ecuación de la planta con el objetivo de analizar en donde ubicar los polos para el controlador. Seguido a esto se crean las matrices de observabilidad y controlabilidad, para su respectivo análisis.

```

polos_bb=[-4 -2];
kbb=place(Abb,Bbb,polos_bb);
A1c_bb=Abb-Bbb*kbb;
eig(A1c_bb);
tvbb=ss(A1c_bb,Bbb,Cbb,Dbb);
tbb=tf(tvbb);
tbb=zpk(tbb);

```

FIGURA 15: Código de diseño del controlador parte 2

En esta sección se definen los polos para el control y se crea la matriz K de realimentación de estados, seguido a esto se calcula la matriz de autovalores y los los eigenvalores de la misma, con el objetivo de analizar la homogeneidad. Se convierte la matriz obtenida en espacio de estados, luego en la forma de función de transferencia y en la forma ZPK, estas ultimas dos son con el objetivo de graficar la función e ir validando si el valor de los polos seleccionados es el correcto y cumple con los objetivos del control y adicional analizar la función con sus ceros y polos.

Finalmente se calcula la ganancia DC de la función en espacio de estados, pues con este valor es con el que se calculan los pre-compensadores. La ganancia DC corresponde al valor de $Thao_0$, el cual se usa en las ecuaciones 7.1 y 7.2.

7.3. Controlador por aprendizaje iterativo

Se selecciona este tipo de controlador basado en su modelo en el dominio de la frecuencia, pues este tiene mejor funcionamiento para sistemas de seguimiento en donde la referencia es repetidamente proporcionada y esta definida para una trayectoria conocida. Por ejemplo sistemas

sistemas con movimientos de tipo punto a punto, impresoras (movimiento del papel y del carrete), escáner, entre otros. También se selecciona este tipo de algoritmo pues la convergencia del lazo del ILC puede ser garantizada si se cumplen las reglas del diseño.

```
controls = (tf(tvss))/((Ps)-(Ps*(tf(tvss))));
controlb = (tf(tvbb))/((Pbb)-(Pbb*(tf(tvbb))));

Psd = c2d(Ps,0.05,'foh');
controlsd = c2d(controls,0.05,'foh');

Pbbd = c2d(Pbb,0.1,'foh');
controlbd = c2d(controlb,0.1,'foh');
```

FIGURA 16: Código de diseño del controlador iterativo 1

Para empezar a diseñar el controlador por aprendizaje iterativo se obtienen las ecuaciones en forma individual, es decir la ecuación de control y de la planta por separado, para calcular la ecuación de control obtenida mediante el método de espacio de estados se emplea el código expuesto en la imagen 16, este código corresponde al despeje de la ecuación de transferencia en lazo cerrado.

Seguido a esto se convierten las ecuaciones de transferencia de continuo a discreto, pues el control por aprendizaje iterativo es aplicable a sistemas discretos. Es importante tener claro que para definir la frecuencia de muestreo para el conversor de continuo a discreto se debe hacer mediante el teorema de Nyquist.

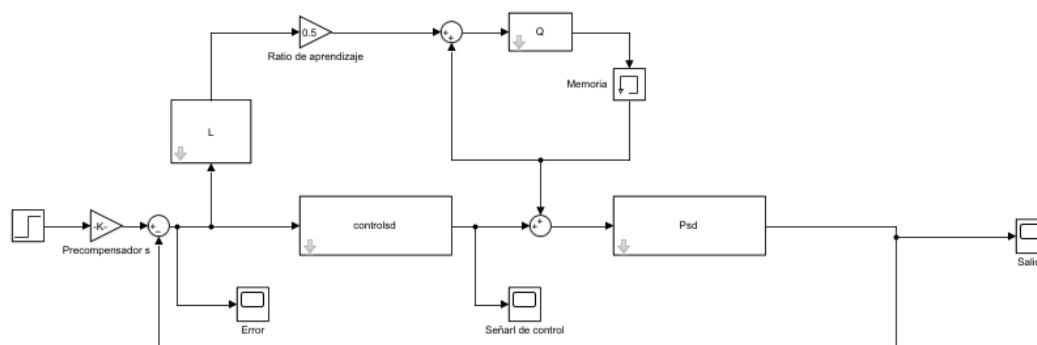


FIGURA 17: Diseño del control por aprendizaje iterativo

De acuerdo al marco teórico se realiza el montaje del controlador en SIMULINK, el cual es similar para el ball and beam y para el servo. El funcionamiento de este controlador es tomar la señal del error y filtrarla con el objetivo de retro-alimentar la señal de control y corregirla en cada iteración.

Para calcular el filtro L , se tiene la ecuación $L = H^{-1}$, en donde inicialmente se debe calcular H la cual se llama sensibilidad del sistema y cuya ecuación es la siguiente:

$$H = \frac{P}{1 + PC} \quad (7.5)$$

En la ecuación 7.5, la P hace referencia a la ecuación de transferencia de la planta y C hace referencia a la ecuación de transferencia del controlador. De acuerdo a la naturaleza de este controlador y que tiene mas polos que ceros, se garantiza la causalidad. Ya teniendo la ecuación de sensibilidad se procede a calcular el valor de L, sin embargo este filtro requiere información del futuro, para solucionar esto se puede diseñar un filtro pasa altas mediante el algoritmo ZPETC o diseñando un filtro pasa altas garantizando que este sea propio.

Para el diseño del filtro L, se utiliza la función de diseño de filtros de MATLAB como se muestra en la imagen 18, a continuación.

```
hpFilt = designfilt('highpassiir', 'FilterOrder', 8, ...
    'PassbandFrequency', 75e3, 'PassbandRipple', 0.2, ...
    'SampleRate', 200e3);

data = rand(1,1);
L = filter(hpFilt,data);
```

FIGURA 18: Código de diseño del filtro L

Respecto al filtro Q, este es necesario ya que el filtro L es un filtro pasa altas, lo que hace que las perturbaciones en altas frecuencias sean amplificadas y por ello el filtro Q debe ser un filtro pasa bajas. Para el diseño de este filtro también se hace uso de la función de diseño de filtros de MATLAB pero esta vez para un filtro pasa bajas como se muestra en la imagen 19.

Para definir el ancho de banda de este filtro, se debe tener en cuenta el ancho de banda del lazo abierto del sistema, pues el ancho de banda del filtro Q debe ser cercano a este valor, para este caso corresponde a un ancho de banda de 2,5Hz.

```
lpFilt = designfilt('lowpassfir', 'PassbandFrequency', 0.05, ...
    'StopbandFrequency', 0.3, 'PassbandRipple', 0.5, ...
    'StopbandAttenuation', 65, 'DesignMethod', 'kaiserwin');

data = rand(1,1);
Q = filter(lpFilt,data);
```

FIGURA 19: Código de diseño del filtro Q

Por ultimo, el ratio de aprendizaje corresponde a un valor entre 0 y 1, en donde si el ratio es igual a 1, tendrá un aprendizaje rápido pero menos minucioso, es decir que no tendrá el

mejor resultado, en cambio si este valor es muy bajo, tardara demasiado en encontrar el valor adecuado, lo recomendable es establecerlo en 0.5 como se hizo en este caso.

Con lo anterior ya explicado, se procede a realizar el diseño completo del sistema quedando como se muestra en la imagen 20, en donde se evidencia un lazo de control externo con su respectivo ILC para el ball and beam y un lazo de control interno con su respectivo ILC en un subsistema para el servo.

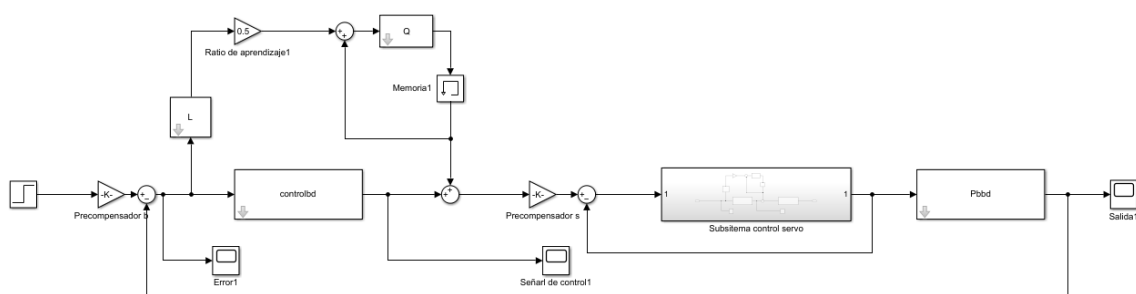


FIGURA 20: Diseño completo del control por aprendizaje iterativo para el ball and beam

Capítulo 8

Resultados

8.1. Controlador Clásico

Para la prueba inicial de este controlador se define la entrada como 1 y se analiza si cumple con los criterios de tiempo de establecimiento y el sobrepaso definidos.

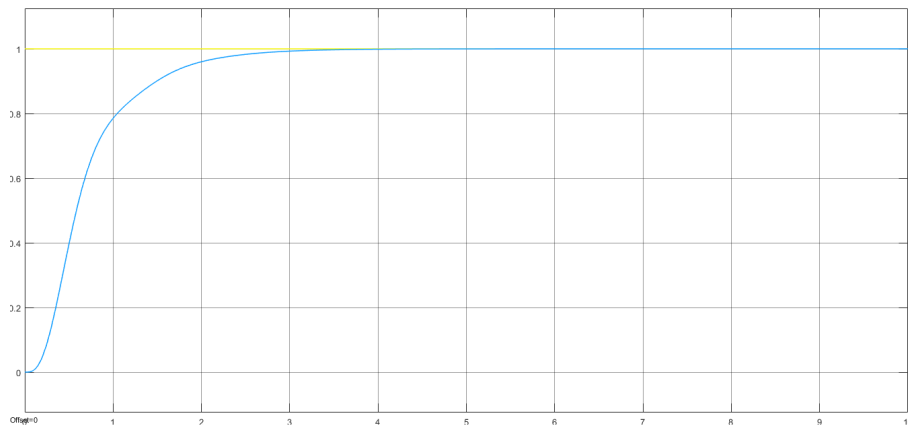


FIGURA 21: Gráfica de salida vs entrada de la posición de la esfera con controlador clásico

En la figura 22 se tiene que la línea amarilla corresponde a la referencia la cual es 1 y la línea azul corresponde a la salida en la cual se evidencia que el tiempo de establecimiento es aproximadamente 2.5 segundos y un porcentaje de sobrepaso de 0.

Adicionalmente se tiene un tiempo de simulación de 10 segundos, sin embargo también se hace una prueba con 500 segundos de ejecución para evidenciar que en tiempos altos el sistema no

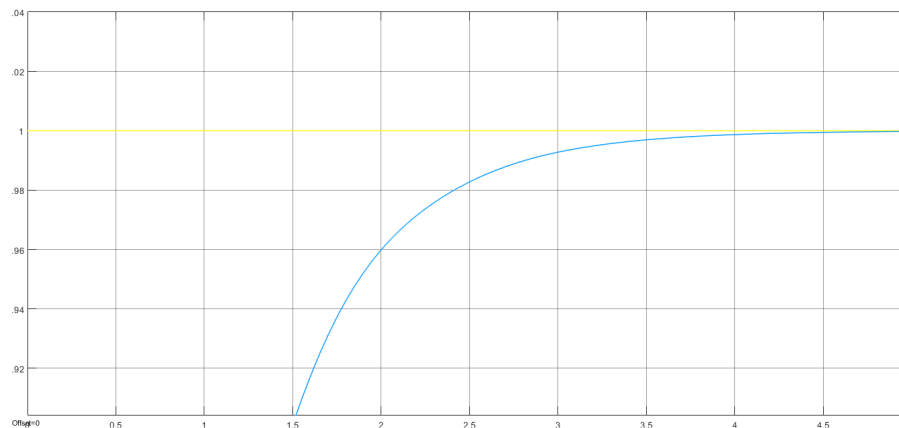


FIGURA 22: Gráfica de salida vs entrada de la posición de la esfera con controlador clásico

se hace inestable, pues este fue uno de los principales problemas que se tuvo durante el diseño, esto se muestra en la figura 23.

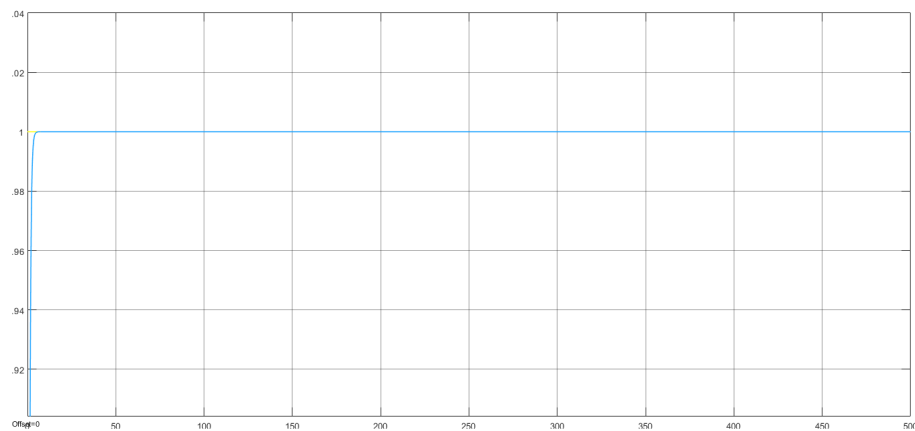


FIGURA 23: Gráfica de salida vs entrada de la posición de la esfera con controlador clásico con tiempo de 500s

Después de evidenciar que el control diseñado funciona correctamente, se procede a hacer dos pruebas:

La primer prueba consiste en variar el valor de la referencia y evidenciar los tiempos que toma en llegar al valor indicado, además de tomar datos del sobrepaso en caso de presentarse y el error de estado estacionario, esta información se plasma en la tabla 1.

En la tabla 1, se evidencia que para el sistema con controlador clásico, independiente del valor de referencia, el tiempo de establecimiento T_s , el porcentaje de sobrepaso P_s y el error de estado estacionario E_{ss} , no cambian.

Referencia	Ts (s)	Ps (%)	Ess
1	2.45	0	0
2	2.45	0	0
3	2.45	0	0
4	2.45	0	0
5	2.45	0	0
6	2.45	0	0
7	2.45	0	0
8	2.45	0	0
9	2.45	0	0
10	2.45	0	0

CUADRO 1: Datos de simulación del controlador clásico con diferentes referencias

La segunda prueba consiste en probar una entrada compuesta por un incremento cada 5 segundos de la referencia, hasta un total de 4 incrementos con el objetivo de evidenciar su comportamiento con una entrada cambiante, esta prueba se ve reflejada en la figura 24

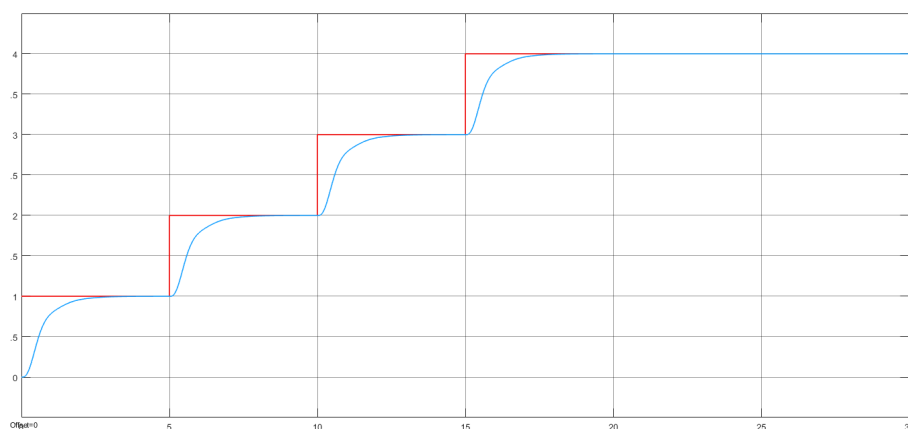


FIGURA 24: Gráfica de salida vs entrada de la posición de la esfera con controlador clásico con entrada variable

Una vez evidenciadas las señales de salida para la posición de la esfera se procede a validar las señales de salida para el ángulo del servo, en este caso se evidencia que independiente del valor de la referencia, la forma de la señal es la misma y se muestra en la imagen pero los valores máximos y mínimos cambian, para evidenciar esto, se toman los datos y se registran en la tabla 2.

En la tabla 2 el valor de diferencia se calcula de la siguiente forma: $Diferencia = \Theta_{max} - \Theta_{min}$, y de esta forma se evidencia que la diferencia tiene un crecimiento lineal de 4.37° , adicionalmente se evidencia que el tiempo de establecimiento de la gráfica del ángulo del servo es similar al de la gráfica de posición.

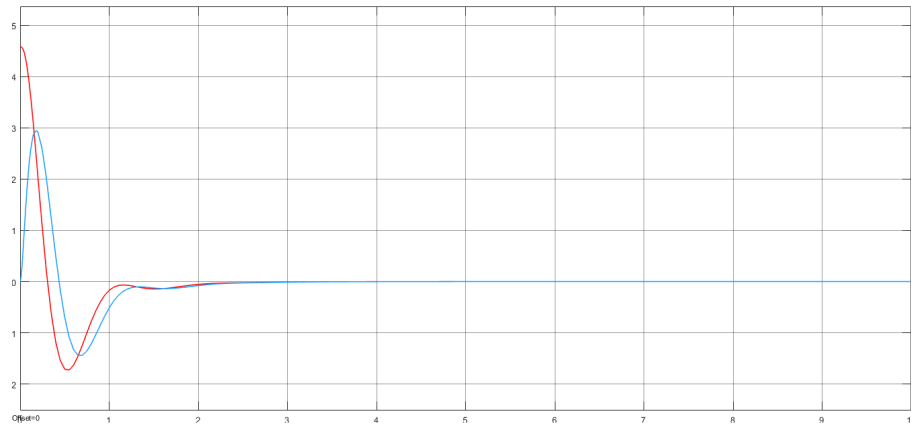


FIGURA 25: Gráfica de salida vs entrada del ángulo del servo con controlador clásico

Referencia	Θ_{min}	Θ_{max}	Diferencia
1	-1.44	2.93	4.37
2	-2.88	5.86	8.74
3	-4.32	8.79	13.11
4	-5.76	11.73	17.49
5	-7.2	14.66	21.86

CUADRO 2: Tabla de datos de ángulos de salida con controlador clásico

8.2. Controlador aprendizaje iterativo

Al igual que con el controlador clásico, la primera prueba para este controlador consiste en tener una entrada con un valor de 1, y se procede a analizar la salida si cumple con los criterios definidos para el control.

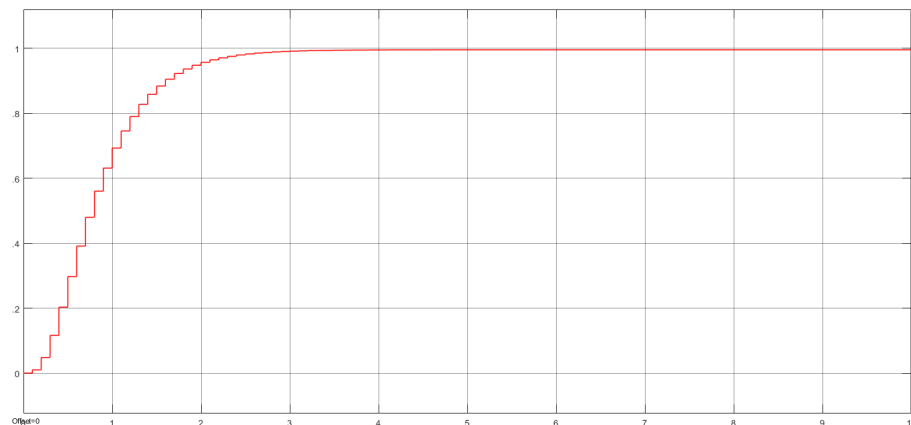


FIGURA 26: Gráfica de salida de posición de la esfera con control por aprendizaje iterativo

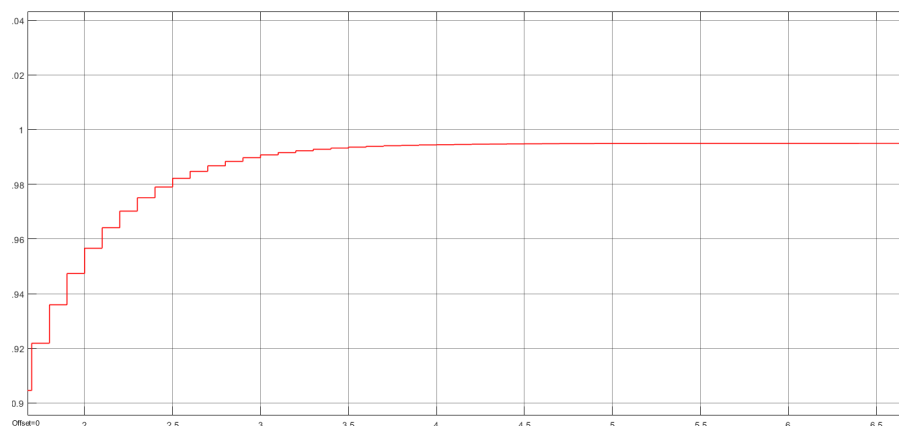


FIGURA 27: Gráfica de salida de posición de la esfera con control por aprendizaje iterativo

En la figura 27 se evidencia que la salida presenta un error de estado estacionario de 0.005 lo cual es un valor bastante bajo para la implementación para la cual se diseñó el controlador, además de esto se puede ver que cumple con el tiempo de establecimiento el cual es de 2.5 segundos y no cuenta con sobrepaso.

Las gráficas anteriores se ejecutaron hasta un tiempo de 10 segundos, sin embargo al igual que con el controlador clásico se realiza una simulación con un tiempo de 500 segundos con el objetivo de evidenciar que a tiempos altos el sistema no se hace inestable, esto se muestra en la imagen 28.

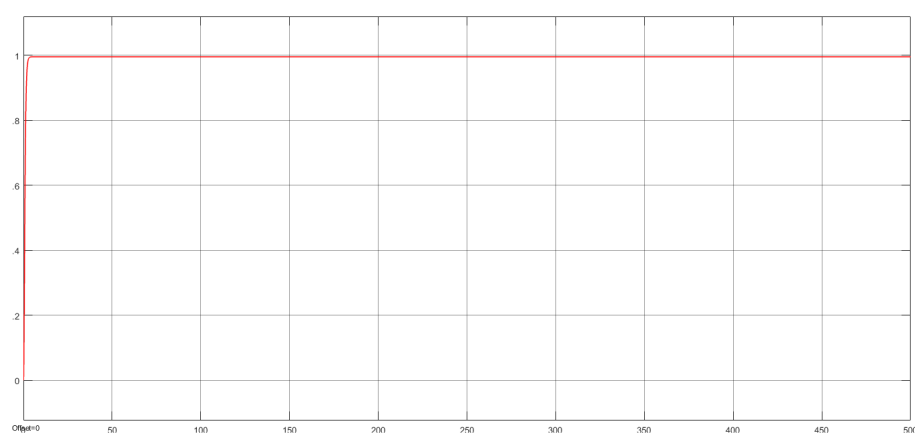


FIGURA 28: Gráfica de salida de posición de la esfera con control por aprendizaje iterativo con tiempo de 500s

Con lo anterior se logra evidenciar que el controlador tiene un funcionamiento correcto y se procede a variar el valor de la referencia y evidenciar los tiempos que toma en llegar al valor

indicado, además de tomar datos del sobrepaso en caso de presentarse y el error de estado estacionario, esta información se plasma en la tabla 3.

Referencia	Ts (s)	Ps (%)	Ess
1	2.5	0	0.005
2	2.5	0	0.01
3	2.5	0	0.015
4	2.5	0	0.02
5	2.5	0	0.025
6	2.5	0	0.03
7	2.5	0	0.035
8	2.5	0	0.04
9	2.5	0	0.045
10	2.5	0	0.5

CUADRO 3: Datos de simulación del controlador por aprendizaje iterativo con diferentes referencias

En la tabla 3, se evidencia que para el sistema con controlador por aprendizaje iterativo, independiente del valor de referencia, el tiempo de establecimiento T_s , el porcentaje de sobrepaso P_s y el error de estado estacionario E_{ss} en porcentaje, no cambian, siendo este ultimo un 0.5 %. Para este controlador también se realizó la prueba con múltiples entradas evidenciando que el comportamiento es igual que con el controlador clásico.

Una vez evidenciadas y analizadas las señales de salida para la posición de la esfera, se procede a validar las señales de salida para el ángulo del servo, en este caso se evidencia que independiente del valor de la referencia al igual que con el controlador clásico, la forma de la señal es la misma y se muestra en la imagen, sin embargo los valores máximos y mínimos del ángulo cambian, para evidenciar esto, se toman los datos y se registran en la tabla 4.

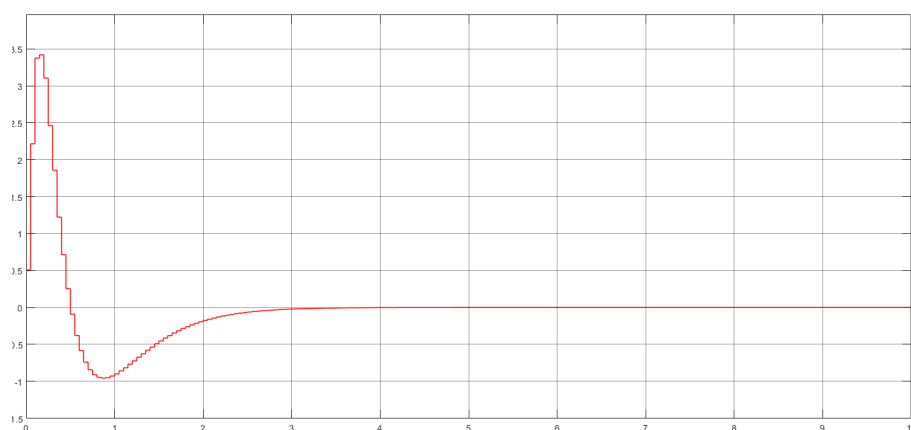


FIGURA 29: Gráfica de salida del ángulo del servo con control por aprendizaje iterativo

Referencia	Θ_{min}	Θ_{max}	Diferencia
1	-0.95	3.42	4.37
2	-1.91	6.84	8.75
3	-2.86	10.26	13.12
4	-3.82	13.68	17.5
5	-4.78	17.1	21.88

CUADRO 4: Tabla de datos de ángulos de salida con controlador por aprendizaje iterativo

En la tabla 4 el valor de diferencia se calcula de la siguiente forma: $Diferencia = \Theta_{max} - \Theta_{min}$, y de esta forma se evidencia que la diferencia tiene un crecimiento lineal de a 4.37° aproximadamente al igual que con el controlador clásico, adicionalmente se evidencia que el tiempo de establecimiento de la gráfica del ángulo del servo de aproximadamente 3 segundos.

8.3. Comparativa

Para realizar la comparativa de los dos diseños se plantea la tabla 5 en la cual se registran los datos de tiempo de establecimiento o tiempo de respuesta y el error en estado estable o en estado estacionario.

Controlador	Ts (s)	Error (%)
Clásico	2.45	0
Aprendizaje iterativo	2.5	0.5

CUADRO 5: Comparativa controlador clásico vs aprendizaje iterativo

Con la tabla 5, se puede ver que en cuanto al tiempo de establecimiento la diferencia no es mucha, sin embargo el controlador clásico es mejor aunque la diferencia es casi insignificante, sin embargo respecto al error en estado estable, el controlador clásico no presenta, mientras que el controlador por aprendizaje automático si tiene, aunque es un error pequeño este dato da mas información de cual controlador se puede definir como mejor, siendo en este caso el controlador clásico diseñado por variables de estado.

Capítulo 9

Conclusiones

En este trabajo se presento el diseño de un controlador de posición para el sistema ball and beam Quanser con un algoritmo de aprendizaje iterativo, el cual puede ser visto como una versión simplificada de una estrategia de aprendizaje por refuerzo [33], este último haciendo parte del aprendizaje automático cumpliendo con el objetivo principal del trabajo. Adicionalmente, se evaluó el desempeño del controlador diseñado respecto a un controlador clásico, el cual fue diseñado mediante el modelo por variables de estados, dando como resultado que en la simulación, el controlador clásico es un poco mejor que el controlador por aprendizaje iterativo.

Se evidenció que aunque se uso el modelo matemático de la planta ball and beam QUANSER proporcionado por el fabricante, estas difieren un poco respecto a las que se tiene en físico, ya que debido a su incorrecta manipulación estas han variado un poco en sus constantes físicas, como la resistencia o movilidad de los engranajes, haciendo que aunque se diseñara un controlador para estas plantas, el resultado no sera igual a la simulación. Aunque mediante simulaciones se halla evidenciado que el controlador diseñado por espacio de estados, halla dado un error de estado estacionario 0, esto en la practica no sera totalmente igual.

Se realizó una investigación de diferentes estrategias de control para sistemas no lineales basadas en algoritmos de aprendizaje automático, en las cuales se evidenció el uso de algoritmos inspirados en la naturaleza, como el algoritmo de colonia de hormigas u otras estrategias como lo es el aprendizaje por refuerzo o el aprendizaje difuso, sin embargo, no se encontró muchas implementaciones de un algoritmo de aprendizaje iterativo para sistemas no lineales y esta fue la principal razón por la cual se opto por usar esta estrategia, ya que por la simplicidad del modelo, es un buen aporte a la comunidad académica pues se evidencia que una estrategia

de control como esta, puede ser implementada para sistemas complejos como lo es el ball and beam.

Este trabajo muestra la implementación de un algoritmo de aprendizaje iterativo en el sistema de computo numérico MATLAB, mediante el entorno de diagramas de bloque SIMULINK, evidenciando que esta estrategia de control puede ser simulada de una manera sencilla, ya que la herramienta ofrece todo lo necesario entre sus extensas librerías.

Se compara y evalúa el desempeño de los controladores diseñados en este trabajo, evidenciando que ambos tiene un resultado muy similar, siendo el controlador clásico un poco mejor respecto al error en estado estacionario, con una diferencia de un 0.5 % en el controlador por aprendizaje iterativo y de 0 con el controlador clásico. Respecto a la diferencia en los tiempo de establecimiento, la cual es de 0.05 segundos entre ambos, siendo un poco mas rápido el controlador clásico, esto se puede deber a un pequeño error en los marcadores de la medición del tiempo, por ende se puede decir que ambos controladores tiene un tiempo de establecimiento igual, siendo la única diferencia significativa el error de estado estacionario. Adicionalmente respecto a la señal del ángulo del servo, se evidencio que para ambos diseños esta es la misma, dando como conclusión que ambos modelos dan casi la misma respuesta pero el controlador diseñado con el algoritmo de aprendizaje iterativo genera un pequeño error de estado estacionario.

Capítulo 10

Trabajos Futuros

Este trabajo deja la puerta abierta para que los controladores diseñados se implementen en la planta física del ball and beam QUANSER. Esta implementación puede dar un nuevo resultado respecto al desempeño de cada controlador, pues aunque la simulación es un fiel modelo de la planta ideal, en la realidad esta planta puede tener pequeños cambios que logren hacer que el resultado sea un poco diferente al obtenido mediante simulación.

En este documento se presentó el modelo de control por aprendizaje iterativo basado en el dominio de la frecuencia con una conexión en paralelo, una de los trabajos a futuro es implementar la conexión en serie, que aunque se entiende que el funcionamiento es similar, puede que este dé otra visión respecto al error de estado estable que se evidencio en los resultados.

Bibliografía

- [1] Felix Berkenkamp y col. «Safe model-based reinforcement learning with stability guarantees». En: *Advances in Neural Information Processing Systems 2017-Decem.Nips* (2017), págs. 909-919. ISSN: 10495258. arXiv: 1705.08551.
- [2] Guillermo Puriel Gil, Wen Yu y Humberto Sossa. «Reinforcement Learning Compensation based PD Control for a Double Inverted Pendulum». En: *IEEE Latin America Transactions* 17.2 (2019), págs. 323-329. ISSN: 15480992. DOI: 10.1109/TLA.2019.8863179.
- [3] Hai Jun Rong y Guang She Zhao. «Direct adaptive neural control of nonlinear systems with extreme learning machine». En: *Neural Computing and Applications* 22.3-4 (2013), págs. 577-586. ISSN: 09410643. DOI: 10.1007/s00521-011-0805-1.
- [4] Zeeshan Shareef Muhammad Asif Rana, Zubair Usman. «Automatic control of ball and beam system using Particle Swarm Optimization». En: (2011).
- [5] B. Tovorni N. Muskinja. «Swinging up and stabilization of a real inverted pendulum». En: (2006).
- [6] Tariq Samad y col. «Industry engagement with control research: Perspective and messages». En: *Annual Reviews in Control* 49 (2020), págs. 1-14. ISSN: 13675788. DOI: 10.1016/j.arcontrol.2020.03.002.
- [7] MATLAB SIMULINK. *Rotary Servo Base Unit Quick Start Guide*. 2012. URL: <https://usermanual.wiki/Document/Rotary20Servo20Base20UnitQuick20Start20Guide.335294577/view>.
- [8] Sen Zhang. «Reinforcement learning Recap : reinforcement learning». En: January (2019), págs. 1-91.
- [9] M. Carreras, J. Yuh y J. Battle. *High-level control of autonomous robots using a behavior-based scheme and reinforcement learning*. Vol. 35. 1. IFAC, 2002, págs. 469-474. ISBN: 9783902661746. DOI: 10.3182/20020721-6-es-1901.01303. URL: <http://dx.doi.org/10.3182/20020721-6-ES-1901.01303>.

-
- [10] M. Carreras, J. Batlle y P. Ridao. «Hybrid coordination of reinforcement learning-based behaviors for AUV control». En: *IEEE International Conference on Intelligent Robots and Systems* 3 (2001), págs. 1410-1415. DOI: 10.1109/iros.2001.977178.
 - [11] N. Kobori y col. «Learning to control a joint driven double inverted pendulum using nested actor/critic algorithm». En: *ICONIP 2002 - Proceedings of the 9th International Conference on Neural Information Processing: Computational Intelligence for the E-Age* 5 (2002), págs. 2610-2614. DOI: 10.1109/ICONIP.2002.1201968.
 - [12] Michail G Lagoudakis y Ronald E Parr. «Least-Squares Methods in Reinforcement Learning for Control». En: May (2002).
 - [13] María López Boada, Beatriz López Boada y Vicente Díaz López. «Algoritmo de aprendizaje por refuerzo continuo para el control de un sistema de suspensión semi-activa». En: *Revista iberoamericana de ingeniería mecánica* 9.2 (2005), págs. 77-92. ISSN: 1137-2729.
 - [14] Xiao Ye y col. «Design and Implementation of Rotary Inverted Pendulum Motion Control Hardware-In-The-Loop Simulation Platform». En: (2010), págs. 155-164.
 - [15] Valeri Mladenov. «Application of neural networks for control of inverted pendulum». En: *WSEAS Transactions on Circuits and Systems* 10.2 (2011), págs. 49-58. ISSN: 11092734.
 - [16] Rafael Figueroa y col. «Reinforcement learning for balancing a flying inverted pendulum». En: *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)* 2015-March.March (2015), págs. 1787-1793. DOI: 10.1109/WCICA.2014.7052991.
 - [17] M. Gómez y col. «Optimal motion planning by reinforcement learning in autonomous mobile vehicles». En: *Robotica* 30.2 (2012), págs. 159-170. ISSN: 02635747. DOI: 10.1017/S0263574711000452.
 - [18] Zhijun Li y Chenguang Yang. «Neural-adaptive output feedback control of a class of transportation vehicles based on wheeled inverted pendulum models». En: *IEEE Transactions on Control Systems Technology* 20.6 (2012), págs. 1583-1591. ISSN: 10636536. DOI: 10.1109/TCST.2011.2168224.
 - [19] L. Adrián León, Ana C. Tenorio y Eduardo F. Morales. «4 - Human Interaction for Effective Reinforcement Learning». En: *European Conf. Mach. Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2013)* (2013). URL: <http://www.ke.tu-darmstadt.de/events/PBRL-13/papers/04-Morales.pdf>.
 - [20] Linglin Wang, Yongxin Liu y Xiaoke Zhai. «Design of reinforce learning control algorithm and verified in inverted pendulum». En: *Chinese Control Conference, CCC* 2015-Septe.Grant 61362002 (2015), págs. 3164-3168. ISSN: 21612927. DOI: 10.1109/ChiCC.2015.7260128.

-
- [21] Guillermo Puriel Gil, Wen Yu y Humberto Sossa. «Reinforcement Learning Compensation based PD Control for a Double Inverted Pendulum». En: *IEEE Latin America Transactions* 17.2 (2019), págs. 323-329. ISSN: 15480992. DOI: 10.1109/TLA.2019.8863179.
 - [22] V. I. Norkin. «Generalized Gradients in Dynamic Optimization, Optimal Control, and Machine Learning Problems*». En: *Cybernetics and Systems Analysis* 56.2 (2020), págs. 243-258. ISSN: 15738337. DOI: 10.1007/s10559-020-00240-x.
 - [23] Angel Miguel Rodriguez Ruiz y col. «Robot Path planning using reinforcement learning and nonlinear approximation function (Planeación de trayectoria utilizando aprendizaje por reforzamiento y función de aproximación)». En: (2017), págs. 56-61.
 - [24] P. V. Mani Maalini ; G. Prabhakar ; S. Selvaperumal. «Modelling and control of ball and beam system using PID controller». En: (2016).
 - [25] Bela Lantos Lorinc Marton. «Stable Adaptive Ball and Beam Control». En: (2006).
 - [26] Z. Shareef, M. Amjad , M.I. Kashif. «Fuzzy logic control of ball and beam system». En: (2010).
 - [27] Yeong-HwaChangaChia-WenChangaChin-WangTaobHung-WeiLincJin-ShiuhTaurd. «Fuzzy sliding-mode control for ball and beam system with fuzzy ant colony optimization». En: (2011).
 - [28] OscarCastillo, EveliaLizárraga, JoseSoria, PatriciaMelin. «New approach using ant colony optimization with ant set partition for fuzzy control design applied to the ball and beam system». En: (2014).
 - [29] Friendly Gui. «Ball and beam». En: (), págs. 3-4.
 - [30] Katsuhiko Ogata. *Ingeniería de control moderna*. Pearson Educación, 2003.
 - [31] Kira Barton, Sandipan Mishra y Enric Xargay. «Robust iterative learning control: L 1 adaptive feedback control in an ilc framework». En: *Proceedings of the 2011 American Control Conference*. IEEE. 2011, págs. 3663-3668.
 - [32] Jian-Xin Xu, Sanjib K Panda y Tong Heng Lee. *Real-time iterative learning control: design and applications*. Springer Science & Business Media, 2008.
 - [33] Fredrik Bagge Carlson y col. «ControlSystems. jl: A Control Toolbox in Julia». En: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, págs. 4847-4853.