CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS

DEL INSTITUTO POLITECNICO NACIONAL

Unidad Zacatenco

Departamento de Control Automático

Control PD / PID de Robots Manipuladores y Sistemas Electromecánicos usando como Compensación de Términos Dinámicos el Aprendizaje por Reforzamiento

Tesis que presenta

M. en C. Guillermo Puriel Gil

Para Obtener el Grado de

Doctor en Ciencias

En la Especialidad de

CONTROL AUTOMÁTICO

Directores de la Tesis

Dr. Wen Yu Liu

Dr. Juan Humberto Sossa Azuela

Ciudad de México. **Diciembre, 2020**

Dedicatoria

A mis padres por haberme forjado como la persona que soy hoy en día; mis logros se los debo a ellos, en donde incluyo este.

Agradecimientos

A dios por cuidarme en cada paso que doy.

A mis padres Alejandro y Genoveva por su amor, su apoyo y sus valiosos consejos.

A mis hermanos Alex y Gaby por su confianza en mí.

A mi hermosa novia Nathllely por su amor.

A mis tías Malena, Marina y Alicia que desde el cielo me acompañan.

Y a toda mi familia por su cariño y amistad.

A Sina por haber sido un excelente amigo y compañero de estudios.

Le agradezco la confianza, apoyo y dedicación de tiempo a mis asesores: el Dr. Wen Yu Liu y el Dr. Juan Humberto Sossa Azuela. Por haber compartido conmigo sus conocimientos y sobre todo su amistad.

Resumen

En este trabajo de tesis, se presenta el aprendizaje por reforzamiento para compensar términos dinámicos de un robot manipulador. Para este fin, se proponen controladores del tipo Proporcional Derivativo (PD), Proporcional Integral Derivativo (PID) y Proporcional Derivativo más la compensación de gravedad que al trabajar conjuntamente con el aprendizaje por reforzamiento dan solución a tareas de regulación y seguimiento de trayectoria. El aprendizaje por reforzamiento tiene un papel importante dentro del área del aprendizaje automático, ampliamente utilizado en la robótica e inspirado en la psicología conductista. Por lo tanto, pertenece a una de las ramas de la inteligencia artificial, donde el controlador determina acciones (entradas de control) en un proceso (planta), para así maximizar su desempeño y su recompensa (función basada en premiar o castigar una acción). Si bien algunos estudios del aprendizaje por reforzamiento han explorado el control de robots o la sintonización de ganancias en controladores, es aún escasa la investigación que aborda la compensación de términos dinámicos en un robot manipulador empleando aprendizaje por reforzamiento y poder mostrar la estabilidad en lazo cerrado.

Se presenta una prueba de estabilidad asintótica semiglobal de un controlador PID usando como compensación el aprendizaje por reforzamiento, cumpliendo con restricciones para que la función de Lyapunov sea definida positiva semiglobalmente, y su derivada sea definida negativa. Esta prueba arroja una sintonización de las ganancias del controlador en forma explícita, y se da el máximo valor que puede tomar la ganancia integral.

En el último capítulo de este documento se presenta un análisis de estabilidad mediante el segundo método de Lyapunov con las condiciones que debe cumplir la ganancia del aprendizaje por reforzamiento en tareas de seguimiento de trayectoria en robots manipuladores, y a diferencia de otros trabajos, se concluye estabilidad asintótica sin invocar el principio de invariancia de Barbashin-Krassovkii-La Salle.

Los algoritmos de control propuestos son evaluados en un doble péndulo invertido sobre un móvil y en un robot manipulador de 2 grados de libertad, diseñado en Solidworks, exportado a Matlab y simulado en un ambiente para sistemas mecánicos 3D.

Los resultados muestran que el aprendizaje por reforzamiento usado para compensar términos dinámicos en un robot manipulador junto con controladores clásicos del tipo PD y PID responde de forma favorable en tareas de regulación y seguimiento de trayectoria.

Abstract

This dissertation presents the reinforcement learning to compensate dynamic terms of a robot manipulator. For this purpose, it is propose controllers of the type Proportional Derivative (PD), Proportional Integral Derivative (PID) and Proportional Derivative plus the compensation of gravity that when working in conjunction with the reinforcement learning gives solution to tasks of regulation and trajectory tracking. Reinforcement learning has an important role within the area of machine learning, widely used in robotics and inspired by behavioral psychology. Therefore, it belongs to one of the branches of artificial intelligence, where the controller determines actions (inputs of control) in a process (controlled system or plant), in order to maximize its performance and its reward (function based on rewarding or punishing an action). While some studies of reinforcement learning have explored the control of robots or the tuning of gains in controllers, there is still little research that addresses the compensation of dynamic terms in a robot manipulator using reinforcement learning and to show the stability in closed loop.

This thesis presents a test of semi-global asymptotic stability of a PID controller using reinforcement learning as compensation, complying with restrictions so that the Lyapunov function is semiglobally positive definite, and its derivative is negative definite. This test yields an explicit tuning of the controller gains, and the maximum value that the integral gain can take is given.

In the last chapter of this document, it presents a stability analysis by means of Lyapunov's second method with the conditions that must fulfill the gain of reinforcement learning for trajectory tracking tasks in manipulative robots, and unlike other works, it is concluded asymptotic stability without invoking Barbashin-Krassovkii-La Salle's invariance principle.

The proposed control algorithms are evaluated in a double inverted pendulum on a cart and in a robot manipulator with 2 degrees of freedom, designed in Solidworks, exported to Matlab and simulated in an environment for 3D mechanical systems.

The results show that reinforcement learning used to compensate dynamic terms in a robot manipulator along with classic controllers of the type PD and PID responds favorably in tasks of regulation and trajectory tracking.

Índice general

Índice de figuras

Capítulo 1

Introducción

El aprendizaje por reforzamiento (RL, por sus siglas en inglés), se ha convertido en una de las más atractivas áreas en el aprendizaje de automático. Su principal objetivo es aprender cómo mapear estados en acciones mientras se maximiza una señal de recompensa [1]. Situado entre el aprendizaje supervisado y el aprendizaje no supervisado, el paradigma del aprendizaje por reforzamiento se ocupa del aprendizaje en problemas de toma de decisiones secuenciales en los que hay una realimentación limitada. Una notable variedad de problemas en robótica pueden expresarse como problemas de aprendizaje por reforzamiento [2]. El aprendizaje por reforzamiento permite a un robot descubrir, de forma autónoma, un comportamiento óptimo a través de interacciones de prueba y error con su entorno. En lugar de detallar explícitamente la solución a un problema, en el aprendizaje por reforzamiento, el diseñador de una tarea de control proporciona realimentación en términos de una función objetivo escalar que mide el rendimiento a cada paso del robot [4].

En los últimos cinco a diez años ha atraído un interés cada vez mayor en las comunidades de aprendizaje automático e inteligencia artificial. Su objetivo consiste en encontrar una forma de programar a los agentes es decir los controlador, mediante recompensas y castigos sin necesidad de especificar cómo se logrará la tarea [6].

El aprendizaje por reforzamiento es "aprender qué hacer", cómo asignar situaciones a acciones para maximizar una señal de recompensa numérica. Al agente no se le dice qué

acciones tomar, como en la mayoría de las formas de aprendizaje automático, sino que debe descubrir qué acciones producen la mayor recompensa al intentarlas [7]. En los casos más interesantes y desafiantes las acciones pueden afectar no sólo la recompensa inmediata sino también la siguiente situación y, a través de ella, todas las recompensas posteriores. Estas dos características, búsqueda mediante prueba y error y recompensa retrasada, son las dos características distintivas más importantes del aprendizaje por reforzamiento. Uno de los desafíos que surgen en el aprendizaje por reforzamiento y no en otros tipos de aprendizaje, es el compromiso entre exploración y explotación [8]. Para obtener una gran recompensa, un agente de aprendizaje por reforzamiento debe preferir las acciones que ha intentado en el pasado y ha encontrado que son efectivas para producir recompensa. Sin embargo, para descubrir tales acciones, un agente tiene que explorar y probar todas las opciones posibles. El agente tiene que explotar lo que ya sabe para obtener una recompensa, pero también tiene que explorar para hacer mejores selecciones de acción en el futuro. El dilema es que ni la exploración ni la explotación pueden realizarse exclusivamente sin fallar en la tarea.

1.1. Motivación y Antecedentes

Para motivar la relevancia de la tesis, se presentan de forma concisa los temas más importantes del aprendizaje por reforzamiento vistos desde la robótica, la incertidumbre del modelo y el diseño de las funciones de recompensa. Además, se dan a conocer brevemente las limitantes de los controladores PD y PID en la compensación de términos dinámicos en un robot manipulador.

1.2. Aprendizaje por reforzamiento en la robótica.

A menudo, dentro del área de la robótica, es poco realista suponer que el estado es completamente observable y libre de ruidos. El agente no podrá saber con precisión en qué estado se encuentra, e incluso, estados muy diferentes pueden parecer muy similares. Por lo tanto, el aprendizaje por reforzamiento en la robótica a menudo se modela como parcialmente

observado. Por tanto, el sistema de aprendizaje debe utilizar filtros para estimar el estado.

La experiencia de controlar un sistema físico real usando el aprendizaje por reforzamiento es tediosa de obtener, incluso cara y a menudo, difícil de reproducir. Cada prueba, es costosa y, como resultado, obligan a centrar a la robótica en las dificultades que no surgen con tanta frecuencia en los ejemplos clásicos de aprendizaje por reforzamiento. Para aprender dentro de un período de tiempo razonable, es necesario introducir aproximaciones adecuadas de estado y/o dinámica del sistema. Sin embargo, si bien la experiencia del mundo real es costosa, generalmente no se puede reemplazar por el aprendizaje sólo en simulaciones. En los modelos analíticos o aprendidos del sistema, incluso los pequeños errores de modelado pueden acumularse en un comportamiento sustancialmente diferente, al menos para tareas dinámicas. Por lo tanto, los algoritmos de control deben ser robustos con respecto a los modelos que no consideran todos los detalles del sistema real.

Un problema que presenta el aprendizaje por reforzamiento de robots es la generación de funciones de recompensa adecuadas. Se necesitan recompensas que guíen el sistema de aprendizaje rápidamente hacia el éxito, para hacer frente al costo de la experiencia del mundo real. Este problema se denomina Formación de Recompensas [11] y representa una contribución manual sustancial; especificar buenas funciones de recompensa en robótica requiere una buena cantidad de conocimiento del dominio y, a menudo, puede ser difícil en la práctica.

1.3. La incertidumbre del modelo dinámico de robots manipuladores.

La incorporación de conocimientos del modelo dinámico del robot es una de las principales herramientas para hacer que el aprendizaje por reforzamiento robótico sea manejable. Sin embargo, a menudo es difícil predecir cuánto conocimiento previo se requiere para permitir que un algoritmo de aprendizaje por reforzamiento, tenga éxito en un número razonable de episodios. Una forma de compensar el costo de la interacción en el mundo real, es utilizar modelos precisos en simuladores. En un entorno ideal, permitiría aprender el comportamiento

en la simulación y luego transferirlo al robot real. Desafortunadamente, crear un modelo suficientemente preciso del robot y su entorno es un desafío y, a menudo no se obtiene. A medida que se acumulan pequeños errores de modelo, el robot simulado diverge rápidamente del robot real. Cuando se entrena un algoritmo de aprendizaje por reforzamiento utilizando un modelo impreciso en un simulador, el comportamiento aprendido no se transferirá al robot de forma adecuada, como lo experimentó [22], cuando aprendió la oscilación del péndulo simple.

Si la tarea es inherentemente estable, es más seguro asumir que los enfoques que se aplicaron en la simulación funcionan de manera similar en el mundo real [23]. Sin embargo, las tareas a menudo se pueden aprender mejor en el mundo real que en la simulación debido a dinámicas complejas incluidos los contactos y la fricción que han demostrado ser difíciles de modelar con precisión.

A lo largo del presente trabajo de investigación se diseña un robot manipulador de 2 grados de libertad en Solidworks, y se transfieren a Matlab sus propiedades físicas, como la masa de los eslabones, centros de masa, longitudes, inercias, fricciones, etc., para ser simulado en un ambiente para sistemas mecánicos 3D, y así crear un modelo dinámico del robot lo suficientemente preciso, y suponer que los enfoques que se aplicaron en la simulación 3D funcionan de manera similar en el mundo real, con la finalidad de incorporar conocimientos previos y transferir conocimientos al algoritmo de control a partir de simulaciones.

Sin embargo, para la mayoría de los sistemas de robots, siempre habrá errores de modelo resultantes. Por lo tanto, el aprendizaje por reforzamiento realizado en simulación, con frecuencia no se puede transferir directamente al algoritmo de control del robot. Este problema puede ser inevitable debido tanto a la incertidumbre sobre la verdadera dinámica del sistema y la incapacidad de cualquier modelo para ser perfectamente preciso respecto de sus dinámicas, lo que ha llevado a una teoría de control robusta [24].

En este trabajo se presentan algoritmos híbridos como el control PID con compensación del aprendizaje por reforzamiento para compensar la dinámica del sistema y reducir el error en estado estacionario en robots manipuladores.

1.4. Especificación de recompensas

En el aprendizaje por reforzamiento, el comportamiento deseado de un robot está implícitamente especificado por la función de recompensa. El objetivo de los algoritmos de aprendizaje por reforzamiento es maximizar la recompensa acumulada a largo plazo. Aunque a menudo es mucho más simple que especificar el comportamiento en sí, en la práctica puede ser sorprendentemente difícil definir una buena función de recompensa. El agente debe conocer la variación en la señal de recompensa para poder mejorar una acción, si siempre se recibe la misma recompensa, no hay forma de determinar qué acción es mejor o más cercana a la óptima. En muchos casos del aprendizaje por reforzamiento, parece natural otorgar recompensas sólo por el logro de una tarea, por ejemplo, cuando un péndulo robot alcanza su equilibrio inestable. Este razonamiento da como resultado una especificación de recompensa binaria (recompensa solamente de dos cifras: cero y uno) aparentemente simple. Sin embargo, raramente un péndulo robot recibe tal recompensa, así que es poco probable que alguna vez tenga éxito en el mundo real. En lugar de depender de recompensas binarias simples, con frecuencia necesitamos incluir recompensas intermedias en la función de recompensa escalar para guiar el proceso de aprendizaje hacia una solución razonable [11]. Por ejemplo, reducir la velocidad del péndulo robot cuando se aproxime a su equilibrio inestable, puede resultar en una recompensa alta, pero es probable que se necesiten considerar más criterios para lograr la tarea. La forma de recompensa le da al sistema una noción de cercanía al comportamiento deseado en lugar de depender de una recompensa que sólo codifica el éxito o el fracaso [25].

Un problema de aprendizaje es potencialmente difícil si la recompensa es escasa, hay retrasos significativos entre una acción y la recompensa, o si la recompensa no es unifrome es decir, cambios muy pequeños en la acción conducen a un resultado drásticamente diferente. En el aprendizaje por refuerzo clásico, a menudo se consideran recompensas discretas por ejemplo, una pequeña recompensa negativa por paso de tiempo y una gran recompensa positiva por alcanzar la meta. Por el contrario, los enfoques de aprendizaje por refuerzo robótico a menudo necesitan una forma de recompensa más motivada físicamente basada en valores continuos.

Por lo tanto, en este trabajo se proponen recompensas en función de las posiciones y velocidades articulares del robot, por ejemplo, si la tarea del robot es alcanzar una posición deseada, entonces, la recompensa estará en base a una función de error que se define como la diferencia entre una posición deseada y la posición articular del robot. Además, si la tarea es el seguimiento de trayectoria, la recompensa se propone en función tanto de los errores de posición como de velocidad. Estos enfoques de recompensa producen una respuesta uniforme y dan solución al problema de recompensa binaria o escasa.

1.5. Compensación de términos dinámicos en robots manipuladores.

El control de robots en el espacio articular el esquema más simple es el algoritmo de control Proporcional Derivativo (PD), el cual sólo con la elección de alguna ganancia positiva, el controlador puede garantizar estabilidad en el problema de regulación [26]. Sin embargo, no se puede garantizar la estabilidad asintótica cuando la dinámica del manipulador incluye: vector de par gravitacional, vector de fricción y de forma general otras incertidumbres, hasta que se aplica una compensación con modelo de referencia, requiriendo el conocimiento de la dinámica del robot. Por ejemplo, en los trabajos de compensación adaptable de gravedad [27][28], compensación por gravedad deseada [29], y control PD más medición de posición [30], es necesaria la información estructural del vector de gravedad en el controlador PD. Por otro lado, algunos controladores PD no lineales también pueden alcanzar la estabilidad asintótica tales como: control PD con ganancias variantes en el tiempo [31], ganancias no lineales [32], y compensación por modos deslizantes [33]. Una compensación para un control PD que no necesita un modelo matemático es conocido como compensador libre de modelo. Dentro de este grupo se encuentran: compensadores difusos [34], controles difusos [35], compensadores neuronales [36] y compensadores neuro-difusos [37][38]. La idea básica de estos controladores es el uso de un filtro de error de seguimiento basado en el análisis de Lyapunov [39]. Para un algoritmo de ajuste de pesos apropiados, son muy similares los métodos de

control adaptable [40], la derivada de la función de Lyapunov es negativa, siempre y cuando el error de seguimiento filtrado esté fuera de una bola de radio b/k_v, aquí b es el límite superior de todas las incertidumbres, k_v es la ganancia derivativa en el control PD. Estos controladores neuronales PD están acotados uniformemente y los errores de seguimiento se hacen más pequeños conforme aumenta la ganancia k_v. El costo de elegir k_v suficientemente grande es que el estado transitorio se vuelve evoluciona lentamente. sólo cuando $k_v \to \infty$ el error de seguimiento converge a cero [41].

Existen varios métodos para minimizar el error en estado estable respecto a incertidumbres en el control de robots, uno de ellos es, utilizando controladores con la ley de control Proporcional Integral Derivativo (PID), en el cuál se incrementa la ganancia integral al hacer al error casi cero. Otro método es utilizando controladores PD en combinación con un compensador neuronal (NN, por sus siglas en inglés) (PD+NN)[36][41] con la condición para una ganancia derivativa suficientemente grande este método puede extenderse a controladores PID más una compensación neuronal (PID+NN) el cual garantiza estabilidad asintótica semiglobal[43]. Además, hay dos enfoques diferentes para combinar el controlador PID con el control inteligente. El primer enfoque es aquel en el que las redes neuronales son formadas dentro de una estructura PID [42][43] y [44]. Para una adecuada actualización de las leyes de control, los parámetros de los controladores PID se eligen de manera que el sistema en lazo cerrado sea estable. Pero la dificultad para implementarlos radica en que las ganancias de la acción PID son variantes en el tiempo.

El segundo enfoque, utiliza técnicas de control inteligente y ajusta parámetros de los controladores PID, tales como sintonización difusa [45], sintonización neuronal [46][47] y sintonización experta [48]. De modo que, para reducir el error generado por el seguimiento de trayectoria, se puede agregar una ganancia integral al controlador PD+NN, es decir, convertir el controlador neuronal PD a un control neuronal PID.

En el caso de los trabajos que abordan el problema de control de robots manipuladores para la cancelación de términos dinámicos usando el aprendizaje por reforzamiento, existe una gran cantidad que utiliza redes neuronales [36][64][8][70][81][82][91], control difuso y neuro difuso [34][37][56][79][92][89] y LQR [49][50][51][52]. Sin embargo, son muy pocos los

trabajos reportados que usan controladores del tipo PD, PID [53][54][55] y que además presenten un análisis formal de estabilidad desarrollando una sintonización explícita de las ganancias del controlador.

En el control PD con compensadores neuronales, difusos o neuro difusos la velocidad de respuesta podría ser pequeña debido al incremento de la ganancia derivativa, además, es complicado definir la estructura de la red neuronal dentro del control PD, etc. En esta tesis se presenta un algoritmo basado en el aprendizaje por reforzamiento para superar estos problemas y garantizar un buen desempeño. La ventaja que presenta es que la ganancia del aprendizaje por reforzamiento se encarga de compensar los términos dinámicos y reducir el error en estado estacionario sin la necesidad de incrementar la ganancia derivativa para que el error pueda converger a cero. La desventaja es que el tiempo de aprendizaje puede tomar desde minutos, horas o hasta días, y así será para cada tarea fijada. Sin embargo, una vez aprendida la tarea ya no se necesitará aprenderla nuevamente siempre y cuando el sistema dinámico no sufra cambios. Si el aprendizaje se realiza con un robot sin que manipule una carga, en el momento en que hace la manipulación cambia su dinámica y el algoritmo debe ejecutarse de nuevo.

Esta tesis, contribuye al control de robots manipuladores, presentando algoritmos de control PD y PID, usando como compensación las técnicas del aprendizaje por reforzamiento para el control de posición de un robot manipulador de 2 grados de libertad. Los algoritmos de control propuestos consideran la compensación de términos dinámicos con base a un método iterativo de prueba y error. Los problemas abordados en este trabajo para la tarea de regulación son: (1) La compensación de términos gravitacionales utilizando únicamente el aprendizaje por reforzamiento. (2) La compensación del término gravitacional y de fricción utilizando el control PD/PID con compensación de aprendizaje por reforzamiento. (3) La estimación de la ganancia del aprendizaje por reforzamiento para compensar los términos gravitacionales y de fricción basados en un análisis de estabilidad de Lyapunov. (4) Utilizar al control PD/PID como una guía o maestro de enseñanza realizando demostraciones como un profesor y agregando conocimientos previos al algoritmo del aprendizaje por reforzamiento.

Se lleva a cabo una prueba de estabilidad asintótica semiglobal de un controlador PID

usando como compensación el aprendizaje por reforzamiento, cumpliendo con restricciones rigurosas de la función de Lyapunov. Este proceso arroja una sintonización de las ganancias en forma explícita, y se da el máximo valor de la ganancia integral para evitar problemas, dado que la ganancia integral toma valores muy grandes al intentar cancelar el error en estado estacionario.

Una contribución más de la tesis es que se presentan las condiciones de la ganancia del aprendizaje por reforzamiento en un controlador PD más la compensación de gravedad (G(q)) y la compensación del aprendizaje por reforzamiento para la tarea del seguimiento de trayectoria. El análisis de estabilidad del sistema en lazo cerrado se lleva a cabo mediante el segundo método de Lyapunov, en donde se toma en cuenta la dinámica completa del robot manipulador, y a diferencia de otros trabajos se concluye estabilidad asintótica sin invocar el principio de invariancia de Barbashin-Krassovskii-La Salle. En este contexto, la contribución de la tesis consiste en probar estabilidad con una función de Lyapunov que es definida positiva, no acotada radialmente y que la derivada sea definida negativa en todos sus estados.

En el aprendizaje por reforzamiento es de gran interés e importancia, el incorporar conocimientos previos y transferir conocimientos a partir de simulaciones [1] [2][3] y [5]. Así, otra contribución más, es el diseño en Solidworks de un robot manipulador de 2 grados de libertad y transferir a Matlab todas sus propiedades físicas para ser simulado en un entorno para sistemas mecánicos 3D, y así tener un modelo dinámico del robot lo suficientemente preciso, y de este modo, se pueda transferir al caso experimental todo el conocimiento aprendido a través de simulaciones. Si bien la simulación nunca igualará a la experimentación [2][4], es una excelente contribución realizar el diseño del robot manipulador y llevar a cabo la simulación del aprendizaje por reforzamiento utilizando el robot diseñado.

En resumen, se propone el algoritmo del aprendizaje por reforzamiento para compensar términos dinámicos de un robot manipulador de 2 grados de libertad diseñado en Solidworks y exportado a Matlab. Para este fin, se proponen controladores PD, PID y PD con compensación de gravedad (G(q)) que al trabajar de forma híbrida con el aprendizaje por reforzamiento dan solución a tareas de regulación y seguimiento de trayectoria. Además, se

compara su desempeño con controladores clásicos basados en el modelo y libres de modelo con la finalidad de observar su respuesta en estado transitorio, en estado estacionario y que tan robusto es. Finalmente, se utiliza un criterio académico ampliamente aceptado en la comunidad científica de robótica, para medir el desempeño de los algoritmos de control determinado por la integral del error absoluto (IAE, por sus siglas en inglés), y por la integral del tiempo por el error absoluto (ITAE, por sus siglas en inglés).

1.6. Objetivos

Diseñar controladores PD, PID, y PD+G(q) usando el aprendizaje por reforzamiento como compensación de términos dinámicos en sistemas electromecánicos y robots manipuladores y dar solución a tareas de regulación y seguimiento de trayectoria. Además, desarrollar de manera explícita las condiciones de sintonización de las ganancias del controlador cumpliendo con restricciones rigurosas de la función Lyapunov.

El objetivo principal es dividido en objetivos particulares para facilitar su cumplimento. Los objetivos particulares a realizar en la presente tesis son:

Objetivos particulares

1. Buscar en la literatura del aprendizaje por reforzamiento algoritmos de control aplicados a robots manipuladores para identificar soluciones al problema de regulación y seguimiento de trayectoria.

2. Diseñar un algoritmo usando el aprendizaje por reforzamiento para controlar un doble péndulo sobre el carro donde se mantenga sobre la vertical invertida mientras el carro conserva un rango de distancia prefijado.

3. Implementar las técnicas de control PD, PID y PD con compensación de gravedad (G(q)) en un robot manipulador de 2 grados de libertad para cumplir con tareas de regulación y seguimiento de trayectoria.

4. Investigar el análisis de estabilidad de controladores PD, PID y PD con compensación de gravedad (G(q)) basados en el segundo método de Lyapunov y en el principio de invariancia de Barbashin-Krassovki-La Salle.

5. Diseñar funciones de recompensas continuas basadas en cumplir tareas de regulación y de seguimiento de trayectoria en un robot manipulador de 2 grados de libertad.

6. Diseñar algoritmos de control PD, PID y PD con compensación de gravedad (G(q)), usando como compensación de términos dinámicos el aprendizaje por reforzamiento y realizar tareas de regulación y seguimiento de trayectoria en un robot manipulador de 2 grados de libertad.

7. Realizar el diseño de un robot manipulador de 2 grados de libertad en Solidworks para ser simulado en un entorno para sistemas mecánicos 3D, con la finalidad de transferir al caso experimental todo el conocimiento aprendido a través de simulaciones.

8. Comparar los algoritmos de control diseñados con controladores clásicos basados en modelo y libres de modelo bajo un índice de desempeño.

1.7. Contribuciones

Se enlistan las contribuciones de la presente tesis:

1. Se presenta una prueba de estabilidad asintótica semiglobal de un controlador PID usando como compensación el aprendizaje por reforzamiento, cumpliendo con restricciones rigurosas de la función de Lyapunov. Este proceso arroja una sintonización de las ganancias en forma explícita, y se da el máximo valor de la ganancia integral y así evitar problemas dado que la ganancia integral toma valores muy grandes al intentar cancelar el error en estado estacionario.

2. Se desarrolla un análisis de estabilidad del controlador PD+G(q) con compensación del aprendizaje por reforzamiento en lazo cerrado con la planta, donde se dan las condiciones de la ganancia del aprendizaje por reforzamiento. El análisis de estabilidad del sistema en lazo cerrado se lleva a cabo mediante el segundo método de Lyapunov, en donde se toma en cuenta la dinámica completa del robot manipulador, y a diferencia de otros trabajos se concluye estabilidad asintótica sin invocar el principio de invariancia de Barbashin Krassovskii-La Salle.

3. Se realiza el diseño en Solidworks de un robot manipulador de 2 grados de libertad y se transfieren a Matlab todas sus propiedades físicas para ser simulado en un entorno para sistemas mecánicos 3D, y así tener un modelo dinámico del robot lo suficientemente preciso, y de este modo, se pueda transferir al caso experimental todo el conocimiento aprendido a través de simulaciones.

1.8. Estructura de la tesis

Esta tesis se encuentra dividida en 7 capítulos.

- **Capítulo 2. Aprendizaje por Reforzamiento.** Este capítulo nos brinda un panorama general de cómo se aborda el aprendizaje por reforzamiento desde el punto de vista de la inteligencia artificial y del control automático. Se presentan las definiciones formales de agente, proceso y se muestra un diagrama de flujo que relaciona al controlador (agente), al proceso (planta) y a la función de recompensa en base a los estados y acciones. Además, se dividen los algoritmos del aprendizaje por reforzamiento en tres sub clases: valor de iteración, política de iteración y política de búsqueda. Finalmente, se clasifica el problema del aprendizaje por reforzamiento como Markoviano/No Markoviano, Determinista/Estocástico, Pequeño/Grande y Episódico/Continuo.

- **Capítulo 3. Control de robots mediante el Aprendizaje por Reforzamiento.** En este capítulo, se presenta el marco teórico, donde se repasan los preliminares

matemáticos y principales conceptos del aprendizaje por reforzamiento, aplicado a robots haciendo especial hincapié en las funciones de valor estado y en la ecuación de Bellman. Igualmente se presenta el proceso de decisión de Markov definido desde las variables de estado del robot y se introduce el concepto de recompensas de horizonte infinito. Finalmente, se muestran las simulaciones para diferentes dominios de evaluación, por ejemplo: (1) La primera simulación es sobre un doble péndulo sobre el carro, donde se desea estabilizar ambos péndulos sobre la vertical invertida mientras se mantiene el carro dentro de un rango deseado de ± 2 metros. (2) La segunda simulación es sobre un Pendubot donde se desea llevar al doble péndulo a su posición vertical invertida partiendo de su equilibrio estable. (3) Como último se presenta un Acrobot en donde el problema se dividió en dos partes: balancearlo (swing-up problem) y estabilizarlo (balancing problem), partiendo de su condición de equilibrio estable.

- **Capítulo 4. Control PD usando como Compensación el Aprendizaje por Reforzamiento.** En el tercer capítulo, se presenta uno de los algoritmos más populares en el aprendizaje por reforzamiento conocido como Q-Learning, el cual se caracteriza por ser un controlador libre de modelo, y se propone usar el control PD con compensación del algoritmo Q-Learning, para formar un control híbrido que tenga una mayor robustez a parámetros no modelados de la dinámica de un robot manipulador, y así, dicho control híbrido, trabaje mejor de manera conjunta que independiente. Se muestran simulaciones: (1) Para un péndulo, donde se busca llevarlo del equilibro estable hacia al equilibrio inestable mientras se compensa una perturbación introducida como un cambio de masa en el péndulo. (2) Un péndulo sobre el carro iniciará en posición vertical invertida y se requiere que el péndulo se mantenga sobre su condición inicial, donde el ángulo del péndulo sólo debe variar un en rango de ± 10 grados y la posición del carro se mantendrá en un rango de ± 2 metros, si el péndulo o el carro rebasan estas restricciones, entonces la tarea se considerará fallida. Finalmente, se presenta la prueba de estabilidad utilizando el segundo método de Lyapunov.

- **Capítulo 5. Control PID usando como Compensación el Aprendizaje por**

Reforzamiento. En el capítulo 5, se extiende el control PD con compensación del aprendizaje por reforzamiento al control PID con compensación del aprendizaje por reforzamiento para cumplir con tareas de regulación en un robot manipulador de 2 grados de libertad. Además, se presenta una prueba de estabilidad asintótica semiglobal cumpliendo con restricciones rigurosas de la función de Lyapunov. Este proceso proporciona una sintonización de las ganancias en forma explícita, y se da el máximo valor de la ganancia integral y así evitar problemas dado que la ganancia integral toma valores muy grandes al intentar cancelar el error en estado estacionario. Se compara el desempeño con controladores basados en el modelo y libres de modelo, donde se utilizó un criterio académico ampliamente usado en la comunidad científica de robótica para medir el desempeño del algoritmos de control, determinado por la integral del error absoluto IAE, y por la integral del tiempo por el error absoluto ITAE. Los resultados muestran que la sintonización de ganancias propuestas arroja los índices de desempeño más bajos, con lo cual estuvieron dentro de los parámetros esperados.

- **Capítulo 6. Control PD+G(q) Compensado con el Aprendizaje por Reforzamiento.** En el capítulo 5, se presenta un control PD+G(q) con compensación del aprendizaje por reforzamiento para el seguimiento de trayectoria. Primero se presenta un análisis de estabilidad del controlador propuesto en lazo cerrado con la planta, donde se dan las condiciones de la ganancia del aprendizaje por reforzamiento. El análisis de estabilidad del sistema en lazo cerrado se lleva a cabo mediante el segundo método de Lyapunov, en donde se toma en cuenta la dinámica completa del robot manipulador, y a diferencia de otros trabajos se concluye estabilidad asintótica sin invocar el principio de invariancia de Barbashin-Krassovskii-La Salle. Como segundo, con el propósito de evaluar mediante simulaciones numéricas la ley de control, se presentan dos casos: (1) Se utiliza un polinomio de 5to grado donde especificamos la posición, la velocidad y la aceleración al inicio y al final de cada segmento de ruta. (2) Se emplea una trayectoria conocida como lemniscata o como el símbolo del infinito, el cual es un tipo de curva descrita en coordenadas paramétricas, donde sus parámetros de diseño fueron seleccionados en función de las dimensiones geométricas del robot. Se hizo uso

de la cinemática inversa para conocer las posiciones articulares deseadas a partir de especificaciones de posición deseada x_d e y_d generadas por la trayectoria de la lemniscata. Se compara el desempeño con controladores basados en el modelo dinámico, donde se utilizó un criterio determinado por la norma L_2 para medir el error de posición y de velocidad en el seguimiento de trayectoria. Los resultados muestran la efectividad del controlador propuesto, y produce los índices de desempeño más bajos, con lo cual estuvieron dentro de los parámetros esperados.

- **Capítulo 7. Conclusiones**. Este capítulo brinda las conclusiones de la presente tesis enfocado en el diseño de controladores PD, PID y PD+G(q) que usan el aprendizaje por reforzamiento para la compensación de términos dinámicos en robots manipuladores, todo esto basado en los resultados obtenidos mediante graficas e índices de desempeño. Además, se dan las perspectivas del trabajo futuro.

1.9. Publicaciones

Revista Internacional

1. G. Puriel-Gil, W. Yu, and H. Sossa, Reinforcement Learning Compensation based PD Control for a Double Inverted Pendulum, IEEE LATIN AMERICA TRANSACTIONS, VOL. 17, NO. 2, FEBRUARY 2019.

2. G.Puriel-Gil, W. Yu, and H. Sossa, Robot PID Control Using Reinforcement Learning, International Journal of Control, Automation and Systems, **(sometido)**.

Congreso Internacional

1. Guillermo Puriel-Gil, Wen Yu, Humberto Sossa Reinforcement Learning Compensation based PD Control for Inverted Pendulum, 2018 $15th$ International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE), Mexico, City. Mexico. September 5-7.

Capítulo 2

Aprendizaje por Reforzamiento

2.1. Introducción

En la programación dinámica (DP) y el aprendizaje por reforzamiento (RL), un controlador (agente, tomador de decisiones) interactúa con el proceso (ambiente), por medio de tres señales: La señal de estado, que describe el estado del proceso, una señal de acción, que le permite al controlador influir en el proceso, y finalmente una señal escalar que le provee al controlador una realimentación de su desempeño inmediato. En cada paso de tiempo discreto, el controlador recibe la medición del estado y aplica una acción, la cual causa la transición hacia un nuevo estado. La recompensa generada evalúa la calidad de la transición. El controlador recibe el nuevo estado y todo el ciclo completo se repite nuevamente. Este flujo de interacción es representado en la figura (2.1).

El comportamiento del controlador está representado por su política, que es una función que mapea estados en acciones. El comportamiento del proceso está descrito por su dinámica, la cual determina como los estados cambian, como resultado de las acciones del controlador. En el caso determinístico, tomar una acción dada en un estado dado, siempre resulta en el mismo estado siguiente, mientras que en el caso estocástico, el estado siguiente es una variable aleatoria. El tipo de recompensa generada es descrita por la función recompensa. La dinámica del proceso y la función recompensa, junto con el conjunto de estados posibles

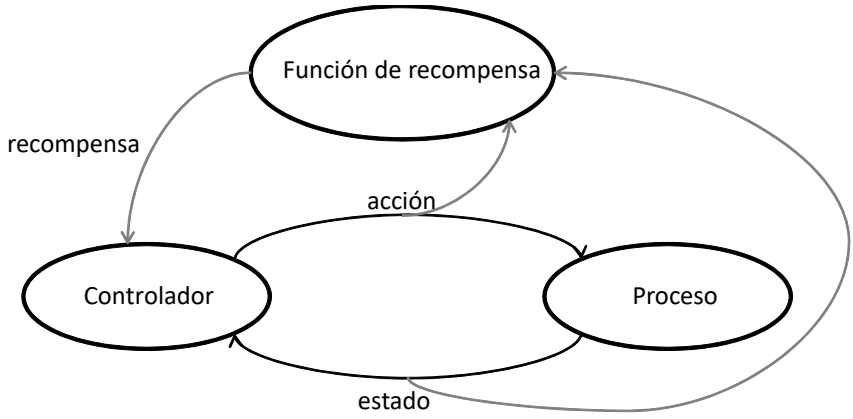


Figura 2.1: Flujo de interacción en el aprendizaje por reforzamiento.

y el conjunto de acciones posibles (espacio de estados y espacio de acciones) constituye el llamado proceso de decisión de Markov (MDP).

En DP y RL, el objetivo es encontrar una política óptima que maximice el retorno consistituído por la recompensa acumulativa durante el curso de la interacción.

El marco DP / RL se puede usar para abordar problemas de una variedad de campos, incluyendo, e. g., control automático, inteligencia artificial, son posiblemente los campos más importantes de origen para DP y RL. En control automático, el DP puede ser usado para resolver problemas de control óptimo lineal y estocástico (Bertsekas, 2007), mientras que RL puede alternativamente ser visto como control óptimo adaptativo (Sutton, 1992). En Inteligencia artificial, RL, ayuda a construir un agente artificial que aprenda como sobrevivir y optimizar su comportamiento en un ambiente desconocido, sin requerir de conocimiento previo (Sutton y Barto, 1998). Finalmente, se tienen nombres y notaciones equivalentes en DP y RL por ejemplo, "Controlador" tiene el mismo significado que "agente" y "proceso" , tiene el mismo significado que "ambiente". En este texto, usaremos la misma terminología y notación que en la Teoría de Control.

Una taxonomía de los algoritmos DP y RL se muestra en la Figura (2.2).

Los algoritmos DP, requieren un modelo del MDP, incluida la dinámica de transición y la

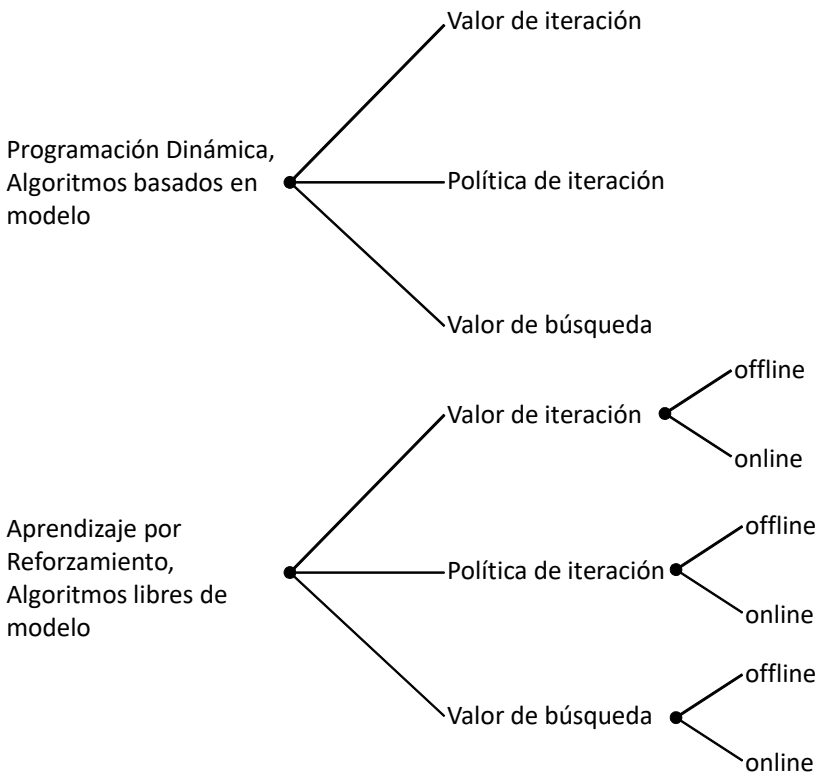


Figura 2.2: Taxonomía de los algortimos DP y RL

función de recompensa, para encontrar una política óptima (Bertsekas, 2007; Powell, 2007). Los algoritmos DP funcionan fuera de línea, produciendo una política que luego se usa para controlar el proceso. Por lo general, no requieren una expresión analítica de la dinámica. En cambio, dado un estado y una acción, el modelo sólo requiere generar un próximo estado y la recompensa correspondiente. La construcción de dicho modelo generativo es a menudo más fácil que obtener una expresión analítica de la dinámica, especialmente cuando la dinámica es estocástica.

Los algoritmos RL no depende de modelos (Bertsekas y Tsitsiklis, 1996; Sutton y Barto, 1998), lo que los hace útiles cuando un modelo es difícil o costoso de construir. Los algoritmos RL usan datos obtenidos del proceso, en forma de un conjunto de muestras, un conjunto de trayectorias de proceso, o una sola trayectoria. Por lo tanto, RL puede verse como libre de modelo, basado en muestras o DP basado en trayectoria, y DP puede verse como RL basado en modelo. Mientras los algoritmos DP pueden usar el modelo para obtener cualquier cantidad de transiciones de muestra desde cualquier par de estados de acción, los algoritmos de RL deben funcionar con datos limitados que se pueden obtener del proceso: un desafío mayor. Nótese, que algunos algoritmos de RL crean un modelo de los datos; llamamos a estos algoritmos "modelo de aprendizaje".

Las clases de algoritmos DP y RL se pueden dividir en tres subclases, de acuerdo con el camino tomado para encontrar una política óptima. Estas tres subclases son: la iteración de valores, la iteración de políticas y la búsqueda de políticas, y se caracterizan como sigue:

•Valor de iteración, algoritmos que buscan la función de valor óptimo, que consiste en los rendimientos máximos de cada estado o de cada par de estados de acción. La función de valor óptimo se utiliza para calcular una política óptima.

•Política de iteración, algoritmos que evalúan las políticas al construir las funciones de valor (en lugar de la función de valor óptimo) y utilizan estas funciones de valor para encontrar nuevas y mejores políticas.

•Política de Búsqueda, algoritmos que utilizan técnicas de optimización para buscar directamente una política óptima.

Dentro de cada una de las tres subclases de algoritmos de RL, se pueden distinguir dos

categorías, a saber, los algoritmos fuera de línea y en línea. Los algoritmos de RL sin conexión, usan datos recopilados de antemano, mientras que los algoritmos de RL en línea aprenden una solución al interactuar con el proceso. En los algoritmos de RL en línea generalmente no se conoce ningún dato de antemano, sino que tienen que depender únicamente de los datos que recopilan durante el aprendizaje, y por lo tanto son útiles cuando es difícil o costoso obtener datos por adelantado. La mayoría de los algoritmos RL en línea funcionan de forma incremental. Por ejemplo, un algoritmo de iteración de valor en línea incremental actualiza su estimación de la función de valor óptimo después de cada muestra recolectada. Incluso antes de que esta estimación sea precisa, se usa para derivar estimaciones de una política óptima, que luego se utilizan para recopilar datos nuevos. Los algoritmos de RL en línea deben equilibrar la necesidad de recopilar datos informativos (mediante la exploración de opciones de acción novedosas o partes nuevas del espacio de estado), con la necesidad de controlar bien el proceso (explotando el conocimiento disponible actualmente). Esta explotación - exploración (a veces va una u otra) hace la RL en línea más desafiante que RL fuera de línea. Tenga en cuenta que, aunque los algoritmos de RL en línea sólo están garantizados (en las condiciones adecuadas) para converger a una política óptima cuando el proceso no cambia con el tiempo, en la práctica a veces también se aplican a procesos que cambian lentamente, en cuyo caso se espera que adapten la solución para que los cambios se tengan en cuenta.

2.2. Conceptos Básicos

2.2.1. Señales de Refuerzo

El concepto mas importante en el aprendizaje por reforzamiento es la idea de señales de refuerzo o recompensas. La prueba y el error sólo pueden arrojar resultados útiles, cuando es posible decidir si la última acción realizada fue un error o no. El controlador necesita algún tipo de realimentación evaluativa, para poder detectar una mejora (o lo contrario) en la situación después de una acción. Por lo tanto, resolver un problema utilizando el aprendizaje por reforzamiento, implica que hay una función de evaluación (recompensa) que

puede asignar estados del entorno a algún tipo de valor numérico.

2.2.2. Agente (Controlador) y Ambiente (Proceso)

Cualquier agente (controlador) en inteligencia artificial se puede especificar definiendo sus percepciones, acciones, objetivos y entorno [Russel y Norving, 1995]. Esto también es válido para el aprendizaje por reforzamiento, como se muestra en la figura (2.3).

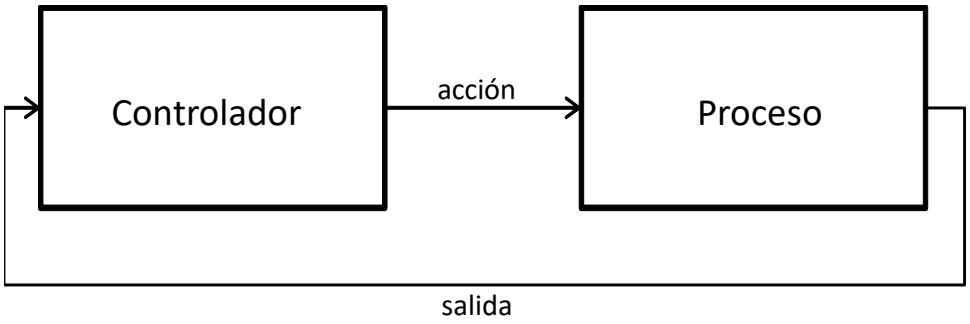


Figura 2.3: (a) Control Automático

Todos los agentes del aprendizaje por reforzamiento, tienen dos características en común:

1.- Sus percepciones incluyen la señal de recompensa que evalúa la situación actual.

2.- Su objetivo es maximizar las recompensas acumuladas, es decir, alcanzar la suma de las recompensas óptimas a lo largo del tiempo.

En la inteligencia artificial, así como en muchos otros casos, la tarea del aprendizaje es una tarea asociativa. El agente inteligente busca aprender una asociación entre situaciones y acciones que se deben tomar, dado que el ambiente se encuentra en esta situación. Generalmente en aprendizaje por reforzamiento asociativo, las percepciones del agente también incluyen información sobre el entorno y el estado del sistema, como se ve en la figura (2.4).

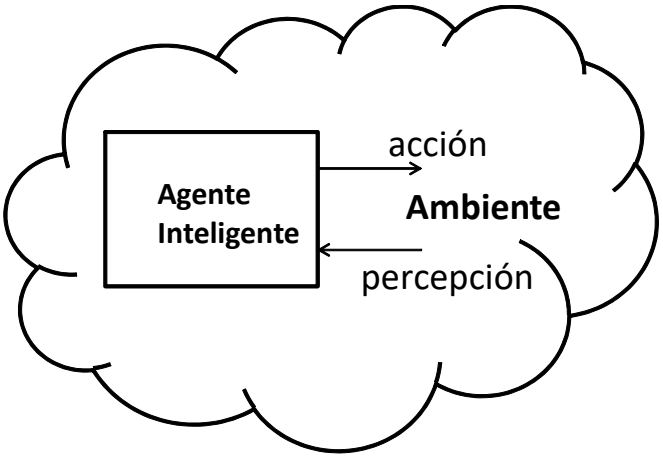


Figura 2.4: (b) Inteligencia Artificial

2.3. Términos y Definiciones.

Para abordar el aprendizaje por reforzamiento sobre una base teórica, las ideas que lo respaldan deben definirse formalmente. Los conceptos básicos en la teoría del aprendizaje por reforzamiento son:

1. El controlador (agente) puede tomar acciones u de un conjunto de todas las posibles acciones U. La acción tomada en el paso de tiempo discreto k se le llama u_k.

2. El controlador puede observar el ambiente, lo cual implica que puede observar el estado actual x desde un conjunto de posibles estados X. El estado en el tiempo discreto es x_k.

3. El controlador recibe recompensas $r \in \mathbb{R}$ que mide lo bueno que es el estado actual x. Nuevamente, la recompensa recibida en el tiempo discreto k se le conoce como $r_{k.}$

4. Para escoger una acción, el controlador sigue una política de control, que es un mapeo de los estados a las acciones $\pi : X \rightarrow U$. Una política determina que acción u el agente debe tomar estando en el estado x.

Basado en los conceptos definidos anteriormente, hay dos términos más importantes: funciones valor y las funciones valor estado.

1. Funciones valor $V^{\pi}(x)$ representa el valor del estado $x \in X$, suponiendo que el controlador sigue una polítca dada π.

2. Funciones de valor-acción $Q^{\pi}(x, u)$ representa el valor de tomar una acción $u \in U$ en el estado $x \in X$, suponiendo que el controlador sigue una política π.

En los problemas de decisión secuenciales (Sequential Decision Problems, SDPs) aparecen cuatro componentes fundamentales [Barreto et al., 2010]: el agente encargado de tomar las decisiones, el entorno con el cual interactúa, el comportamiento que exhibe, y los refuerzos que recibe. Aunque los problemas de decisión secuencial pueden ser tratados en diferentes niveles de abstracción, en el modelo considerado en este trabajo, un agente simplemente es el sistema responsable de interactuar con el mundo y tomar las decisiones. De forma general, el entorno será todo aquello externo al agente. El entorno cambia de estado en estado en respuesta a las acciones ejecutadas por el agente de acuerdo a una dinámica, que en la mayoría de los casos es de naturaleza estocástica (aunque podrá ser determinista). La interacción entre el agente y el entorno se realiza en intervalos discretos de tiempo. Las acciones de los agentes, además, sirven a un propósito en los problemas que se consideran aquí, como el propósito de maximizar un refuerzo. Todas estas decisiones responden a un comportamiento llevado a cabo por el agente, una política. Atendiendo a estas definiciones, un problema de decisión secuencial puede ser clasificado como:

Markoviano/No Markoviano: En un problema de decisión de Markov las transiciones y las recompensas dependen únicamente del estado actual y la acción seleccionada por el agente [Puterman, 1994]. Otra forma de decir esto mismo es que un estado de Markov contiene toda la información relativa a la dinámica de una tarea: una vez conocido el estado actual, la historia de las transiciones que llevaron al agente hasta esta posición es irrelevante a efectos de la toma de decisiones del problema. En el ajedrez por ejemplo, una configuración particular de la partida en cualquier momento proporciona toda la información necesaria para realizar el próximo movimiento. Por otro lado, al decidir si procede o no conceder el

empate al adversario, el jugador podrá beneficiarse sobre la historia de los juegos anteriores. En este caso, nos encontramos con un problema de decisión no Markoviano. Los algoritmos de aprendizaje por refuerzo que se consideran en este trabajo están desarrollados basándose en esta suposición Markoviana. No es difícil ver que los algoritmos de aprendizaje por refuerzo son particularmente sensibles a la propiedad de Markov: si la dinámica de la tarea depende de la historia de transiciones ejecutada por el agente, tiene poco o ningún sentido asociar estado-acción con una secuencia específica de recompensas. Esto no quiere decir que sea imposible emplear algoritmos de aprendizaje por refuerzo a tareas no markovianas, aunque actualmente representan un verdadero obstáculo para los algoritmos de aprendizaje por refuerzo en general.

Determinista/Estocástico: En un problema de decisión determinista la ejecución de una determinada acción, u_k, en un determinado estado, x_k, siempre lleva al agente a un mismo estado, x_{k+1}. En contraste, en un entorno estocástico, cada transición está asociada con una distribución de probabilidad sobre el espacio de estados X, es decir, el agente podrá acabar en estados diferentes en dos ejecuciones distintas de la misma acción, u, en x_k. El ajedrez, por ejemplo, es un juego determinista mientras que el blackjack es estocástico [Sutton and Barto, 1998].

Pequeño/Grande: Aquí los términos pequeño y grande se refieren al tamaño de los espacios de estados y acciones. Si para un problema de decisión secuencial se pueden almacenar cada par estado-acción del problema, teniendo en cuenta la capacidad de almacenamiento de los computadores actuales, el problema puede considerarse pequeño. En caso contrario, el problema se considera grande. Obviamente, un problema de decisión secuencial con unos espacios de estados y acciones continuos es siempre grande [Barreto et al., 2010].

Episódico/Continuo: En una tarea episódica, la interacción entre el agente y el entorno está dividida en episodios. Cada episodio comienza en un estado inicial y termina en un estado especial llamado estado terminal (terminal state) [Sutton and Barto, 1998]. Los distintos episodios se van ejecutando secuencialmente. En problemas continuos, no existe la división en episodios y la tarea continua sin un criterio claro de interrupción.

Capítulo 3

Control de Robots mediante Aprendizaje por Reforzamiento

En este capítulo, se presentan las técnicas en el aprendizaje por reforzamiento y el modelo formal detrás del problema que resuelven: Tal como es el Proceso de Decisión de Markov. Además se considera el Proceso de Decisión de Markov desde el caso determinístico y estocástico, así como también su solución óptima y finalmente, llegar a una ecuación de recursividad de Bellman.

3.1. Control del Robot con Recompensas y Retornos

En el aprendizaje por reforzamiento, el propósito u objetivo del controlador, es el recibir una señal de recompensa que pasa del proceso al controlador. En cada lapso de tiempo, la recompensa es un simple número, $r_k \in \mathbb{R}$. En palabras simples, el objetivo del controlador será el de maximizar la cantidad total de recompensas que recibe, donde la recompensa no se maximiza de manera inmediata pero sí a largo plazo.

En general, se busca maximizar la esperanza del retorno, donde el retorno $R(x)$, se define como una función específica de la recompensa. El caso más simple del retorno es la suma de las recompensas:

$$R(x) = r_1 + r_2 + r_3 + \cdots + r_T \in \mathbb{R} \tag{3.1}$$

Donde T es el tiempo final. Este enfoque tiene mucho sentido en aplicaciones donde existe un tiempo final, que es, cuando la interacción entre el controlador y el proceso termina en una subsecuencia, llamada episodios. Cada episodio termina en un estado especial llamado *estado terminal*, seguido de una reinicialización a los estados iniciales prefijados o a una muestra de una distribución estándar de los estados iniciales.

Por otra parte, en muchos casos la interacción entre el controlador y el proceso no termina de manera natural en algún episodio, sino que se sigue continuamente sin límite. Por ejemplo, este sería el caso de una tarea de control de procesos continuo, o una aplicación a un robot con larga vida útil. Se le conoce como *tareas continuas.* La ecuación (3.1) resulta problemática para tareas continuas ya que el tiempo final sería $T = \infty$, y el retorno, que es lo que tratamos de maximizar, podría fácilmente ser infinito. (Por ejemplo: suponer que el controlador recibe una recompensa de $+1$ en cada instante de tiempo). Entonces el concepto adicional que necesitamos es el de *descuento.* En este enfoque el controlador trata de seleccionar acciones tal que la suma de las recompensas con descuento que recibe en el futuro sea maximizada. En particular se escoge u_k para maximizar la esperanza del retorno con descuento:

$$R(x) = r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{k+1}, \tag{3.2}$$

donde γ es un parámetro, $0 \leq \gamma \leq 1$, llamado *factor de descuento* y la ecuación (3.2) es ligeramente más compleja conceptualmente pero mucho más simple matemáticamente. El factor de descuento determina los valores presentes de las futuras recompensas: por ejemplo, una recompensa recibida en el tiempo k en el futuro sólo tendrá un valor de γ^{k-1} veces lo que valdría si fuera recibida de manera inmediata. Si $\gamma < 1$, la suma infinita tiene ahora un valor finito, siempre y cuando la secuencia de la recompensa $\{R\}$ esté acotada. Si $\gamma = 0$, al agente se le conoce como *miope* ya que sólo se enfoca en maximizar la recompensa inmediata: su objetivo en este caso es aprender como escoger u_k para maximizar sólo r_{k+1}. Si cada acción del controlador tuviera una influencia sólo en la recompensa inmediata y no de las futuras

recompensas, entonces el controlador miope podría maximizar (3.2) de manera separada únicamente maximizando cada recompensa inmediata. Pero en general, el sólo maximizar la recompensa inmediata puede reducir el acceso a futuras recompensas de modo que el retorno puede reducirse. Mientras γ se aproxime a 1, el objetivo toma en consideración las recompensas futuras con más fuerza, el controlador llega a ser más clarividente.

3.2. Control del Robot en el Proceso de Decisión de Markov

En particular, formalmente definimos la propiedad del ambiente y sus señales de estado conocida como la *propiedad de Markov*. Se supone que existe un número finito de estados y recompensas, lo cual permite trabajar en términos de sumas y probabilidades en lugar de integrales y densidad de probabilidades, pero el argumento fácilmente puede ser extendido a incluir recompensas y estados continuos. Consideramos como el ambiente podría responder en el tiempo $k+1$ a la acción tomada en el tiempo k. En el caso causal más general, esta respuesta podría depender de todo lo sucedido antes. En este caso, la dinámica puede ser definida solamente especificando la distribución de probabilidad:

$$\Pr\{r_{k+1} = r, x_{k+1} = x' \mid x_0, u_0, r_1, \ldots, x_{k-1}, u_{k-1}, r_k, x_k, u_k\}, \tag{3.3}$$

para todo x' , r, y todos los posibles valores de los eventos pasados, entradas, salidas y recompensas: $x_0, u_0, r_1, \ldots, x_{k-1}, u_{k-1}, r_k, x_k, u_k$. Si el estado tiene la propiedad de Markov, entonces la respuesta del ambiente al tiempo $k+1$ dependerá únicamente del estado y la acción tomada al tiempo k, por lo que en este caso la dinámica del ambiente puede ser definida únicamente por:

$$\Pr\{r_{k+1} = r, x_{k+1} = x' \mid x_k, u_k\}. \tag{3.4}$$

Para todo x', r, x_k, y u_k. En otras palabras, el estado tiene la propiedad de Markov, y es un estado de Markov, si y sólo si la ecuación (3.3) es igual a la ecuación (3.4) para todo x', r

y las historias de $x_0, u_0, r_1, \ldots, x_{k-1}, u_{k-1}, r_k, x_k, u_k$. En este caso, se dice que el ambiente y las tareas tienen la propiedad de Markov.

Los problemas de DP y RL se pueden formalizar con la ayuda de MDP (Puterman, 1994). Primero presentamos el caso más simple de MDP con transiciones de estado deterministas. Luego, extendemos la teoría al caso estocástico.

3.3. Control de un Robot en el Caso Determinístico

Un MDP determinístico está definido por el espacio de estados X del proceso, el espacio de acción U del controlador, la función de transición f del proceso (la cual describe cómo cambia el estado como resultado de las acciones de control), y la función de recompensa ρ (la cual evalúa el desempeño del control inmediato). Como un resultado de la acción u_k aplicada en el estado x_k en el tiempo discreto k, el estado cambia a x_{k+1}, de acuerdo a la función de transición $f : X \times U \rightarrow X$:

$$x_{k+1} = f(x_k, u_k).$$

Al mismo tiempo el controlador recibe la señal de recompensa escalar r_{k+1}, de acuerdo con la función de recompensa $\rho : X \times U \rightarrow \mathbb{R}$:

$$r_{k+1} = \rho(x_k, u_k),$$

donde se asume que $\| \rho \|_\infty = \sup_{x,u} \mid \rho(x,u) \mid$ es finita. La recompensa evalúa el efecto inmediato de la acción u_k, conocida como la transición de x_k a x_{k+1}, pero en general, no nos dice nada sobre los efectos a largo plazo.

El controlador escoge acciones de acuerdo con su política $h : X \rightarrow U$, usando:

$$u_k = h(x_k).$$

Dados f y ρ, el estado actual x_k y la acción actual u_k son suficientes para determinar tanto el siguiente estado x_{k+1} como la recompensa r_{k+1}. Esta es la propiedad de Markov que

es esencial para proporcionar las garantías teorícas sobre los algoritmos DP/RL. Algunos procesos de desición de Markov tienen estados terminales que, una vez que son alcanzados ya no los pueden dejar; y todas las recompensas recibidas en el estado terminal son 0. En la literatura del RL frecuentemente se usan ensayos o episodios para referirse a trayectorias que empiezan en un estado inicial y finalizan en un estado terminal.

3.3.1. Optimización en el Caso Determinístico

En DP y RL, el objetivo es encontrar una política óptima que maximice el retorno desde cualquier estado inicial x_0. El retorno es una agregación acumulativa de las recompensas a lo largo de las trayectorias empezando desde x_0. Representa de forma concisa la recompensa obtenida por el controlador a largo plazo. Diferentes tipos de retornos existen, dependiendo de la manera en que se acumula la recompensa. (Bertsekas y Tsitsiklis, 1996; Kaelbling 1996). El objetivo del aprendizaje en los procesos de decisión de Markov MDP es acumular recompensas. Si el controlador sólo tomara en cuenta la recompensa inmediata, un simple criterio de optimización sería optimizar (R). Sin embargo, hay varias maneras de tomar esto en cuenta. Existen básicamente tres modelos de optimización en los MDP que son suficientes para cubrir la mayoría de los enfoques en la literatura.

Horizonte infinito:

$$R(x_0) = \sum_{k=0}^{\infty} \gamma^k r_{k+1} = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, h(x_k)), \tag{3.5}$$

donde $\gamma \in [0, 1)$ es el factor de descuento y $x_{k+1} = f(x_k, h(x_k))$ para $k \geq 0$. El factor de descuento puede ser interpretado intuitivamente como que tan clarividente es el controlador al considerar sus recompensas, o también como una manera de tomar en cuenta la creciente incertidumbre sobre las recompensas futuras. Desde un punto de vista matemático, el descuento asegura que el retorno siempre estará acotado si las recompensas son acotadas. El objetivo es por lo tanto maximizar el desempeño a largo plazo (retornos), mientras sólo usamos la realimentación del desempeño inmediato (recompensa). Esto nos lleva al conocido reto de recompensas con retardo (Sutton y Barto, 1998): donde las acciones tomadas

en el presente afectan el potencial de lograr una buena recompensa en el futuro, pero las recompensas inmediatas no proveen información sobre los efectos a largo plazo.

Igualmente, otro tipo de retornos pueden ser definidos, como el retorno sin descuento, obtenido al dejar γ con un valor de 1 en la ecuación (3.5), donde simplemente suma las recompensas sin realizar algún descuento. Desafortunadamente, el retorno horizonte finito sin descuento frecuentmente no está acotado. Una alternativa es usar el retorno *horizonte finito promedio*:

$$\lim_{k\to\infty} \frac{1}{k} \sum_{k=0}^{k} \rho(x_k, h(x_k)),$$

el cual es acotado en muchos casos. Retornos de horizonte finito pueden ser obtenidos mediante la acumulación de las recompensas a lo largo de las trayectorias finitas de longitud K (el horizonte), en lugar de trayectorias infinitas. De hecho, el retorno de *horizonte finito* con descuento puede ser definido como:

$$\sum_{k=0}^{K} \gamma^k \rho(x_k, h(x_k)).$$

El retorno sin descuento ($\gamma = 1$), puede ser usado de manera más fácil en el caso del horizonte finito, que está acotado cuando las recompensas son acotadas. En este trabajo principalmente utilizaremos el retorno con descuento de Horizonte Infinito (3.5) por que cuenta con muchas propiedades teóricas y aunque es ligeramente más complejo conceptualmente, también mucho más simple matemáticamente. En particular, para este tipo de retornos, bajo ciertas suposiciones técnicas existe por lo menos una política óptima determinista estacionaria $h^* : X \to U$ (Bertsekas y Shreve, 1978, Capítulo 9). En contraste con el caso del horizonte finito, las políticas óptimas dependen en general del paso de muestreo k, i.e., no son estacionarias (Bertsekas, 2005a, Capítulo 1). Mientras que el factor de descuento γ puede ser considerado teóricamente como una parte del problema, en la práctica, se debe elegir un buen valor de γ. Escoger γ a menudo implica una compensación entre la calidad de la solución y la tasa de convergencia del algortimo DP/RL, por las siguientes razones: algunos

algoritmos importantes convergen más rápido cuando γ es más pequeño (este es el caso de iteraciones basadas en modelo). Sin embargo, si γ es muy pequeño, la solución puede llegar a ser no satisfactoria por que no toma suficientemente en cuenta las recompensas obtenidas después de una gran cantidad de pasos. Asimismo, se puede interpretar γ de varias maneras: puede ser visto como un factor de interés, una probabilidad de que exista otro paso, o como un truco matemático para acotar la suma infinita. Además el modelo con descuento es matemáticamente más tratable que el modelo del horizonte finito. Esta es una de las razones por la cual este modelo ha recibido gran atención.

3.3.2. Funciones Valor y Ecuación de Bellman

Una manera conveniente de caracterizar las políticas es por medio de sus funciones valor. Dos tipos de funciones valor existen: funciones de valor estado-acción (funciones Q) y funciones de valor estado (funciones V). Frecuentemente en la literatura el nombre de funciones valor se utiliza tanto para funciones Q como para funciones V, en este texto se utilizará el nombre de función Q y función V para diferenciarlos claramente uno del otro. Primero definiremos las funciones Q y más adelante se explicarán las funciones V.

La función Q que está definida como $Q^h : X \times U \rightarrow \mathbb{R}$ de una política h nos da como resultado el retorno obtenido cuando empezamos desde un estado dado, aplicando una acción dada, y siguiendo una política h por lo tanto tenemos:

$$Q^h(x,u) = \rho(x,u) + \gamma R^h(f(x,u)), \tag{3.6}$$

donde $R^h(f(x,u))$ es el retorno del siguiente estado $f(x,u)$. Esta representación de la fórmula puede ser obtenida si se escribe $Q^h(x,u)$ de manera explícita como una suma de las recompensas con descuento obtenido, tomando la acción u en el estado x y siguiendo la política h,

$$Q^h(x,u) = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k),$$

donde $(x_0, u_0) = (x, u)$, $x_{k+1} = f(x_k, u_k)$ para $k \geq 0$, y $u_k = h(x_k)$ para $k \geq 1$. Entonces, podemos separar el primer termino de la sumatoria:

$$
\begin{aligned}
Q^h(x,u) &= \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k) \\
Q^h(x,u) &= \gamma^0 r_1 + \gamma^1 r_2 + \gamma^2 r_3 + \cdots \\
Q^h(x,u) &= r_1 + \gamma^1 r_2 + \gamma^2 r_3 + \cdots \\
Q^h(x,u) &= r_1 + \sum_{k=1}^{\infty} \gamma^k r_{k+1} \\
Q^h(x,u) &= \rho(x,u) + \sum_{k=1}^{\infty} \gamma^k \rho(x_k, u_k) \\
Q^h(x,u) &= \rho(x,u) + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} \rho(x_k, h(x_k)) \\
Q^h(x,u) &= \rho(x,u) + \gamma R^h(x_{k+1}) \\
Q^h(x,u) &= \rho(x,u) + \gamma R^h(f(x,u))
\end{aligned}
$$

La función óptima Q se define como la mejor función Q que puede ser obtenida por cualquier política de la siguiente manera:

$$Q^*(x,u) = \max_h Q^h(x,u). \tag{3.7}$$

Cualquier política h^* que seleccione en cada estado una acción que genere el valor más grande de la función óptima Q :

$$h^*(x) \in \arg\max_u Q^*(x,u), \tag{3.8}$$

es óptima (nos dice que maximiza el retorno). En general, para una función Q dada, tener una política h que satisface

$$h(x) \in \arg\max_u Q(x,u), \tag{3.9}$$

se le conoce como una acción ambiciosa en Q. Entonces para encontrar una política óptima basta con encontrar una Q^* y después aplicar la ecuación (3.8) para calcular una política en

Q^*.

Las funciones Q^h y Q^* se caracterizan recursivamente por la ecuación de Bellman, y que además son de importancia central para los algoritmos de valor de iteración y de política de iteración. La ecuación de Bellman para Q^h, nos dice que el valor de tomar una acción u en el estado x bajo la política h es igual a la suma de las recompensas inmediatas y el valor descontado alcanzado por h en el siguiente estado:

$$Q^h(x,u) = \rho(x,u) + \gamma Q^h(f(x,u), h(f(x,u))) \tag{3.10}$$

Esta ecuación de Bellman puede ser obtenida de la ecuación (3.6) como sigue a continuación:

$$\begin{aligned}
Q^h(x,u) &= r_1 + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} r_{k+1} \\
Q^h(x,u) &= r_1 + \gamma \left[r_2 + \gamma \sum_{k=2}^{\infty} \gamma^{k-2} r_{k+1} \right] \\
Q^h(x,u) &= \rho(x,u) + \gamma \left[\rho(f(x,u), h(f(x,u))) + \gamma \sum_{k=2}^{\infty} \gamma^{k-2} \rho(x_k, h(x_k)) \right] \\
Q^h(x,u) &= \rho(x,u) + \gamma Q^h(f(x,u), h(f(x,u)))
\end{aligned}$$

donde $(x_0, u_0) = (x, u)$, $x_{k+1} = f(x_k, u_k)$ para $k \geq 0$, y $u_k = h(x_k)$ para $k \geq 1$.

La ecuación de Bellman que representa a Q^*, donde establece que el valor óptimo de la acción u tomada en el estado x es igual a la suma de las recompensas inmediatas y al valor óptimo con descuento obtenido por la mejor acción en el siguiente estado:

$$Q^*(x,u) = \rho(x,u) + \gamma \max_{\acute{u}} Q^*(f(x,u), u^{'})). \tag{3.11}$$

La función $V^h : X \rightarrow \mathbb{R}$ de una política h es el retorno obtenido empezando desde un estado particular y siguiendo la política h :

$$V^h(x) = R^h(x) = Q^h(x, h(x)). \tag{3.12}$$

La función óptima V se obtiene como la mejor función V que puede ser obtenida por cualquier política, y puede ser calculada desde la función óptima Q:

$$V^*(x) = \max_h V^h(x) = \max_u Q^*(x,u). \tag{3.13}$$

Y finalmente, una polítca óptima h^* puede ser calculada desde V^*, usando el hecho de que satisface:

$$h^*(x) \in \arg\max_u \left[\rho(x,u) + \gamma V^*(f(x,u))\right]. \tag{3.14}$$

Usando esta formula es más difícil que usar la ecuación (3.8); en particular, un modelo de MDP se necesita en la forma de la dinámica f y la recompensa ρ. Debido a que la función Q también depende de la acción, esta ya incluye la información sobre la calidad de la transición. Por el contrario, la función V sólo describe la calidad del estado, y para inferir sobre la calidad de las transiciones, estas deben tenerse en cuenta de manera explícita. Esto es lo que vemos en la ecuación (3.14), y esto explica por que es más difícil calcular políticas desde las funciones V. Debido a estas diferencias las funciones Q se preferirán a las funciones V, aunque sea más costoso que representar funciones V, ya que la función Q depende tanto de x y u.

Las funciones V^h y V^* satisfacen las siguientes ecuaciones de Bellman, que son similares a las ecuaciones (3.10) y (3.11):

$$V^h(x) = \rho(x, h(x)) + \gamma V^h(f(x, h(x)))$$

$$V^*(x) = \max_u \left[\rho(x,u) + \gamma V^*(f(x,u))\right]$$

3.4. Control de un Robot en el Caso Estocástico

Una tarea del aprendizaje por reforzamiento que satisface la propiedad de Markov se le conoce como Proceso de decisión de Markov MDP. Si el espacio de estados y las acciones son

finitos entonces se le conoce como proceso de decisión de Markov finito (MDP finito). MDP finito son muy importantes en la teoría del aprendizaje por reforzamiento. Dada cualquier estado y acción, x y u, la probabilidad de que el siguiente estado sea $x^{'}$ es:

$$p(x_{k+1} \mid x_k, u_k) = \tilde{f}(x_k, u_k, x^{'}) = \Pr\{x_{k+1} = x' \mid x_k = x, u_k = u\}. \tag{3.15}$$

Estas cantidades se llaman *probabilidades de transición*. Similarmente dado cualquier estado o acción actual, x, u, junto con el siguiente estado, $x^{'}$, la esperanza del valor de la siguiente recompensa es:

$$r_{k+1} = \tilde{\rho}(x_k, u_k, x_{k+1}) = E\{r_{k+1} \mid x_k = x, u_k = u, x_{k+1} = x'\}. \tag{3.16}$$

Estas cantidades $p(x_{k+1} \mid x_k, u_k)$ y $\tilde{\rho}(x_k, u_k, x_{k+1})$, especifican completamente los aspectos más importantes de la dinámica de un MDP finito.

3.4.1. Funciones de Valor y Ecuación de Bellman

Uno de los objetivos en el aprendizaje por reforzamiento es estimar qué *tan bueno* es estar en un estado (o estar en un estado y realizar una acción). La noción de qué "tan bueno"se define en términos de las futuras recompensas o la esperanza de las recompensas acumuladas que son representadas como funciones de valor. La función valor de un estado x denotado por $V^h(x)$, representa la esperanza total de la recompensa acumulada que el agente puede recibir iniciando en el estado x y siguiendo la política h. De manera similar, la función valor del estado x, tomando la acción u, denotado por $Q^h(x, u)$ representa la esperanza total de la recompensa acumulada que el agente puede recibir iniciando en el estado x, tomando la acción u y siguiendo la política h. La idea es encontrar una política que produzca el máximo de la función valor en lugar del máximo de una recompensa inmediata. Las recompensas son dadas por el proceso, pero las funciones valor necesitan ser estimadas (aprendidas) con la experiencia [5]. En consecuencia, las funciones valor son definidas con respecto a políticas particulares. Recordar que una política h, es un mapeo de cada estado $x \in X$, y acción $u \in U(x)$, a una probabilidad h de tomar una acción u encontrándose en el estado x. De manera

informal, el valor de un estado x bajo la política h, denotado por $V^h(x)$, es la esperanza del retorno cuando se inicia en el estado x y se sigue h. Para MDP, podemos definir formalmente $V^h(x)$ como:

$$V^h(x) = E\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid x_k = x\right], \tag{3.17}$$

donde $E[\cdot]$, representa la esperanza del valor dado cuando el agente sigue una política h, y k es cualquier paso de tiempo. Llamamos a la función $V^h(x)$ *la función del valor estado* para la política h.

Una propiedad fundamental de las funciones de valor es que satisfacen ciertas propiedades de recursividad. Para cualquier política h y cualquier estado x la expresión de la ecuación (3.17) puede ser definida recursivamente en términos de la ecuación de Bellman (1957):

$$\begin{aligned}
V^h(x) &= E\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid x_k = x\right] \\
V^h(x) &= E\left[r_1 + \gamma r_2 + \gamma^2 r_3 + \gamma^3 r_4 + \cdots \mid x_k = x\right] \\
V^h(x) &= E\left[r_1 + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} r_{k+1} \mid x_k = x\right] \\
V^h(x) &= E\left[\tilde{\rho}(x, u, x') + \gamma E\left[\sum_{k=1}^{\infty} \gamma^{k-1} r_{k+1} \mid x_{k+1} = x'\right] \mid x_k = x\right] \\
V^h(x) &= E\left[\tilde{\rho}(x, u, x') + \gamma V^h(x')\right]
\end{aligned}, \tag{3.18}$$

donde está implícito que las acciones u, son tomadas del conjunto $U(x)$, y los siguientes estados x', son tomados del conjunto de estados X. La ecuación (3.18) es la ecuación de Bellman para V. Expresa una relación entre el valor del estado y el valor de su estado sucesor.

Similarmente, se define el valor de tomar una acción u en el estado x bajo la política h, denotado por $Q^h(x, u)$, como la esperanza del retorno iniciando en x, tomando la acción u, y siguiendo la política h.

$$Q^h(x,u) = E\left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} \mid x_k = x, u_k = u\right],$$

donde Q^h se le conoce como *función valor-acción* para la política h. *(Sutton and Barto 2012).*

Haciendo un análisis de recursividad como el anterior podemos derivar una ecuación equivalente para $Q^h(x,u)$:

$$Q^h(x,u) = E\left[\tilde{\rho}(x,u,x^{'}) + \gamma Q^h(x^{'}, h(x^{'}))\right]$$

3.4.2. Ecuación de Optimización de Bellman

En la práctica, las mejores políticas son obtenidas por aquellas que producen la esperanza de la recompensa acumulada más grande. Una política h se considera mejor o igual que otra política $h^{'}$ si la esperanza de los retornos es más grande o igual que los de $h^{'}$ para todos los estados.

$h \geq h'$ si $V^h(x) \geq V^{h'}$ para todo $x \in X$.

Existe al menos una política que es mejor o igual que todas las demás políticas, llamada *política óptima*, aunque podría existir más de una, representamos todas las políticas óptimas como h^*. Su función valor conocida como *función de valor óptimo*, representado por V^* y definida como:

$$V^*(x) = \max_h V^h(x), \tag{3.19}$$

para todo $x \in X$.

Las políticas óptimas también comparten la misma *función de valor-acción* representada por Q^* y definida como:

$$Q^*(x,u) = \max_h Q^h(x,u). \tag{3.20}$$

Para todo $x \in X$ y $u \in U(x)$, para el par estado-acción (x,u), donde esta función da la esperanza del retorno al tomar la acción u en el estado x y luego seguir una política óptima. Entonces, se puede escribir Q^* en términos de V^* de la siguiente manera:

$$Q^*(x,u) = E[\tilde{\rho}(x,u,x^{'}) + \gamma V^*(x_{k+1}) \mid x_k = x, u_k = u]. \tag{3.21}$$

Considerando las ecuaciones (3.19) y (3.20), las funciones de valor óptimo pueden ser expresadas recursivamente con la ecuación de optimización de Bellman de la siguiente manera:

$$\begin{aligned} V^*(x) &= \max_u Q^h(x,u) \\ V^*(x) &= \max_u E\left[\sum_{k=0}^{\infty}\gamma^k r_{k+1} \mid x_k = x, u_k = u\right] \\ V^*(x) &= \max_u E\left[r_{k+1} + \gamma\sum_{k=1}^{\infty}\gamma^{k-1} r_{k+1} \mid x_k = x, u_k = u\right] \\ V^*(x) &= \max_u E\left[\tilde{\rho}(x,u,x') + \gamma V^*(x_{k+1}) \mid x_k = x, u_k = u\right] \\ V^*(x) &= \max_u E\left[\tilde{\rho}(x,u,x') + \gamma V^*(x')\right] \end{aligned} \tag{3.22}$$

y similarmente para Q:

$$\begin{aligned} Q^*(x,u) &= E\left[r_{k+1} + \gamma \max_{u'} Q^*(x_{k+1},u') \mid x_k = x, u_k = u\right] \\ Q^*(x,u) &= E\left[\tilde{\rho}(x,u,x') + \gamma \max_{u'} Q^*(x',u')\right]. \end{aligned} \tag{3.23}$$

Uno de los principales avances en la investigación de métodos de aprendizaje por refuerzo, fue el de la introducción de los métodos de diferencia temporal (TD), los cuales son una clase de procedimientos de aprendizaje incremental especializados en problemas de predicción. Dichos métodos son conducidos por el error o diferencia entre predicciones sucesivas temporales de los estados. El aprendizaje ocurre en cada momento que se produce un cambio en la predicción al paso del tiempo.

El método más simple conocido como **TD (0)** actualiza la estimación de la función de valor, después de ir del estado x al estado x_{k+1} y de recibir la recompensa r, utilizando la siguiente regla:

$$V_{k+1}(x) \rightarrow V_k(x) + \alpha[r + \gamma V_k(x') - V_k(x)]$$

Varios métodos populares tales como Q-Learning, Sarsa y los métodos de Actor-Crítico fueron desarrollados basándose en dicha regla.

Aunque el aprendizaje por reforzamiento es una estrategia difícil para resolver el problema de aprendizaje automático, promete el desarrollo de sistemas computacionales capaces

de auto-mejorar sus desempeños, lo cual es un objetivo principal de la comunidad de investigadores en inteligencia artificial. Dos de las aplicaciones más impresionantes de estos métodos son el jugador TD-Gammon (Gerald Tesauro, IBM, 1992), y el algoritmo de control de helicóptero. El primer trabajo inyectó nuevo interés en el estudio de los métodos de aprendizaje por refuerzo; mientras que el segundo trabajo es una de las mejores aplicaciones del mundo real que se han logrado con los métodos de RL.

3.5. Control de un Robot con Q-Learning

Q-Learning es un algoritmo de control por diferencia temporal off -policy que aproxima directamente la función de valor estado-acción óptima, independientemente de la política seguida. Es uno de los algoritmos de aprendizaje por refuerzo más populares. El algoritmo del Esquema 1. muestra los pasos a seguir de manera secuencial. Si en el límite los valores de las acciones para todos los pares de estado-acción son actualizados un número infinito de veces, con un valor decreciente de α entonces el algoritmo converge a Q con probabilidad 1.

Esquema 1. Algoritmo Q-Learning

Inicializar $Q(x, u)$ arbitrariamente
para cada episodio de entrenamiento **hacer**
inicializar x
repetir para cada paso del episodio
escoger u desde x usando la política derivada de Q (ej..u ambiciosa)
realizar la acción u, observar $r, x^{'}$
$Q(x, u) \leftarrow Q(x, u) + \alpha[r + \gamma \max_u Q(x^{'}, u) - Q(x, u)]$
$x \leftarrow x^{'}$
hasta x es terminal
fin

También podemos encontrar dentro del algoritmo Q-Learning dos aspectos importantes a considerar: La representación tabular y la discretización del espacio de estados.

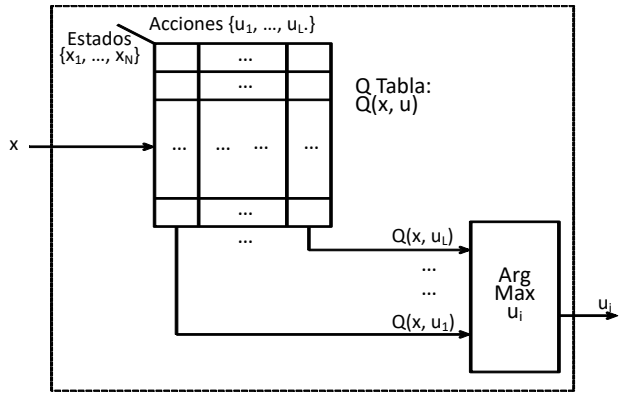


Figura 3.1: Representación tabular

En la figura (3.1) podemos ver la representación tabular:

La figura (3.2) muestra una discretización del espacio de estados.

3.6. Simulaciones.

3.6.1. Doble péndulo invertido sobre un móvil

Como una planta no lineal subactuada, el péndulo doble invertido sobre el carro plantea un problema de control interesante para algoritmos que no depende de modelo. Además, es una de las herramientas más atractivas para probar leyes de control lineal y no lineal.

Descripción y planteamiento del problema

El doble péndulo invertido sobre el carro es una extensión del péndulo invertido en el carro. El objetivo es estabilizar ambos péndulos en la posición vertical invertida mientras se mantiene una posición deseada sobre el desplazamiento del carro. La dificultad para controlar el doble péndulo invertido es el hecho de que es un sistema caótico por naturaleza. Por lo tanto, es extremadamente difícil, casi imposible predecir con precisión el movimiento de los péndulos. El mecanismo está formado por tres cuerpos rígidos, como se muestra en la figura

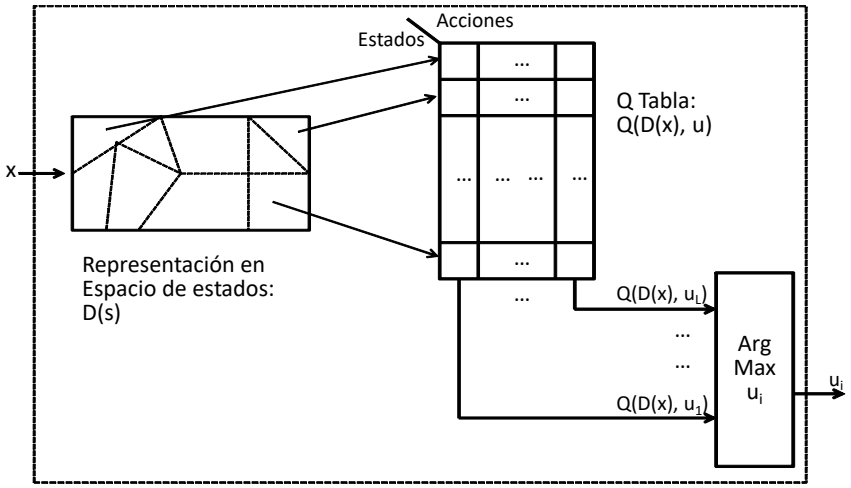


Figura 3.2: Discretización del espacio de estados.

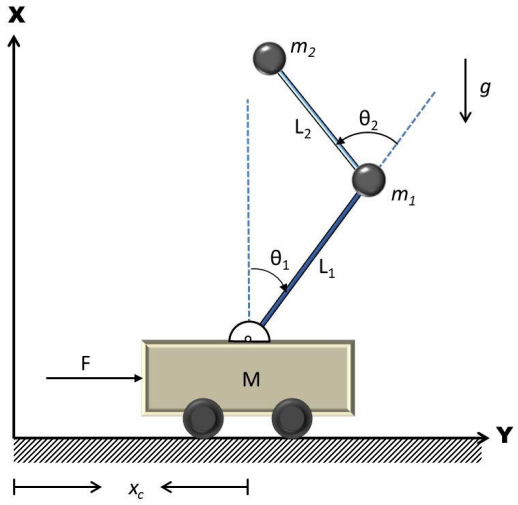


Figura 3.3: Doble péndulo invertido sobre el carro

(3.3) un carro de masa M, acoplado a través de una articulación de rotación a una barra con masa m_1, y longitud l_1. A su vez, a la primera barra esta acoplada, en el otro extremo y también a través de una articulación de rotación, una segunda barra de masa m_2 y longitud l_2.

Planteamiento del problema

Se desea mantener el segundo péndulo (péndulo superior) sobre la vertical invertida mientras el primer péndulo no mantiene alguna restricción en su movimiento. Además, el carro conserva un rango de ± 2 metros en su desplazamiento.

Las condiciones iniciales son que ambos péndulos empiecen en su equilibrio inestable (origen) y con velocidad cero.

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ x_c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \quad \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Su modelo matemático está representado de la siguiente manera:

$$\begin{bmatrix} m_1 + m_2 + M & m_1 l_1 \cos(\theta_1) + m_2 l_1 \cos(\theta_1) + m_2 l_2 \cos(\theta_1 + \theta_2) & m_2 l_2 \cos(\theta_1 + \theta_2) \\ m_1 l_1 \cos(\theta_1) + m_2 l_1 \cos(\theta_1) + m_2 l_2 \cos(\theta_1 + \theta_2) & m_1 l_1^2 + m_2 l_1^2 + m_2 l_2^2 + 2 m_2 l_1 l_2 \cos(\theta_2) & m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2) \\ m_2 l_2 \cos(\theta_1 + \theta_2) & m_2 l_2^2 + m_2 l_1 l_2 \cos(\theta_2) & m_2 l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{x}_c \\ \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix}$$
$$+ \begin{bmatrix} 0 & -(m_1 l_1 \sin(\theta) + m_2 l_1 \sin(\theta_1) + m_2 l_2 \sin(\theta_1 + \theta_2))\dot{\theta} & -m_2 l_2 \sin(\theta_1 + \theta_2)(2\dot{\theta}_1 + \dot{\theta}_2) \\ 0 & -2 m_2 l_1 l_2 \dot{\theta}_2 \sin(\theta_2) & -m_2 l_1 l_2 \dot{\theta}_2 \sin(\theta_2) \\ 0 & m_2 l_1 l_2 \dot{\theta}_1 \sin(\theta_2) & 0 \end{bmatrix} \begin{bmatrix} \dot{x}_c \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix}$$
$$+ \begin{bmatrix} 0 \\ -m_1 g l_1 \sin(\theta_1) - m_2 g l_1 \sin(\theta_1) - m_2 g l_2 \sin(\theta_1 + \theta_2) \\ -m_2 g l_2 \sin(\theta_1 + \theta_2) \end{bmatrix} = \begin{bmatrix} f \\ 0 \\ 0 \end{bmatrix}$$

Tenemos un sistema dinámico de tres ecuaciones diferenciales de segundo orden o también se puede ver como seis ecuaciones diferenciales de primer orden (espacio de estados) donde θ_1 y $\dot{\theta}_1$ representan la posición y velocidad del primer péndulo, θ_2 y $\dot{\theta}_2$, representa la posición y velocidad del segundo péndulo y finalmente x_c y $\dot{x}_c$ son la posición y velocidad del carro.

Diseño de la ley de control

El algoritmo del aprendizaje por reforzamiento Q-Learning no necesita de la dinámica del sistema, sin embargo si se necesita de una discretización del espacio de estados x_d y una representación tabular Q para la matriz Q-Learning. Las posiciones están limitadas a $\theta_1 \in [-\pi/4, \pi/4]\ rad, \theta_2 \in [-\pi/4, \pi/4]\ rad$ y $x_c \in [-2, 2]$ m, las velocidades están restringidas a $\dot{\theta}_1 \in [-\pi, \pi]\ rad/s, \dot{\theta}_2 \in [-\pi, \pi]\ rad/s$ y $\dot{x}_c \in [-0{,}5, 0{,}5]$ m/s y la fuerza aplicada al carro está representada por $f \in \{-10, 0, 10\}\ N$.

Se genera una discretización del espacio de estados basado en las posibles posiciones y velocidades en las que el péndulo superior podría estar. Las posiciones se definen como $z_1 = [-\pi/4, \pi/4]$ discretizado en pasos de 0,01 los cual produce 158 estados. Las velocidades se definen como $z_2 = [-\pi, \pi]$ discretizado en pasos de 0,01 lo cual produce 629 estados. La discretización del espacio de estados tendrá $158 \times 629 = 99382$ filas que representan todas las posibles combinaciones de posiciones y velocidades que podría tomar el péndulo superior. Además, se tendrán dos columnas una para definir posiciones y otra para definir las velocidades. Finalmente, se tiene una discretización del espacio de estados como $x_d \in \mathbb{R}^{99382\times 2}$.

La matriz Q tendrá una dimensión 99382 filas y 3 columnas. Se tienen 3 columnas asociadas a las 3 entradas de control $f \in \{-10, 0, 10\}$, por lo tanto $Q \in \mathbb{R}^{99382\times 3}$.

La posición que se va a controlar es la sombra que proyecta el péndulo superior sobre el piso definido de la siguiente manera:

$$x_{sombra} = x_c + l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2)$$

y su velocidad está definida como:

$$\dot{x}_{sombra} = \dot{x}_c + l_1\dot{\theta}_1\cos(\theta_1) + l_2(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1 + \theta_2).$$

En cada muestreo se genera un vector con las posiciones y velocidades que proyecta la sombra del péndulo superior y con la misma dimensión del espacio de estados discreto.

$$x_s = \begin{bmatrix} x_{sombra} \\ \dot{x}_{sombra} \end{bmatrix}^T \in \mathbb{R}^{99382\times 2}.$$

El error está definido como:

$$e_x = x_d - x_s$$

y lo que se busca es la fila con el error más cercano a cero, min e_x.

Las acciones están definidas como la fuerza aplicada al péndulo invertido

$$u = \{-10, 0, 10\},$$

y nuestro objetivo es encontrar una política f (ley de control) que maximice el retorno esperado.

$$f = \arg\max_u \left[Q_{t+1}(e_{x_t}, u_t)\right].$$

La matriz Q se inicializa en ceros y el algoritmo Q-Learning está definido de la siguiente forma:

$$Q_{t+1}(e_{x_t}, u_t) = Q_t(e_{x_t}, u_t) + \alpha\left[r_{t+1} + \gamma\max_{u'}Q_t(e_{x_{t+1}}, u') - Q_t(e_{x_t}, u_t)\right],$$

la recompensa estará definida de la forma:

$$r_{t+1} = -\left|\theta_2\right|^2 - 0{,}25\left|\dot{\theta}_2\right|^2,$$

donde la mayor recompensa se produce cuando la posición θ_2 y la velocidad $\dot{\theta}_2$ son cero, es decir, cuando el péndulo superior se encuentra sobre la vertical invertida y su velocidad es cero. Además, en la función de recompensa, la velocidad se encuentra escalada por un factor de 0,25 con la finalidad de no castigar al algoritmo de control a cambios bruscos de velocidad.

Resultados del aprendizaje

Para mostrar la efectividad del desempeño del controlador se realizaron las simulaciones bajo la plataforma Matlab. Los parámetros utilizados se muestran en la Tabla [1].

Tabla 1. Parámetros del doble péndulo sobre el carro y del controlador

Parámetros	Descripción	Valor
m_1	Masa del péndulo 1	0,5 kg
m_2	Masa del péndulo 2	0,5 kg
M	Masa del carro	0,1 kg
l_1	Longitud del péndulo 1	0,5 m
l_2	Longitud del péndulo 2	0,5 m
g	Gravedad	9,8 m/s^2
α	Tasa de aprendizaje	0,99
γ	Factor de descuento	0,9
f	Acciones de entrada	$\{0, -10, 10\}$ N

En la Tabla [2] se muestran todos los intentos que se realizaron para que el algoritmo lograra estabilizar al péndulo.

Tabla 2. Episodios e Iteraciones del péndulo doble sobre el carro

Episodio	Iterar	Posición $rads$ Péndulo 1	Velocidad Péndulo 1 $\frac{rads}{s}$	Posición $rads$ Péndulo 2	Velocidad Péndulo 2 $\frac{rads}{s}$
1	1	5	±20	±,8	±18
2	1	5	±14	±,8	±14
3	180	2,5	−18	±0,5	±10
4	150	3	−17	±0,4	±10
5	350	±0,05	±0,1	±0,028	±0,9
6	400	±0,04	±0,9	±0,025	±0,8
7	400	±0,02	±0,8	±0,023	±0,8
8	400	±0,02	±0,8	±0,020	±0,7

Continuación de la Tabla 2				
Episodios	Iteraciones	Posición Carro m	Velocidad Carro m/s	Éxito\Falla
1	1	-6	-17	Falla
2	1	-7	-15	Falla
3	180	-4	10	Falla
4	150	-4	9,5	Falla
5	350	0,05	$\pm$0,41	Falla
6	400	0,05	$\pm$0,41	Éxito
7	400	0,04	$\pm$0,40	Éxito
8	400	0,04	$\pm$0,40	Éxito

En la figura (3.4) y (3.5) podemos ver que hasta el episodio 6 se logró estabilizar el primer y segundo péndulo, donde en los primeros episodios con mucha facilidad se venían a bajo los péndulos por no poder controlar el carro de manera correcta.

Finalmente, en la figura (3.6) observamos que el desplazamiento del carro tuvo una pequeña deriva sobre la posición y que la velocidad se mantuvo estable en un rango de $\pm$0,8.

Se observa que controlar dos péndulos invertidos sobre un carro resulta en una tarea interesante para el algoritmo Q-Learning por que el diseño de las recompensas tiene que ser ingenioso y debe estar en función de la posición y velocidad angular del péndulo superior. Además, la forma de discretizar el espacio de trabajo no sólo está en función de los dos ángulos del péndulo sino también en función de la velocidad del carro. Finalmente, vemos que en los primeros episodios los péndulos se caen en las primeras iteraciones pero pasado un número de episodios el algoritmo corrige y empieza a estabilizar a la planta de forma correcta.

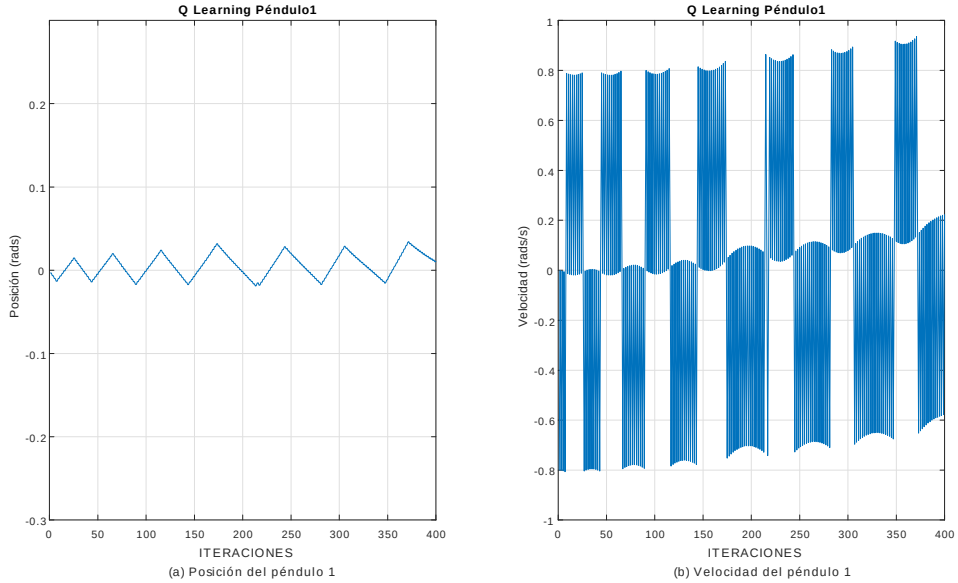


Figura 3.4: Posición y velocidad del péndulo 1

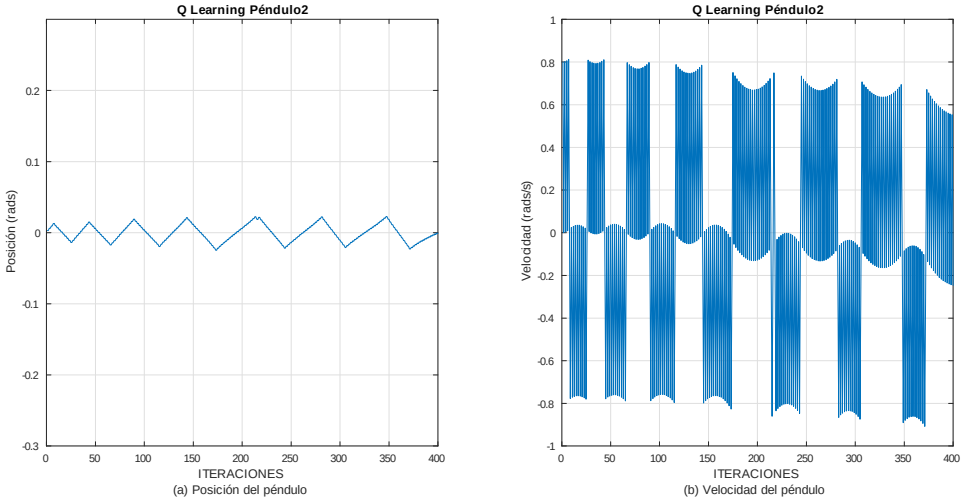


Figura 3.5: Posición y velocidad del péndulo 2

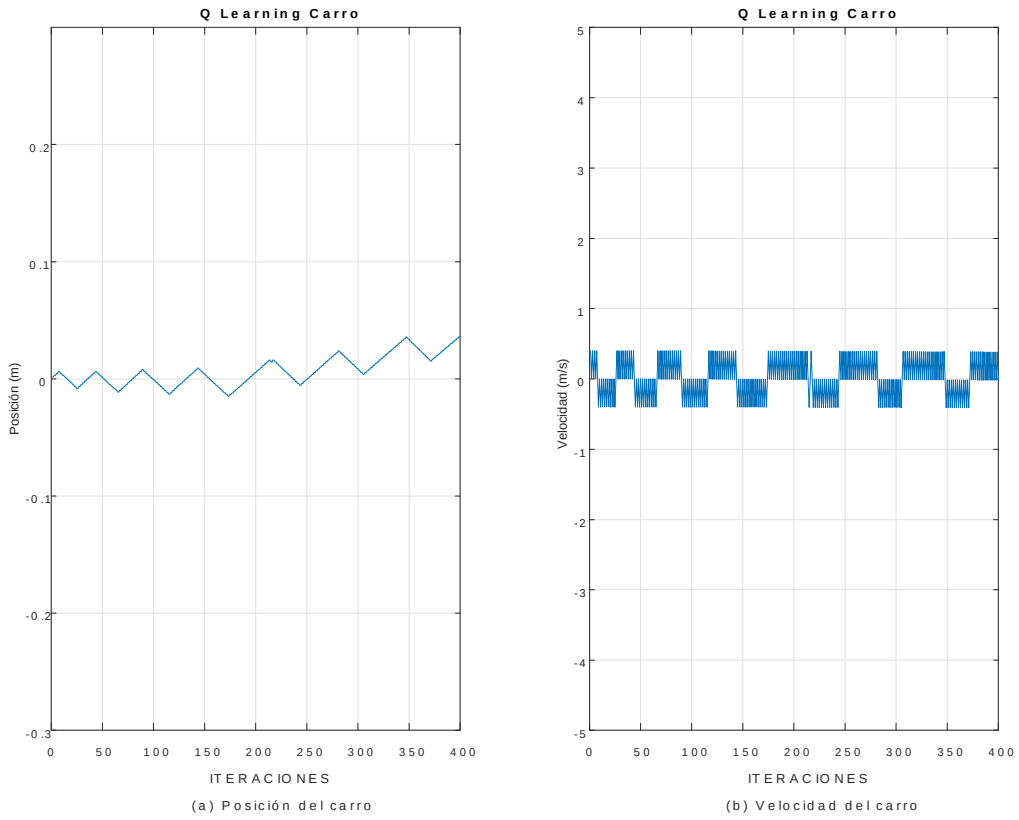


Figura 3.6: Posición y velocidad del carro

Capítulo 4

Control PD de sistemas electromecánicos usando como Compensación el Aprendizaje por Reforzamiento.

4.1. Introducción

En este capítulo se presentan los resultados de combinar las técnicas de control PD clásico con técnicas de aprendizaje por reforzamiento Q-Learning. Además, se detalla como este control híbrido tiene mejores resultados frente a sistemas o procesos que han tenído cambios en los parámetros de su modelo dinámico.

4.2. Control PD+QL

El algoritmo PD+QL tiene la ventaja de ser un control híbrido usando técnicas del aprendizaje por reforzamiento en conjunto con el clásico control PD. Este esquema de control híbrido brinda una mayor robustez a parámetros no modelados en la dinámica del sistema,

como lo es: la matriz de masas, inercias, longitudes, centros de masa, fricciones, términos gravitacionales, etc., además, también da solución al problema del error en estado estacionario. Otra de las ventajas de este algoritmo de control es que no necesita del conocimiento del modelo dinámico del sistema a controlar, lo cual resulta de lo más favorable al momento de seleccionar un esquema de control, debido a que el control es mucho más simple al usuario, y sin necesidad de tener los parámetros del sistema. Una ventaja más encontrada, es que este algoritmo híbrido resulta robusto ante perturbaciones generadas de forma externa, y que no fueron presentadas durante su aprendizaje.

Una de las desventajas que presenta este controlador, es debido a que utiliza algoritmos de inteligencia artificial, donde primero pasan por una etapa de entrenamiento que puede tomar segundos, minutos, horas o días, para llegar a dar solución al problema en curso. Otra de las desventajas que presenta, es buscar la mejor discretización del espacio de trabajo, ya que un espacio de trabajo donde el robot no va a operar, resulta en un tiempo de aprendizaje invertido que no va a ser reflejado al momento de realizar la tarea. Otra desventaja que encontramos, es en la selección de las recompensas que se le darán al robot cuando alcance una posición deseada de forma correcta. Como se mencionó anteriormente el aprendizaje por reforzamiento funciona en base al premio o al castigo, entonces, el determinar cómo premiar o como castigar al robot manipulador durante su aprendizaje, resulta en una tarea no sencilla y que para cada caso es diferente. Una última desventaja que podemos mencionar, es que para cada posición deseada asignada, el algoritmo tendrá que realizar el aprendizaje para ese punto asignado, y no se podrá asignar una nueva posición deseada, sin que pase por el aprendizaje primero.

En el caso de control de posición, el término derivativo en el esquema de control PD+QL, se interpreta como un efecto de amortiguamiento en la respuesta del robot, teniendo como finalidad disminuir los sobretiros y por lo tanto mejorar la respuesta. La velocidad $\dot{q}$ se emplea como inyección de amortiguamiento cuya principal acción es sobre el estado transitorio, puesto que en el estado estacionario la posición es una constante y la velocidad es cero. Si durante el estado estacionario se presentan oscilaciones, perturbaciones o cambios de referencia, entonces, la acción de control derivativa actúa de manera inmediata. La acción

derivativa no reduce la magnitud del error de posición en estado estacionario. Finalmente, la compensación QL disminuye el error en estado estacionario.

Este esquema de control nace de la necesidad de buscar un control robusto usando técnicas de la inteligencia artificial como lo es el aprendizaje por reforzamiento. Frecuentemente se proponen nuevas técnicas de control que buscan de alguna manera optimizar algún criterio, ya sea en el desempeño o en el esquema de control. Se pretende tener un controlador libre del modelo dinámico, y que en el caso del control de posición (regulación), tenga mejores resultados que los controladores clásicos [106], basados en un índice de desempeño de la integral del error absoluto, y la integral del tiempo por el error absoluto conocidos como *IAE, ITAE.*

La dinámica de un robot manipulador rígido de n eslabones de cadena abierta se escribe como [Spong, Vidyasagar 1989]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) = \tau, \tag{4.1}$$

donde $q \in \mathbb{R}^n$ es el vector de variables articulares y determina la posición de los eslabones, $\dot{q} \in \mathbb{R}^n$ es el vector de variables articulares, $M(q) \in \mathbb{R}^{n\times n}$ es la matriz de inercia, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ es la matriz de fuerzas centripetas y de coriolis, $G(q) \in \mathbb{R}^n$ representa el vector de gravedad, $F(\dot{q}) \in \mathbb{R}^n$ es un vector que contiene los términos de fricción, y $\tau \in \mathbb{R}^n$ es el vector de entradas de control.

Se puede realizar la combinación del controlador propuesto mediante técnicas del aprendizaje por reforzamiento con técnicas de control clásico como lo es el control Proporcional Derivativo (PD). A continuación se muestra este control híbrido PD+QL para el caso de regulación $\dot{q}_d = 0$.

$$\tau = K_p\tilde{q} - K_d\dot{q} + u_r, \tag{4.2}$$

donde $\tilde{q} = q_d - q$ está definido como el error entre la posción del robot y la variable deseada, donde $K_p, K_d \in \mathbb{R}^{n\times n}$, son matrices diagonales definidas positivas, simétricas y constantes, que representan las ganancias del control PD.

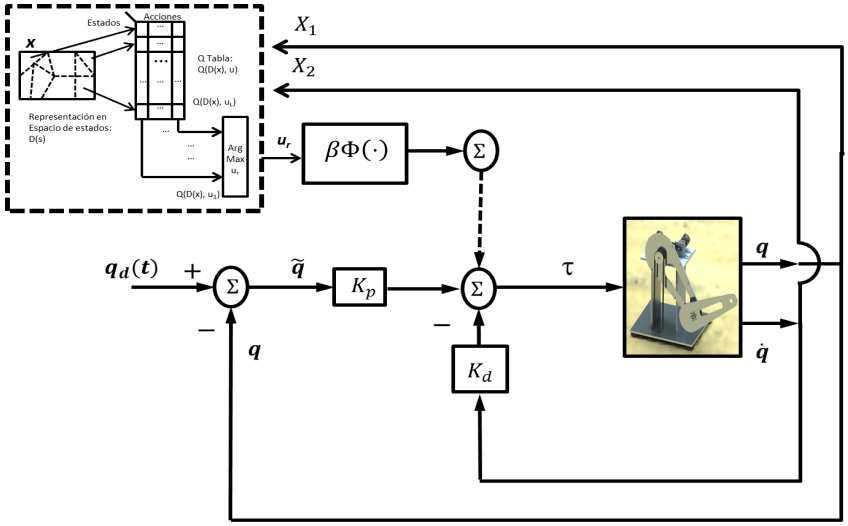


Figura 4.1: Esquema de control PD con compensación QL

Además u_r es la entrada de control obtenida del aprendizaje por reforzamiento y se representa de la siguiente manera:

$$u_r = \beta\Phi(\cdot)$$

donde $\Phi(\cdot)$ es una función que representan las acciones de control $\{-1, 0, 1\}$ del controlador Q-Learning, y β es una matriz diagonal definida positiva, simetrica y constante que representa la ganancia del control QL.

El diagrama de bloques se muestra en la figura (4.1), el cual muestra al control QL+PD en lazo cerrado con el robot.

Ecuación en malla cerrada

La ecuación que describe el comportamiento en malla cerrada se obtiene al combinar las ecuaciones (4.1) y (4.2).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) = K_p\tilde{q} - K_d\dot{q} + \beta\Phi(\cdot),$$

despejando $\ddot{q}$ se tiene

$$\ddot{q} = M(q)^{-1}\left[K_p\tilde{q} - K_d\dot{q} + K_i\xi + \beta\Phi(\cdot) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\right],$$

la cual puede expresarse en términos del vector de estado $\left[\tilde{q}^T, \dot{q}^T\right]^T$ como:

$$\frac{d}{dt}\begin{bmatrix} \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(q)^{-1}\left[K_p\tilde{q} - K_d\dot{q} + \beta\Phi(\cdot) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\right] \end{bmatrix}, \tag{4.3}$$

Nótese que la ecuación anterior es autónoma y su único equilibrio es el origen $\left[\tilde{q}^T, \dot{q}^T\right]^T = 0 \in \mathbb{R}^{2n}$

Teorema 4.1 *Dada la dinámica (4.1) con la entrada de control (4.2), entonces el sistema en lazo cerrado (4.3) es estable en el punto de equilibrio*

$$x = [\tilde{q}^T, \dot{q}^T]^T = 0$$

con la condición de que $\beta > \beta\varepsilon + (G + F)$, donde β es una matriz constante definida positiva que representa la ganancia del controlador Q-Learning.

Demostración. *Construyamos una función de Lyapunov de la siguiente manera:*

$$V(\tilde{q},\dot{q}) = \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}\tilde{q}^T K_p\tilde{q}$$

su derivada temporal

$$\dot{V}(\tilde{q},\dot{q}) = \dot{q}^T M(q)\ddot{q} + \frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} - \tilde{q}^T K_p\dot{q}$$

$$\dot{V}(\tilde{q},\dot{q}) = \dot{q}^T M(q)M^{-1}(q)\left[K_p\tilde{q} - K_d\dot{q} + \beta\Phi(\cdot) - C(q,\dot{q})\dot{q} - G(q) - F(\dot{q})\right] + \frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} - \tilde{q}^T K_p\dot{q}$$

usando la propiedad de antisimetría

$$\frac{1}{2}\dot{q}^T[\dot{M}(q) - 2C(q,\dot{q})]\dot{q} = 0$$

$$\dot{V}(\tilde{q},\dot{q}) = -\dot{q}^T K_d \dot{q} + \dot{q}^T[\beta\Phi(\cdot) - G(q) - F(\dot{q})]$$

Si la acción $\Phi(\cdot)$ *usando la política de control* Q *se representa de la siguiente manera*

$$\Phi(\cdot) = sign(\dot{q}) \begin{cases} 1 & si \quad \dot{q} > 0 \\ 0 & si \quad \dot{q} = 0 \\ -1 & si \quad \dot{q} < 0 \end{cases}$$

entonces la política de control u_r *determinada por el algoritmo Q-Learning se expresa de la siguiente manera:*

$$u_r = \beta(-sign(\dot{q}) + \varepsilon)$$

donde ε *representa el error de la aproximación de la función* $\Phi(\cdot)$ *con la función* $sign(\dot{q})$.

$$\dot{V}(\tilde{q},\dot{q}) = -\dot{q}^T K_d \dot{q} + \dot{q}^T[\beta(-sign(\dot{q}) + \varepsilon) - G(q) - F(\dot{q})]$$

$$\dot{V}(\tilde{q},\dot{q}) = -\dot{q}^T K_d \dot{q} - \dot{q}^T \beta sign(\dot{q}) + \dot{q}^T \beta\varepsilon - \dot{q}^T G(q) - \dot{q}^T F(\dot{q})]$$

Si $G(q)$ *y* $F(\dot{q})$ *son desconocidos pero se conoce su cota como* $(G+F)$, *y además, usando* $\dot{q}^T sign(\dot{q}) = \|\dot{q}\|_1$, *con* $\|\dot{q}\|_1 = |q_1| + |q_2| + \cdots + |q_m| = \sum_{i=1}^{m} \|\dot{q}_i\|_1$, *donde* $\|\cdot\|_1$ *es la norma 1, el valor absoluto, tenemos:*

$$\dot{V}(\tilde{q},\dot{q}) \leq -K_d \parallel \dot{q} \parallel_2^2 - \beta \parallel \dot{q} \parallel_1 + \beta\varepsilon \parallel \dot{q} \parallel_1 + (G+F) \parallel \dot{q} \parallel_1$$

$$\dot{V}(\tilde{q},\dot{q}) \leq -K_d \parallel \dot{q} \parallel_2^2 - (\beta - \beta\varepsilon - G - F) \parallel \dot{q} \parallel_1$$

Si se cumple que $\beta > \beta\varepsilon + G + F$, *entonces el punto de equilibrio* $[\tilde{q}^T, \dot{q}^T]^T$ *es estable y las soluciones* $\tilde{q}(t)$ *y* $\dot{q}(t)$ *están acotadas.* ■

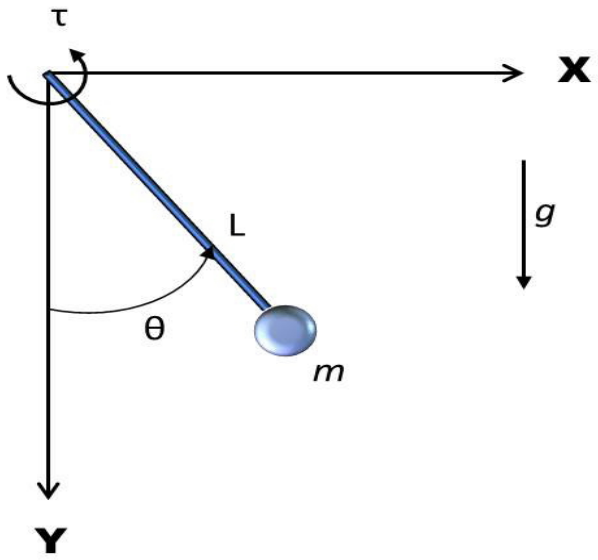


Figura 4.2: Péndulo invertido

4.3. Simulaciones

4.3.1. Péndulo

El péndulo es un sistema mecánico clásico para probar nuevas ideas en la disciplina del control inteligente [56]. Tiene la ventaja de ser, por un lado, un mecanismo relativamente sencillo, y por el otro, un sistema que contiene puntos inestables. El péndulo se ha usado ampliamente como patrón para comparar tanto algoritmos de control [65], como el hardware para implementarlos. Otra de las cualidades de este dispositivo es que su dinámica es no lineal, similar a la de un robot de un grado de libertad. Los algoritmos utilizados para controlar el péndulo pueden ser adaptados al control de otros mecanismos más complejos [70]. Hay varias formas de implementar estos procesos de aprendizaje. En esta tesis nos centraremos en lo que se conoce como Q-Learning, una forma de aprendizaje por reforzamiento en la que el agente aprende a asignar valores a los pares (estado, acción).

Descripción y planteamiento del problema.

El péndulo es un servo mecanismo que consta de una base sobre el cual el péndulo rota los 360° de libertad y en nuestro caso puede girar libremente sin restricciones. El péndulo iniciará en su condición de equilibrio estable y con velocidad cero y se desea que el péndulo alcance la vertical invertida. Las condiciones iniciales están dadas de la siguiente manera:

$$\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

La ecuación dinámica que describe el comportamiento del péndulo está representada de la siguiente manera:

$$ml^2\ddot{\theta} + b\dot{\theta} + mgl\sin(\theta) = \tau, \tag{4.4}$$

tomando como variables de estado $q = \theta$, y $\dot{q} = \dot{\theta}$

$$\ddot{q} = \frac{\tau}{ml^2} - \frac{b}{ml^2}\dot{q} - \frac{g}{l}\sin(q).$$

Diseño de la ley de control

El algoritmo Q-Learning requiere que el péndulo alcance la posición vertical superior, por lo que se propone como objetivo primordial diseñar un algoritmo de control que lleve al péndulo a la condición deseada $q_d = \pi$.

La posición del péndulo está limitada a $q \in [-\pi, \pi]\ rad$, la velocidad está restringida a $\dot{q} \in [-\pi, \pi]\ rad/s$.

Se genera una discretización del espacio de estados x_d basado en las posibles posiciones y velocidades en las que el péndulo podría estar. Las posiciones se definen como $z_1 = [-\pi, \pi]$ discretizado en pasos de 0,01 los cual produce 629 estados. Las velocidades se definen como $z_2 = [-\pi, \pi]$ discretizado en pasos de 0,1 lo cual produce 63 estados. La discretización del espacio de estados tendrá $629 \times 63 = 39627$ filas que representan todas las posibles combinaciones de posiciones y velocidades que podría tomar el péndulo. Además, se tendrán dos columnas una para definir posición y otra para definir velocidad. Finalmente, se tiene una discretización del espacio de estados $x_d \in \mathbb{R}^{39627\times 2}$.

La matriz Q tendrá una dimensión 39627 filas y 3 columnas. Se tienen 3 columnas asociadas a las 3 entradas de control $\{-1, 0, 1\}$, por lo tanto $Q \in \mathbb{R}^{39627\times 3}$.

En cada muestreo se genera un vector con las posiciones y velocidades del péndulo y con la misma dimensión del espacio de estados discreto x_d.

$$x_s = \begin{bmatrix} q \\ \dot{q} \end{bmatrix}^T \in \mathbb{R}^{39627\times 2}.$$

El error está definido como:

$$e_x = x_d - x_s,$$

y lo que se busca es la fila con el error más cercano a cero, min e_x.

Las acciones están definidas como el par aplicado al péndulo

$$u = \{-1, 0, 1\},$$

y el objetivo es encontrar una política u_r (ley de control) que maximice el retorno esperado.

$$u_r = \arg\max_u \left[Q_{t+1}(e_{x_t}, u_t)\right],$$

La matriz Q se inicializa en ceros y el algoritmo Q-Learning está definido de la siguiente forma:

$$Q_{t+1}(e_{x_t}, u_t) = Q_t(e_{x_t}, u_t) + \alpha \left[r_{t+1} + \gamma \max_{u'} Q_t(e_{x_{t+1}}, u') - Q_t(e_{x_t}, u_t)\right],$$

la recompensa estará definida de la forma:

$$r_{_{t+1}} = -\left|\tilde{q}\right|^2 - 0{,}25\left|\dot{q}\right|^2,$$

donde $\tilde{q} = q_d - q$, y la mayor recompensa se produce cuando el error $\tilde{q}$ y la velocidad $\dot{q}$ son cero, es decir, cuando el péndulo se encuentra sobre la vertical superior y su velocidad es cero. Además, en la función de recompensa, la velocidad se encuentra escalada por un factor de 0,25 con la finalidad de no castigar al algoritmo de control a cambios bruscos de velocidad.

La tarea se realizará en 1000 Episodios con 1500 iteraciones por episodio. Cada vez que un episodio termine, el péndulo regresará a su condición inicial (vertical inferior).

La Tabla [3] muestra los valores de los parámetros de control del algoritmo PD+QL

Tabla 3. Párametros del péndulo y el control PD+QL

Parámetros	Descripción	Valor
m	Masa del péndulo	1 kg
l	Longitud	0,5 m
b	Fricción	$0{,}01\frac{Nm}{ras/s}$
g	Gravedad	9,8 m/s^2
K_p	Ganancia proporcional	1
K_d	Ganancia derivativa	0,2
β	Ganancia del control QL	5
u_r	Acciones de entrada	$\{-1, 0, 1\}$ N

Las ganancias del controlador PD fueron ajustadas de tal manera que el péndulo se pudiera estabilizar lo mejor posible en su posición deseada y la ganancia del controlador Q-Learning $\beta = 5$ fue calculada cumpliendo $\beta > \beta\varepsilon + G + F$, donde $G = mgl\sin(\frac{\pi}{2}) = 4{,}9,\ F = 0{,}01$ y $\varepsilon = 0{,}01$ es el error de aproximación de la función signo.

Entonces, representamos el control híbrido de la siguiente manera:

$$\tau = K_p\tilde{q} - K_d\dot{q} + u_r,$$

donde

$$u_r = \beta\Phi(\cdot).$$

Resultados del aprendizaje

Para mostrar la efectividad del desempeño del controlador se realizaron las simulaciones bajo la plataforma Matlab. En la figura (4.3) se presentan las gráficas de salida para el ángulo de posición y velocidad. En la posición observamos una trayectoria suave que alcanza su condición deseada en 300 iteraciones. Para la velocidad observamos una trayectoria suave

y además un sobre impulso en la iteración 300 generado por el cambio de posición cuando el péndulo alcanzó la condición deseada. Finalmente, podemos ver que el control híbrido cumple de forma satisfactoria la tarea de regulación

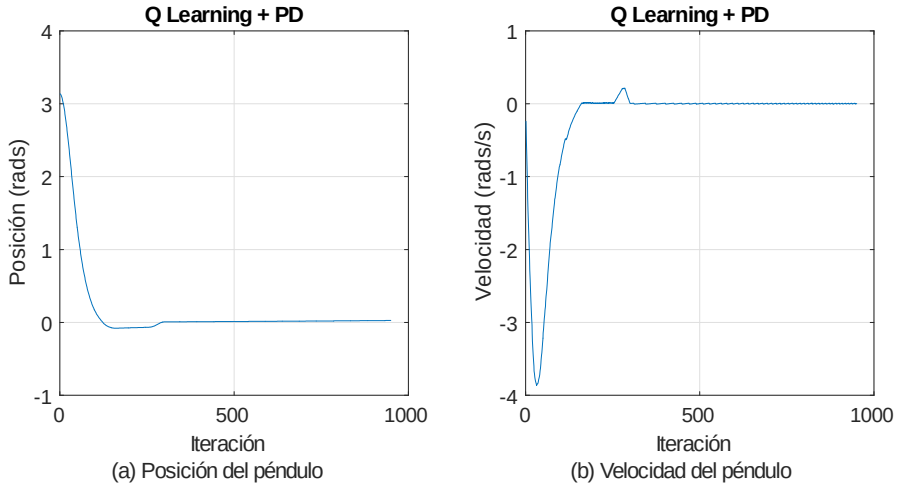


Figura 4.3: Gráficas de posición y velocidad aplicando el control PD+QL

4.3.2. Doble péndulo invertido

Descripción y planteamiento del problema

Se desea controlar un doble péndulo invertido visto en la figura (3.3). Como primer punto se controlará al doble péndulo invertido sólo usando el algoritmo Q-Learning. Como segundo punto se realizará un incremento en los parámetros de las masas y longitudes del doble péndulo invertido con la finalidad de introducir perturbaciones que afecten al algoritmo QL previamente aprendido. Como tercer punto, se utilizará la matriz Q previamente aprendida conjuntamente con el control PD para compensar los cambios introducidos en los parámetros del modelo. Al utilizar el control PD+QL, el algoritmo QL ya no realizará el aprendizaje, en vez de eso, utilizará el conocimiento aprendido y trabajará conjuntamente con el control PD para compensar los incrementos en los parámetros del modelo.

Planteamiento del problema: Se desea que el segundo péndulo se mantenga en la posición vertical invertida sin restricción del primer pendulo, mientras el carro mantiene una restricción impuesta de encontrarse en un rango de ± 1 metro desde su punto de origen referencial ($x_{c\_inicial} = 0$). Las condiciones iniciales son que el primer péndulo (péndulo inferior) empiece en su equilibrio inestable (origen) y con velocidad cero, mientras que el segundo péndulo (péndulo superior) podrá tener una condición inicial de $\pm 5°$, donde θ_1 y $\dot{\theta}_1$ representan la posición y velocidad del primer péndulo, θ_2 y $\dot{\theta}_2$, representa la posición y velocidad del segundo péndulo y finalmente x_c y $\dot{x}_c$ son la posición y velocidad del carro.

$$\begin{bmatrix} \theta_1 \\ \theta_2 \\ x_c \end{bmatrix} = \begin{bmatrix} 0{,}087\ rad \\ 0\ rad \\ 0\ m \end{bmatrix}; \quad \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{x}_c \end{bmatrix} = \begin{bmatrix} 0\ rad/s \\ 0\ rad/s \\ 0\ m/s \end{bmatrix}$$

Diseño de la ley de control

Las posiciones están limitadas a $\theta_1 \in [-\pi/4, \pi/4]\ rad, \theta_2 \in [-\pi/4, \pi/4]\ rad$ y $x_c \in [-1, 1]$ m, las velocidades están restringidas a $\dot{\theta}_1 \in [-\pi, \pi]\ rad/s, \dot{\theta}_2 \in [-\pi, \pi]\ rad/s$ y $\dot{x}_c \in [-0{,}5, 0{,}5]$ m/s y la fuerza aplicada al carro está representada por $f \in \{-100, 0, 100\}$ N.

Se genera una discretización del espacio de estados basado en las posibles posiciones y velocidades en las que el péndulo superior podría estar. Las posiciones se definen como $z_1 = [-\pi/4, \pi/4]$ discretizado en pasos de 0,01 los cual produce 158 estados. Las velocidades se definen como $z_2 = [-\pi, \pi]$ discretizado en pasos de 0,01 lo cual produce 629 estados. La discretización del espacio de estados tendrá $158 \times 629 = 99382$ filas que representan todas las posibles combinaciones de posiciones y velocidades que podría tomar el péndulo superior. Además, se tendrán dos columnas una para definir posiciones y otra para definir las velocidades. Finalmente, se tiene una discretización del espacio de estados como $x_d \in \mathbb{R}^{99382\times 2}$.

La matriz Q tendrá una dimensión 99382 filas y 3 columnas. Se tienen 3 columnas asociadas a las 3 entradas de control $f \in \{-100, 0, 100\}$, por lo tanto $Q \in \mathbb{R}^{99382\times 3}$.

La posición que se va a controlar es la sombra que proyecta el péndulo superior sobre el piso definido de la siguiente manera:

$$x_{sombra} = x_c + l_2 \sin(\theta_1 + \theta_2),$$

y su velocidad está definida como:

$$\dot{x}_{sombra} = \dot{x}_c + l_2(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1 + \theta_2).$$

En cada muestreo se genera un vector con las posiciones y velocidades que proyecta la sombra del péndulo superior y tendrá la misma dimensión del espacio de estados discreto.

$$x_s = \begin{bmatrix} x_{sombra} \\ \dot{x}_{sombra} \end{bmatrix}^T \in \mathbb{R}^{99382\times 2}.$$

El error del espacio discretizado con respecto a la posición del péndulo superior está definido como:

$$e_x = x_d - x_s,$$

y lo que se busca es la fila con el error más cercano a cero, min e_x.

Las acciones están definidas como la fuerza aplicada al péndulo invertido

$$u = \{-100, 0, 100\},$$

y nuestro objetivo es encontrar una política f (ley de control) que maximice el retorno esperado.

$$f = \arg\max_u \left[Q_{t+1}(e_{x_t}, u_t)\right].$$

La matriz Q se inicializa en ceros y el algoritmo Q-Learning está definido de la siguiente forma:

$$Q_{t+1}(e_{x_t}, u_t) = Q_t(e_{x_t}, u_t) + \alpha\left[r_{t+1} + \gamma \max_{u'} Q_t(e_{x_{t+1}}, u') - Q_t(e_{x_t}, u_t)\right],$$

la recompensa estará definida de la forma:

$$r_{1_{k+1}} = -\left|x_c + \sin\left(\theta_1{+}\theta_2\right)\right|^2 - 0{,}25\left|\dot{x}_c{+}(\dot{\theta}_1{+}\dot{\theta}_2)\cos\left(\theta_1{+}\theta_2\right)\right|^2,$$

donde la mayor recompensa se produce cuando la posición θ_2, x_c y la velocidad $\dot{\theta}_2, \dot{x}_c$ son cero, es decir, cuando el carro se está en el origen y péndulo superior se encuentra sobre la vertical invertida y su velocidad es cero. Además, en la función de recompensa, la velocidad se encuentra escalada por un factor de 0,25 con la finalidad de no castigar al algoritmo de control a cambios bruscos de velocidad.

Resultados del aprendizaje

Para mostrar la efectividad del desempeño del controlador se realizarán las simulaciones bajo la plataforma Matlab. Los parámetros utilizados tanto en el doble péndulo invertido como en el aprendizaje por reforzamiento se muestran en la Tabla [4].

Tabla 4. Parámetros del doble péndulo y del controlador QL

Parámetros	Descripción	Valor
m_1	Masa del péndulo 1	0,3 kg
m_2	Masa del péndulo 2	0,2kg
M	Masa del carro	0,8kg
l_1	Longitud del péndulo 1	0,5 m
l_2	Longitud del péndulo 2	0,4m
g	Gravedad	9,8 m/s^2
α	Tasa de aprendizaje	0,99
γ	Factor de descuento	0,9
f	Acciones de entrada	$\{-100, 0, 100\}N$

La Tabla [5] muestra los parámetros tanto de la planta como del controlador al agregar el control PD a nuestra matriz Q previamente calculada con los parámetros iniciales de la

Tabla [4].

Tabla 5. Parámetros del péndulo y el controlador QL+PD

Parámetros	Descripción	Valor
m_1	Masa del péndulo 1	0,5 kg
m_2	Masa del péndulo 2	0,5kg
M	Masa del carro	1,2kg
l_1	Longitud del péndulo 1	0,5 m
l_2	Longitud del péndulo 2	0,5m
g	Gravedad	9,8 m/s^2
α	Tasa de aprendizaje	0,99
γ	Factor de descuento	0,9
K_p	Ganancia Proporcional	10
K_d	Ganancia Derivativa	155
β	Ganancia control QL	100
Γ	Conjunto de acciones	$\{-1, 0, 1\}N$

El control híbrido final está definido de la siguiente manera:

$$f = K_p\tilde{q} - K_d\dot{q} + \beta u_r, \tag{4.5}$$

donde $\tilde{q} = q_d - x_{sombra}, q_d = 0$, y la velocidad está representada como $\dot{q} = \dot{x}_{sombra}$. Las ganancias del controlador PD fueron ajustadas de tal manera que no causara una inestabilidad en el péndulo superior, y así que el control PD sólo proporcionará la energía suficiente para reducir el error de posición y el del velocidad evitando así la deriva de los péndulos y del carro mismo. Finalmente, en el control Q-Learning sólo una ganancia β fue incrementada a la entrada, βu_r, mientras se mantiene la misma matriz $Q_t(e_{x_t}, u_t)$.

Las Figuras (4.4) y (4.5), muestran el desempeño del algoritmo propuesto. Para el cálculo de las ganancias del controlador PID se realizaron modificaciones sucesivas en los parámetros de control hasta conseguir los valores más óptimos en función de la comparación entre los controladores QL y QL+PD, resultando en: $kp = 9, kd = 160, ki = 6$, donde se pudo observar, que el mejor desempeño fue del controlador QL+PD, demostrando que mantiene

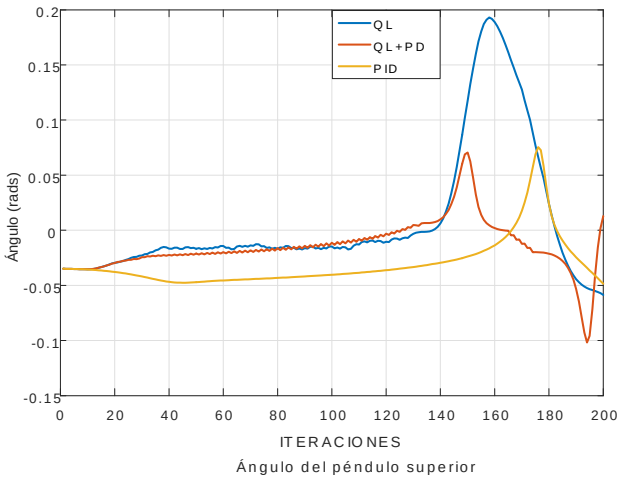


Figura 4.4: Posición del péndulo superior

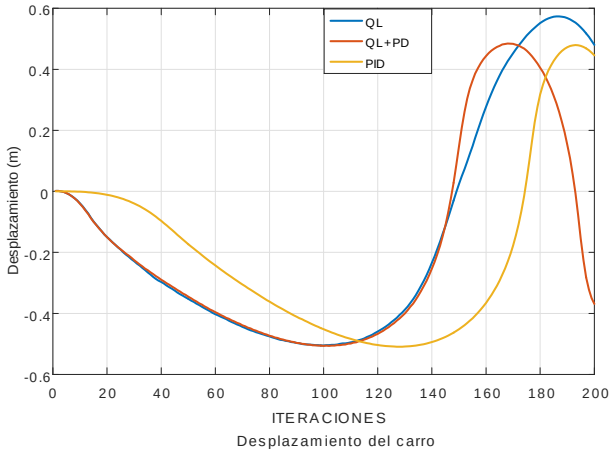


Figura 4.5: Desplazamiento del carro

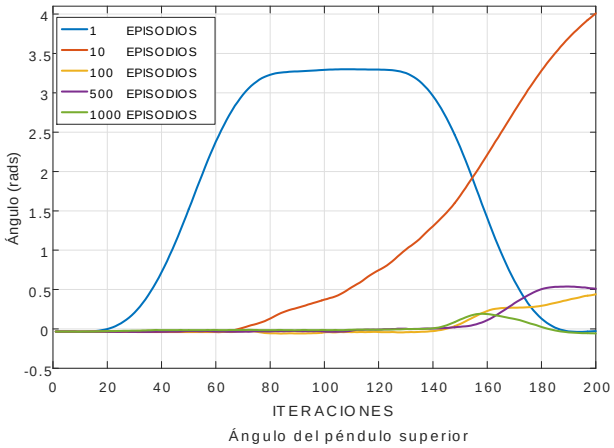


Figura 4.6: Posición del péndulo superior en los episodios 1, 10, 100, 500 y 1000

el péndulo superior sobre la vertical con un ángulo de desplazamiento mucho más pequeño en comparación con los otros controladores. La Figura (4.5) muestra como la restricción del desplazamiento del carro se mantiene en el rango de ± 1 a partir de su punto de origen referencial, arrojando la respuesta esperada. La Figura (4.6) muestra la evolución del aprendizaje del ángulo de salida del péndulo superior θ_2, para el episodio 1, el episodio 10, el episodio 100, el episodio 500 y finalmente el episodio 1000, esto con la intención de revelar como va mejorando el aprendizaje a lo largo del tiempo.

Índices de desempeño: Con la finalidad de cuantificar el comportamiento de los controladores, utilizamos los criterios integrales conocidos como Integral del error absoluto *(IAE)* e Integral del tiempo por el error absoluto *(ITAE)*:

$$IAE = \int_0^\infty \mid e(t) \mid dt \tag{4.6}$$

$$ITAE = \int_0^\infty t \mid e(t) \mid dt \tag{4.7}$$

donde el error está dado por $e(t) = \theta_d - \theta_2$, donde $\theta_d = 0$.

Tabla 6. Índices de desempeño

	IAE	ITAE
QL	7,688	0,0231
PID	7,136	0,0214
QL+PD	4,241	0,0127

Se realizó la comparación entre el aprendizaje por reforzamiento QL, el control QL+PD y el control PID, donde la Tabla [6] muestra que los valores más pequeños en los índices de desempeño son para el controlador QL+PD, lo cual nos dice que el ángulo de salida del péndulo superior se mantiene sobre la referencia (vertical invertida) por más tiempo en comparación con los otros controladores, además en este controlador QL+PD propuesto se aprecian ventajas como por ejemplo: la respuesta híbrida [68] trabaja mucho mejor que un controlador PID, así mismo, la sintonización del control PD resulta mucho más simple ya que el aprendizaje por reforzamiento absorbe toda la dinámica del sistema, dejando que el control PD sólo se encargue de mantener el error del ángulo del péndulo superior en cero. La desventaja que se presenta es que el tiempo que tarda en aprender el algoritmo de aprendizaje por reforzamiento es de aproximadamente 1hr con 30 minutos lo que representa 1000 episodios de 200 Iteraciones por episodio.

Conclusión

Se observa que controlar una planta tal como el doble péndulo invertido sobre el carro resulta una tarea más complicada para el algoritmo Q-Learning por que el diseño de la recompensa tiene que ser más ingenioso y deber estar en función de la proyección que el péndulo superior tiene sobre el eje de las abscisas. Además, la forma de discretizar el espacio de trabajo no sólo está en función de los dos ángulos, sino también en función de la velocidad del carro. Finalmente, las graficas muestran el desempeño del ángulo θ_2 y del desplazamiento del carro x, para el contolador QL, el control híbrido QL+PD, y para el control PID, donde se aprecia que el algoritmo de aprendizaje por reforzamiento más el control PD trabajan muy bien de manera cooperativa dando mejores resultados en su forma híbrida, que de manera

individual.

Capítulo 5

Control PID usando como Compensación el Aprendizaje por Reforzamiento.

5.1. Introducción

Hoy en día, la idea del control de robots manipuladores ha atraído la atención de la comunidad científica de robótica. Un punto de interés se ubica en el diseño de sistemas de control para robots manipuladores en aplicaciones industriales tales como el estibar cargamento, ensamblaje, traslado, pintado de objetos, etc. Puesto que los robots industriales son capaces de realizar correctamente una variedad, a simple vista parecería innecesario desarrollar investigación sobre el tema de control de robots manipuladores. Sin embargo, es importante resaltar que la ejecución de la tarea encomendada al robot requiere alto desempeño y exactitud en sus movimientos.

El diseño de nuevos esquemas de control requiere de grandes retos teóricos que mejoran sustancialmente problemas de origen práctico. Además, su estudio resulta indispensable en aplicaciones que no pueden ser llevadas a cabo por medio de algoritmos de control tradicionales. De ahí que resulta muy importante y de gran interés el diseño de nuevas técnicas

de control que solventen las necesidades de tener un control con mayor robustez. Este capítulo está destinado a presentar un algoritmo de control basado en la técnica del aprendizaje por reforzamiento, y compararlo contra una familia de controladores clásicos basados en la metodología de modelo de energía.

El control de posición o regulación de control de robots manipuladores es un caso particular de control de movimiento en el cual no hay una referencia variante en el tiempo que el robot haga seguimiento como en el caso de control de trayectoria, más bien, es un punto constante en el tiempo al que se le denomina posición deseada o set point. El objetivo de control es posicionar el extremo final del robot en ese punto y que permanezca ahí de manera indefinida. Por supuesto, para propósitos prácticos, una vez que el extremo final del robot alcanza el punto deseado, deberá pasar uno o más periodos de muestreo para cambiar de valor el punto deseado, entonces, el actual punto deseado, tomará el papel de condición inicial y el extremo final del robot se moverá al nuevo punto deseado, y así sucesivamente. Este concepto da la posibilidad de interpolar punto a punto para que el robot pueda seguir la trayectoria a través de un esquema de control de posición con puntos cercanos entre sí. En control automático se le conoce como control punto a punto.

5.2. Control PID en caso de Regulación

El problema de control de posición o regulación consiste en mover el extremo final del robot manipulador desde cualquier posición inicial hacia una posición deseada. Esto significa que la *i- ésima* articulación del robot deberá moverse hacia la respectiva *i-ésima* posición deseada. Un ejemplo ilustrativo se muestra en la figura (5.1) donde el robot parte de la posición de reposo (posición de casa) para llegar a la configuración deseada permaneciendo indefinidamente en el punto de equilibrio. Formalmente, el objetivo de control de posición está determinado por encontrar una ley de control τ que proporciona los pares aplicados a las articulaciones o servomotores del robot, de tal forma que la posición actual del robot $q(t)$ y la velocidad articular de movimiento $\dot{q}(t)$ tiende asintóticamente hacia la posición deseada q_d y velocidad cero, respectivamente, sin importar las condiciones iniciales. Es decir,

$$\lim_{t\to\infty} \begin{bmatrix} q(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} q_d \\ 0 \end{bmatrix}. \tag{5.1}$$

Nótese que en la figura (5.1) el robot se encuentra en su punto de equilibrio estable, lo que significa que el objetivo de control (5.1) se cumplió sin depender de las condiciones iniciales, entonces la posición deseada se alcanza, por lo que la posición del extremo final del robot permanece constante $q(t) = q_d$ y por lo tanto la velocidad de movimiento es cero ($\dot{q} = 0$).

Finalmente, poder decir que un algoritmo de control de posición o regulación es una formulación cuya principal característica es generar un atractor en la ecuación de lazo cerrado formada por el modelo dinámico del robot manipulador y la estructura matemática del algoritmo de control. Lo anterior significa que el punto de equilibrio sea asintóticamente estable. La importancia de esta problemática radica en proponer estrategias de control que no sólo cumpla con el objetivo de control $\lim_{\to\infty} \begin{bmatrix} \tilde{q}^T(t) & \dot{q}^T(t) \end{bmatrix}^T = \begin{bmatrix} 0^T & 0^T \end{bmatrix}^T$, sino también que el desempeño práctico sea alto.

El *desempeño* de un algoritmo de control se refiere a realizar de manera correcta y exacta la tarea programada al robot, lo que lo habilita a llevar a cabo diversas aplicaciones de control punto a punto. Por lo tanto, el espectro de aplicaciones comerciales, domesticas, científicas e industriales se incrementa.

5.3. Control PID con compensación QL

Considérese el modelo dinámico que describe el comportamiento de un robot manipulador de n grados de libertad

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) = \tau, \tag{5.2}$$

en términos del vector de estado $\begin{bmatrix} q^T, & \dot{q}^T \end{bmatrix}^T$

$$\frac{d}{dt} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M(q)^{-1}\left[\tau - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\right] \end{bmatrix},$$

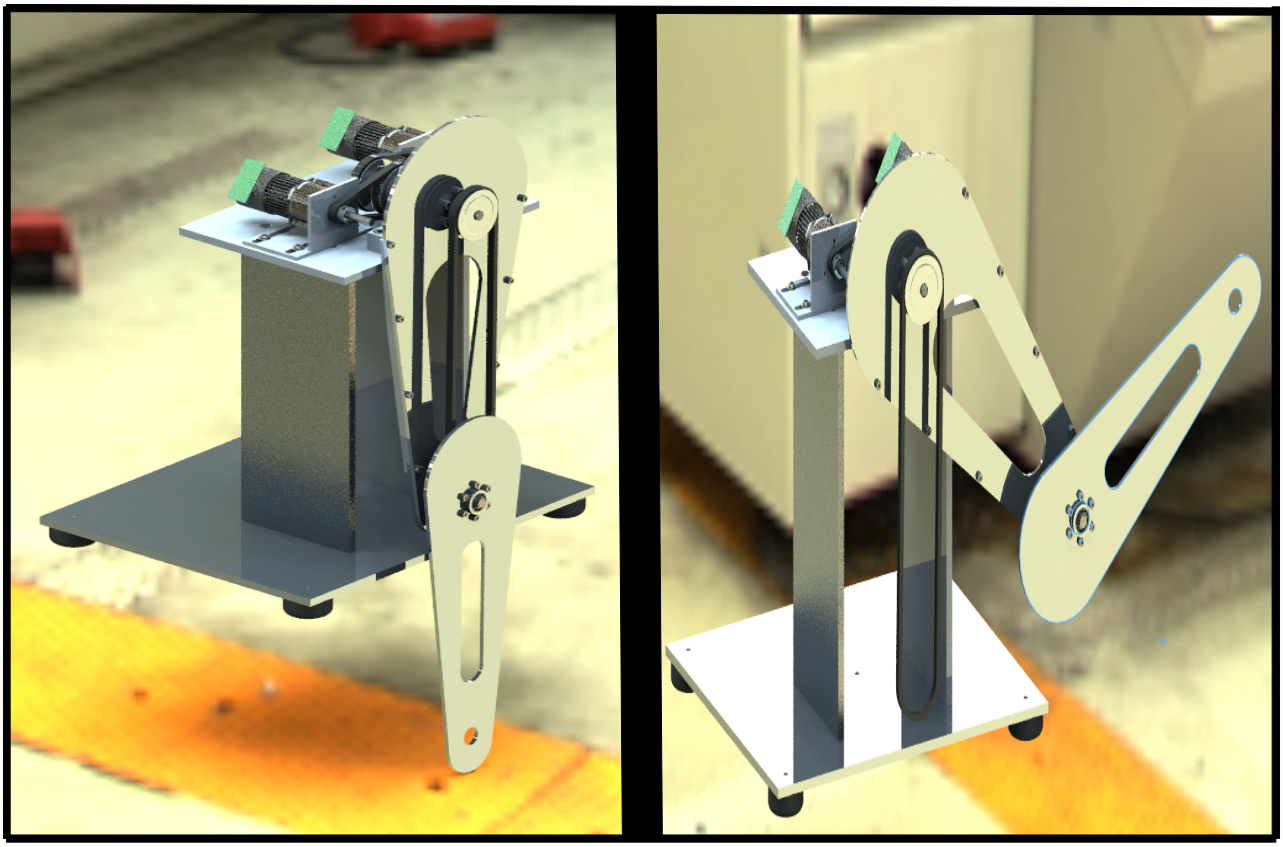


Figura 5.1: En la figura de la izquierda vemos la condición inicial en la posición de equilibrio estable (posición de casa). En la figura de la derecha vemos la condición final deseada.

donde $q\ \dot{q}\ \ddot{q} \in \mathbb{R}^n$ denotan la posición, velocidad y aceleración articular, respectivamente, $\tau \in \mathbb{R}^n$ es un vector de fuerzas y pares aplicados en las uniones mediante los actuadores, $M(q) \in \mathbb{R}^{n\times n}$ es la matriz de inercia, $C(q,\dot{q})\dot{q} \in \mathbb{R}^n$ es el vector de fuerzas centrifugas y de coriolis, $G(q) \in \mathbb{R}^n$ es el vector de pares gravitacionales, y $F(\dot{q})$ representa la función de coulomb, y se representa de la forma:

$$F(\dot{q}) = B_{f1}\dot{q} + B_{f2}\text{sign}(\dot{q}),$$

donde B_{f1} y B_{f2} son matrices positivas $\in \mathbb{R}^{n\times n}$, por simplicidad usaremos el modelo siguiente:

$$F(\dot{q}) = B_{f1}\dot{q}.$$

El objetivo de control se puede definir formalmente de la siguiente manera: dada la posición angular deseada $q_d \in \mathbb{R}^n$ constante para todo $t \geq 0$, el problema es diseñar una ley de control τ tal que la posición angular del robot $q(t)$ se aproxima a $q_d \in \mathbb{R}^n$ asintóticamente, es decir:

$$\lim_{t\to\infty} \|\tilde{q}\| = 0.$$

El vector $q_d \in \mathbb{R}^n$ es la posición articular deseada , y el vector $\tilde{q} = q_d - q \in \mathbb{R}^n$ es el vector de error de posición.

Ley de Control

La ley de control PID+QL puede expresarse de la siguiente manera:

$$\tau = K_p\tilde{q} + K_d\dot{\tilde{q}} + K_i \int_0^t \tilde{q}(\psi)d\psi + u_r,$$

donde las matrices de diseño $K_p, K_d, K_i \in \mathbb{R}^{n\times n}$ llamadas respectivamente las ganancias proporcional, derivativa e integral, son matrices simétricas y definidas positivas convencionalmente elegidas. $u_r \in \mathbb{R}^n$ es el algoritmo de control llamado Q-Learning, que tiene la forma:

$$u_r = \beta(-\text{sign}(\dot{q}) + \varepsilon),$$

donde β es una matriz diagonal y definida positiva seleccionada por el diseñador, y ε representa el error de aproximación de la función $\Phi(\dot{q})$ con la función $sign(\dot{q})$, donde la función signo se representa de la siguiente manera:

$$\Phi(\dot{q}) = \text{sign}(\dot{q}) = \begin{cases} 1 \text{ si } \dot{q} > 0 \\ 0 \text{ si } \dot{q} = 0 \\ -1 \text{ si } \dot{q} < 0 \end{cases}$$

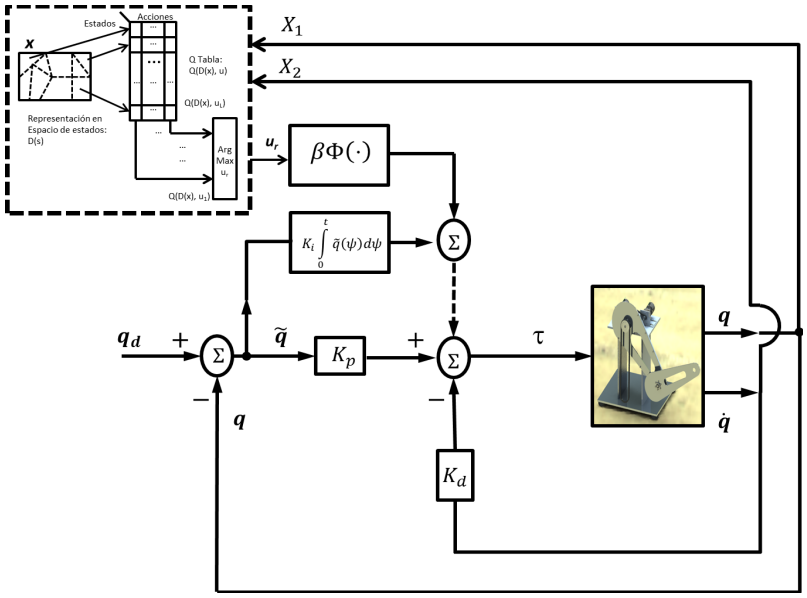


Figura 5.2: Control PID+QL

El vector $\text{sign}(\dot{q})$ está definido por $\text{sign}(\dot{q}) = [\text{sign}(\dot{q}_1), \ldots, \text{sign}(\dot{q}_n)]^T$.

La acción integral del controlador PID+QL introduce una nueva variable de estado adicional que será denotada por ξ y cuya derivada temporal es $\dot{\xi} = K_i\tilde{q}$. Además, en el caso de regulación $\dot{q}_d = 0$, $\tilde{q} = -\dot{q}$, entonces, la ley de control PID+QL en el caso de regulación puede expresarse por medio de las ecuaciones siguientes:

$$\begin{aligned} \tau &= K_p\tilde{q} - K_d\dot{q} + \xi + \beta(-\text{sign}(\dot{q}) + \varepsilon) \\ \dot{\xi} &= K_i\tilde{q} \end{aligned} \tag{5.3}$$

Ecuación en malla cerrada

La ecuación que describe el comportamiento en malla cerrada se obtiene al combinar las ecuaciones (5.2) y (5.3).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(\dot{q}) + G(q) = K_p\tilde{q} - K_d\dot{q} + \xi + \beta(-\text{sign}(\dot{q}) + \varepsilon),$$

despejando $\ddot{q}$ se tiene

$$\begin{array}{l} \ddot{q} = M(q)^{-1}\left[K_p\tilde{q} - K_d\dot{q} + K_i\xi + \beta(-\text{sign}(\dot{q}) + \varepsilon) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\right] \\ \dot{\xi} = K_i\tilde{q} \end{array},$$

la cual puede expresarse en términos del vector de estado $\left[\xi^T, \tilde{q}^T, \dot{q}^T\right]^T$ como:

$$\frac{d}{dt}\begin{bmatrix} \xi \\ \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} K_i\tilde{q} \\ -\dot{q} \\ M(q)^{-1}\left[K_p\tilde{q} - K_d\dot{q} + \xi + \beta(-\text{sign}(\dot{q}) + \varepsilon) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\right] \end{bmatrix},$$

los equilibrios tienen la forma

$$\begin{bmatrix} \xi \\ \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} G(q_d) \\ 0 \\ 0 \end{bmatrix}.$$

El equilibrio anterior puede trasladarse al origen mediante el siguiente cambio de variable $\tilde{\xi} = \xi - G(q_d)$, y la malla cerrada podrá expresarse en términos del vector de estado $\left[\tilde{\xi}^T, \tilde{q}^T, \dot{q}^T\right]^T$ como:

$$\frac{d}{dt}\begin{bmatrix} \tilde{\xi} \\ \tilde{q} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} K_i\tilde{q} \\ -\dot{q} \\ M(q)^{-1}\left[K_p\tilde{q} - K_d\dot{q} + \tilde{\xi} - G(q_d) + \beta(-\text{sign}(\dot{q}) + \varepsilon) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q)\right] \end{bmatrix}. \tag{5.4}$$

Nótese que la ecuación anterior es autónoma y su único equilibrio es el origen $\left[\tilde{\xi}^T, \tilde{q}^T, \dot{q}^T\right]^T = 0 \in \mathbb{R}^{3n}$.

Prueba de Estabilidad

Para estudiar la estabilidad del origen del espacio de estados, se propone la siguiente función candidata de Lyapunov:

$$\begin{aligned} V(\tilde{\xi},\tilde{q},\dot{q}) &= \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2}\tilde{q}^T K_p \tilde{q} + U(q_d - q) - k_u + \\ &+\tilde{q}^T G(q_d) + \tilde{q}^T\tilde{\xi} + \frac{3}{2}G(q_d)^T K_p^{-1} G(q_d) + \frac{\alpha}{2}\tilde{\xi}^T K_i^{-1}\tilde{\xi} + \\ &-\alpha\dot{q}^T M(q)\tilde{q} + \frac{\alpha}{2}\tilde{q}^T \left[\ K_d + B_{f1}\ \right]\tilde{q} + \alpha K_i^{-1}\int_0^t \Phi(\dot{q})d\xi \end{aligned}$$

donde $U(q_d - q)$ denota, como ya es costumbre, la energía potencial del robot, $k_u = \text{mín}_q \left\{U(q_d - q)\right\}$, que se agrega de modo que $V(0) = 0$, y α es una constante positiva que satisface condiciones bien definidas para que la función candidata de Lyapunov sea definida positiva.

Se probará que la función candidata de Lyapunov es definida positiva, $V(\tilde{\xi},\tilde{q},\dot{q}) \geq 0$.

El término $\frac{1}{2}\tilde{q}^T K_p \tilde{q}$ los dividiremos en tres partes, y la función candidata de Lyapunov la dividiremos en cuatro partes $V(\tilde{\xi},\tilde{q},\dot{q}) = \sum_{i=1}^{4} V(\tilde{\xi},\tilde{q},\dot{q})_i$.

$$\begin{aligned} V(\tilde{\xi},\tilde{q},\dot{q})_1 &= \tfrac{1}{6}\tilde{q}^T K_p\tilde{q} + \tilde{q}^T G(q_d) + \tfrac{3}{2}G(q_d)^T K_p^{-1} G(q_d) \\ V(\tilde{\xi},\tilde{q},\dot{q})_2 &= \tfrac{1}{6}\tilde{q}^T K_p\tilde{q} + \tilde{q}^T\tilde{\xi} + \tfrac{\alpha}{2}\tilde{\xi}^T K_i^{-1}\tilde{\xi} \\ V(\tilde{\xi},\tilde{q},\dot{q})_3 &= \tfrac{1}{6}\tilde{q}^T K_p\tilde{q} - \alpha\dot{q}^T M(q)\tilde{q} + \tfrac{1}{2}\dot{q}^T M(q)\dot{q} \\ V(\tilde{\xi},\tilde{q},\dot{q})_4 &= U(q_d - q) - k_u + \tfrac{\alpha}{2}\tilde{q}^T \left[\ K_d + B_{f1}\ \right]\tilde{q} + \alpha K_i^{-1}\textstyle\int_0^t \Phi(\dot{q})d\tilde{\xi} \end{aligned}$$

Del primer término $V(\tilde{\xi},\tilde{q},\dot{q})_1$ expresado en forma matricial podemos facílmente ver que si $K_p > 0$, entonces $V(\tilde{\xi},\tilde{q},\dot{q})_1$ es semidefinida positiva.

$$V(\tilde{\xi},\tilde{q},\dot{q})_1 = \frac{1}{2}\begin{bmatrix} \tilde{q} \\ G(q_d) \end{bmatrix}\begin{bmatrix} \frac{1}{3}K_p & I \\ I & 3K_p^{-1} \end{bmatrix}\begin{bmatrix} \tilde{q} \\ G(q_d) \end{bmatrix} \geq 0.$$

Del segundo término $V(\tilde{\xi},\tilde{q},\dot{q})_2$ expresado en forma matricial obtenemos la primera condición de α para que la función sea definida positiva.

$$V(\tilde{\xi},\tilde{q},\dot{q})_2 = \frac{1}{2}\begin{bmatrix}\tilde{q}\\ \tilde{\xi}\end{bmatrix}\begin{bmatrix}\frac{1}{3}K_p & I\\ I & \alpha K_i^{-1}\end{bmatrix}\begin{bmatrix}\tilde{q}\\ \tilde{\xi}\end{bmatrix},$$

usando el criterio de Sylvester, para probar que la matriz es definida positiva, el determinante debe ser positivo, por lo cual tenemos:

$$V(\tilde{\xi},\tilde{q},\dot{q})_2 = \frac{1}{2}\begin{bmatrix}\tilde{q}\\ \tilde{\xi}\end{bmatrix}\begin{bmatrix}\frac{1}{3}\lambda_{\text{mín}}(K_p) & 1\\ 1 & \alpha\lambda_{\text{mín}}(K_i^{-1})\end{bmatrix}\begin{bmatrix}\tilde{q}\\ \tilde{\xi}\end{bmatrix}$$

$$\begin{aligned}&\det = \tfrac{1}{3}\lambda_{\text{mín}}(K_p)\alpha\lambda_{\text{mín}}(K_i^{-1}) - 1 \geq 0\\ &\tfrac{\alpha}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(K_i^{-1}) \geq 1 \qquad\qquad ,\\ &\alpha \geq \tfrac{3}{\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(K_i^{-1})}\end{aligned}$$

por lo tanto si hacemos que α cumpla con la restricción anterior, entonces $V(\tilde{\xi},\tilde{q},\dot{q})_2$ es definida positiva.

Del tercer término $V(\tilde{\xi},\tilde{q},\dot{q})_3$ expresado en forma matricial obtenemos la segunda condición de α para que la función sea definida positiva.

$$V(\tilde{\xi},\tilde{q},\dot{q})_3 = \frac{1}{2}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix}\begin{bmatrix}\frac{1}{3}K_p & -\alpha M(q)\\ -\alpha M(q) & M(q)\end{bmatrix}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix},$$

usando nuevamente el criterio de Sylvester, para probar que la matriz es definida positiva, el determinante debe ser positivo, por lo cual tenemos:,

$$V(\tilde{\xi},\tilde{q},\dot{q})_3 = \frac{1}{2}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix}\begin{bmatrix}\frac{1}{3}\lambda_{\text{mín}}(K_p) & -\alpha\lambda_{\text{máx}}(M)\\ -\alpha\lambda_{\text{máx}}(M) & \lambda_{\text{mín}}(M)\end{bmatrix}\begin{bmatrix}\tilde{q}\\ \dot{q}\end{bmatrix}$$

$$\det = \tfrac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M) - \alpha^2\lambda_{\text{máx}}(M)^2 \geq 0$$

$$\tfrac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M) \geq \alpha^2\lambda_{\text{máx}}(M)^2$$

$$\frac{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}{\lambda_{\text{máx}}(M)^2} \geq \alpha^2$$

$$\frac{\sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}}{\lambda_{\text{máx}}(M)} \geq \alpha$$

,

por lo tanto si hacemos que α cumpla con la restricción anterior, entonces $V(\tilde{\xi}, \tilde{q}, \dot{q})_3$ es definida positiva.

Finalmente, es fácil ver que $V(\tilde{\xi}, \tilde{q}, \dot{q})_4 \geq 0$.

Si hacemos cumplir a α las restricciones anteriores, tenemos:

$$\frac{\sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}}{\lambda_{\text{máx}}(M)} \geq \alpha \geq \frac{3}{\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(K_i^{-1})}, \tag{5.5}$$

donde vemos que si K_p es suficientemente grande o K_i suficientemente pequeño, entonces $V(\tilde{\xi}, \tilde{q}, \dot{q})$ es definida positiva semiglobalmente.

La derivada temporal de $V(\tilde{\xi}, \tilde{q}, \dot{q})$ a lo largo de las trayectorias del sistema en lazo cerrado y usando $\frac{d}{dt}\int_0^t \Phi(\dot{q})d\tilde{\xi} = \frac{\partial \int_0^t \Phi(\dot{q})d\tilde{\xi}}{\partial \tilde{\xi}}\frac{\partial \tilde{\xi}}{\partial t} = \dot{\tilde{\xi}}^T\Phi(\dot{q})$, tenemos:

$$\begin{aligned}\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) &= \dot{q}^T M(q)\ddot{q} + \frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} + \tilde{q}^T K_p\dot{\tilde{q}} + \dot{q}G(q) + \dot{\tilde{q}}^T G(q_d) + \\ &\quad + \dot{\tilde{q}}^T\tilde{\xi} + \tilde{q}^T\dot{\tilde{\xi}} + \alpha\tilde{\xi}^T K_i^{-1}\dot{\tilde{\xi}} - \alpha\dot{\tilde{q}}^T M(q)\dot{q} - \alpha\tilde{q}^T\dot{M}(q)\dot{q} + \\ &\quad - \alpha\tilde{q}^T M(q)\ddot{q} + \alpha\tilde{q}^T\left[\ K_d + B_{f1}\ \right]\dot{\tilde{q}} + \alpha\dot{\tilde{\xi}}^T K_i^{-1}\Phi(\dot{q})\end{aligned}$$

sustituyendo $\dot{\tilde{\xi}}$, $\dot{\tilde{q}}$, y $\ddot{q}$ en la ecuación anterior resulta:

$$\begin{aligned}\dot{V}(\tilde{\xi},\tilde{q},\dot{q}) &= \dot{q}^T M(q)M(q)^{-1}\left[\begin{array}{c} K_p\tilde{q} - K_d\dot{q} + \tilde{\xi} - G(q_d) + \\ +\beta(-sign(\dot{q}) + \varepsilon) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q) \end{array}\right] \\ &+\frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} + \tilde{q}^T K_p\left[-\dot{q}\right] + \dot{q}G(q) + \left[-\dot{q}\right]^T G(q_d) + \\ &+\left[-\dot{q}\right]^T\tilde{\xi} + \tilde{q}^T\left[K_i\tilde{q}\right] + \alpha\tilde{\xi}^T K_i^{-1}\left[K_i\tilde{q}\right] - \alpha\left[-\dot{q}\right]^T M(q)\dot{q} - \alpha\tilde{q}^T\dot{M}(q)\dot{q} + \\ &-\alpha\tilde{q}^T M(q)M(q)^{-1}\left[\begin{array}{c} K_p\tilde{q} - K_d\dot{q} + \tilde{\xi} - G(q_d) + \\ +\beta(-\text{sign}(\dot{q}) + \varepsilon) - C(q,\dot{q})\dot{q} - F(\dot{q}) - G(q) \end{array}\right] \\ &+\alpha\tilde{q}^T\left[\begin{array}{c} K_d + B_{f1} \end{array}\right]\left[-\dot{q}\right] + \alpha\left[\tilde{q}^T K_i\right]K_i^{-1}\Phi(\dot{q})\end{aligned}$$

reagrupando términos, usando la propiedad de antisimetría $\frac{1}{2}\dot{q}^T\dot{M}(q)\dot{q} - \dot{q}^T C(q,\dot{q})\dot{q} = 0$, la igualdad $M(\dot{q}) = C(q,\dot{q}) + C(q,\dot{q})^T$ y simplificando, tenemos lo siguiente:

$$\begin{aligned}\dot{V}(\tilde{\xi},\tilde{q},\dot{q}) &= -\dot{q}^T\left[K_d - \alpha M(q)\right]\dot{q} - \tilde{q}^T[\alpha K_p - K_i]\tilde{q} + \\ &-\alpha\tilde{q}^T C(q,\dot{q})^T\dot{q} - \alpha\tilde{q}^T\left[G(q_d) - G(q)\right] + \\ &+\dot{q}^T\left[\beta(-\text{sign}(\dot{q}) + \varepsilon) - F(\dot{q})\right].\end{aligned}$$

Normando la función de Lyapunov.

$$\begin{aligned}-\dot{q}^T\left[K_d - \alpha M(q)\right]\dot{q} &\leq -\left[\begin{array}{c} \lambda_{\text{mín}}(K_d) - \alpha\lambda_{\text{máx}}(M) \end{array}\right]\ \|\dot{q}\|_2^2 \\ -\tilde{q}^T\left[\alpha K_p - K_i\right]\tilde{q} &\leq -\left[\begin{array}{c} \alpha\lambda_{\text{mín}}(K_p) - \lambda_{\text{máx}}(K_i) \end{array}\right]\ \|\tilde{q}\|_2^2\ ,\end{aligned}$$

además, usando las propiedades

$\|C(x,y)z\| \leq k_{C1}\|y\|\|z\|$

$\|G(q_d) - G(q)\| \leq k_g\|x - y\|$

los siguientes términos cumplen con:

$$-\alpha\tilde{q}^T C(q,\dot{q})^T\dot{q} \leq \alpha k_{C1}\|\tilde{q}\|_2\|\dot{q}\|_2^2$$

$$-\alpha\tilde{q}^T\left[G(q_d) - G(q)\right] \leq \alpha k_g\|\tilde{q}\|_2^2$$

finalmente, usando $\dot{q}^T \text{sign}(\dot{q}) = \|\dot{q}\|_1$, con $\|\dot{q}\|_1 = |q_1| + |q_2| + \cdots + |q_m| = \sum_{i=1}^{m} \|\dot{q}_i\|_1$, donde $\|\cdot\|_1$ es la norma 1, el valor absoluto.

$$-\dot{q}^T \beta \text{sign}(\dot{q}) \leq -\lambda_{\text{mín}}(\beta) \|\dot{q}\|_1$$

$$\dot{q}^T \beta \varepsilon \leq \varepsilon \lambda_{\text{mín}}(\beta) \|\dot{q}\|_1 \qquad ,$$

$$-\dot{q}^T F(\dot{q}) \leq \lambda_{\text{máx}}(B_{f1}) \|\dot{q}\|_1$$

después de mayorar la función de Lyapunov, tenemos:

$$\begin{array}{c} \dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) \leq -\left[\lambda_{\text{mín}}(K_d) - \alpha\lambda_{\text{máx}}(M)\right] \|\dot{q}\|_2^2 - \left[\alpha\lambda_{\text{mín}}(K_p) - \lambda_{\text{máx}}(K_i)\right] \|\tilde{q}\|_2^2 \\ \alpha k_{C1} \|\tilde{q}\|_2 \|\dot{q}\|_2^2 + \alpha k_g \|\tilde{q}\|_2^2 \\ -\lambda_{\text{mín}}(\beta) \|\dot{q}\|_1 + \varepsilon\lambda_{\text{mín}}(\beta) \|\dot{q}\|_1 + \lambda_{\text{máx}}(B_{f1}) \|\dot{q}\|_1 \end{array} ,$$

agrupando,

$$\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) \leq -\left[\lambda_{\text{mín}}(K_d) - \alpha\lambda_{\text{máx}}(M) - \alpha k_{C1} \|\tilde{q}\|_2\right] \|\dot{q}\|_2^2$$

$$-\left[\alpha\lambda_{\text{mín}}(K_p) - \lambda_{\text{máx}}(K_i) - \alpha k_g\right] \|\tilde{q}\|_2^2$$

$$-\left[\lambda_{\text{mín}}(\beta) - \varepsilon\lambda_{\text{mín}}(\beta) - \lambda_{\text{máx}}(B_{f1})\right] \|\dot{q}\|_1$$

si elegimos la norma del error de posición de la siguiente manera $\|\tilde{q}\|_2$

$$\|\tilde{q}\|_2 \leq \frac{\lambda_{\text{máx}}(M)}{\alpha k_{C1}},$$

entonces, tomando el primer término de $\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q})$, se obtiene la siguiente relación:

$$\lambda_{\text{mín}}(K_d) - \alpha\lambda_{\text{máx}}(M) - \alpha k_{C1} \|\tilde{q}\|_2 > 0$$

usando
$\|\tilde{q}\|_2 \leq \frac{\lambda_{\text{máx}}(M)}{\alpha k_{C1}}$ y $\alpha = \frac{\sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}}{\lambda_{\text{máx}}(M)}$

$\lambda_{\text{mín}}(K_d) - \alpha\lambda_{\text{máx}}(M) - \alpha k_{C1}\frac{\lambda_{\text{máx}}(M)}{\alpha k_{C1}} \geq 0$
$\lambda_{\text{mín}}(K_d) - \alpha\lambda_{\text{máx}}(M) - \lambda_{\text{máx}}(M) \geq 0$
$\lambda_{\text{mín}}(K_d) \geq \alpha\lambda_{\text{máx}}(M) + \lambda_{\text{máx}}(M)$
$\lambda_{\text{mín}}(K_d) \geq \frac{\sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}}{\lambda_{\text{máx}}(M)}\lambda_{\text{máx}}(M) + \lambda_{\text{máx}}(M)$
$\lambda_{\text{mín}}(K_d) \geq \sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)} + \lambda_{\text{máx}}(M)$

$$\lambda_{\text{mín}}(K_d) \geq \eta + \lambda_{\text{máx}}(M)$$

donde $\eta = \sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}$.

Ahora, si tomamos el segundo término de $\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q})$, y usando $\alpha = \frac{3}{\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(K_i^{-1})}$ tenemos:

$\alpha\lambda_{\text{mín}}(K_p) - \lambda_{\text{máx}}(K_i) - \alpha k_g \geq 0$
$\alpha\lambda_{\text{mín}}(K_p) \geq \lambda_{\text{máx}}(K_i) + \alpha k_g$
$\lambda_{\text{mín}}(K_p) \geq \frac{1}{\alpha}\lambda_{\text{máx}}(K_i) + k_g$
$\lambda_{\text{mín}}(K_p) \geq \frac{\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(K_i^{-1})\lambda_{\text{máx}}(K_i)}{3} + k_g$

si $\lambda_{\text{mín}}(K_i^{-1}) = \frac{1}{\lambda_{\text{máx}}(K_i)}$

$\lambda_{\text{mín}}(K_p) \geq \frac{1}{3}\lambda_{\text{mín}}(K_p) + k_g$

$$\lambda_{\text{mín}}(K_p) \geq \frac{3}{2}k_g.$$

Si tomamos la relación (5.5) encontramos el valor mínimo para K_i

$\frac{\sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M(q))}}{\lambda_{\text{máx}}(M)} \geq \alpha \geq \frac{3}{\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(K_i^{-1})}$
$\frac{\eta}{\lambda_{\text{máx}}(M)} \geq \frac{3}{\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(K_i^{-1})}$
$\frac{\eta\lambda_{\text{mín}}(K_p)}{3\lambda_{\text{máx}}(M)} \geq \frac{1}{\lambda_{\text{mín}}(K_i^{-1})}$

si nuevamente tomamos $\lambda_{\text{mín}}(K_i^{-1}) = \frac{1}{\lambda_{\text{máx}}(K_i)}$

$$\eta \frac{\lambda_{\text{mín}}(K_p)}{3\lambda_{\text{máx}}(M)} \geq \lambda_{\text{máx}}(K_i).$$

Por lo tanto, si elegimos la siguiente sintonización para las ganancias K_p, K_d, K_i y β

$$\lambda_{\text{mín}}(K_p) \geq \tfrac{3}{2} k_g$$

$$\lambda_{\text{mín}}(K_d) \geq \eta + \lambda_{\text{máx}}(M)$$

$$\lambda_{\text{máx}}(K_i) \leq \eta \tfrac{\lambda_{\text{mín}}(K_p)}{3\lambda_{\text{máx}}(M)}$$

$$\lambda_{\text{mín}}(\beta) \geq \lambda_{\text{mín}}(\beta)\varepsilon + \lambda_{\text{máx}}(B_{f1})$$

entonces,

$$\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) \leq 0$$

Estabilidad asintótica

Definamos una bola $\sum$ de radio $\sigma > 0$ centrada en el origen del espacio de estados del tipo:

$$\sum \left\{ \ \tilde{q} : \|\tilde{q}\| \leq \tfrac{\lambda_{\text{máx}}(M)}{\alpha k_{C1}} = \sigma \ \right\}.$$

Luego si $\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q})$ es semidefinida negativa en la bola $\sum$. Entonces existe una bola $\sum$ de radio $\sigma > 0$ centrada en el origen del espacio de estados que satisface que $\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) \leq 0$.

Haciendo uso del principio de invariancia de Barbashin-Krassovskii-La Salle para sistemas dinámicos autónomos definimos primero Ω como:

$$\Omega = \left\{ x(t) = \left[\tilde{\xi}, \tilde{q}, \dot{q} \right] \in \mathbb{R}^{3n} : \dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) = 0 \right\}$$

Para la derivada de la función de Lyapunov se cumple que:

$$\dot{V}(\tilde{\xi}, \tilde{q}, \dot{q}) = 0.$$

Entonces, para una solución $x(t)$ contenida en Ω para todo $t \geq 0$, es necesario y suficiente que el error de regulación sea cero, es decir, $\tilde{q} = \dot{q} = 0$ para todo $t \geq 0$.

Por lo tanto también se cumple que $\ddot{q} = 0$ para todo $t \geq 0$.

Entonces, concluimos que del sistema en lazo cerrado (5.4), si la solución $x(t) \in \Omega$ para todo $t \geq 0$, entonces $G(q) = G(q_d) = \tilde{\xi} + G(q_d)$ y $\dot{\tilde{\xi}} = 0$. Esto implica que $\tilde{\xi} = 0$ para todo $t \geq 0$. De modo que $x(t) = \left[\tilde{\xi}, \tilde{q}, \dot{q} \right] = 0 \in \mathbb{R}^{3n}$ es sólo la condición inicial en Ω para la cuál $x(t) \in \Omega$ para todo $t \geq 0$. De aquí se concluye finalmente que el origen de la ecuación de malla cerrada (5.4) es un equilibrio asintoticamente estable en forma local.

Finalmente, con estos resultados se puede formular el siguiente teorema:

Teorema 5.1 *Dada la dinámica del robot (5.2) controlada por el la ley de control PID+QL (5.3), entonces el sistema en lazo cerrado (5.4) es semiglobalmente asintóticamente estable en el punto de equilibrio:*

$$x = \left[\xi^T - G(q_d), \tilde{q}^T, \dot{q}^T \right]^T = 0 \in \mathbb{R}^{3n}$$

Con las siguientes condiciones para las ganancias:

$$\lambda_{\text{mín}}(K_p) \geq \tfrac{3}{2}k_g$$

$$\lambda_{\text{mín}}(K_d) \geq \eta + \lambda_{\text{máx}}(M)$$

$$\lambda_{\text{máx}}(K_i) \leq \eta\tfrac{\lambda_{\text{mín}}(K_p)}{3\lambda_{\text{máx}}(M)}$$

$$\lambda_{\text{mín}}(\beta) \geq \varepsilon\lambda_{\text{mín}}(\beta) + \lambda_{\text{máx}}(B_{f1})$$

,

donde $\eta = \sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}$, k_g *satisfacen la condición de Lipschitz.*

5.4. Simulaciones

Ejemplo 1

Considérese el péndulo robot de la figura (8.4), cuyas características y valores numéricos se resumen en la Tabla [13]. El objetivo de control consiste en satisfacer:

$$\lim_{t\to\infty} q(t) = q_d = \left[\frac{3\pi}{4}\right] rad.$$

Para seleccionar el valor de k_g se toma el vector de gravedad $G(q) = mgl_c$ y se busca el valor máximo cuando el péndulo robot se encuentra extendido sobre la horizontal, $q_d = \pi/2$, lo cual produce el valor de $k_g = 0{,}1607$, entonces:

$\lambda_{\text{mín}}(K_p) \geq \frac{3}{2}k_g$, por lo tanto, $\lambda_{\text{mín}}(K_p) \geq 0{,}2411$, así K_p queda:

$$K_p = 0{,}3.$$

Para el cálculo de la ganancia derivativa tenemos: $\lambda_{\text{mín}}(K_d) \geq \eta + \lambda_{\text{máx}}(M)$. Primero empezamos con el cálculo de la masa $M(q) = (ml_c^2 + I)$, por lo cual $\lambda_{\text{máx}}(M) = 0{,}0037$, y como el sistema es escalar $\lambda_{\text{máx}}(M) = \lambda\,\text{mín}(M) = 0{,}0037$.

El valor de $\eta = \sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}$, si $\lambda_{\text{mín}}(K_p) = 0{,}2411$, y $\lambda_{\text{mín}}(M) = 0{,}0037$, entonces, $\eta = \sqrt{\frac{1}{3}(0{,}2411)(0{,}0037)} = 0{,}0172$, por lo tanto, $\lambda_{\text{mín}}(K_d) \geq 0{,}0172 + 0{,}0037 \geq$

0,0361. Seleccionamos valores de K_d ligeramente más altos para evitar sobre tiros, entonces, la ganancia derivativa queda:

$$K_d = 0{,}1.$$

Para el cálculo de la ganancia del término integral, tenemos: $\lambda_{\text{mín}}(K_i) \leq \eta \frac{\lambda_{\text{mín}}(K_p)}{3\lambda_{\text{máx}}(M)} \leq 0{,}3736$, donde $\eta = 0{,}0172, \lambda_{\text{mín}}(Kp) = 0{,}2411$ y $\lambda_{\text{máx}}(M) = 0{,}0037$. La ganancia integral queda:

$$K_i = 0{,}1.$$

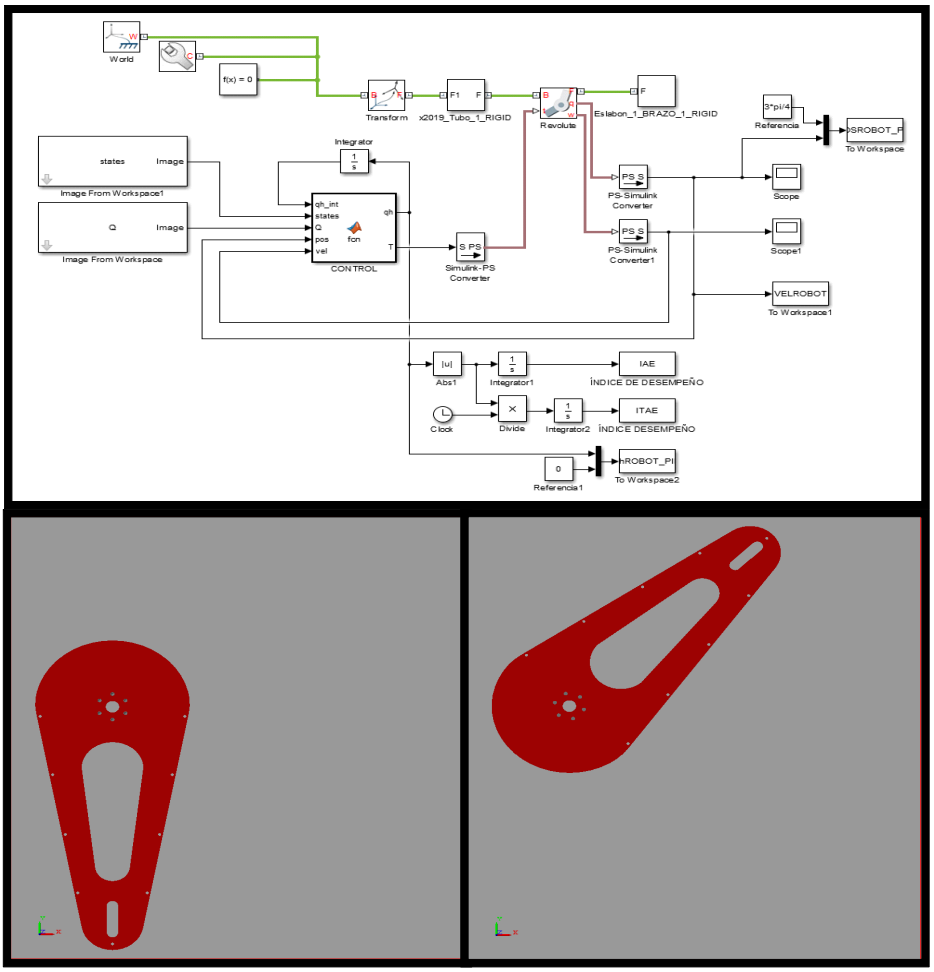
El cálculo de la última ganancia, la del aprendizaje por reforzamiento, se obtiene por medio de las condiciones en el teorema [5.1], sin embargo, le damos un valor un poco más alto para que cancele las perturbaciones o parámetros no modelados en el péndulo robot $\lambda_{\text{mín}}(\beta) \geq \varepsilon\lambda_{\text{mín}}(\beta) + \lambda_{\text{máx}}(B_{f1})$. Si el error de aproximación de la función signo lo tomamos como $\varepsilon = 0{,}01$, y el eigenvalor máximo de la matriz de fricción como $\lambda_{\text{máx}}(B_{f_1}) = 0{,}0017$, entonces $\lambda_{\text{mín}}(\beta) \geq 0{,}0117$, donde ganancia del algoritmo Q-Learning es:

$$\beta = 0{,}1.$$

Por lo tanto con esta selección de ganancias, se cumple con los requisitos impuestos para el caso de regulación en el control PID+QL vistos en el teorema [5,1].

Además, se muestra una comparación entre las leyes de control clásicas para el control de posición vistas en [104] y [106], con la misma sintonización de ganancias en la parte proporcional y derivativa, además, para los controladores que hacen uso parcial de la dinámica, como lo es, el término gravitacional $G(q)$, se considera que la masa, la inercia y el centro de masa, varían en un 5 % de su valor real.

Las condiciones iniciales son fijadas en $q(0) = 0$ y $\dot{q}(0) = 0$. Los resultados de la evaluación se realizan para seis controladores de posición sobre el ángulo de salida del hombro, que se presentan en la figura (5.4) donde se grafica el ángulo q del péndulo robot para cada controlador.



(a) Posición de Casa (b) Posición Final

Figura 5.3: Diagrama a bloques del controlador (arriba), y brazo del péndulo robot de su condición inicial $q = 0$, a su condición final $q_d = 3\pi/4$, desde Simulink Matlab.

El objetivo de control consiste en mover la articulación del péndulo robot $q_d = 3\pi/4$, usando los algoritmos de control antes citados para el control de posición, con lo cual se busca posicionar al robot en la configuración deseada.

La evolución en el tiempo del ángulo de salida y del error de posición se presentan en las figuras (5.4) y (5.5), donde el efecto de amortiguamiento atenúa de forma considerable los sobretiros en la respuesta del robot. El error en la articulación no presenta sobre impulsos como se observa en la figura (5.5). El régimen transitorio tiene una duración de 1 segundo.

Índices de desempeño: Con la finalidad de cuantificar el comportamiento de los controladores, utilizamos los criterios integrales conocidos como Integral del error absoluto *(IAE)* e Integral del tiempo por el error absoluto *(ITAE)*:

$$IAE = \int_0^\infty \mid \tilde{q}(t) \mid dt \tag{5.6}$$

$$ITAE = \int_0^\infty t \mid \tilde{q}(t) \mid dt \tag{5.7}$$

donde el error está dado por $\tilde{q}(t) = q_d - q$

Tabla 7. Índices de desempeño

	IAE	ITAE
PD+QL	2,8319	5,6762
PD+G(q)	1,7144	1,7134
PD+G(q_d)	1,9943	2,6500
PID	2,2342	3,7003
PID+QL	1,4943	0,8439
PD	6,3101	26,5947

Tal como se muestra en la Tabla [7], los índices de desempeño IAE, ITAE se encargan evaluar los seis controladores de posición, arrojando la siguiente información: el controlador que tuvo el desempeño más bajo fue el control P con realimentación de velocidad, esto es debido a que el controlador dentro de su algoritmo no incluye ninguna información sobre la

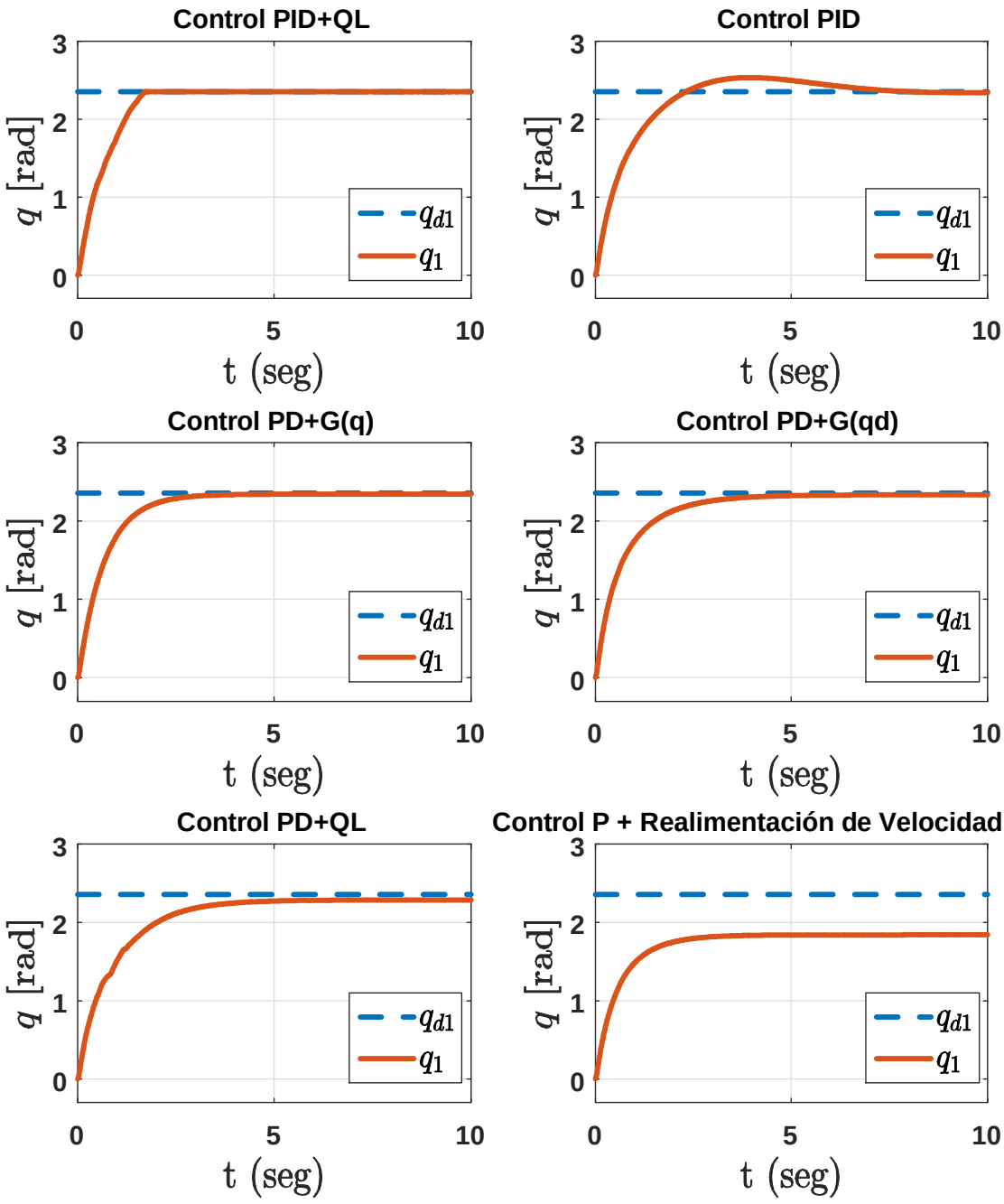


Figura 5.4: Gráfica del ángulo de posición q_1

dinámica del sistema a controlar, provocando que su desempeño sea bastante bajo. El controlador que toma el lugar siguiente es el control PID, donde la parte integral elimina el error en estado estacionario que no se puede eliminar con el controlador anterior, sin embargo, la sintonización de K_i no siempre resulta mejor en comparación de tener conocimiento de una parte de la dinámica del sistema, como lo es, el término gravitacional. Los controladores PD con precompensación y compensación de gravedad, además de mostrar una respuesta parecida, también presentan una respuesta favorable al alcanzar la posición deseada, debido a su término gravitacional en la ley de control que cancela la dinámica de G en todo momento. Finalmente, tanto el control PD+QL como el control PID+QL compensados con el aprendizaje por reforzamiento, son leyes de control que hacen uso de la inteligencia artificial, sin embargo, el control PID+QL toma el primer puesto al tener mejores resultados comparado con los otros controladores, al alcanzar la posición final deseada en un menor tiempo y sin sobre impulsos, debido a la sintonización explícita de ganancias y a que la aportación del QL logra compensar la dinámica no modelada, mucho mejor que tener el conocimiento de G en el control, y además disminuir el error en estado estacionario en un menor tiempo.

Se realizó la comparación entre el control PID con compensación QL y los controladores clásicos de control de posición para robots [104] y [106], donde la Tabla [7] muestra que los valores más pequeños en los índices de desempeño son para el controlador PID+QL, lo cual nos dice que el ángulo de salida del péndulo robot alcanza la referencia y se mantiene sobre ella en un menor tiempo y con menor error en el estado estacionario en comparación con los otros controladores, además en este controlador PID+QL propuesto se aprecian ventajas como por ejemplo: la respuesta híbrida trabaja mucho mejor que un controlador PID, PD+G o PD+QL, así mismo, la sintonización del control PID está basada en condiciones explícitas dadas en el teorema [5.1], lo cual resulta mucho más simple. La desventaja que se presenta, es que para cada posición deseada diferente, el algoritmo QL tiene que realizar un nuevo aprendizaje, por lo tanto, si cambiamos la q_d, se necesita un nuevo QL.

Finalmente, el tiempo que tarda en aprender el algoritmo de aprendizaje por reforzamiento para el control de posición es de aproximadamente 30 minutos por q_d, lo que representa 1000 episodios de 1000 Iteraciones por episodio.

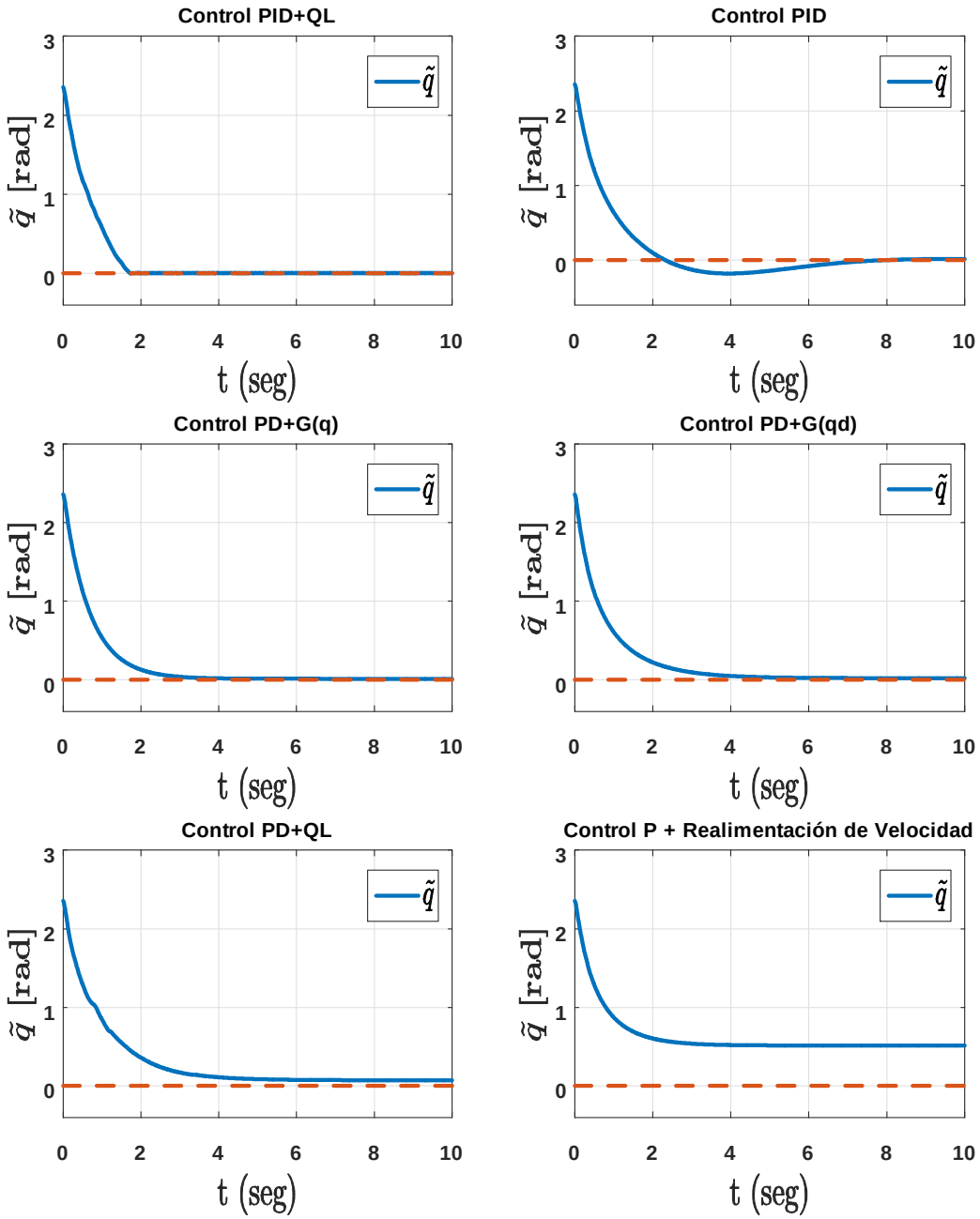


Figura 5.5: Gráfica de errores $\tilde{q}$

Ejemplo 2

Considérese el robot manipulador de 2 GDL estudiado en la ecuación (8.1), y mostrado en la figura (8.7). Los diversos elementos que conforman el modelo dinámico se describen en el apéndice A, y el cálculo de las ganancias se obtienen de la siguiente manera:

Para seleccionar el valor de k_g se toma el vector de gravedad $G(q)$ y se busca el valor máximo cuando los eslabones se encuentran extendidos $q_{d1} = \pi/2$ y $q_{d2} = 0$, lo cual produce el valor de $k_g = 0{,}6104$, entonces:

$\lambda_{\text{mín}}(K_p) \geq \frac{3}{2}k_g$, por lo tanto, $\lambda_{\text{mín}}(K_p) \geq 0{,}9156$, así la matriz de K_p queda:

$$K_p = \begin{bmatrix} 1 & 0 \\ 0 & 0{,}9156 \end{bmatrix}.$$

Para el cálculo de la ganancia derivativa tenemos: $\lambda_{\text{mín}}(K_d) \geq \eta + \lambda_{\text{máx}}(M)$. Primero empezamos con el cálculo de matriz $M(q)$, donde sus eigenvalores son: $\lambda_1 = 0{,}0178$, y $\lambda_2 = 0{,}0011$, por lo cual $\lambda_{\text{máx}}(M) = 0{,}0178$.

El valor de $\eta = \sqrt{\frac{1}{3}\lambda_{\text{mín}}(K_p)\lambda_{\text{mín}}(M)}$, si $\lambda_{\text{mín}}(K_p) = 0{,}9156$, y $\lambda_{\text{mín}}(M) = 0{,}0011$, entonces $\eta = \sqrt{\frac{1}{3}(0{,}9156)(0{,}0011)} = 0{,}0183$, por lo tanto, $\lambda_{\text{mín}}(K_d) \geq 0{,}0183{+}0{,}0178 \geq 0{,}0361$. Seleccionamos valores de K_d ligeramente más altos para evitar sobre tiros, entonces, la matriz queda:

$$K_d = \begin{bmatrix} 0{,}2 & 0 \\ 0 & 0{,}1 \end{bmatrix}.$$

Para el cálculo de la ganancia del término integral, tenemos: $\lambda_{\text{mín}}(K_i) \leq 0{,}3138$, donde $\eta = 0{,}0183, \lambda_{\text{mín}}(Kp) = 0{,}9156, \lambda_{\text{máx}}(M) = 0{,}0178$. La matriz de ganancia integral queda:

$$K_i = \begin{bmatrix} 0{,}3 & 0 \\ 0 & 0{,}1 \end{bmatrix}.$$

El cálculo de la última ganancia, la del aprendizaje por reforzamiento, se obtiene por medio de las condiciones en el teorema [5.1], sin embargo, le damos un valor un poco más alto para que cancele las perturbaciones o parámetros no modelados en el brazo robótico

$\lambda_{\text{mín}}(\beta) \geq \varepsilon\lambda_{\text{mín}}(\beta) + \lambda_{\text{máx}}(B_{f1})$. Si el error de aproximación de la función signo lo tomamos como $\varepsilon = 0{,}01$, y el eigenvalor máximo de la matriz de fricción como $\lambda_{\text{máx}}(B_{f_1}) = 0{,}0017$, entonces $\lambda_{\text{mín}}(\beta) \geq 0{,}0117$, donde la matriz de ganancia del algoritmo Q-Learning es:

$$\beta = \begin{bmatrix} 0{,}4 & 0 \\ 0 & 0{,}05 \end{bmatrix}.$$

Por lo tanto con esta selección de ganancias se cumple con los requisitos impuestos para el caso de regulación en el control PID+QL vistos en el teorema [5.1].

También, considérese las leyes de control (5.3), (4.2), y los controles clásicos vistos en [104], donde sus matrices de diseño usan la sintonización arriba explicada, tanto para la ganancia proporcional como la derivativa, además, para los controladores que hacen uso parcial de la dinámica, como lo es, el término gravitacional $G(q)$, se considera que la masa, las inercias y los centros de masas, varían en un 5 % de su valor real.

Las condiciones iniciales correspondientes a las posiciones y velocidades se escogen nulas:

$$\begin{aligned} q_1(0) = 0, \quad q_2(0) = 0, \\ \dot{q}_1(0) = 0, \quad \dot{q}2(0) = 0. \end{aligned}$$

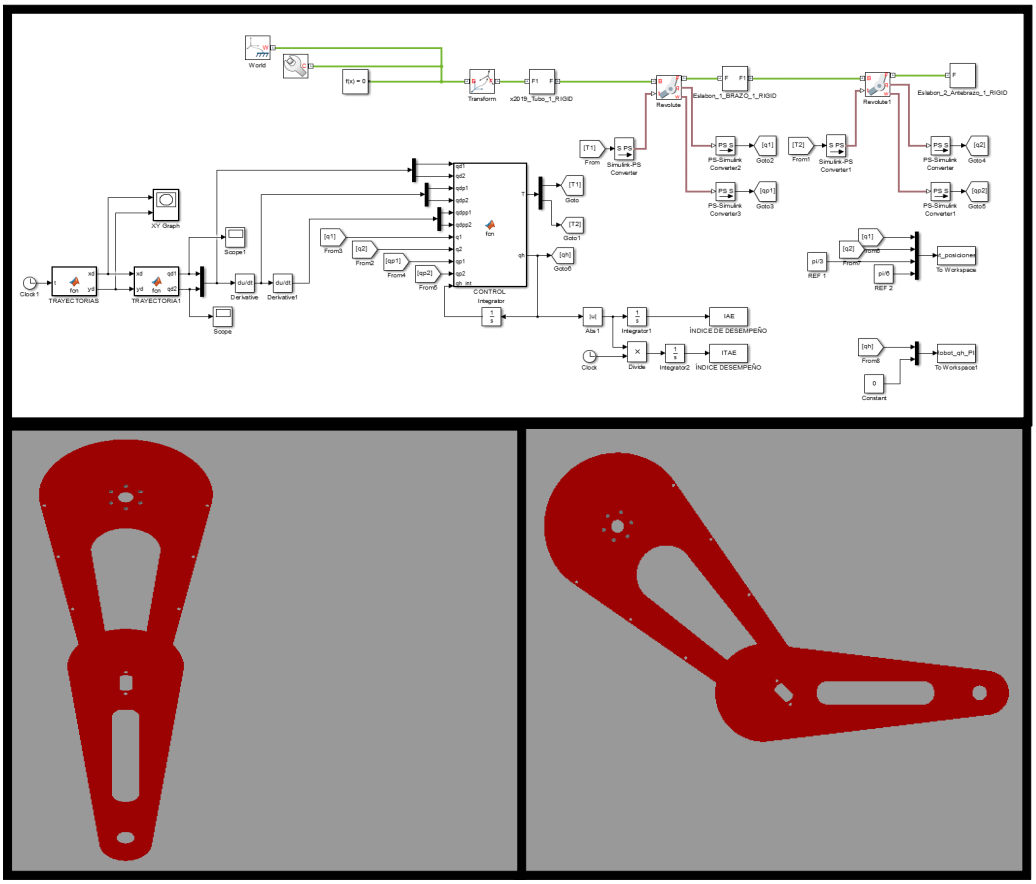
Las posiciones articulares deseadas son:

$$q_{d1} = \pi/4 \ [rad], \qquad q_{d2} = \pi/4 \ [rad].$$

En términos del vector de estado de la ecuación del error $\tilde{q}$ y de velocidad $\dot{q}$, el estado inicial correspondiente es

$$\begin{bmatrix} \tilde{q}_1(0) \\ \tilde{q}_2(0) \\ \dot{q}_1(0) \\ \dot{q}_2(0) \end{bmatrix} = \begin{bmatrix} \pi/4 \\ \pi/4 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0{,}7854 \\ 0{,}7854 \\ 0 \\ 0 \end{bmatrix}$$

La figuras (5.7) y (5.8) presentan los resultados de simulación. En particular, se muestra cómo los elementos que conforman el error de posición $\tilde{q}$ tienden a cero.



(a) Posición inicial (b) Posición final

Figura 5.6: Diagrama a bloques del controlador (arriba), y robot manipulador de su condición inicial $q_1 = 0, q_2 = 0$ a su condición final $q_{d1} = \pi/4,\ q_{d2} = \pi/4$, desde Simulink Matlab.

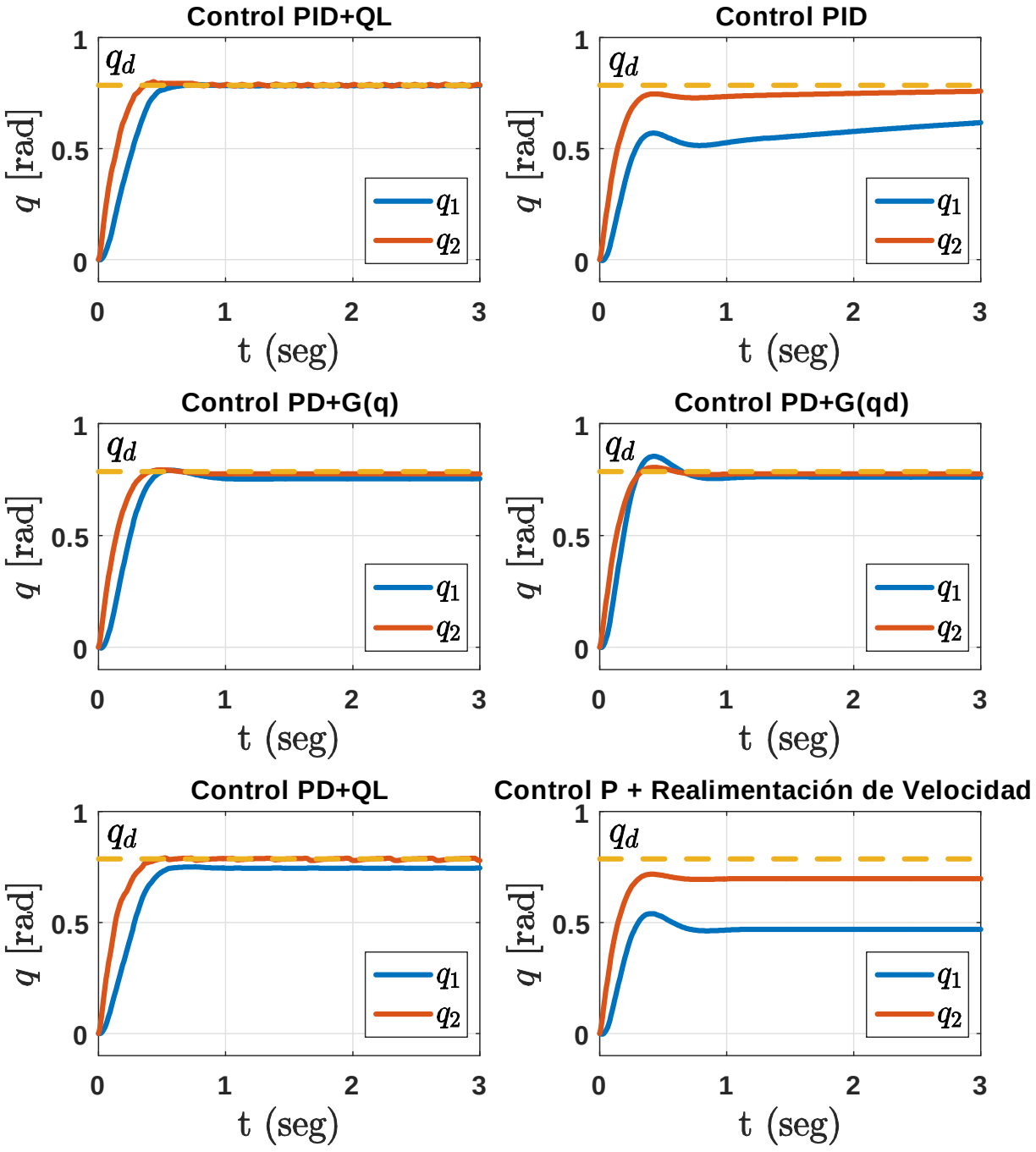


Figura 5.7: Gráfica de los ángulos del hombro q_1 y codo q_2.

La figura (5.7) muestra los ángulos de salida tanto para el codo como para el hombro evaluados para los seis controladores de posición. Se observa que el algoritmo de aprendizaje por reforzamiento trabajando de forma híbrida con el control PID brinda mejores resultados al alcanzar las condiciones deseadas en un tiempo menor y reduciendo el error en estado estacionario. Los controladores PD+G(q) y PD+G(qd) tienen un comportamiento bastante similar, siendo que de manera cualitativa no hay diferencia. Los ángulos de salida utilizando el control PID tiende a su condición deseada mientras el tiempo tienda a infinito, sin embargo, se tomaría un mayor tiempo alcanzar la condición final y disminuir el error en estado estacionario. Finalmente, el peor desempeño lo presenta el control P con realimentación de velocidad, mostrando que los ángulos de salida se encuentran muy por debajo de las condiciones deseadas finales, debido a que el controlador no tiene conocimiento alguno de la dinámica del sistema, sino que sólo conoce el estado de la posición y la velocidad.

En la figura (5.8) se presentan las gráficas del error de posición para las seis leyes de control en el caso de regulación. Las variables $\tilde{q}_1$ y $\tilde{q}_2$ representa el error entre la condición deseada q_d y el estado q, tanto para el hombro como para el codo. En todos los casos, a excepción del control PD, se tiende a disminuir el error en estado estacionario, siendo el de mejor desempeño el controlador PID+QL al lograr disminuir el error en estado estacionario y alcanzar la referencia en un menor tiempo, comparado con los demás controladores. La gráfica del error de salida para la ley de control P con realimentación de velocidad muestra que $\tilde{q}_1$ y $\tilde{q}_2$ una vez que pasan el estado transitorio se estabilizan en una condición diferente a $\tilde{q} = 0$, y no se reduce el error en estado estacionario.

Índices de desempeño: En el área de robótica no hay un criterio estándar para medir el desempeño (performance) de un algoritmo de control. Nyquist, Root-locus, Bode, etc., son criterios específicos de sistemas lineales los cuales no aplican a robots manipuladores debido a su naturaleza no lineal.

Algunos investigadores miden el desempeño del algoritmo de control por inspección visual de las gráficas del error de posición y a su juicio determinan si el desempeño es adecuado. Sin embargo, dicha medición es muy subjetiva y queda a interpretación del mismo.

Un criterio académico ampliamente aceptado en la comunidad científica de robótica para

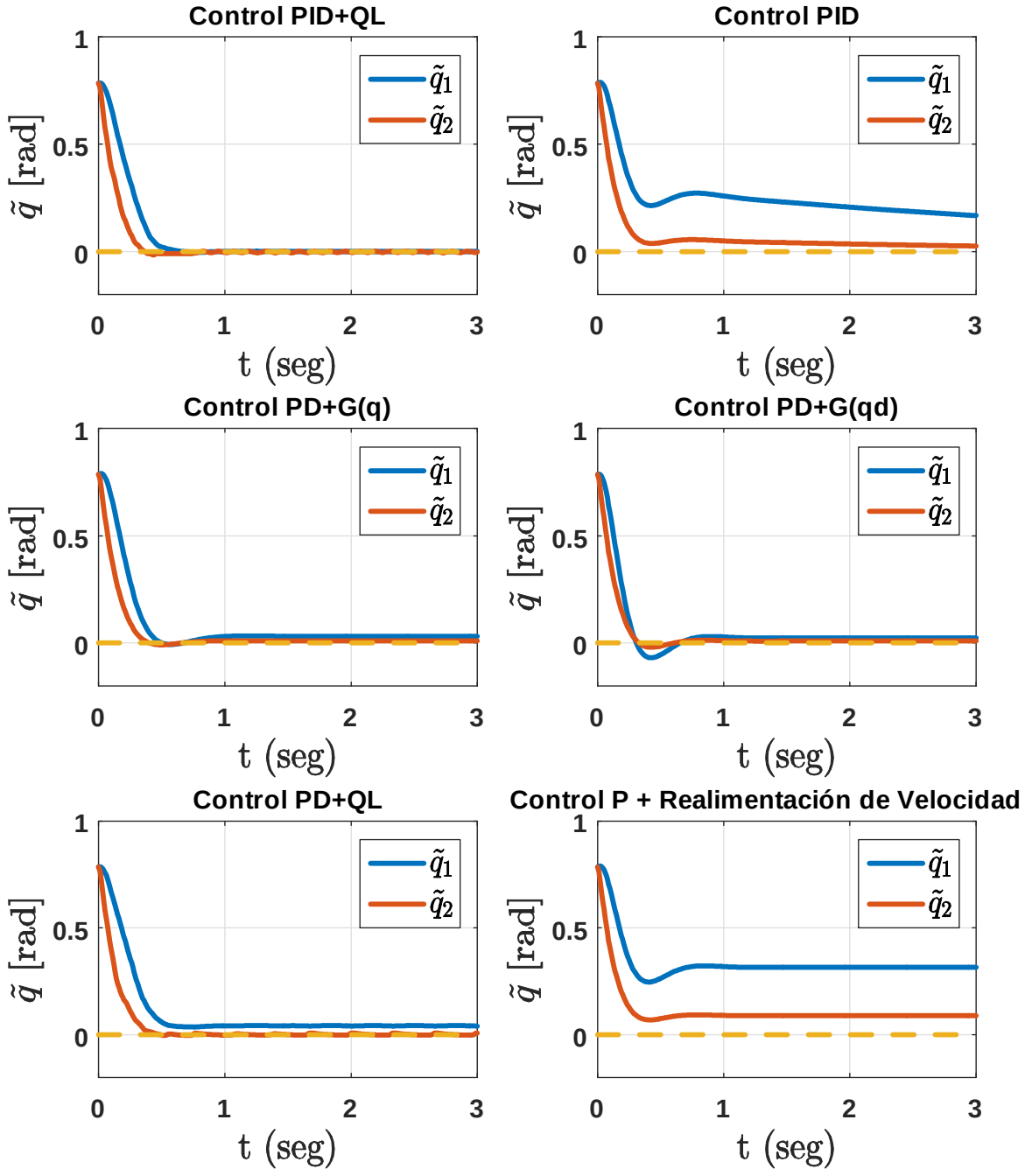


Figura 5.8: Gráfica de los errores de posición $\tilde{q}$ para el hombro y el codo.

medir el desempeño de un algoritmo de control es por medio de la obtención del índice de desempeño determinado por la integral del error absoluto (IAE), y por la integral del tiempo por el error absoluto (ITAE).

$$IAE = \int_0^\infty \mid \tilde{q}(t) \mid dt$$

$$ITAE = \int_0^\infty t \mid \tilde{q}(t) \mid dt$$

donde el error está dado por $\tilde{q}(t) = \begin{bmatrix} q_{d1} - q_1 \\ q_{d2} - q_2 \end{bmatrix}$

Tabla 8. Índices de desempeño

	IAE(q_1)	ITAE(q_1)	IAE(q_2)	ITAE(q_2)
PD	1,0065	1,4238	0,3375	0,4037
PID	0,7592	0,9423	0,2070	0,1722
PID+QL	0,1861	0,0329	0,1048	0,0204
PD+$G(q)$	0,2428	0,1560	0,1206	0,0501
PD+ $G(q_d)$	0,1975	0,1204	0,1173	0,0502
PD+QL	0,2957	0,2108	0,1063	0,0215

El análisis comparativo del índice de desempeño se realiza con respecto a seis esquemas de control de referencia, generalmente son los controladores clásicos de control de posición pura debido a que se conocen ampliamente. El índice de desempeño de los esquemas de control se refiere a la exactitud que debe tener la respuesta del robot manipulador. Claramente se ve en la Tabla [8] que los controladores que hacen uso de las propiedades matemáticas del gradiente de la energía potencial $\frac{\partial U(q)}{\partial q}$ como el control PD con compensación de gravedad que permiten exhibir un corto transitorio, sobre impulsos y vibraciones mecánicas atenuadas por la función de amortiguamiento, facilidad en la sintonía de las ganancias proporcional y derivativa, robustez frente a incertidumbres paramétricas que son algunas de las cualidades del esquema de control y se ven reflejadas en el desempeño del algoritmo de control y por lo tanto repercuten en la medición de su indicador *IAE* e *ITAE*. El control híbrido PID+QL

brinda una respuesta con un alto desempeño, lo cual nos dice que la respuesta del robot es de calidad, es decir, presenta una curva con perfil suave, picos o sobre impulsos atenuados, rápido estado transitorio, no hay vibración mecánica, ni oscilaciones. Aunque este controlador híbrido no presenta el término de la energía potencial en su ley de control, esta es compensada con el aprendizaje por reforzamiento QL que otorga la energía necesaria para compensar las dinámicas no modeladas del sistema a controlar, y además, disminuir el error en estado estacionario. Cuando el desempeño de un algoritmo de control no es el adecuado, como lo muestra la Tabla [8] para el control PD, es necesario incrementar las ganancias o agregarle términos derivados de la energía potencial.

Finalmente, el tiempo que tarda en aprender el algoritmo de aprendizaje por reforzamiento para el control de posición es de aproximadamente 30 minutos por q_d, lo que representa 1000 episodios de 1000 Iteraciones por episodio.

Capítulo 6

Control PD+G(q) Compensado con el Aprendizaje por Reforzamiento.

6.1. Introducción.

Hoy en día, el interés en los robots manipuladores radica en su facultad de realizar movimientos de alta velocidad y con alto grado de exactitud. El control de movimiento mueve al robot libremente en su espacio de trabajo siguiendo una trayectoria deseada en posición y velocidad sin interactuar con su medio ambiente.

El problema del control de movimiento es uno de los temas más importantes en robótica. Recientemente ha recibido la atención en la comunidad científica y como resultado se han reportado en la literatura diversos controladores; entre los que han mostrado tener mejor desempeño se encuentran: control Par Calculado, control PD+, PD con precompensación calculada y PD con compensación.

El control de movimiento de robots manipuladores ha sido ampliamente estudiado en simulaciones. Sin embargo, la evaluación experimental de controladores basados en el modelo del robot manipulador es un problema de origen práctico, el cual ha quedado evidenciado en la literatura científica.

Lo anterior se debe a la falta de robots experimentales adecuados, así como a la dificultad

que presenta conocer el valor nominal de los parámetros dinámicos del robot manipulador.

Los algoritmos del control de trayectoria incluyen la dinámica completa del robot manipulador en la estructura matemática del controlador, es decir, se basan en el modelo dinámico del robot. La exactitud, desempeño y robustez de esos controladores dependen del grado de precisión con que se conozcan los parámetros dinámicos que describen el modelo.

El problema del control de trayectoria o control de movimiento puede plantearse formalmente en los siguientes términos.

Considérese el modelo dinámico visto en (5.2) de un robot manipulador de n GDL con eslabones rígidos, con fricción viscosa en sus uniones y con actuadores ideales:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + B_{f1}\dot{q} + G(q) = \tau,$$

o en términos del vector de estado $\left[q^T,\ \dot{q}^T\right]^T$:

$$\frac{d}{dt}\begin{bmatrix} q \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \dot{q} \\ M(q)^{-1}\left(\tau(t) - C(q,\dot{q})\dot{q} - B_{f1}\dot{q} - G(q)\right) \end{bmatrix},$$

donde $M(q) \in \mathbb{R}^{n\times n}$ es la matriz de inercia, $C(q,\dot{q})\dot{q} \in \mathbb{R}^n$ es el vector de fuerzas centrifugas y de coriolis, $G(q) \in \mathbb{R}^n$ es el vector de pares gravitacionales y $\tau \in \mathbb{R}^n$ es un vector de fuerzas y pares aplicados en las uniones. Los vectores $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ denotan la posición, velocidad y aceleración articular, respectivamente.

El problema de control de movimiento de robots manipuladores puede formularse de la siguiente manera:

Considérese la ecuación dinámica de un robot de n GDL (5.2). Dado un conjunto de funciones vectoriales acotadas $q_d, \dot{q}_d, \ddot{q}_d$ referidas como posiciones, velocidades y aceleraciones articulares deseadas, se trata de determinar una función vectorial τ, de tal forma que las posiciones q asociadas a las coordenadas articulares del robot sigan con precisión a q_d.

En términos más formales, el objetivo de control de movimiento consiste en determinar τ de tal forma que:

$$\lim_{t\to\infty} \begin{bmatrix} \tilde{q}(t) \\ \dot{\tilde{q}}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix},$$

donde $\tilde{q}, \dot{\tilde{q}} \in \mathbb{R}^n$ denota el vector de errores de posiciones y velocidades articulares, simplemente denominado error de posición, y error de velocidad definido como:

$$\begin{bmatrix} \tilde{q}(t) \\ \dot{\tilde{q}}(t) \end{bmatrix} = \begin{bmatrix} q_d(t) - q(t) \\ \dot{q}_d(t) - \dot{q}(t) \end{bmatrix}.$$

Si el objetivo de control se cumple, significará que las articulaciones del robot manipulador siguen asintóticamente la trayectoria de movimiento deseado.

El cálculo del vector τ involucra generalmente una función vectorial no lineal de $q, \dot{q}, \ddot{q}$. Esta función se denominará "ley de control" o simplemente controlador. Es importante recordar que los robots manipuladores disponen de sensores de posición y velocidad para cada articulación por lo que los vectores q y $\dot{q}$ son medibles y pueden emplearse en los controladores. El controlador puede expresarse como:

$$\tau = \tau\left(q, \dot{q}, \ddot{q}, q_d, \dot{q}_d, \ddot{q}_d, M(q), C(q,\dot{q}), B_{f1}, G(q)\right).$$

Para fines prácticos es deseable que el controlador no dependa de la aceleración articular $\ddot{q}$

La figura (6.1) presenta un diagrama de bloques formado por un controlador en malla cerrada con un robot. Donde se muestra el diagrama del control de trayectoria de robots manipuladores. Obsérvese que las variables que definen el problema de control tales como el error de posición y error de velocidad son procesadas por la estructura matemática del esquema de control, la cual requiere del conocimiento completo de la dinámica del robot.

La ecuación en lazo cerrado que determina el problema del control de movimiento queda expresada en variables de estado articulares $\begin{bmatrix} \tilde{q}^T, & \dot{\tilde{q}}^T \end{bmatrix}^T$ de la siguiente forma:

$$\frac{d}{dt} = \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ M(q)^{-1}\left(\tau(K_p, K_d, \tilde{q}, \dot{\tilde{q}}, M(q), C(q,\dot{q}), B_{f1}, G(q)) - C(q,\dot{q})\dot{q} - B_{f1}\dot{q} - G(q)\right) \end{bmatrix} \tag{6.1}$$

La ecuación en lazo cerrado (6.1) es una ecuación diferencial ordinaria no lineal y no autónoma. La incorporación de la trayectoria $q_d(t)$ es lo que determina que la naturaleza de la ecuación diferencial en lazo cerrado sea no autónoma.

6.2. Control de un Robot para Seguimiento de Trayectoria

El diseño de la trayectoria de seguimiento, velocidad y aceleración deseada deben incorporar aspectos de planificación de trayectoria para una adecuada operatividad de la tarea que va a desempeñar el robot.

1) Las funciones de seguimiento $q_d(t), \dot{q}_d(t), \ddot{q}_d(t) \in \mathbb{R}^n$ representan la posición, velocidad y aceleración, respectivamente.

2) El error de seguimiento $\tilde{q} \in \mathbb{R}^n$ se define $\tilde{q}(t) = q_d(t) - q(t)$.

3) El error de velocidad $\dot{\tilde{q}} \in \mathbb{R}^n$ se define como $\dot{\tilde{q}}(t) = \dot{q}_d(t) - \dot{q}(t)$.

4) Las ganancias proporcional y derivativa, respectivamente, son $K_p, K_d \in \mathbb{R}^{n\times n}$, ambas matrices definidas positivas.

Nótese la diferencia sustancial que existe entre el control de posición y control de trayectoria.

El control de posición se define en términos del error de posición $\tilde{q}$ y la velocidad articular $\dot{q}$ para propósitos de inyección de amortiguamiento. Únicamente se requiere conocimiento parcial de la dinámica del robot, como es el par gravitacional $G(q)$.

$$\tau = \tau\left(K_p, K_d, \tilde{q}, \dot{q}, G(q)\right)$$

En contraste, en control de trayectoria se controla el error de posición $\tilde{q}$ y el error de

velocidad $\dot{\tilde{q}}$. Además, se requiere del conocimiento completo de la dinámica del robot manipulador (efecto inercial, vector de fuerzas centrípetas y de coriolis, par gravitacional y pares de fricción).

$$\tau = \tau\left(K_p, K_d, \tilde{q}, \dot{\tilde{q}}, M(q), C(q,\dot{q}), B_{f_1}, G(q)\right)$$

La notación requerida para la descripción de los esquemas de control se determina de la siguiente manera: sea $q_d(t)$ denota la trayectoria de seguimiento, es una función continua, suave y acotada, además es doblemente diferenciable. La velocidad de seguimiento y la aceleración deseada deben ser funciones continuas y diferenciables suaves y acotadas en sus magnitudes de acuerdo con las características propias del robot.

6.3. Control PD+G(q) con compensación QL

Considérese el modelo dinámico visto en (5.2), que describe el comportamiento de un robot manipulador de n grados de libertad.

El objetivo de control de movimiento se puede definir formalmente de la siguiente manera: dada la posición, la velocidad y la aceleración angular deseada $q_d, \dot{q}_d, \ddot{q}_d \in \mathbb{R}^n$, el problema es diseñar una ley de control τ tal que:

$$\begin{aligned} \lim_{t\to\infty} \|\tilde{q}(t)\| &= 0. \\ \lim_{t\to\infty} \left\|\dot{\tilde{q}}(t)\right\| &= 0 \end{aligned}$$

donde el vector $\tilde{q}, \dot{\tilde{q}} \in \mathbb{R}^n$ denotan el vector de errores de posición y velocidades articulares, definido como: $\tilde{q} = q - q_d$, $\dot{\tilde{q}} = \dot{q}_d - \dot{q} \in \mathbb{R}^n$.

Ley de Control

La ley de control PD+G(q) con compensación QL puede expresarse de la siguiente manera:

$$\tau = K_p\tilde{q} + K_d\dot{\tilde{q}} + G(q) + u_r$$

donde las matrices de diseño $K_p, K_d, \in \mathbb{R}^{n\times n}$ llamadas respectivamente las ganancias proporcional y derivativa, son matrices simétricas y definidas positivas convencionalmente elegidas.

$u_r \in \mathbb{R}^n$ es el algoritmo de control llamado Q-Learning, que tiene la forma:

$$u_r = \beta(\text{sign}(\tilde{q}) - \varepsilon),$$

donde β es una matriz diagonal y definida positiva seleccionadas por el diseñador, y ε representa el error de aproximación de la función $\Phi(\tilde{q})$ con la función $sign(\tilde{q})$, donde la función signo se representa de la siguiente manera:

$$\Phi(\tilde{q}) = \text{sign}(\tilde{q}) = \begin{cases} 1 \text{ si } \tilde{q} > 0 \\ 0 \text{ si } \tilde{q} = 0 \\ -1 \text{ si } \tilde{q} < 0 \end{cases}$$

El vector $\text{sign}(\tilde{q})$ está definido por $\text{sign}(\tilde{q}) = [\text{sign}(\tilde{q}_1), \ldots, \text{sign}(\tilde{q}_n)]^T$.

Entonces, la ley de control para el seguimiento de trayectoria queda de la siguiente manera:

$$\tau = K_p\tilde{q} + K_d\dot{\tilde{q}} + G(q) + \beta(\text{sign}(\tilde{q}) - \varepsilon). \tag{6.2}$$

Ecuación en malla cerrada

La ecuación que describe el comportamiento en malla cerrada se obtiene al combinar las ecuaciones (5.2) y (6.2).

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + B_{f1}\dot{q} + G(q) = K_p\tilde{q} + K_d\dot{\tilde{q}} + G(q) + \beta(\text{sign}(\tilde{q}) - \varepsilon),$$

agrupando y eliminando términos:

$$M(q)\ddot{q} = K_p\tilde{q} + K_d\dot{\tilde{q}} - C(q,\dot{q})\dot{q} - B_{f1}\dot{q} + \beta(\text{sign}(\tilde{q}) - \varepsilon),$$

despejando $\ddot{q}$ se tiene:

$$\ddot{q} = M(q)^{-1}\left[K_p\tilde{q} + K_d\dot{\tilde{q}} - C(q,\dot{q})\dot{q} - B_{f1}\dot{q} + \beta(\text{sign}(\tilde{q}) - \varepsilon)\right],$$

la cual puede expresarse en términos del vector de estado $\left[\tilde{q}^T, \dot{\tilde{q}}^T\right]^T$ como:

$$\frac{d}{dt}\begin{bmatrix}\tilde{q}\\ \dot{\tilde{q}}\end{bmatrix} = \begin{bmatrix}\dot{\tilde{q}}\\ \ddot{q}_d - M(q)^{-1}\left[K_p\tilde{q} + K_d\dot{\tilde{q}} - C(q_d-\tilde{q},\dot{q}_d-\dot{\tilde{q}})\dot{q} - B_{f1}\dot{q} + \beta(\text{sign}(\tilde{q}) - \varepsilon)\right]\end{bmatrix} \tag{6.3}$$

la cual es una ecuación diferencial no lineal y no autónoma. Esta última propiedad se debe a que la ecuación depende explícitamente de las funciones del tiempo $q_d(t)$ y $\dot{q}_d(t)$.

Cabe constatar además que la ecuación de malla cerrada tiene como único estado de equilibrio al origen $\left[\tilde{q}^T, \dot{\tilde{q}}^T\right]^T = 0 \in \mathbb{R}^{2n}$.

Prueba de Estabilidad

Para estudiar la estabilidad del origen del espacio de estados, se propone la siguiente función candidata de Lyapunov:

$$V(t,\tilde{q},\dot{\tilde{q}}) = \frac{1}{2}\dot{\tilde{q}}^T M(q)\dot{\tilde{q}} + \frac{1}{2}\tilde{q}^T K_p\tilde{q} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T M(q)\dot{\tilde{q}} + \int_0^t \Phi(\tilde{q})d\tilde{q}.$$

La prueba de que la función candidata de Lyapunov es definida positiva se puede ver en [106].

La derivada temporal de $V(t,\tilde{q},\dot{\tilde{q}})$ a lo largo de las trayectorias del sistema en lazo cerrado y usando $\frac{d}{dt}\int_0^t \Phi(\tilde{q})d\tilde{q} = \frac{\partial \int_0^t \Phi(\tilde{q})d\tilde{q}}{\partial \tilde{q}}\frac{\partial \tilde{q}}{\partial t} = \dot{\tilde{q}}^T\Phi(\tilde{q})$, tenemos:

$$\begin{aligned}\dot{V}(t,\tilde{q},\dot{\tilde{q}}) &= \dot{\tilde{q}}^T M(q)\ddot{\tilde{q}} + \frac{1}{2}\dot{\tilde{q}}^T\dot{M}(q)\dot{\tilde{q}} + \tilde{q}^T K_p\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\dot{\tilde{q}}^T M(q)\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\dot{M}(q)\dot{\tilde{q}}\\ &\quad + \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T M(q)\ddot{\tilde{q}} - \frac{\epsilon_0}{\|\tilde{q}\|\left[1+\|\tilde{q}\|\right]^2}\tilde{q}^T\dot{\tilde{q}}\tilde{q}^T M(q)\dot{\tilde{q}} + \dot{\tilde{q}}^T\Phi(\tilde{q})\end{aligned}$$

sustituyendo $\dot{\tilde{q}}$ y $\ddot{\tilde{q}}$ en la ecuación anterior resulta:

$$\begin{aligned}\dot{V}(t,\tilde{q},\dot{\tilde{q}}) &= \dot{\tilde{q}}^T M(q)\left[\ddot{q}_d - M(q)^{-1}[K_p\tilde{q} + K_d\dot{\tilde{q}} - C(q,\dot{q})\dot{q} - B_{f1}\dot{q} + \beta(\text{sign}(\tilde{q}) - \varepsilon)]\right] \\ &+\frac{1}{2}\dot{\tilde{q}}^T\dot{M}(q)\dot{\tilde{q}} + \tilde{q}^T K_p\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\dot{\tilde{q}}^T M(q)\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\dot{M}(q)\dot{\tilde{q}} \\ &\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T M(q)[\ddot{q}_d - M(q)^{-1}\left[K_p\tilde{q} + K_d\dot{\tilde{q}} - C(q,\dot{q})\dot{q} - B_{f1}\dot{q} + \beta(\text{sign}(\tilde{q}) - \varepsilon)\right]] \\ &-\frac{\epsilon_0}{\|\tilde{q}\|\left[1+\|\tilde{q}\|\right]^2}\tilde{q}^T\dot{\tilde{q}}\tilde{q}^T M(q)\dot{\tilde{q}} + \dot{\tilde{q}}^T\Phi(\tilde{q})\end{aligned}$$

eliminando $M(q)M(q)^{-1}$,

$$\begin{aligned}\dot{V}(t,\tilde{q},\dot{\tilde{q}}) &= \dot{\tilde{q}}^T M(q)\ddot{q}_d - \dot{\tilde{q}}^T\left[K_p\tilde{q} + K_d\dot{\tilde{q}} - C(q,\dot{q})\dot{q} - B_{f1}\dot{q} + \beta(\text{sign}(\tilde{q}) - \varepsilon)\right] \\ &+\frac{1}{2}\dot{\tilde{q}}^T\dot{M}(q)\dot{\tilde{q}} + \tilde{q}^T K_p\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\dot{\tilde{q}}^T M(q)\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\dot{M}(q)\dot{\tilde{q}} \\ &\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T M(q)\ddot{q}_d - \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\left[K_p\tilde{q} + K_d\dot{\tilde{q}} - C(q,\dot{q})\dot{q} - B_{f1}\dot{q} + \beta(\text{sign}(\tilde{q}) - \varepsilon)\right] \\ &-\frac{\epsilon_0}{\|\tilde{q}\|\left[1+\|\tilde{q}\|\right]^2}\tilde{q}^T\dot{\tilde{q}}\tilde{q}^T M(q)\dot{\tilde{q}} + \dot{\tilde{q}}^T\Phi(\tilde{q})\end{aligned}$$

eliminando y reagrupando términos, además, usando $\Phi(\tilde{q}) = \beta(sign(\tilde{q}) - \varepsilon)$, y $\dot{M}(q) = C(q,\dot{q})^T + C(q,\dot{q})$

$$\begin{aligned}\dot{V}(t,\tilde{q},\dot{\tilde{q}}) &= \dot{\tilde{q}}^T M(q)\ddot{q}_d - \dot{\tilde{q}}^T K_d\dot{\tilde{q}} + \frac{1}{2}\dot{\tilde{q}}^T\dot{M}(q)\dot{\tilde{q}} + \dot{\tilde{q}}^T C(q,\dot{q})\dot{q} + \dot{\tilde{q}}B_{f1}\dot{q} - \dot{\tilde{q}}^T\beta(\text{sign}(\tilde{q}) - \varepsilon) \\ &+\frac{\epsilon_0}{1+\|\tilde{q}\|}\dot{\tilde{q}}^T M(q)\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\left[C(q,\dot{q})^T + C(q,\dot{q})\right]\dot{\tilde{q}} \\ &\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T M(q)\ddot{q}_d - \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T K_p\tilde{q} - \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T K_d\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T C(q,\dot{q})\dot{q} \\ &+\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T B_{f1}\dot{q} - \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\beta(\text{sign}(\tilde{q}) - \varepsilon) \\ &-\frac{\epsilon_0}{\|\tilde{q}\|\left[1+\|\tilde{q}\|\right]^2}\tilde{q}^T\dot{\tilde{q}}\tilde{q}^T M(q)\dot{\tilde{q}} + \dot{\tilde{q}}^T\beta(\text{sign}(\tilde{q}) - \varepsilon)\end{aligned}$$

si $\dot{q} = \dot{q}_d - \dot{\tilde{q}}$, entonces $\dot{\tilde{q}}^T C(q,\dot{q})\dot{q} = \dot{\tilde{q}}^T C(q,\dot{q})\dot{q}_d - \dot{\tilde{q}}^T C(q,\dot{q})\dot{\tilde{q}}$, y $\tilde{q}B_{f1}\dot{q} = \tilde{q}B_{f1}\dot{q}_d - \tilde{q}B_{f1}\dot{\tilde{q}}$.

$$
\begin{aligned}
\dot{V}(t,\tilde{q},\dot{\tilde{q}}) &= \dot{\tilde{q}}^T M(q)\ddot{q}_d - \dot{\tilde{q}}^T K_d\dot{\tilde{q}} + \dot{\tilde{q}}^T[\frac{1}{2}\dot{M}(q) - C(q,\dot{q})]\dot{\tilde{q}} + \dot{\tilde{q}}^T C(q,\dot{q})\dot{q}_d + \dot{\tilde{q}}B_{f1}\dot{q}_d - \dot{\tilde{q}}B_{f1}\dot{\tilde{q}} \\
&\quad -\dot{\tilde{q}}^T\beta(sign(\tilde{q}) - \varepsilon) + \frac{\epsilon_0}{1+\|\tilde{q}\|}\dot{\tilde{q}}^T M(q)\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\left[C(q,\dot{q})^T + C(q,\dot{q})\right]\dot{\tilde{q}} \\
&\quad \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T M(q)\ddot{q}_d - \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T K_p\tilde{q} - \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T K_d\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}[\tilde{q}^T C(q,\dot{q})\dot{q}_d - \tilde{q}^T C(q,\dot{q})\dot{\tilde{q}}] \\
&\quad +\frac{\epsilon_0}{1+\|\tilde{q}\|}[\tilde{q}B_{f1}\dot{q}_d - \tilde{q}B_{f1}\dot{\tilde{q}}] - \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\beta(\text{sign}(\tilde{q}) - \varepsilon) \\
&\quad -\frac{\epsilon_0}{\|\tilde{q}\|\left[1+\|\tilde{q}\|\right]^2}\tilde{q}^T\dot{\tilde{q}}\tilde{q}^T M(q)\dot{\tilde{q}} + \dot{\tilde{q}}^T\beta(\text{sign}(\tilde{q}) - \varepsilon)
\end{aligned}
$$

utilizando la propiedad de antisimetría $\frac{1}{2}\dot{M}(q) - C(q,\dot{q}) = 0$, y simplificando, tenemos:

$$
\begin{aligned}
\dot{V}(t,\tilde{q},\dot{\tilde{q}}) &= \dot{\tilde{q}}^T M(q)\ddot{q}_d - \dot{\tilde{q}}^T[K_d + B_{f1}]\dot{\tilde{q}} + \dot{\tilde{q}}^T C(q,\dot{q})\dot{q}_d + \dot{\tilde{q}}B_{f1}\dot{q}_d \\
&\quad +\frac{\epsilon_0}{1+\|\tilde{q}\|}\dot{\tilde{q}}^T M(q)\dot{\tilde{q}} + \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T C(q,\dot{q})^T\dot{\tilde{q}} \\
&\quad \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T M(q)\ddot{q}_d - \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T K_p\tilde{q} - \frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T K_d\dot{\tilde{q}} \\
&\quad +\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T C(q,\dot{q})\dot{q}_d + \frac{\epsilon_0}{1+\|\tilde{q}\|}[\tilde{q}B_{f1}\dot{q}_d - \tilde{q}B_{f1}\dot{\tilde{q}}] \\
&\quad -\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\beta(\text{sign}(\tilde{q}) - \varepsilon) - \frac{\epsilon_0}{\|\tilde{q}\|\left[1+\|\tilde{q}\|\right]^2}\tilde{q}^T\dot{\tilde{q}}\tilde{q}^T M(q)\dot{\tilde{q}}
\end{aligned}
$$

tomando las siguientes desigualdades:

$$\dot{\tilde{q}}^T M(q)\ddot{q}_d \leq \left\|\dot{\tilde{q}}\right\| \lambda_{\text{máx}}(M) \left\|\ddot{q}_d\right\|_{\text{M}} \leq \left\|\dot{\tilde{q}}\right\|^2 \lambda_{\text{máx}}(M) \left\|\ddot{q}_d\right\|_{\text{M}}$$

$$-\dot{\tilde{q}}^T[K_d + B_{f1}]\dot{\tilde{q}} \leq -\left[\lambda_{\text{mín}}(K_d) + \lambda_{\text{mín}}(B_{f1})\right] \left\|\dot{\tilde{q}}\right\|^2$$

$$\tfrac{\epsilon_0}{1+\|\tilde{q}\|}\dot{\tilde{q}}^T M(q)\dot{\tilde{q}} \leq \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \left\|\dot{\tilde{q}}\right\|^2 \lambda_{\text{máx}}(M) \leq \epsilon_0 \left\|\dot{\tilde{q}}\right\|^2 \lambda_{\text{máx}}(M)$$

$$-\tfrac{\epsilon_0}{\|\tilde{q}\|[1+\|\tilde{q}\|]^2}\tilde{q}^T\dot{\tilde{q}}\tilde{q}^T M(q)\dot{\tilde{q}} \leq \left|\tfrac{\epsilon_0}{\|\tilde{q}\|[1+\|\tilde{q}\|]^2}\tilde{q}^T\dot{\tilde{q}}\tilde{q}^T M(q)\dot{\tilde{q}}\right| \leq \tfrac{\epsilon_0\|\tilde{q}\|^2\left\|\dot{\tilde{q}}\right\|^2\lambda_{\text{máx}}(M)}{\|\tilde{q}\|[1+\|\tilde{q}\|]^2} \leq \epsilon_0 \left\|\dot{\tilde{q}}\right\|^2 \lambda_{\text{máx}}(M)$$

$$\tfrac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T C(q,\dot{q})^T\dot{\tilde{q}} = \tfrac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T C(q,\dot{q})\dot{\tilde{q}} \leq \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \|\dot{q}\| \left\|\dot{\tilde{q}}\right\|,$$

si tomamos $\|\dot{q}\| = \left\|\dot{q}_d - \dot{\tilde{q}}\right\|$, entonces, $\|\dot{q}\| \left\|\dot{\tilde{q}}\right\| = \left\|\dot{q}_d - \dot{\tilde{q}}\right\| \left\|\dot{\tilde{q}}\right\| \leq \|\dot{q}_d\| \left\|\dot{\tilde{q}}\right\| + \left\|\dot{\tilde{q}}\right\|^2$.

$$\begin{aligned}
&\tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \|\dot{q}\| \left\|\dot{\tilde{q}}\right\| \leq \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \left[\|\dot{q}_d\| \left\|\dot{\tilde{q}}\right\| + \left\|\dot{\tilde{q}}\right\|^2\right] \\
&\tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \left[\|\dot{q}_d\| \left\|\dot{\tilde{q}}\right\| + \left\|\dot{\tilde{q}}\right\|^2\right] \leq \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \|\dot{q}_d\| \left\|\dot{\tilde{q}}\right\| + \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \left\|\dot{\tilde{q}}\right\|^2, \\
&\tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \|\dot{q}_d\| \left\|\dot{\tilde{q}}\right\| + \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \left\|\dot{\tilde{q}}\right\|^2 \leq \epsilon_0 k_c \|\tilde{q}\| \|\dot{q}_d\| \left\|\dot{\tilde{q}}\right\| + \epsilon_0 k_c \left\|\dot{\tilde{q}}\right\|^2
\end{aligned}$$

además

$$\begin{aligned}
&\dot{\tilde{q}}^T C(q,\dot{q})\dot{q}_d \leq \left\|\dot{\tilde{q}}\right\| k_c \|\dot{q}\| \|\dot{q}_d\| \leq \left\|\dot{\tilde{q}}\right\| k_c \left\|\dot{q}_d - \dot{\tilde{q}}\right\| \|\dot{q}_d\| \leq \left\|\dot{\tilde{q}}\right\| k_c[\|\dot{q}_d\|^2 - \left\|\dot{\tilde{q}}\right\| \|\dot{q}_d\|] \\
&\left\|\dot{\tilde{q}}\right\| k_c[\|\dot{q}_d\|^2 - \left\|\dot{\tilde{q}}\right\| \|\dot{q}_d\|] \leq \left\|\dot{\tilde{q}}\right\| k_c \|\dot{q}_d\|^2 + k_c \left\|\dot{\tilde{q}}\right\|^2 \|\dot{q}_d\| \\
&\left\|\dot{\tilde{q}}\right\| k_c \|\dot{q}_d\|^2 + k_c \left\|\dot{\tilde{q}}\right\|^2 \|\dot{q}_d\| \leq \left\|\dot{\tilde{q}}\right\|^2 k_c \|\dot{q}_d\|^2 + k_c \left\|\dot{\tilde{q}}\right\|^2 \|\dot{q}_d\|^2 \leq 2k_c \left\|\dot{\tilde{q}}\right\|^2 \|\dot{q}_d\|^2
\end{aligned}$$

$$\dot{\tilde{q}} B_{f1}\dot{q}_d \leq \left\|\dot{\tilde{q}}\right\| \lambda_{\text{máx}}(B_{f1}) \|\dot{q}_d\| \leq \left\|\dot{\tilde{q}}\right\|^2 \lambda_{\text{máx}}(B_{f1}) \|\dot{q}_d\|$$

$$\begin{aligned}
&\tfrac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T C(q,\dot{q})\dot{q}_d \leq \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \|\dot{q}\| \|\dot{q}_d\| \leq \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \left\|\dot{q}_d - \dot{\tilde{q}}\right\| \|\dot{q}_d\| \\
&\tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \left\|\dot{q}_d - \dot{\tilde{q}}\right\| \|\dot{q}_d\| \leq \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \|\dot{q}_d\|^2 + \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \|\dot{q}_d\| \left\|\dot{\tilde{q}}\right\| \\
&\tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \|\dot{q}_d\|^2 + \tfrac{\epsilon_0}{1+\|\tilde{q}\|} \|\tilde{q}\| k_c \|\dot{q}_d\| \left\|\dot{\tilde{q}}\right\| \leq \epsilon_0 \|\tilde{q}\| k_c \|\dot{q}_d\|_{\text{M}}^2 + \epsilon_0 \|\tilde{q}\| k_c \|\dot{q}_d\|_{\text{M}} \left\|\dot{\tilde{q}}\right\|
\end{aligned}$$

$$\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}B_{f1}\dot{q}_d \leq \frac{\epsilon_0}{1+\|\tilde{q}\|}\|\tilde{q}\|\,\lambda_{\text{máx}}(B_{f1})\,\|\dot{q}_d\|_{\text{M}} \leq \epsilon_0\|\tilde{q}\|\,\lambda_{\text{máx}}(B_{f1})\,\|\dot{q}_d\|_{\text{M}}$$

$$\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T M(q)\ddot{q}_d \leq \frac{\epsilon_0}{1+\|\tilde{q}\|}\|\tilde{q}\|\,\lambda_{\text{máx}}(M)\,\|\ddot{q}_d\|_{\text{M}} \leq \epsilon_0\|\tilde{q}\|\,\lambda_{\text{máx}}(M)\,\|\ddot{q}_d\|_{\text{M}}$$

entonces, la derivada queda:

$$\begin{aligned}\dot{V}(t,\tilde{q},\dot{\tilde{q}}) \;\leq\;& -\left[\lambda_{\text{mín}}(K_d)+\lambda_{\text{mín}}(B_{f1})\right]\left\|\dot{\tilde{q}}\right\|^2+\epsilon_0\left\|\dot{\tilde{q}}\right\|^2\lambda_{\text{máx}}(M)+\epsilon_0\left\|\dot{\tilde{q}}\right\|^2\lambda_{\text{máx}}(M)\\
&+\left\|\dot{\tilde{q}}\right\|^2\lambda_{\text{máx}}(M)\,\|\ddot{q}_d\|+\epsilon_0 k_c\,\|\tilde{q}\|\,\|\dot{q}_d\|\,\left\|\dot{\tilde{q}}\right\|+\epsilon_0 k_c\left\|\dot{\tilde{q}}\right\|^2-\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T K_p\tilde{q}\\
&+2k_c\left\|\dot{\tilde{q}}\right\|^2\|\dot{q}_d\|^2+\left\|\dot{\tilde{q}}\right\|^2\lambda_{\text{máx}}(B_{f1})\,\|\dot{q}_d\|+\epsilon_0\|\tilde{q}\|\,k_c\,\|\dot{q}_d\|\,\left\|\dot{\tilde{q}}\right\|\\
&-\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\left[K_d+B_{f1}\right]\dot{\tilde{q}}-\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\beta(\text{sign}(\tilde{q})-\varepsilon)\\
&+\epsilon_0\|\tilde{q}\|\,\lambda_{\text{máx}}(B_{f1})\,\|\dot{q}_d\|+\epsilon_0\|\tilde{q}\|\,k_c\,\|\dot{q}_d\|^2+\epsilon_0\|\tilde{q}\|\,\lambda_{\text{máx}}(M)\,\|\ddot{q}_d\|\end{aligned}$$

ahora considérese las siguientes desigualdades:

$$\begin{aligned}-\tfrac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T K_p\tilde{q} &\leq -\tfrac{\epsilon_0}{1+\|\tilde{q}\|}\lambda_{\text{mín}}(K_p)\,\|\tilde{q}\|^2\\
-\tfrac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\left[K_d+B_{f1}\right]\dot{\tilde{q}} \leq \left|-\tfrac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\left[K_d+B_{f1}\right]\dot{\tilde{q}}\right| &\leq \epsilon_0\left[\lambda_{\text{máx}}(K_d)+\lambda_{\text{máx}}(B_{f1})\right]\|\tilde{q}\|\,\left\|\dot{\tilde{q}}\right\|.\end{aligned}$$

Finalmente, usando $\tilde{q}^T\text{sign}(\tilde{q}) = \|\tilde{q}\|_1$, con $\|\tilde{q}\|_1 = |\tilde{q}_1|+|\tilde{q}_2|+\cdots+|\tilde{q}_m| = \sum_{i=1}^m \|\tilde{q}_i\|_1$, donde $\|\cdot\|_1$ es la norma 1, el valor absoluto

$$-\frac{\epsilon_0}{1+\|\tilde{q}\|}\tilde{q}^T\beta(\text{sign}(\tilde{q})-\varepsilon) \leq -\frac{\epsilon_0}{1+\|\tilde{q}\|}\lambda_{\text{mín}}(\beta)\,\|\tilde{q}\|_1+\frac{\epsilon_0}{1+\|\tilde{q}\|}\lambda_{\text{mín}}(\beta)\varepsilon\,\|\tilde{q}\|_1.$$

Incorporando las cotas anteriores a la derivada de Lyapunov, tenemos:

$$
\begin{aligned}
\dot{V}(t,\tilde{q},\dot{\tilde{q}}) \;\leq\; & -\left[\lambda_{\text{mín}}(K_d)+\lambda_{\text{mín}}(B_{f1})\right]\left\|\dot{\tilde{q}}\right\|^2+2\epsilon_0\left\|\dot{\tilde{q}}\right\|^2\lambda_{\text{máx}}(M)+\epsilon_0 k_c\left\|\dot{\tilde{q}}\right\|^2 \\
& +2k_c\left\|\dot{\tilde{q}}\right\|^2\left\|\dot{q}_d\right\|^2+\left\|\dot{\tilde{q}}\right\|^2\lambda_{\text{máx}}(B_{f1})\left\|\dot{q}_d\right\|+\left\|\dot{\tilde{q}}\right\|^2\lambda_{\text{máx}}(M)\left\|\ddot{q}_d\right\| \\
& +\epsilon_0 k_c\left\|\tilde{q}\right\|\left\|\dot{q}_d\right\|\left\|\dot{\tilde{q}}\right\|-\frac{\epsilon_0}{1+\left\|\tilde{q}\right\|}\lambda_{\text{mín}}(K_p)\left\|\tilde{q}\right\|^2 \\
& +\epsilon_0\left[\lambda_{\text{máx}}(K_d)+\lambda_{\text{máx}}(B_{f1})\right]\left\|\tilde{q}\right\|\left\|\dot{\tilde{q}}\right\|+\epsilon_0\left\|\tilde{q}\right\|k_c\left\|\dot{q}_d\right\|\left\|\dot{\tilde{q}}\right\| \\
& -\frac{\epsilon_0}{1+\left\|\tilde{q}\right\|}\lambda_{\text{mín}}(\beta)\left\|\tilde{q}\right\|_1+\frac{\epsilon_0}{1+\left\|\tilde{q}\right\|}\lambda_{\text{mín}}(\beta)\varepsilon\left\|\tilde{q}\right\|_1 \\
& +\epsilon_0\left\|\tilde{q}\right\|\lambda_{\text{máx}}(B_{f1})\left\|\dot{q}_d\right\|_{\text{M}}+\epsilon_0\left\|\tilde{q}\right\|k_c\left\|\dot{q}_d\right\|_{\text{M}}^2+\epsilon_0\left\|\tilde{q}\right\|\lambda_{\text{máx}}(M)\left\|\ddot{q}_d\right\|_{\text{M}}
\end{aligned}
$$

agrupando:

$$
\begin{aligned}
\dot{V}(t,\tilde{q},\dot{\tilde{q}}) \;\leq\; & -\left[\begin{array}{c}\lambda_{\text{mín}}(K_d)+\lambda_{\text{mín}}(B_{f1})-2\epsilon_0\lambda_{\text{máx}}(M)-\epsilon_0 k_c \\ -2k_c\left\|\dot{q}_d\right\|_{\text{M}}^2-\lambda_{\text{máx}}(B_{f1})\left\|\dot{q}_d\right\|_{\text{M}}-\lambda_{\text{máx}}(M)\left\|\ddot{q}_d\right\|_{\text{M}}\end{array}\right]\left\|\dot{\tilde{q}}\right\|^2 \\
& -\frac{\epsilon_0}{1+\left\|\tilde{q}\right\|}\lambda_{\text{mín}}(K_p)\left\|\tilde{q}\right\|^2 \\
& -\frac{\epsilon_0}{1+\left\|\tilde{q}\right\|}\left[\begin{array}{c}\lambda_{\text{mín}}(\beta)-\lambda_{\text{mín}}(\beta)\varepsilon-\epsilon_0\lambda_{\text{máx}}(B_{f1})\left\|\dot{q}_d\right\|_{\text{M}} \\ -\epsilon_0 k_c\left\|\dot{q}_d\right\|_{\text{M}}^2-\epsilon_0\lambda_{\text{máx}}(M)\left\|\ddot{q}_d\right\|_{\text{M}}\end{array}\right]\left\|\tilde{q}\right\|_1 \\
& +\epsilon_0\left[\lambda_{\text{máx}}(K_d)+\lambda_{\text{máx}}(B_{f1})+2k_c\left\|\dot{q}_d\right\|_{\text{M}}\right]\left\|\tilde{q}\right\|\left\|\dot{\tilde{q}}\right\|.
\end{aligned}
$$

$$
\dot{V}(t,\tilde{q},\dot{\tilde{q}}) \leq -\left[\begin{array}{c}\left\|\dot{\tilde{q}}\right\| \\ \left\|\tilde{q}\right\|\end{array}\right]^T\left[\begin{array}{cc}\begin{array}{c}\lambda_{\text{mín}}(K_d)+\lambda_{\text{mín}}(B_{f1}) \\ -2\epsilon_0\lambda_{\text{máx}}(M)-\epsilon_0 k_c \\ -2k_c\left\|\dot{q}_d\right\|_{\text{M}}^2-\lambda_{\text{máx}}(B_{f1})\left\|\dot{q}_d\right\|_{\text{M}} \\ -\lambda_{\text{máx}}(M)\left\|\ddot{q}_d\right\|_{\text{M}}\end{array} & -\frac{\epsilon_0}{2}\left[\begin{array}{c}\lambda_{\text{máx}}(K_d)+ \\ \lambda_{\text{máx}}(B_{f1}) \\ +2k_c\left\|\dot{q}_d\right\|_{\text{M}}\end{array}\right] \\ -\frac{\epsilon_0}{2}\left[\begin{array}{c}\lambda_{\text{máx}}(K_d) \\ +\lambda_{\text{máx}}(B_{f1}) \\ +2k_c\left\|\dot{q}_d\right\|_{\text{M}}\end{array}\right] & \frac{\epsilon_0}{1+\left\|\tilde{q}\right\|}\lambda_{\text{mín}}(K_p)\end{array}\right]\left[\begin{array}{c}\left\|\dot{\tilde{q}}\right\| \\ \left\|\tilde{q}\right\|\end{array}\right]
$$

$$
-\tfrac{\epsilon_0}{1+\left\|\tilde{q}\right\|}\left[\lambda_{\text{mín}}(\beta)-\lambda_{\text{mín}}(\beta)\varepsilon-\epsilon_0\lambda_{\text{máx}}(B_{f1})\left\|\dot{q}_d\right\|_{\text{M}}-\epsilon_0 k_c\left\|\dot{q}_d\right\|_{\text{M}}^2-\epsilon_0\lambda_{\text{máx}}(M)\left\|\ddot{q}_d\right\|_{\text{M}}\right]\left\|\tilde{q}\right\|_1<0 \tag{6.4}
$$

Para que $\dot{V}(t, \tilde{q}, \dot{\tilde{q}}) < 0$

$$\lambda_{\text{mín}}(\beta) > \lambda_{\text{mín}}(\beta)\varepsilon + \epsilon_0 \lambda_{\text{máx}}(B_{f1}) \left\|\dot{q}_d\right\|_{\text{M}} + \epsilon_0 k_c \left\|\dot{q}_d\right\|_{\text{M}}^2 + \epsilon_0 \lambda_{\text{máx}}(M) \left\|\ddot{q}_d\right\|_{\text{M}},$$

y además la matriz debe satisfacer que:

$$\frac{4\lambda_{\text{mín}}(K_p)\left(\lambda_{\text{mín}}(K_d) + \lambda_{\text{mín}}(B_{f1})\right)}{\left(\lambda_{\text{máx}}(K_d) + \lambda_{\text{máx}}(B_{f1}) + k_c \left\|\dot{q}_d\right\|_{\text{M}}\right)^2 \left[1 + \|\tilde{q}\|\right] + 4\left(\lambda_{\text{mín}}(K_d)k_c + 2\lambda_{\text{mín}}(K_p)\lambda_{\text{máx}}(M)\right)} > \epsilon_0 > 0.$$

No es necesario conocer el valor de ϵ_0, sólo es suficiente su existencia.

Entonces, la función (6.4) es definida negativa y por lo tanto se demuestra la estabilidad asintótica global del punto de equilibrio, es decir:

$$\lim_{t\to\infty} \left[\tilde{q}^T, \dot{\tilde{q}}^T\right]^T \Rightarrow 0 \in \mathbb{R}^{2n}$$

Teorema 6.1 *Dada la dinámica del robot (5.2) controlada por el la ley de control PD+G(q) con compensación QL (6.2), entonces el sistema en lazo cerrado (6.3) es asintóticamente globalmente estable en el punto de equilibrio:*

$$x = \left[\tilde{q}^T, \dot{\tilde{q}}^T\right]^T = 0 \in \mathbb{R}^{2n}.$$

El diagrama de bloques se muestra en la figura (6.1), el cual muestra al control PD+G(q) con compensación QL en lazo cerrado con el robot.

6.4. Simulaciones

Ejemplo 1)

Considérese el péndulo robot de la figura (8.4) donde sus parámetros mostrados en la Tabla [13] presentan una variación del $\pm 5\,\%$, en la masa m_1, la inercia I_1, centro de masa l_{c1} y fricción viscosa b_1, cuyo modelo viene dado de la siguiente manera:

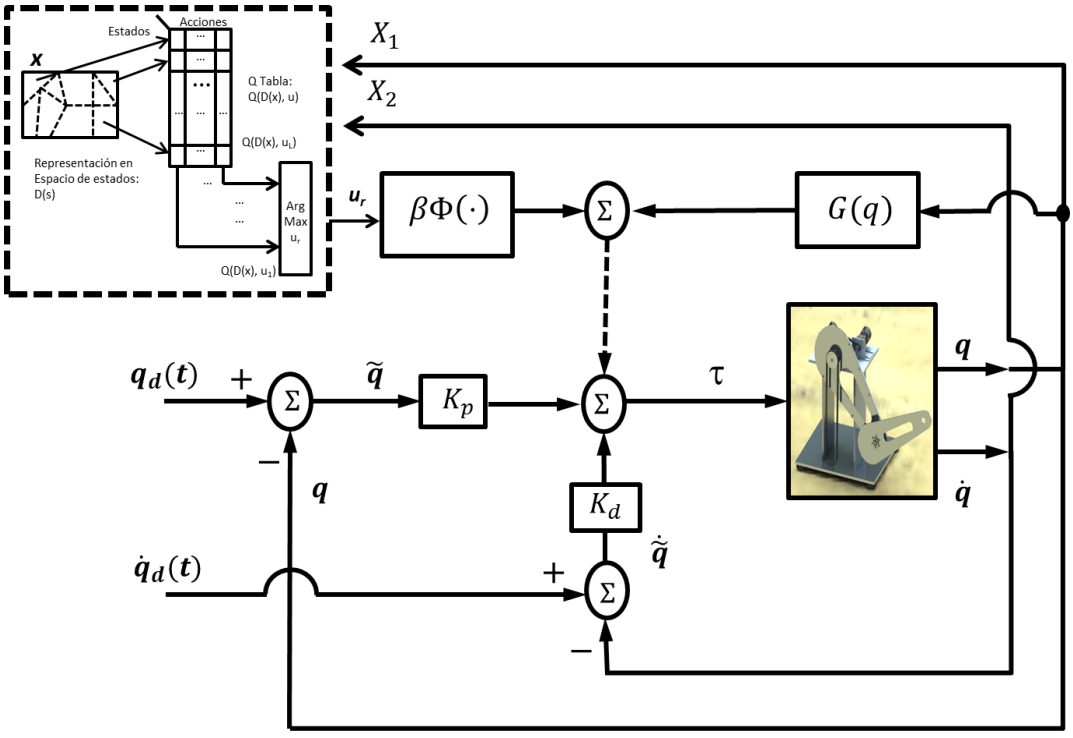


Figura 6.1: Esquema de control PD+G(q) con compensación QL para el control de movimiento.

$$\left(m_1 l_{c1}^2 + I_1\right) \ddot{q}_1 + b_1 \dot{q}_1 + m_1 g l_{c1} \sin(q_1) = \tau,$$

donde q_1 es la posición angular con respecto a la vertical y τ es el par aplicado en la unión. Para este ejemplo se identifica $M(q_1) = (m_1 l_{c1}^2 + I_1)$, $C(q_1, \dot{q}_1) = 0$, $b_1 = 0{,}001785$, $G(q_1) = m_1 g l_{c1} \sin(q_1)$.

Las ganancias son:

$$K_p = 1{,}5\ [Nm/rad], \quad K_d = 0{,}3\ [Nms/rad].$$

Las condiciones iniciales son fijadas en $q_1(0) = 0$ y $\dot{q}_1(0) = 0$.

Con el propósito de evaluar mediante simulaciones numéricas la prestación del controlador de movimiento descrito anteriormente, se ha elegido un polinomio de 5to grado definido en el apéndice A.

Se pretende que el robot siga las trayectorias deseadas $q_d(t), \dot{q}_d(t), \ddot{q}_d(t)$ representadas respectivamente en (8.2), y con las siguientes restricciones:

$$\begin{bmatrix} q_d(0) \\ qd(t_f) \\ \dot{q}_d(0) \\ \dot{q}d(t_f) \\ \ddot{q}_d(0) \\ \ddot{q}_d(t_f) \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

Los resultados de la evaluación para el seguimiento de trayectoria sobre el ángulo de salida q se presentan en la figura (6.2) y (6.3), donde se grafica el ángulo q_1 y la velocidad $\dot{q}_1$ para cada controlador.

Los resultados del seguimiento de posición se presentan en la figura (6.2), donde se muestra una comparación entre controladores clasicos vistos en [106] y [104]. En color azul punteado se presenta la trayectoria de posición deseada obtenida por medio del polinomio de 5to grado, y en color rojo se presenta el ángulo de salida q_1 para la articulación rotacional del péndulo robot. Se observa un buen seguimiento de trayectoria para los controles PD+G(q)+QL,

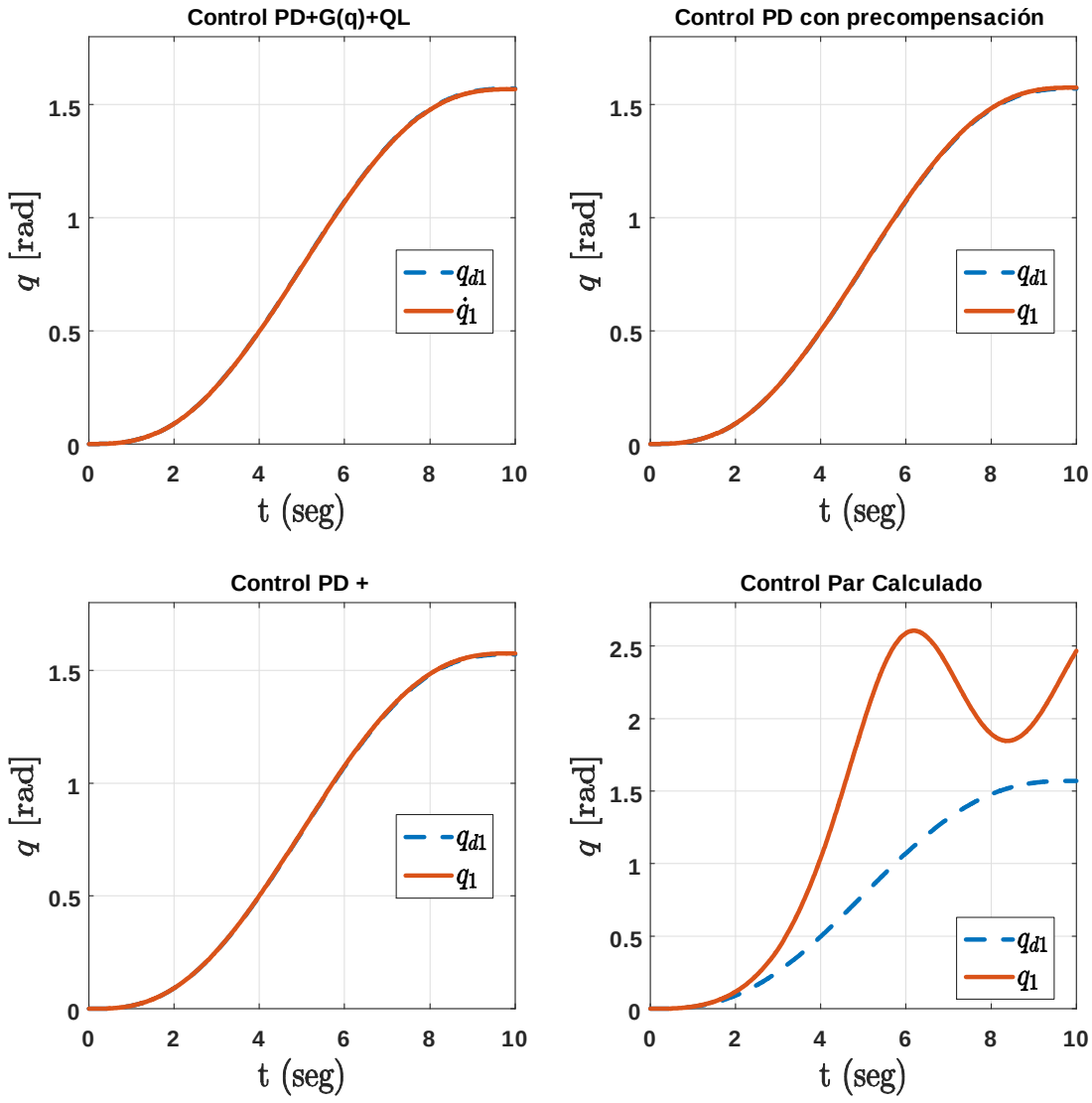


Figura 6.2: Seguimiento de trayectoria del polinomio de 5*to* grado para la posición $q_{d1}(t)$.

PD con precompensación y PD+ donde el índice de desempeño (6.5) revela que el mejor seguimiento es para el control PD+G(q)+QL, siguiendo el control PD con precompensación. El desempeño más bajo es para el control Par Calculado, esto es debido a la variación en los parámetros del modelo dinámico, que afectan directamente a la ley de control que no presentan de forma explícita el término lineal PD, por el contrario, el término lineal PD se encuentra multiplicado por la masa del sistema.

La figura (6.3) presenta el seguimiento de trayectoria para la velocidad $q_{d1}(t)$, donde se observa una trayectoria suave en forma de campana con una velocidad tanto inicial como final de 0 rad/s. El control Par Calculado muestra un pobre desempeño al presentar un comportamiento oscilatorio sobre la referencia deseada.

Índice de desempeño.

Un criterio académico ampliamente aceptado en la comunidad científica de robótica para medir el desempeño de un algoritmo de control es por medio de la obtención de desempeño determinado por la norma L_2.

En control de trayectoria el índice de desempeño se mide usando la norma L_2 del error de posición y del error de velocidad de la siguiente forma:

$$L_2\left[\tilde{q},\dot{\tilde{q}}\right] = \sqrt{\frac{1}{T}\int_0^T \left(\|\tilde{q}(t)\|^2 + \left\|\dot{\tilde{q}}(t)\right\|^2\right) dt}, \tag{6.5}$$

donde T representa el tiempo de simulación o de experimentación.

Un valor grande en el índice de desempeño, significa pobre desempeño. Alto desempeño se ve reflejado en un valor pequeño del índice L_2, donde el error de posición está dado por $\tilde{q}(t) = q_{d_1} - q_1$, y el error de velocidad por $\dot{\tilde{q}}(t) = \dot{q}_{d_1} - \dot{q}_1$.

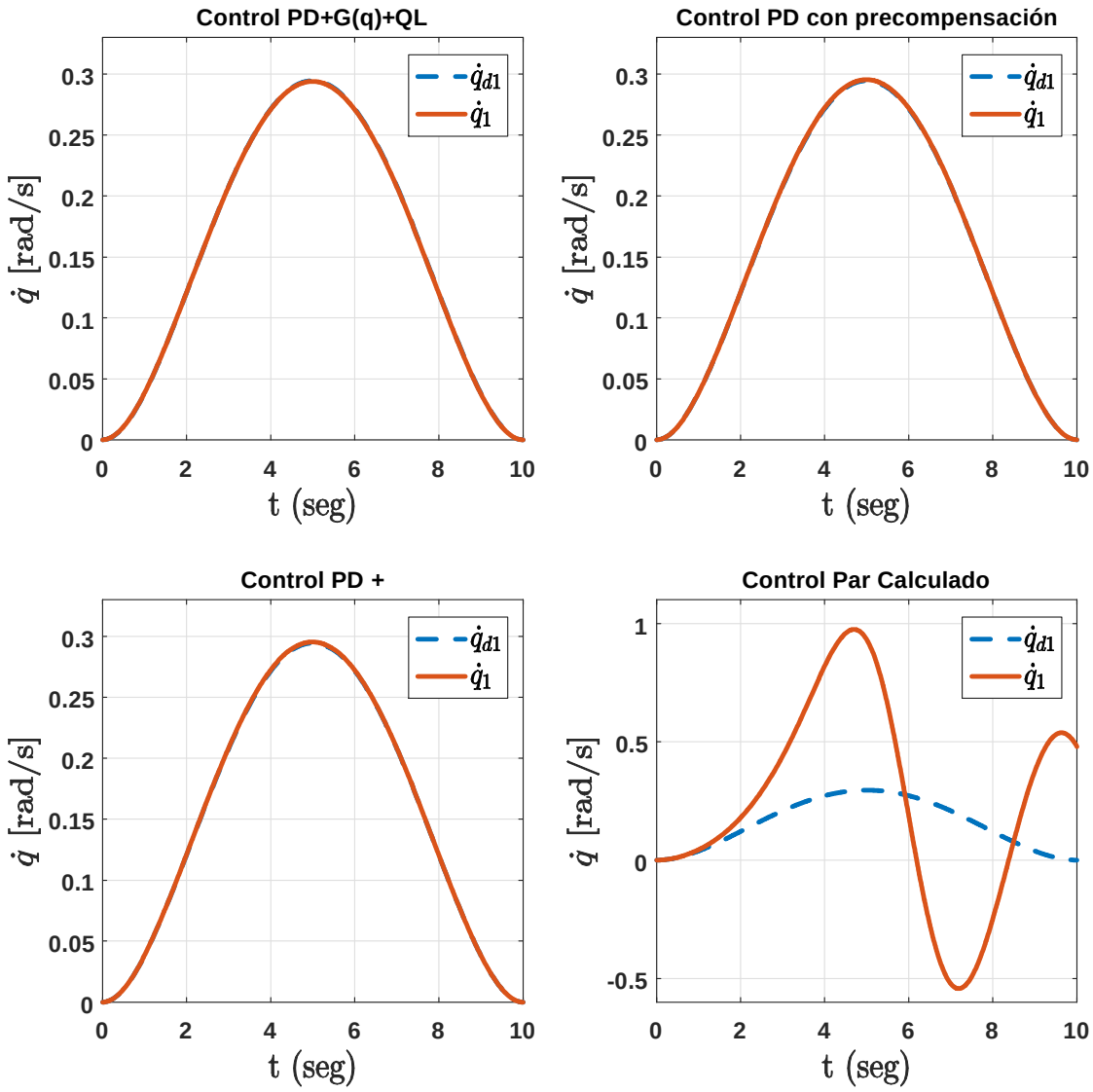


Figura 6.3: Seguimiento de trayectoria del polinomio de $5to$ grado para la velocidad $\dot{q}_{d1}(t)$.

Tabla 9. Índice de desempeño

	Norma L_2
PD+G(q)+QL	0,0020
PD con precompensación	0,0028
PD +	0,0029
Par Calculado	0,8644

Tal como se muestra en la Tabla [9], el índice de desempeño se encarga de evaluar los cuatro controladores de movimiento, arrojando la siguiente información: el controlador que tuvo el desempeño más bajo fue el control Par Calculado, esto es debido a que el controlador dentro de su algoritmo multiplica a la parte proporcional y derivativa por la matriz de masas, y si los parámetros de esta matriz difieren aunque sea un poco del valor real, produce resultados no deseados en el seguimiento de la trayectoria, provocando que su desempeño sea bastante bajo. Los resultados del control PD + y del PD con precompensación muestran resultados muy similares, puesto que los algoritmos de control sólo difieren por el uso de estados deseados en lugar de usar los estados reales del sistema. Finalmente, el algoritmo de control obtenido a través de técnicas del aprendizaje por reforzamiento ofrece mejores resultados en el seguimiento del polinomio de 5to grado para la posición, velocidad y aceleración.

Ejemplo 2) Lemniscata

Considérese el robot manipulador de 2GDL y mostrado en la figura (8.7). El modelo dinámico completo incluyendo los valores numéricos de los parámetros se describen en el apéndice A, y en la Tabla [11]. Se propone utilizar la trayectoria de la lemniscata cuyas ecuaciones vienen dadas también en el apéndice A, y las pruebas de los controladores se realizaron con una variación del $\pm 5\,\%$ en los parámetros del modelo dinámico.

Las matrices simétricas definidas positivas K_p y K_d se escogen como:

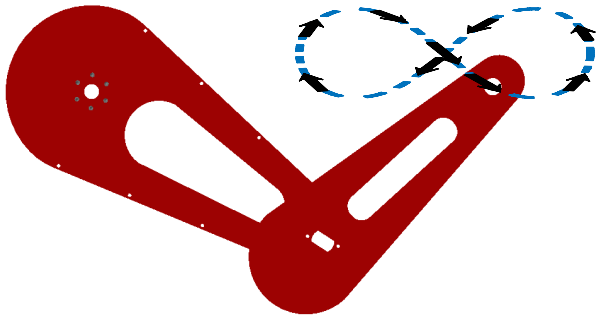


Figura 6.4: Seguimiento de la trayectoria Lemniscata.

$$K_p = \begin{bmatrix} 3 & 0 \\ 0 & 1 \end{bmatrix} [Nm/rad], \qquad K_d = \begin{bmatrix} 0{,}3 & 0 \\ 0 & 0{,}2 \end{bmatrix} [Nm\ s/rad].$$

Las condiciones iniciales correspondientes a las posiciones y velocidades se escogen nulas:

$$q_1(0) = 0, \quad q_2(0) = 0$$
$$\dot{q}_1(0) = 0, \quad \dot{q}_2(0) = 0.$$

Los resultados de la evaluación para el seguimiento de trayectoria tanto para la posición como de la velocidad sobre el ángulo de salida q, se presentan en las figuras (6.5) y (6.6), donde se grafican el ángulo q_1, q_2 y la velocidad $\dot{q}_1, \dot{q}_2$ para cada controlador.

La figura (6.5) muestra los resultados del seguimiento de trayectoria de la lemniscata en el caso de la posición. El control Par Calculado, nuevamente presenta los desempeños más deficientes al seguir la trayectoria deseada. Esto es debido a que los parámetros de la ley de control difieren en un 5 % con respecto a los parámetros del robot, lo cual le produce un mal seguimiento de trayectoria.

No obstante, los otros controladores que también fueron evaluados, presentan un buen seguimiento de trayectoria sin que les afecte tanto la variación en los parámetros del modelo

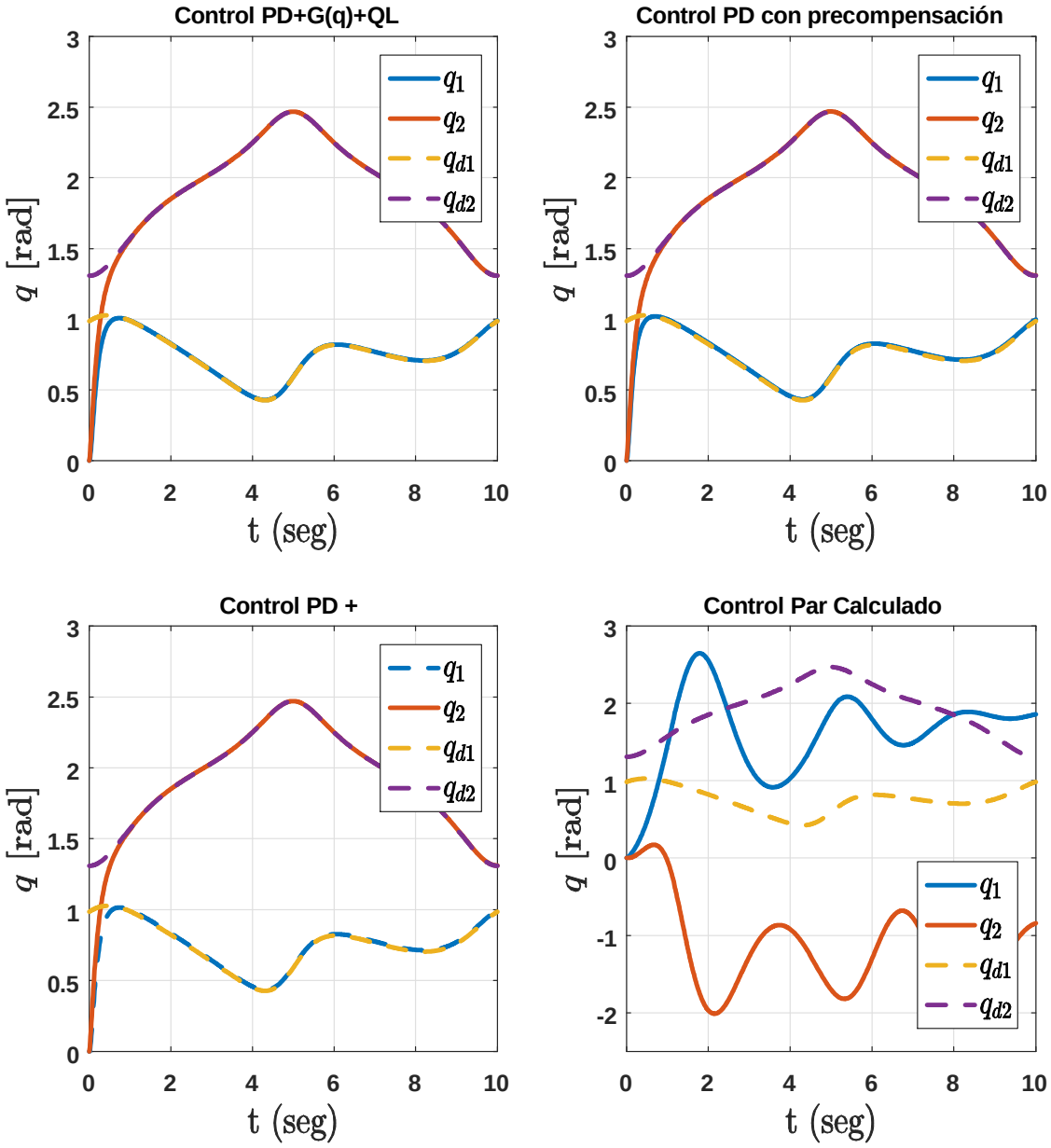


Figura 6.5: Seguimiento de trayectoria de la posición para el hombro q_1 y para el codo q_2.

que se usan en el algoritmo de control. Por último, vemos que el control PD+G(q)+QL ofrece mejores resultados en el control de movimiento, lo cual es debido al tiempo de aprendizaje que tomó el algoritmo QL en aprender la trayectoria deseada.

Los resultados del control de movimiento para la trayectoria de velocidad se presentan en la figura (6.6). Donde observamos en todos los casos un pico de velocidad al arranque, debido a que el robot empieza en su condición inicial de casa, mientras que la trayectoria comienza a 0,2 m de distancia del robot. El seguimiento de la velocidad para el control Par Calculado resulta en una respuesta oscilatoria, provocando que el robot ya no pueda regresar a la trayectoria deseada.

Índice de desempeño.

Un valor grande en el índice de desempeño, significa pobre desempeño. Alto desempeño se ve reflejado en un valor pequeño del índice L_2, donde el error de posición y velocidad está dado por:

$$\tilde{q}(t) = \begin{bmatrix} q_{d1} - q_1 \\ q_{d2} - q2 \end{bmatrix}, \qquad \dot{\tilde{q}}(t) = \begin{bmatrix} \dot{q}_{d1} - \dot{q}_1 \\ \dot{q}_{d2} - \dot{q}_2 \end{bmatrix}$$

Tabla 10. Índice de desempeño

	Norma L_2
PD+G(q)+QL	0,8464
PD precompensación	0,8660
PD +	0,8484
Par Calculado	3,6504

El índice de desempeño L2 para el seguimiento de la lemniscata se presenta en la Tabla [10]. Los resultados nos muestran que el índice más bajo es para el control PD+G(q)+QL, y el más alto para el control Par Calculado. Un índice alto ofrece un mal desempeño, por lo cual el Par Calculado es el que muestra el peor seguimiento entre todos los controladores evaluados en la prueba. La diferencia entre el resto de los controladores es mínima, exponiendo gráficas muy similares al momento de seguir la trayectoria, lo cual nos dice que el controlador propuesto tiene un buen desempeño a pesar de no conocer la dinámica del robot.

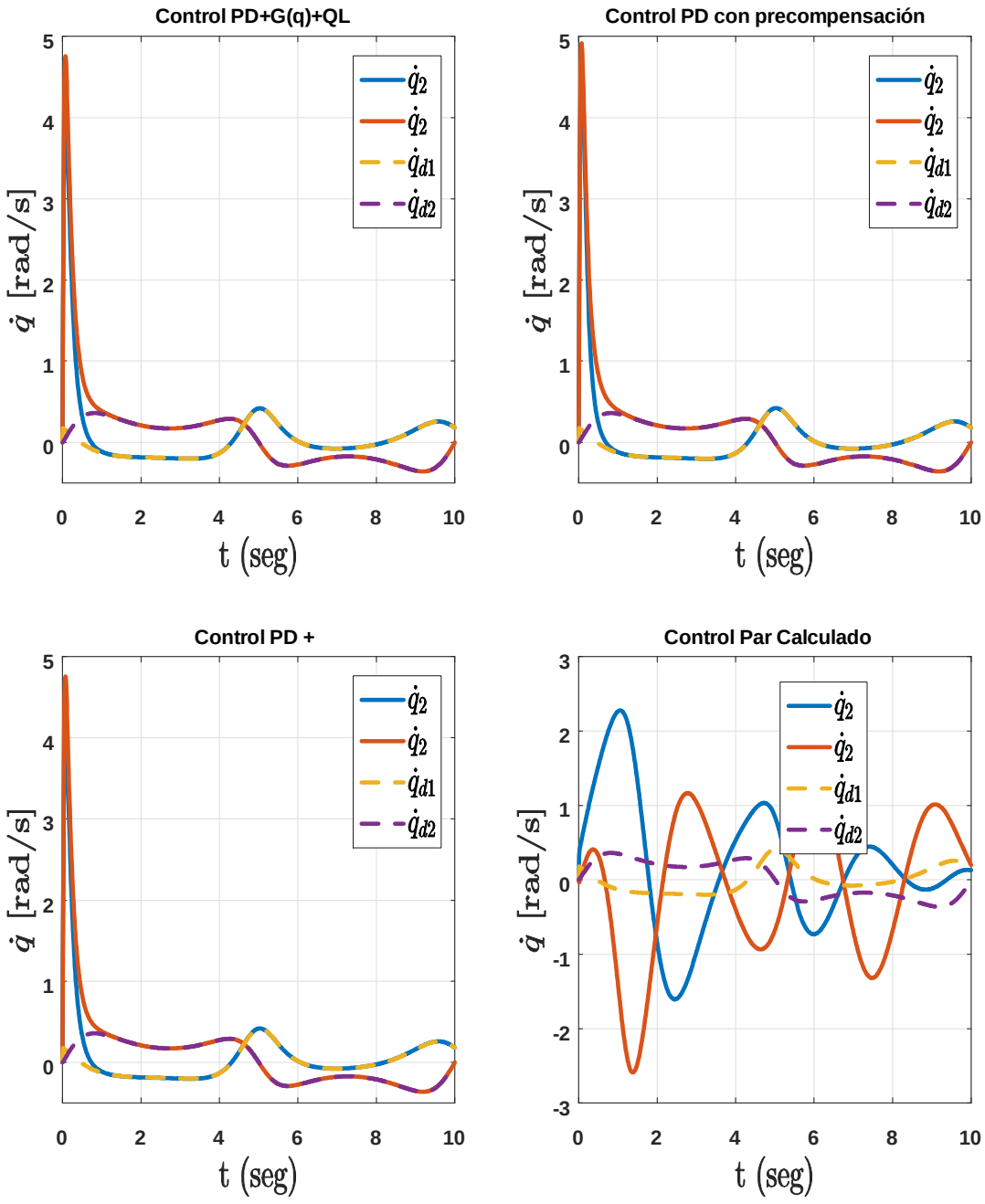


Figura 6.6: Seguimiento de trayectoria de la velocidad para el hombro $\dot{q}_1$ y para el codo $\dot{q}_2$.

Capítulo 7

Conclusiones

7.1. Conclusiones

Lo expuesto a lo largo de este trabajo permite arribar a las siguientes conclusiones:

De manera inicial, se diseñó un control PD con compensación Q-Learning basado en la teoría del aprendizaje por reforzamiento y el control clásico, y se realizaron pruebas de control en casos de estudio, tal como el doble péndulo invertido sobre el carro, el acrobot, el pendubot, etc., además, se realizó la discretización del espacio de estado y la generación de recompensas que brindaron resultados favorables al momento de simularlos. Este controlador sirvió de base para generar las primeras pruebas de estabilidad local y estabilidad asintótica.

La ventaja de este algoritmo de control es que no necesita del conocimiento del modelo dinámico del sistema a controlar, lo cual resulta de lo más favorable al momento de seleccionar un esquema de control, debido a que el control es mucho más simple al usuario, y sin necesidad de conocer los parámetros del sistema. Otra ventaja, es que este algoritmo híbrido brinda robustez ante perturbaciones generadas de forma externa, y que no fueron presentadas durante su aprendizaje, siempre y cuando la ganancia del aprendizaje por refuerzo sea mayor a la perturbación más la dinámica a compensar.

Por lo tanto, las prestaciones de este controlador son satisfactorias y la combinación del

control híbrido trabajó mejor de manera conjunta que de forma separada.

2.- En cuanto al control PID con compensación QL, se presenta una sintonización explícita de las ganancias del controlador, donde el valor máximo de la ganancia integral se da de forma explícita, y así se evitan problemas debido a valores muy grandes de la ganancia integral al momento de cancelar el error en estado estacionario. La principal contribución de este controlador es que la ganancia del aprendizaje por reforzamiento se utiliza para cancelar la dinámica del robot 2GDL, dando mejores resultados en su forma híbrida PID con compensación QL que de forma individual. Además, se presenta la prueba de que el sistema en lazo cerrado es semiglobal asintóticamente estable. Finalmente, los resultados experimentales en un entorno de simulación para sistemas mecánicos 3D muestran la efectividad del controlador.

3.- También se presenta un controlador para el seguimiento de trayectoria basado en la combinación del control PD+G(q) y el control Q-Learning. Se sabe que el control PD+G(q) hace uso explícito del término de gravedad, sin embargo, al sumar la compensación QL, este controlador híbrido sólo necesita de la trayectoria deseada y del vector de gravedad para cumplir con la tarea del seguimiento de trayectoria, esto quiere decir que la falta de conocimiento del modelo es compensada con el algoritmo del aprendizaje por reforzamiento.

4.- Finalmente, se realizó el diseño de un robot manipulador de 2 GDL en Solidworks y se exportó a Matlab por medio del complemento Simscape Multibody. Se buscó que el diseño del robot manipulador tuviera cierta simetría con el brazo humano, es decir, que fuera antropomórfico. Este prototipo sirvió de base para implementar las leyes de control PD, PID y PD+G(q) con compensación QL.

7.2. Trabajo a futuro.

Las perspectivas de este trabajo pueden ser, la implementación de algoritmos de aprendizaje por reforzamiento profundo (Deep-Learning) para lograr tareas más complejas, donde ya no se utilizarán tablas o matrices que pueden resultar inefectivas en ambientes con espacios de estados muy grandes, en vez de eso implementar una red neuronal que reciba los

estados de la planta y estime la función Q basado en cada estado. Esto nos dará la oportunidad de resolver problemas del mundo real aplicando algoritmos avanzados del aprendizaje por reforzamiento.

Encontrar una discretización del espacio de estado más precisa para el seguimiento de trayectoria brindaría un menor tiempo de aprendizaje, lo cual nos lleva a pensar que se puede expandir el trabajo en ese tópico. Además, la generación de recompensas en función no sólo del error de seguimiento, sino también del error de velocidad y/o de aceleración se propone como trabajo futuro para conseguir un seguimiento de trayectoria que no necesite del modelo parcial del robot ni de las ganancias proporcional y derivativa.

También, como trabajo futuro se busca aplicar estos algoritmos de control en la robótica móvil, para el control de posición como el estacionado, y el seguimiento de trayectoria en un esquema líder seguidor.

Finalmente, al momento del aprendizaje, la matriz Q-Learning no siempre se usa por completo en todo su espacio de trabajo, con lo cual se puede reutilizar este espacio y no generar otra matriz Q cuando el robot se encuentre pasando por la misma posición debido a la trayectoria impuesta.

Capítulo 8

Apéndice

8.1. A.-Robot de 2 grados de libertad

En este apéndice se resumen varios tópicos sobre el robot de 2 g.d.l. que se ha usado como robot prototipo en diversos capítulos de la tesis. Específicamente, se abordan los siguientes temas:

Modelo dinámico.

Modelo cinemático directo.

Modelo cinemático inverso.

El robot considerado puede verse en la figura (8.1). Este consiste de 2 eslabones conectados a través de articulaciones rotacionales. El significado de las diversas constantes así como el de sus valores numéricos están resumidos en la Tabla [11] que muestra los parámetros del robot manipulador como la masa, longitud, centro de masa, inercia, fricción viscosa, etc. Además, para el brazo y el antebrazo se observa que las dimensiones del brazo son mayores que las del antebrazo, esto debido a la relación antropomórfica que se buscó. Para obtener

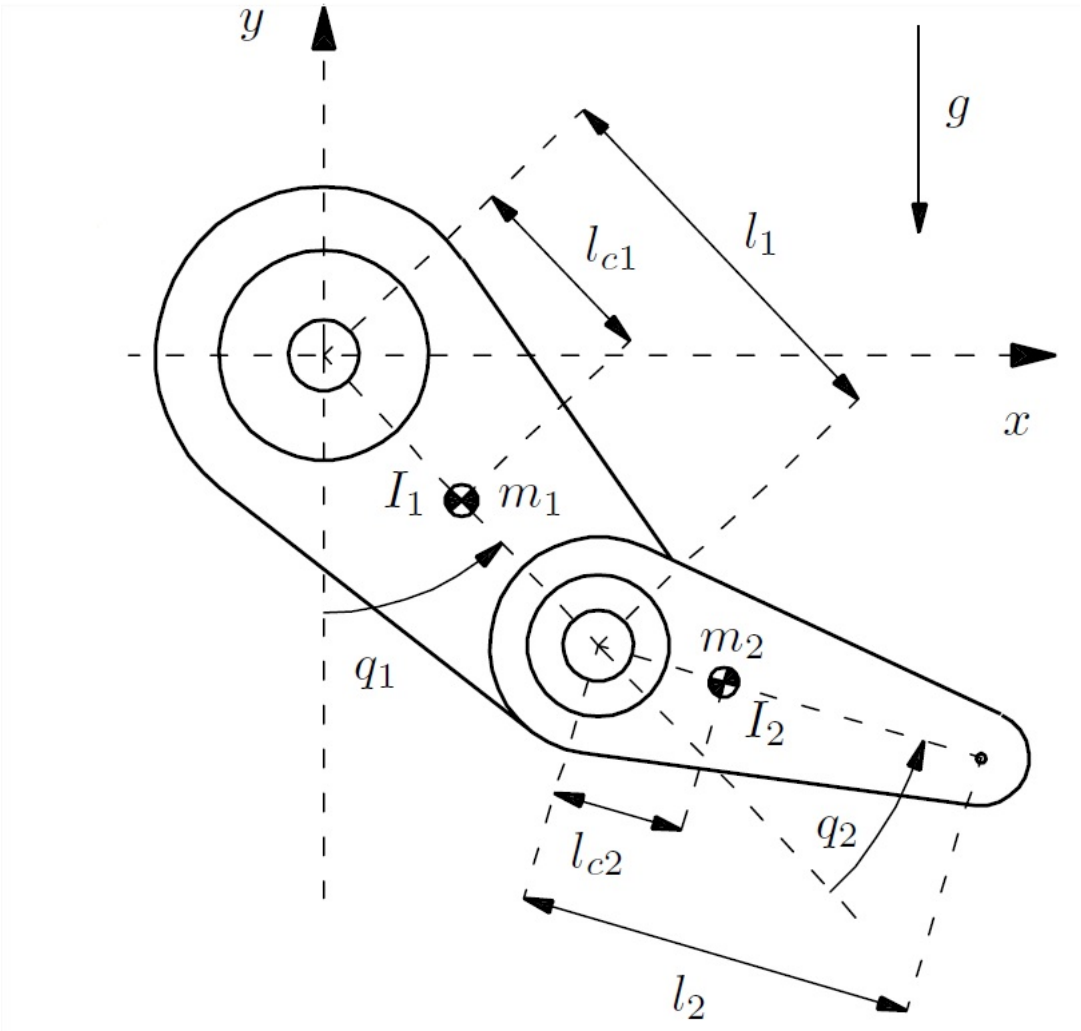


Figura 8.1: Robot prototipo 2 g.d.l.

los valores numéricos se hizo uso de programas como Solidworks y Matlab.

Tabla 11. Párametros del Robot Mamipulador de 2GDL.

	Parámetros	Descripción	Valor
	m_1	Masa del Brazo	0,2393 kg
	l_1	Longitud del Brazo	0,240 m
(HOMBRO)	l_{c1}	Centro de masa del Brazo	0,0684 m
	I_1	Inercia del Brazo	0,002547 kgm^2
	b_1	Coeficiente de fricción viscosa en el hombro	0,0017$\frac{Nm}{rad/s}$
	q_1	Posición articular del Brazo	q_1 rad
	m_2	Masa del Antebrazo	0,1541 kg
	l_2	Longitud del Antebrazo	0,200 m
	l_{c2}	Centro de masa del Antebrazo	0,0574 m
(CODO)	I_2	Inercia del Antebrazo	0,001153 kgm^2
	b_2	Coeficiente de fricción viscosa en el codo	0,0013$\frac{Nm}{rad/s}$
	q_2	Posición articular del Antebrazo	q_2 rad
	g	Acción debida a la gravedad	9,81 m/s^2

8.1.1. Modelo dinámico

El robot manipulador de 2 g.d.l. mostrado en la figura (8.1) fue objeto de estudio detallado en el tema 3.5.2 Robot Planar. Se obtuvo su modelo dinámico a partir del empleo de las ecuaciones de movimiento de Euler-Lagrange. Por lo tanto, el modelo dinámico puede ser expresado de la siguiente forma:

$$\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix} \ddot{q} + \begin{bmatrix} C_{11}(q,\dot{q}) & C_{12}(q,\dot{q}) \\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q}) \end{bmatrix} \dot{q} + \begin{bmatrix} B_{11} & 0 \\ 0 & B_{22} \end{bmatrix} \dot{q} + \begin{bmatrix} G_{11}(q) \\ G_{21}(q) \end{bmatrix} = \tau, \quad (8.1)$$

donde,

Matriz de masas

$$
\begin{aligned}
M_{11}(q) &= m_1 l_{c1}^2 + m_2 \left[l_1^2 + l_{c2}^2 + 2 l_1 l_{c2} \cos(q_2) \right] + I_1 + I_2, \\
M_{12}(q) &= m_2 \left[l_{c2}^2 + l_1 l_{c2} \cos(q_2) \right] + I_2, \\
M_{21}(q) &= m_2 \left[l_{c2}^2 + l_1 l_{c2} \cos(q_2) \right] + I_2, \\
M_{22}(q) &= m_2 l_{c2}^2 + I_2.
\end{aligned}
$$

Matriz de coriolis

$$
\begin{aligned}
C_{11}(q, \dot{q}) &= -m_2 l_1 l_{c2} \sin(q_2) \dot{q}_2, \\
C_{11}(q, \dot{q}) &= -m_2 l_1 l_{c2} \sin(q_2) [\dot{q}_1 + \dot{q}_2], \\
C_{21}(q, \dot{q}) &= m_2 l_1 l_{c2} \sin(q_2) \dot{q}_1, \\
C2(q, \dot{q}) &= 0.
\end{aligned}
$$

Matriz de fricción

$$
\begin{aligned}
B_{11} &= b_1, \\
B_{12} &= 0, \\
B_{21} &= 0, \\
B_{22} &= b_2.
\end{aligned}
$$

Vector de gravedad

$$
\begin{aligned}
G_{11} &= \left[m_1 l_{c1} + m_2 l_1 \right] g \sin(q_1) + m_2 l_{c2} g \sin(q_1 + q_2), \\
G_{21} &= m_2 l_{c2} g \sin(q_1 + q_2).
\end{aligned}
$$

Las variables de estado adecuadas para describir el modelo dinámico del robot son las posiciones q_1 y q_2 y las velocidades $\dot{q}_1$ y $\dot{q}_2$.

8.1.2. Modelo cinemático directo

El modelado cinemático directo de robots manipuladores se plantea en los siguientes términos. Considérese un robot manipulador de n g.d.l. colocado en una superficie fija. Defínase un marco de referencia también fijo en algún lugar de la superficie. Dicho marco de referencia suele denominarse marco referencial de base. El problema de la determinación

del modelo cinemático directo del robot consiste en expresar la posición y orientación de un marco de referencia sólidamente colocado en la parte terminal del último eslabón del robot referida al marco referencial de base en términos de las coordenadas articulares del robot.

Con relación al robot de 2 g.d.l., defínase primeramente el marco referencial de base como un sistema cartesiano de 2 dimensiones cuyo origen se localiza exactamente en la primera articulación del robot, tal y como se muestra en la figura (8.1). Las coordenadas cartesianas x e y denotan la posición del extremo final del segundo eslabón con respecto al marco referencial de base. Para este caso sencillo de 2 g.d.l., la orientación del extremo final del robot carece de sentido. Por supuesto, se aprecia claramente que ambas coordenadas cartesianas x e y dependen de las coordenadas q_1 y q_2. La relación entre ellas define al modelo cinemático directo propiamente dicho:

$$\begin{bmatrix} x \\ y \end{bmatrix} = f(q_1, q_2),$$

donde $f : \mathbb{R}^2 \to \mathbb{R}^2$.

Para el caso del robot de 2 g.d.l., es inmediato verificar que el modelo cinemático directo viene dado por:

$$\begin{aligned} x &= l_1 \sin(q_1) + l_2 \sin(q_1 + q_2), \\ y &= -l_1 \cos(q_1) - l_2 \cos(q_1 + q_2). \end{aligned}$$

8.1.3. Modelo cinemático inverso

El modelo cinemático inverso de robots manipuladores resulta de gran importancia desde un punto de vista práctico. Dicho modelo permite obtener las posiciones articulares q en términos de la posición y orientación del extremo final del último eslabón referido al marco referencial cartesiano de base. Para el caso del robot de 2 g.d.l., el modelo cinemático inverso tiene la forma:

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = f^{-1}(x, y),$$

donde $f^{-1} : \mathbb{R}^2 \to \mathbb{R}^2$. La obtención del modelo cinemático inverso resulta ser, en general laborioso.

El interés práctico del modelo cinemático inverso consiste en su empleo para obtener las especificaciones de posiciones articulares deseadas $q_d = \begin{bmatrix} q_{d1}, & q_{d2} \end{bmatrix}^T$ a partir de especificaciones de posición deseada x_d e y_d del extremo final del último eslabón del robot.

A partir de esta información pueden obtenerse las posiciones, velocidades y aceleraciones articulares deseadas:

$$q_{d1} = \tan^{-1}\left(\frac{x_d}{-y_d}\right) - \tan^{-1}\left(\frac{l_2 \sin(q_{d2})}{l_1 + l_2 \cos(q_{d2})}\right),$$

$$q_{d2} = \tan^{-1}\left(\frac{\pm\sqrt{(1-\cos(q_2)^2)}}{\cos(q_2)}\right),$$

$$\cos(q_2) = \left(\frac{x_d^2 + y_d^2 - l_1^2 - l_2^2}{2 l_1 l_2}\right).$$

8.2. B.-Trayectoria deseada

En este apéndice nos enfocaremos en los métodos para calcular una trayectoria que describa el movimiento deseado de un manipulador. Aquí trayectoria se refiere a un historial en el tiempo de la posición, la velocidad y la aceleración para cada grado de libertad.

8.2.1. Polinomio 5to grado

Con el propósito de evaluar mediante simulaciones numéricas la prestación de los controladores de movimiento descritos anteriormente, se ha elegido la siguiente trayectoria de movimiento articular. Si deseamos especificar la posición, la velocidad y la aceleración al inicio y al final de cada segmento de ruta, se requiere un polinomio de quinto grado, a saber,

$$q_d(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5,$$

en donde las restricciones se dan así:

$$\begin{aligned}
q_{d0} &= a_0, \\
q_{df} &= a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5, \\
\dot{q}_{d0} &= a_1, \\
\dot{q}_{df} &= a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4, \\
\ddot{q}_{d0} &= 2a_2, \\
\ddot{q}_{df} &= 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3.
\end{aligned}$$

Estas restricciones especifican un conjunto lineal de seis ecuaciones con seis variables desconocidas cuya solución es:

$$\begin{aligned}
a_0 &= q_{d0} \\
a_1 &= \dot{q}_{d0} \\
a_2 &= \tfrac{\ddot{q}_{d0}}{2} \\
a_3 &= \tfrac{20q_{df} - 20q_{d0} - \left(8\dot{q}_{df} + 12\dot{q}_{d0}\right)t_f - \left(3\ddot{q}_{d0} + \ddot{q}_{df}\right)t_f^2}{2t_f^3} \\
a_4 &= \tfrac{30q_{d0} - 30q_{df} + \left(14\dot{q}_{df} + 16\dot{q}_{d0}\right)t_f + \left(3\ddot{q}_{d0} - 2\ddot{q}_{df}\right)t_f^2}{2t_f^4} \\
a_5 &= \tfrac{12q_{df} - 12q_{d0} - \left(6\dot{q}_{df} + 6\dot{q}_{d0}\right)t_f - \left(\ddot{q}_{d0} - \ddot{q}_{df}\right)t_f^2}{2t_f^5}.
\end{aligned}$$

Finalmente, tenemos las siguientes ecuaciones:

$$\begin{aligned}
q_d(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5, \\
\dot{q}_d(t) &= a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4, \\
\ddot{q}_d(t) &= 2a_2 t + 6a_3 t + 12a_4 t^2 + 20a_5 t^3.
\end{aligned} \tag{8.2}$$

8.2.2. Lemniscata

La lemniscata, conocida comúnmente como el símbolo del infinito, es uno de los símbolos matemáticos, que todos hemos visto alguna vez o escuchado. Este símbolo es parecido a un ocho en forma horizontal y fue descrito por primera vez hace más de 300 años por Jakob Bernoulli.

La representación gráfica de esta ecuación genera una curva similar a (∞). La curva se ha convertido en el símbolo del infinito y es ampliamente utilizada en matemática. En matemática, una lemniscata es un tipo de curva descrita por la siguiente ecuación en coordenadas paramétricas:

$$\begin{aligned} x_d &= h_x + \tfrac{a\cos(\omega t)}{1+\sin(\omega t)^2} \\ y_d &= h_y + \tfrac{a\sin(\omega t)\cos(\omega t)}{1+\sin(\omega t)^2} \end{aligned} ,$$

donde los parámetros de diseño se presentan en la Tabla [12]

Tabla 12. Lemniscata

ω	$2\pi f\ rad/s$
f	$1/T$
T	$10\ seg$
a	$0{,}15\ m$
h_x	$0{,}2\ m$
h_y	$0\ m$

8.3. C.-Diseño del robot de 2 grados de libertad

8.3.1. Péndulo Robot

Diseño Solidworks-Simscape Multibody

En esta sección se muestra el diseño de un péndulo robot realizado en Solidworks y exportado a Matlab por medio del complemento Simscape Multibody. Una vez que un mecanismo se traduce en un modelo SimMechanics, podemos interactuar con Simulink para realizar una amplia gama de tareas de análisis o diseño que no están disponibles en la mayoría del software de diseño asistido por computadora (CAD).

Simscape Multibody (anteriormente SimMechanics) proporciona un entorno de simulación multicuerpo para sistemas mecánicos 3D, como robots, suspensiones de vehículos, maquinaria de construcción y trenes de aterrizaje de aeronaves. Se pueden modelar sistemas multicuerpo utilizando bloques que representan cuerpos, articulaciones, restricciones, elementos de fuerza y sensores. Simscape Multibody formula y resuelve las ecuaciones de movimiento de todo el sistema mecánico. Puede importar en el modelo montajes CAD completos, incluidas todas las masas, inercias, articulaciones, restricciones y geometría 3D. Una animación 3D generada automáticamente permite visualizar la dinámica del sistema.

Simscape Multibody ayuda a desarrollar sistemas de control y a probar el rendimiento a nivel de sistema. Permite parametrizar modelos utilizando variables y expresiones de MATLAB®, así como diseñar sistemas de control para su sistema multicuerpo en Simulink®. Es posible integrar los sistemas eléctricos, hidráulicos, neumáticos y otros tipos de sistemas físicos en el modelo mediante los componentes de la familia de productos Simscape™. Para desplegar los modelos en otros entornos de simulación, como sistemas de tipo hardware-in-the-loop (HIL), Simscape Multibody soporta la generación de código C.

En las figuras (8.2) y (8.4) se muestra una vista isométrica y frontal del diseño realizado en Solidworks del péndulo robot, este robot fue exportado a Matlab por medio del complemento Simscape Multibody, lo cual nos brinda toda la información completa del péndulo robot, como lo es: longitudes, masas, inercias, centros de mas, etc.

Figura 8.2: Vista Isométrica lado izquierdo del Péndulo robot

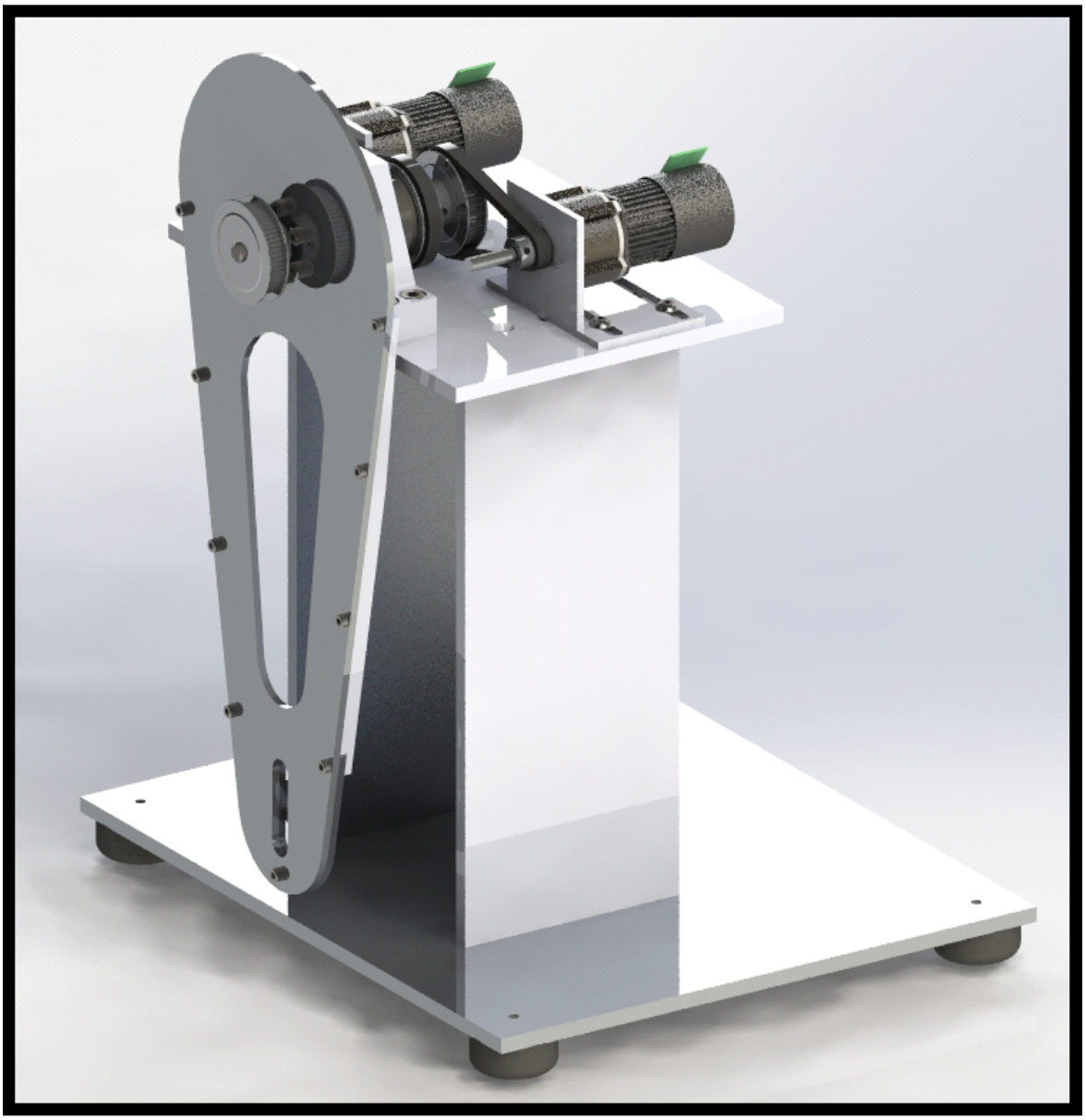


Figura 8.3: Vista Isométrica lado derecho del Péndulo robot.

El beneficio de traer todas las propiedades físicas del robot a Matlab, resultan de gran interés, debido a que nuestro algoritmo de control es evaluado en un sistema físico, que asemeja muy cercanamente a un experimento real.

El brazo tiene como material aluminio con una aleación 6061, y con un peso aproximado de 250 gramos y una longitud total aproximada de 35 cm, y además cuenta con una transmisión por medio de poleas y bandas conectadas directamente a motores de corriente continua Faulhaber. En la Tabla [13] se muestran las características del péndulo robot.

Tabla 13. Parámetros del Péndulo Robot

Parámetros	Descripción	Valor
m_1	Masa del Brazo	0,2393 kg
l_1	Longitud del Brazo	0,240 m
l_{c1}	Centro de masa del Brazo	0,0684 m
I_1	Inercia	0,002547 kgm^2
b_1	Coeficiente de fricción viscosa	0,0017$\frac{Nm}{rad/s}$
q_1	Posición articular del Brazo	q_1 rad
g	Acción debida a la gravedad	9,81 m/s^2

El tamaño total del péndulo robot es de aproximadamente 50 cm y el peso aproximado es de 4.5 kg, que incluye los motores de CD, la base de los motores, la plancha sobre la que descansan los motores, las piernas de la estructura, la plancha base sobre la que descansa el robot, poleas, bandas, tornillos, tuercas, gomas, etc. La vista frontal junto con la vista isométrica, las estaremos utilizando para presentar nuestros resultados en el control de posición. Estas dos vistas nos ayudan a percibir si se alcanza el punto final deseado.

8.3.2. Robot Manipulador.

Diseño Solidworks-Simscape Multibody

En esta sección se muestra el diseño de un robot manipulador de 2 GDL realizado en Solidworks y exportado a Matlab por medio del complemento Simscape Multibody. Una vez

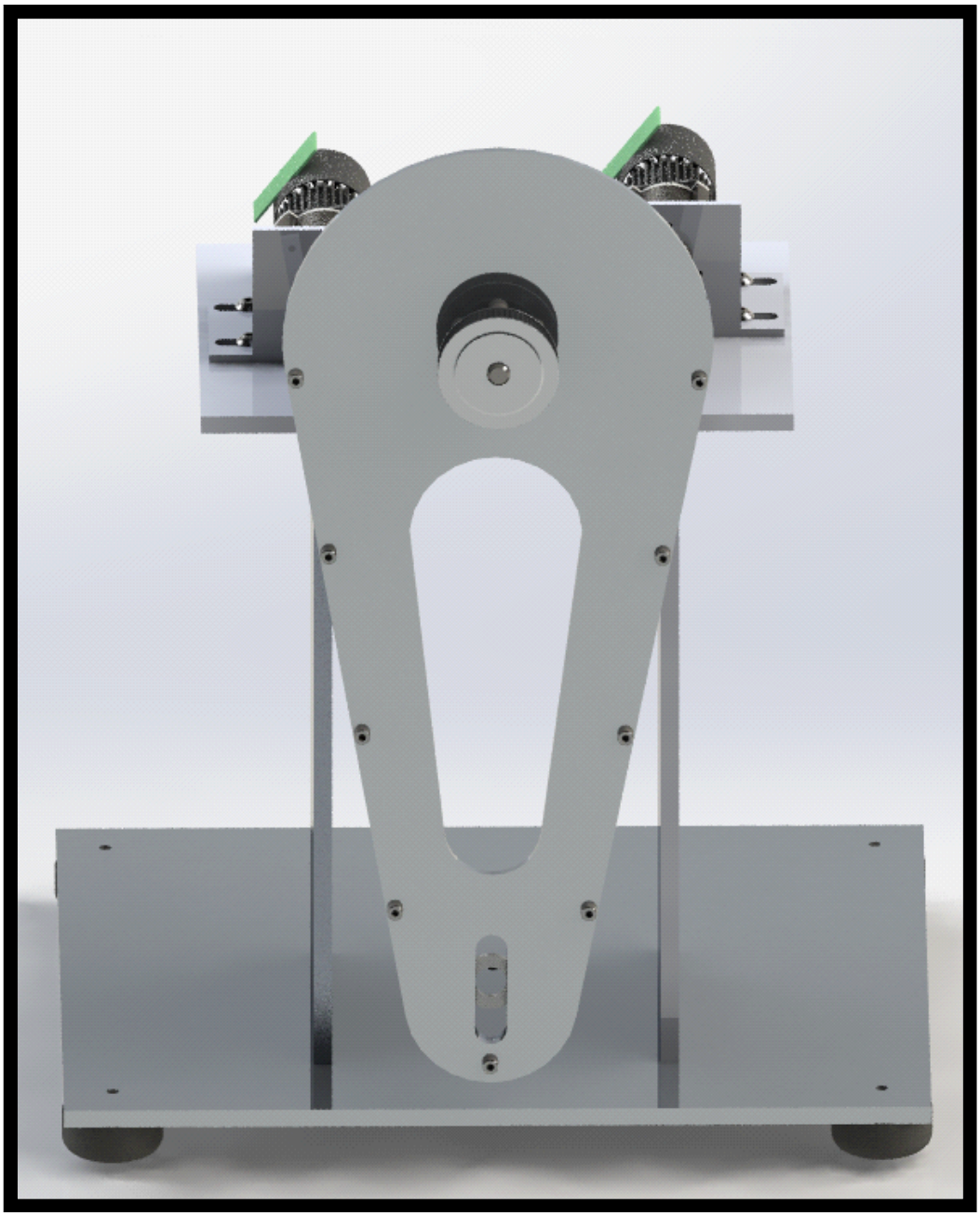
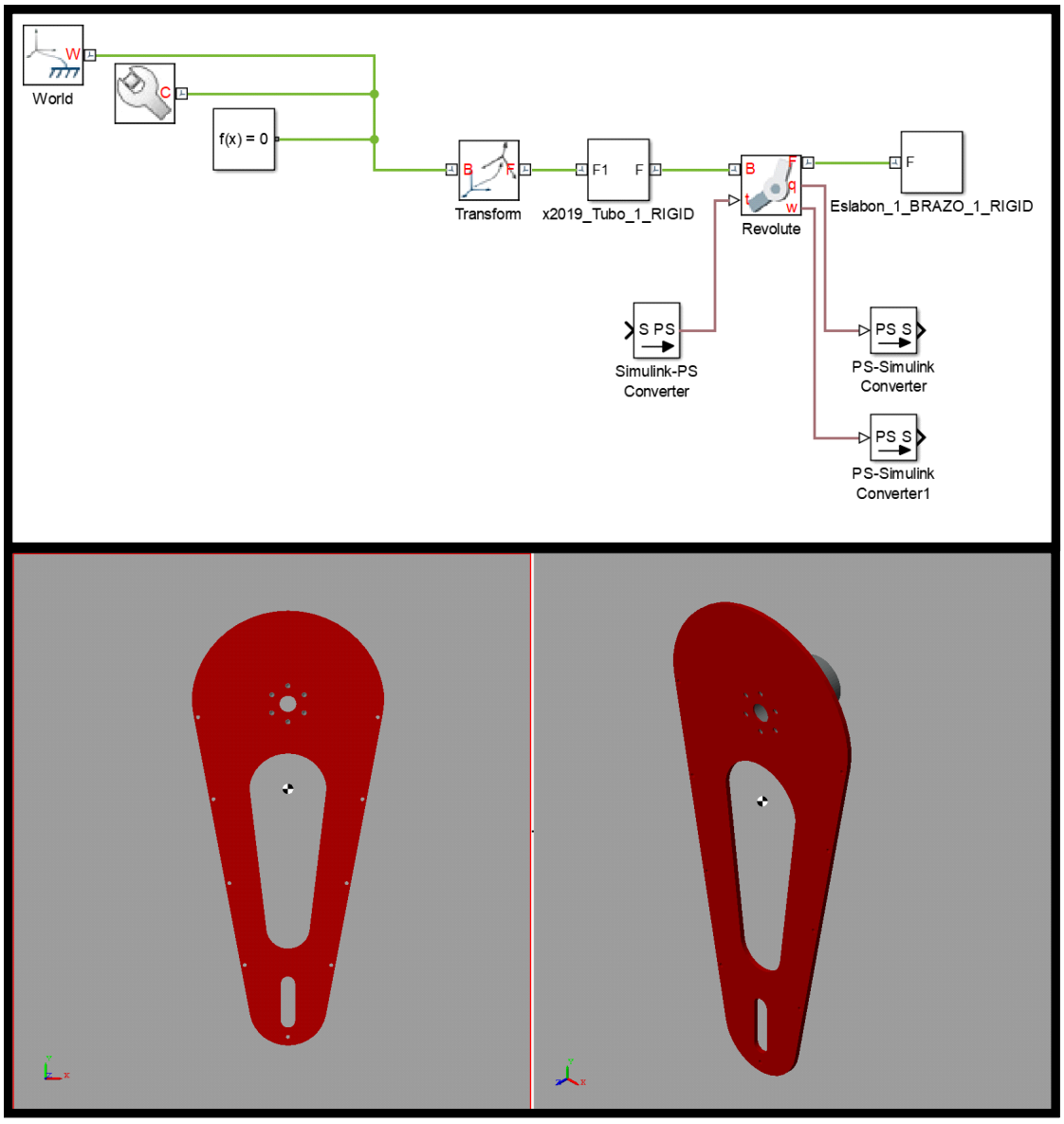


Figura 8.4: Vista frontal del Péndulo robot.



(a) Vista Frontal

(b) Vista Isométrica

Figura 8.5: En la parte superior vemos el diagrama de bloques generado por Matlab, y en la parte inferior el péndulo robot desde el ambiente de Simscape.

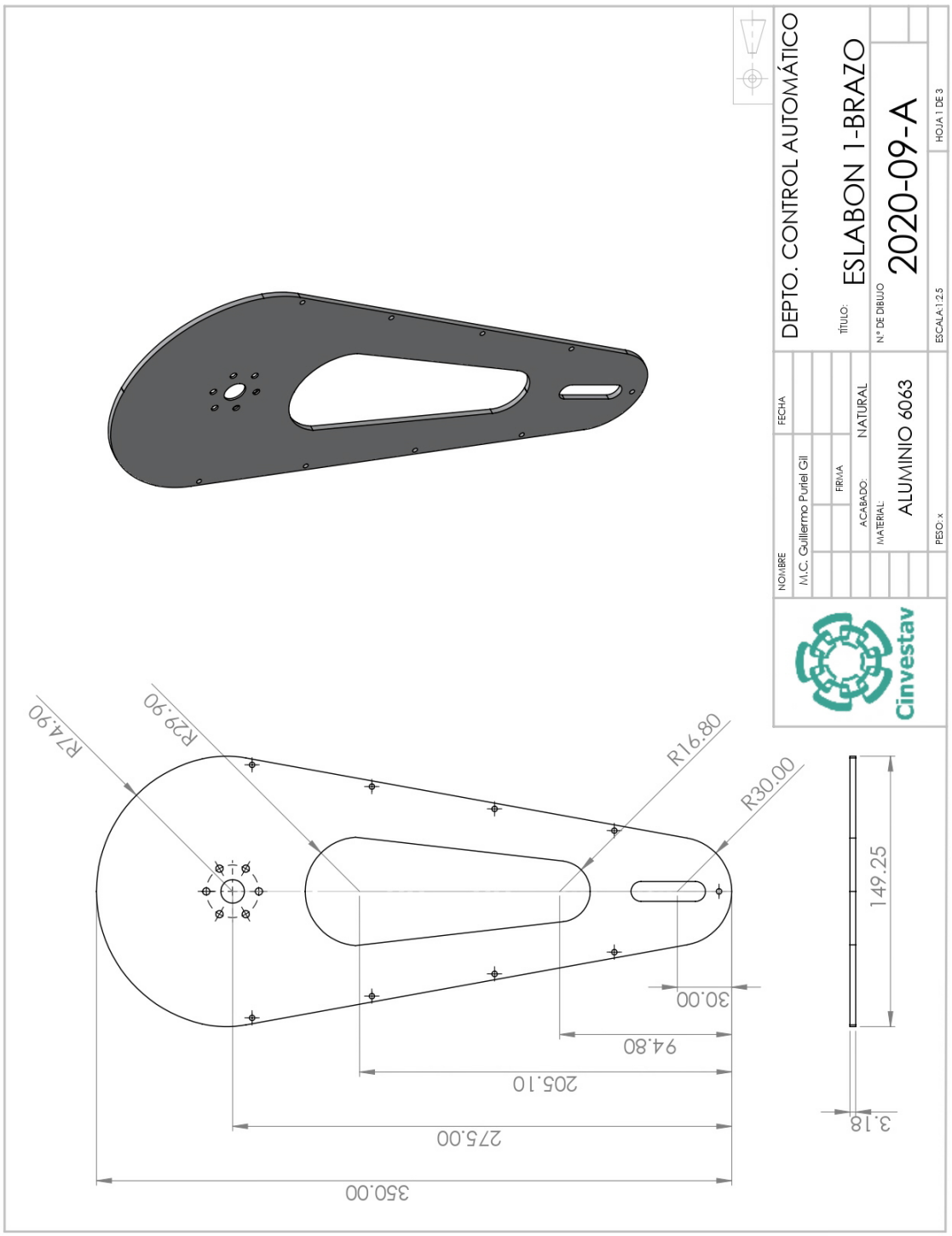


Figura 8.6: Eslabón 1 Brazo. Vsita Frontal (A)

que un mecanismo se traduce en un modelo SimMechanics, podemos interactuar con Simulink para realizar una amplia gama de tareas de análisis o diseño que no están disponibles en la mayoría del software CAD. Se hizo el diseño del robot manipulador mediante la realización de un brazo y un antebrazo que tuviera una cierta simetría con el brazo humano, es decir, que fuera antropomórfico, además, se buscó que los pesos y dimensiones de los eslabones estuvieran dentro del rango de torque máximo de los motores seleccionados. Los motorreductores que se encargan de realizar la tarea de mover al brazo y al antebrazo son motores de corriente continua con encoder y caja reductora con una relación 64:1 de la marca Faulhaber modelo 2342L012CR. Este es un motor de precisión de 12 VDC de Alemania que ofrecen un potente par y una marcha suave y lenta. Estos pueden funcionar bien en escalas más grandes o en proyectos de robótica también, donde no es requerida una alta velocidad pero si un constante funcionamiento, como máquinas de producción.

La figura (8.7) muestra el robot manipulador desde una vista isométrica en su condición inicial de casa, donde tanto el brazo como el antebrazo se encuentran extendidos y descansando en su posición de equilibrio estable. La posición de casa (equilibrio estable) será la posición donde el manipulador iniciará para llegar a su condición final deseada.

La figura (8.8) también es una vista isométrica, pero el manipulador ya no se encuentra en la posición de casa, sino que, por el contrario, tanto el brazo como el antebrazo se encuentran extendidos con ángulos diferentes de cero. El espacio de trabajo del robot manipulador comprende un círculo con un diámetro aproximado de 1 metro. Con lo cual, si se necesitara algún punto deseado en el plano x-y o en condiciones paramétricas, se puede proponer cualquier condición deseada que se encuentre dentro del área de trabajo del robot.

La Tabla [11] muestra los parámetros del robot manipulador, donde se puede apreciar la masa, longitud, centro de masa, inercia, fricción viscosa, etc. Para el brazo y el antebrazo, donde se observa que las dimensiones del brazo son mayores que las del antebrazo, esto debido a la relación antropomórfica que se buscaba. Para conseguir los parámetros se hizo uso de programas como Solidworks y Matlab, donde en el primero, arrojó los parámetros, y en el segundo los validamos mediante una simulación.

La figura (8.9) muestra una vista superior del robot manipulador, donde se muestran los

Figura 8.7: Robot Manipulador de 2GDL

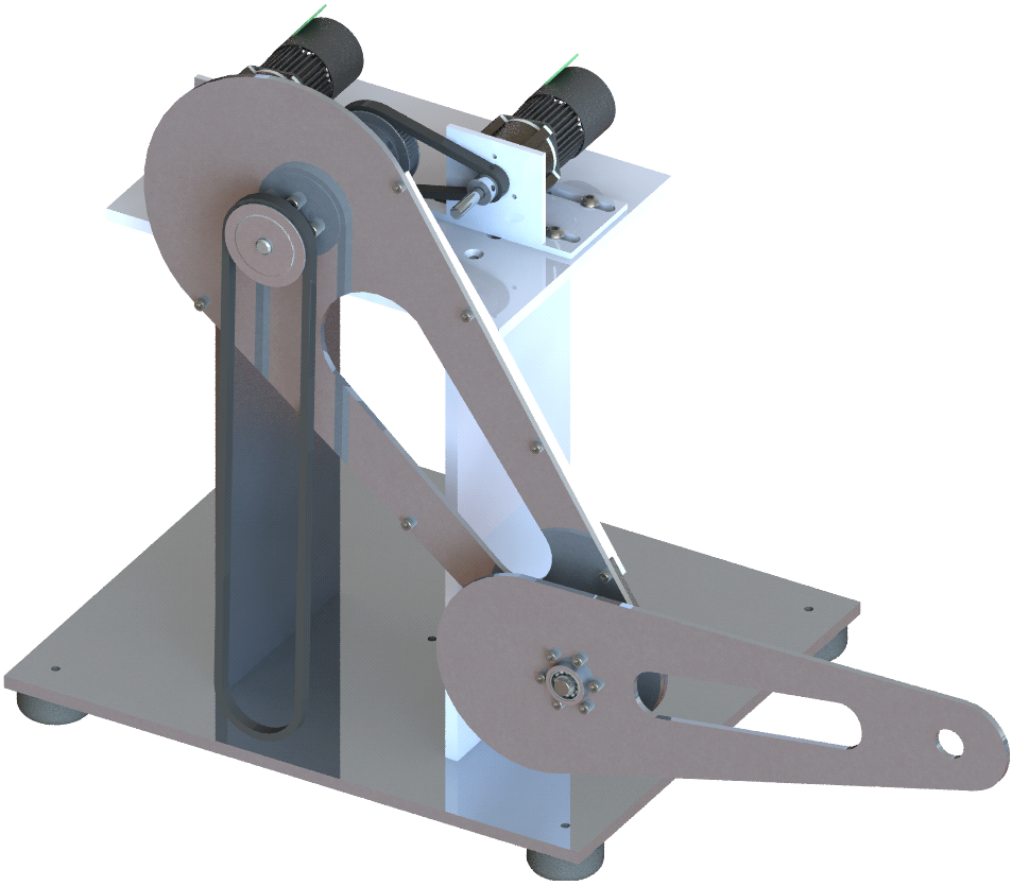


Figura 8.8: Robot Manipualdor vista lateral

motores Faulhaber 2342L012CR descansando en una plancha de aluminio que además alberga todo el sistema de transmisión de potencia para mover tanto el brazo como el antebrazo. La transmisión de potencia se realiza a través de ejes, baleros, poleas dentadas y bandas que se encuentran conectadas en una configuración de rombo con los motores, buscando simpleza al montarlo o desmontarlo, y un alto desempeño en su funcionamiento.

El diagrama a bloques generado por Matlab se muestra en la figura (8.10), donde en la parte superior observamos los bloques generados por Simscape Simulink, como las articulaciones rotacionales del hombro y del codo, los sensores de posición y velocidad, las entradas de torque, las piezas diseñadas del brazo, antebrazo, y finalmente, los bloques de control. En la parte inferior se muestran los eslabones del robot en una vista frontal descansando en su posición de casa lado (a), y en el lado (b) se presenta el robot alcanzando su condición final deseada ($q_{d1} = \pi/3$, y $q_{d2} = \pi/6$). Estos eslabones son generados a través de Simulink-Matlab una vez que se exportó el ensamble desde Solidworks. El color rojo se colocó para hacer notar más las piezas, debido al fondo gris, y sólo cumple un objetivo estético y de presentación.

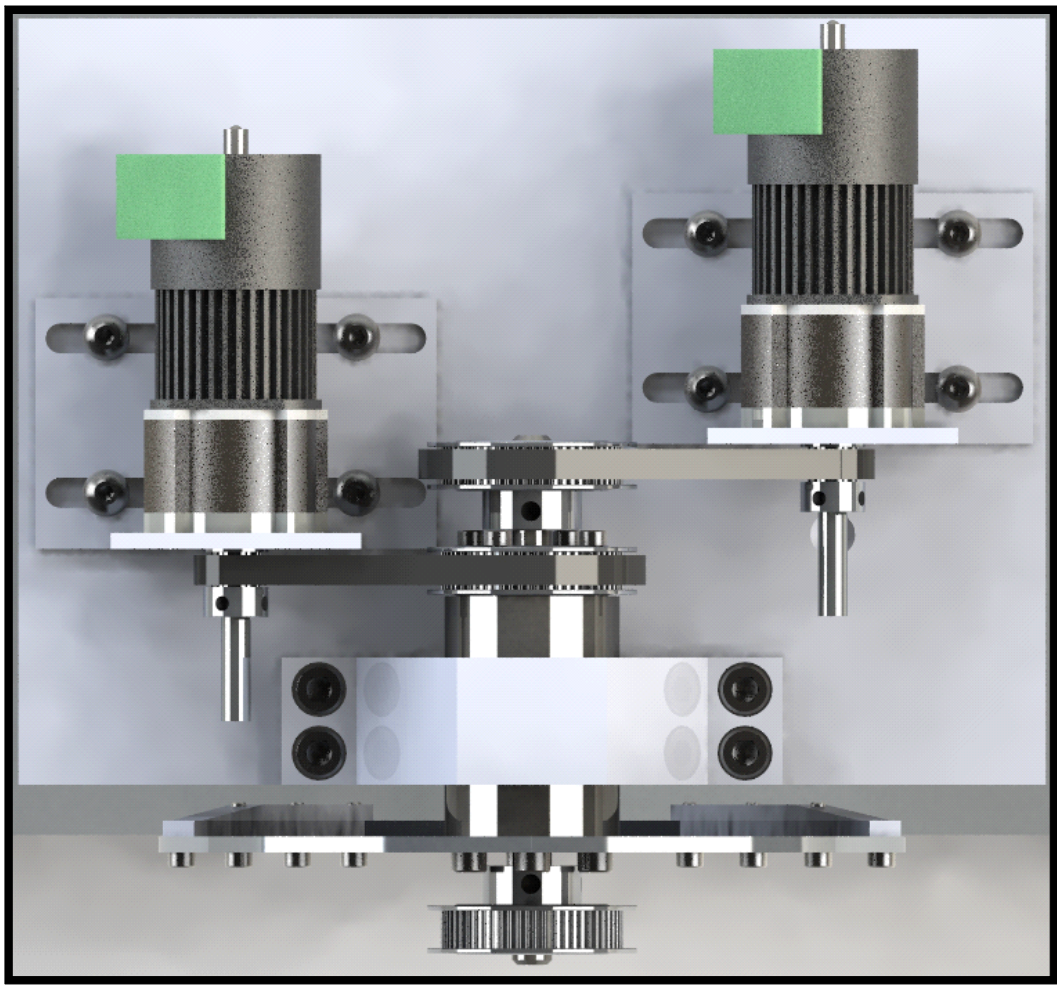
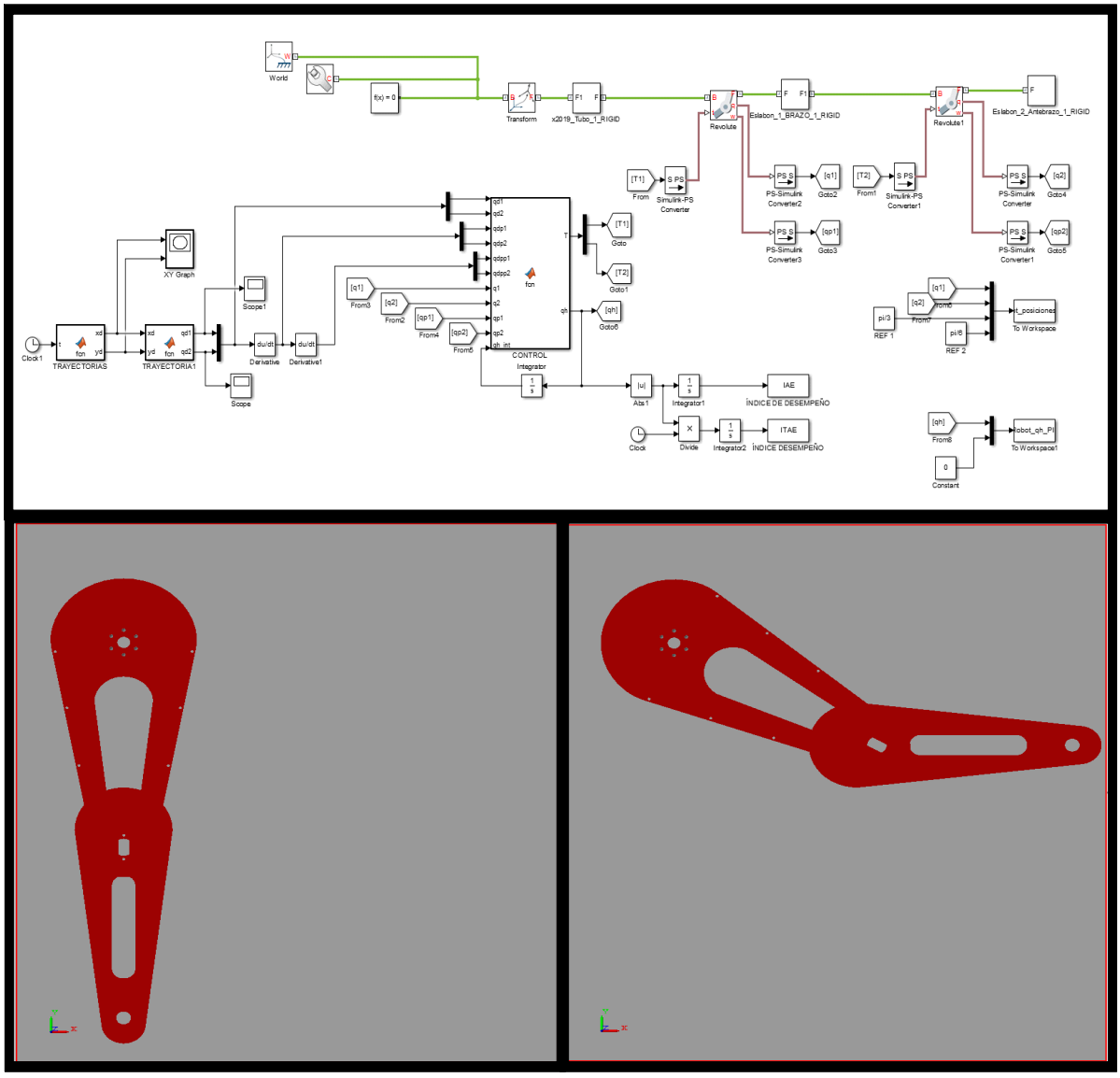


Figura 8.9: Vista superior del robot manipulador, (plancha de los motores).



(a) Posición de Casa (b) Posición Final

Figura 8.10: Diagrama a bloques y eslabones del manipulador desde Simulink Matlab.

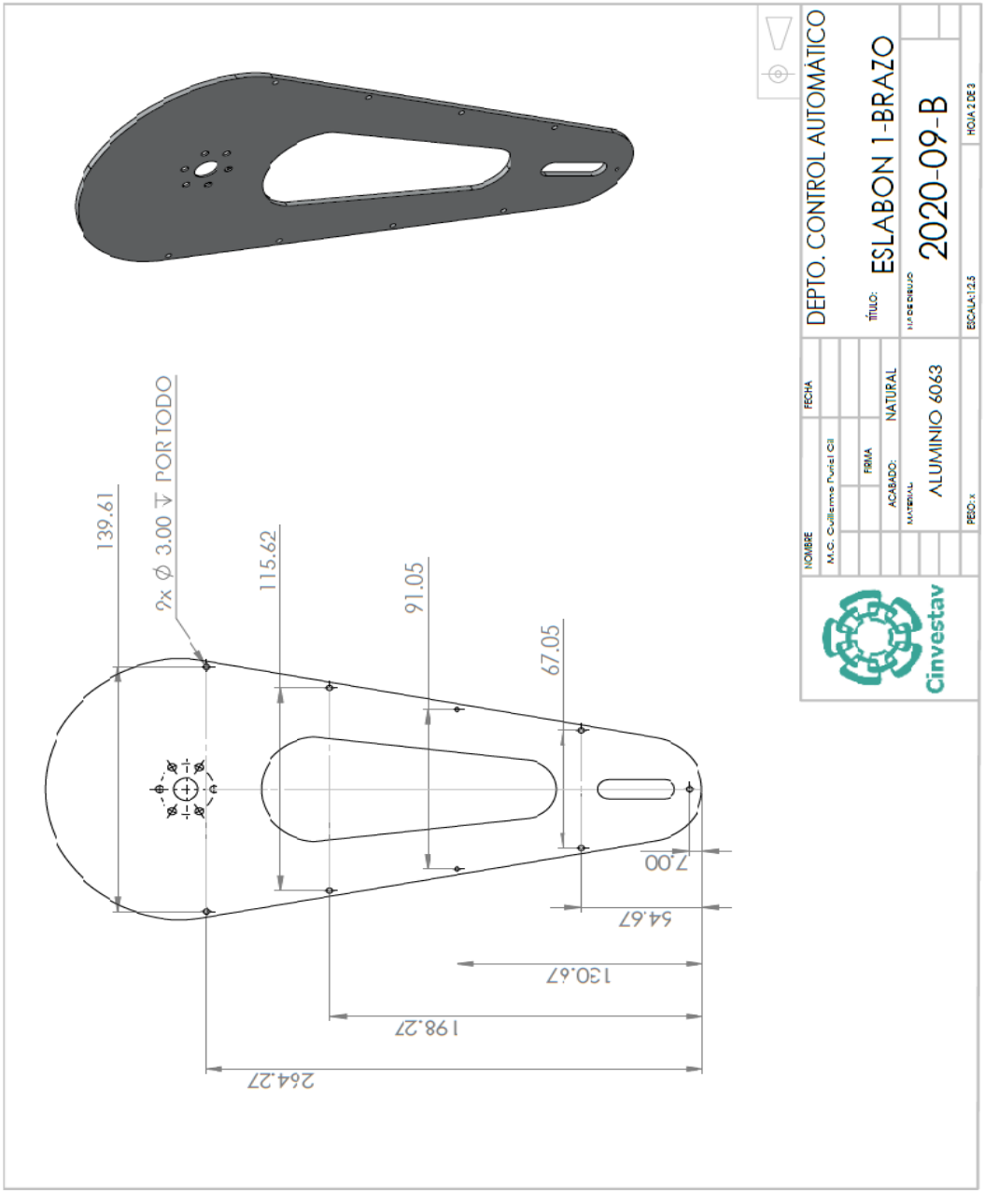


Figura 8.11: Eslabón 1 Brazo, Vista Frontal (B)

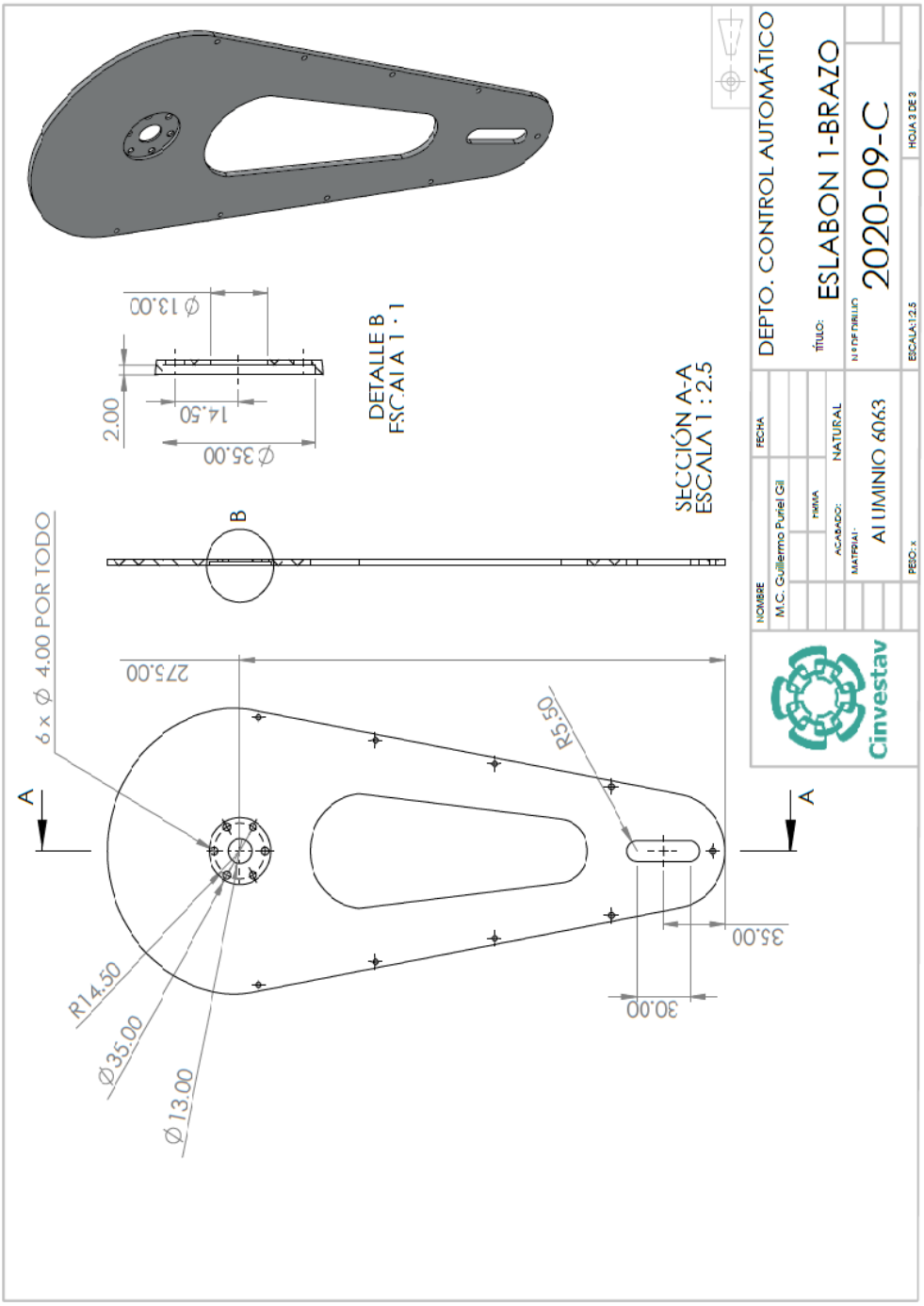


Figura 8.12: Eslabón 1-Brazo Vista Posterior.

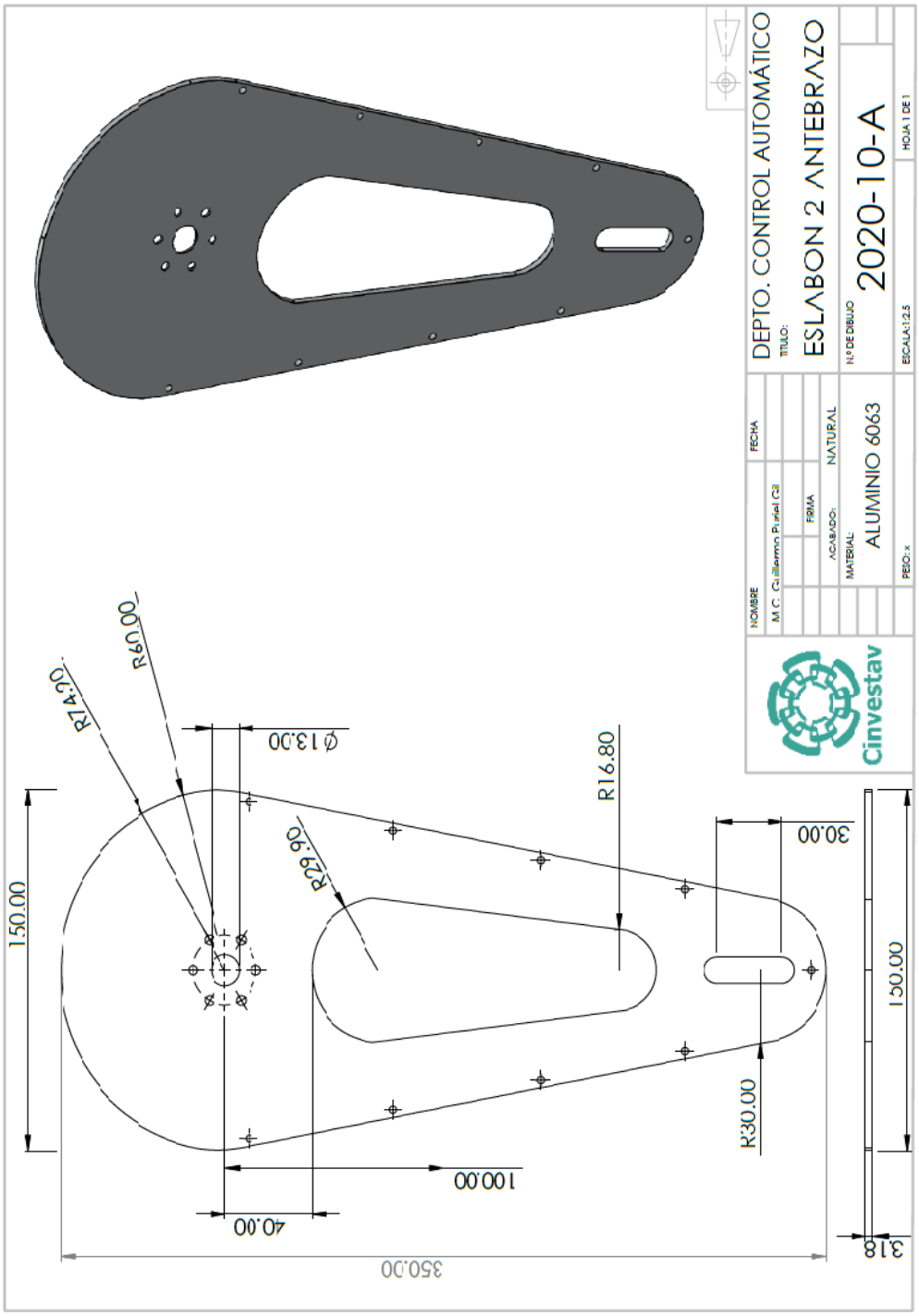


Figura 8.13: Eslabón 2 Antebrazo

Bibliografía

[1] Richard S. Sutton and Andrew G. Barto. Reinforcement learning: an introduction. The MIT Press, March 1998. ISBN 0262193981

[2] Deisenroth, Marc Peter, Gerhard Neumann, and Jan Peters. A survey on policy search for robotics. Foundations and Trends®in Robotics 2.1–2 (2013): 1-142.

[3] Polydoros, Athanasios S., and Lazaros Nalpantidis. Survey of Model-Based Reinforcement Learning: Applications on Robotics. Journal of Intelligent & Robotic Systems 86.2 (2017): 153-173.

[4] Kober, Jens, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. The International Journal of Robotics Research 32.11 (2013): 1238-1274.

[5] Kaelbling, Leslie Pack, Michael L. Littman, and Andrew W. Moore. Reinforcement learning: A survey. Journal of artificial intelligence research 4 (1996): 237-285.

[6] Moerland, Thomas M., Joost Broekens, and Catholijn M. Jonker. Emotion in Reinforcement Learning Agents and Robots: A Survey. arXiv preprint arXiv:1705.05172 (2017).

[7] Ghavamzadeh, Mohammad, et al. Bayesian reinforcement learning: A survey. Foundations and Trends®in Machine Learning 8.5-6 (2015): 359-483.

[8] Zheng, Yu, Siwei Luo, and Ziang Lv. Control double inverted pendulum by reinforcement learning with double cmac network. Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. Vol. 4. IEEE, 2006.

[9] Hosokawa, Shu, and Kazushi Nakano. A reward allocation method for reinforcement learning in stabilizing control of T-inverted pendulum. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTICON), 2012 9th International Conference on. IEEE, 2012.

[10] Hosokawa, Shu, Joji Kato, and Kazushi Nakano. A reward allocation method for reinforcement learning in stabilizing control tasks. Artificial Life and Robotics 19.2 (2014): 109-114.

[11] Laud AD (2004) Theory and Application of Reward Shaping in Reinforcement Learning. Ph.D. thesis, University of Illinois at Urbana-Champaign.

[12] Atkeson CG and Schaal S (1997) Robot learning from demonstration.In: International Conference on Machine Learning (ICML).

[13] Abbeel P, Coates A, Quigley M and Ng AY (2007) An application of reinforcement learning to aerobatic helicopter flight. In: Advances in Neural Information Processing Systems (NIPS).

[14] Deisenroth MP, Rasmussen CE and Fox D (2011) Learning to control a low-cost manipulator using data-efficient reinforcement learning. In: Robotics: Science and Systems (RSS).

[15] Gullapalli V, Franklin J and Benbrahim H (1994) Acquiring robot skills via reinforcement learning. IEEE Control Systems Magazine 14(1): 13–24.

[16] Miyamoto H, Schaal S, Gandolfo F, et al. (1996) A Kendama learning robot based on bi-directional theory. Neural Networks 9(8): 1281–1302.

[17] Bagnell JA and Schneider JC (2001) Autonomous helicopter control using reinforcement learning policy search methods. In: IEEE International Conference on Robotics and Automation (ICRA).

[18] Kohl N and Stone P (2004) Policy gradient reinforcement learning for fast quadrupedal locomotion. In: IEEE International Conference on Robotics and Automation (ICRA).

[19] Tedrake R, Zhang TW and Seung HS (2005) Learning to walk in 20 minutes. In: Yale Workshop on Adaptive and Learning Systems.

[20] Kober J and Peters J (2009) Policy search for motor primitives in robotics. In: Advances in Neural Information Processing Systems (NIPS).

[21] Peters J and Schaal S (2008a) Learning to control in operational space. The International Journal of Robotics Research 27(2): 197–212.

[22] Atkeson CG (1994) Using local trajectory optimizers to speed up global optimization in dynamic programming. In: Advances in Neural Information Processing Systems (NIPS).

[23] Kober J and Peters J (2010) Policy search for motor primitives in robotics. Machine Learning 84(1–2): 171–203.

[24] Zhou K and Doyle JC (1997) Essentials of Robust Control. Englewood Cliffs, NJ: Prentice-Hall.

[25] Ng AY, Harada D and Russell SJ (1999) Policy invariance under reward transformations: Theory and application to reward shaping. In: International Conference on Machine Learning (ICML).

[26] Spong M. Vidyasagar and S. Hutchinson. Robot Modeling and Control". John Wiley and Sons, 2006.

[27] Tomei P, Adaptive PD Controller for Robot Manipulators, IEEE Transactions on Robotics and Automation, VoL. 7, No. 4 , 565-570,1991.

[28] Slotine J. J.,W.Li, .Adaptive manipulator control: A case study", IEEE Transactions on Automatic Control, Vol.33, No.11, 995.1003, 1988.

[29] Kelly R., "PD control with desired gravity compensation of robotic manipulators: A review", International Journal of Robotics Research, Vol. 16, No.5, pp. 660.672 , 1997.

[30] Paden B., R.Panja, "Globally asymptotically stable PD+ controller for robot manipulators", International Journal of Control, Vol. 47, No. 6, 1697.1712 , 1988.

[31] Qu z., D.M. Dawson, S. Y. Lim, J.F. Dorsey, .A New Class of Robust Control Laws for Tracking of Robots", International Journal of Robotics Research, Vol. 13, No.4, 355.363, 1994.

[32] Nunes E.V. L., L.Hsu, F.Lizarralde, Arbitrarily small damping allows global output feedback tracking of a class of Euler-Lagrange systems", 2008 American Control Conference„ Seattle, USA, 378-382, 2008.

[33] Parra-Vega, S.Arimoto, Y.-H.Liu,G.Hirzinger, P.Akella, "Dynamic Sliding PID Control for Tracking of Robot Manipulators: Theory and Experiments", IEEE Transactions on Robotics and Automation, Vol.19, No.6, 967-976, 2003.

[34] Li H-X, L.Zhang, K-Y.Cai, G.Chen, An Improved Robust Fuzzy-PID ControllerWith Optimal Fuzzy Reasoning, IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol.35, No.6, 1283-1294, 2005.

[35] Harinath E.,G.Mann, "Design and Tuning of Standard Additive Model Based Fuzzy PID Controllers for Multivariable Process Systems", IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol.38, No.8, 667-674, 2008.

[36] Lewis, F.L., Dawson, D.M. and Abdallah, C.T., Robot Manipulator Control: Theory and Practice", Marcel Dekker, Inc, New York, NY 10016, 2004.

[37] Chen C., "Dynamic Structure Neural-Fuzzy Networks for Robust Adaptive Control of Robot Manipulators", IEEE Trans. Industrial Electronics, VOL. 55, NO. 9,3402-3414, 2008.

[38] Er M.J. and Y.Gao, Robust Adaptive Control of Robot Manipulators Using Generalized Fuzzy Neural Networks", IEEE Trans. Industrial Electronics, VOL. 50, NO. 3,620-628, 2003.

[39] Lewis, F.L., Liu, K. and Yesildirek, A., "Neural net robot controller with guaranteed tracking performance", IEEE Trans. Neural Networks, vol. 6, no. 3, pp. 703.715, 1995.

[40] Ioannou P.A. and J.Sun, Robust Adaptive Control, Prentice-Hall", Inc, Upper Saddle River: NJ, 1996.

[41] Gutierrez L.B. and F. L. Lewis, "Implementation of a neural net tracking controller for a single .exible link: comparisonwith PD and PID controllers", IEEE Trans. Industrial Electronics, VOL. 45, NO. 2, 307-318, 1998.No.1, 131-147, 2002.

[42] Cong S. and Y. Liang, "PID-Like Neural Network Nonlinear Adaptive Control for Uncertain Multivariable Motion Control Systems", IEEE Trans. Ind. Electron., vol. 56, no. 10, pp. 3872-3879, 2009.

[43] Scott G.M., J. W.Shavlik, W. H. Ray, Re.ning PID Controllers Using Neural Networks", Neural Computation, Vol. 4, No. 5, 746-757, 1992.

[44] Uang H. J. and C. C. Lien, "Mixed H2/H1 PID tracking control design for uncertain spacecraft systems using a cerebellar model articulation controller", IEEE Proceedings Control Theory and Applications„ vol. 153, no. 1, pp. 1.13, 2006.

[45] Mann G. K. I., B-G. Hu, R.G. Gosine, "Two-Level Tuning of Fuzzy PID Controllers", IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol.31, No.2, 263-269, 2001.

[46] Ho S-J., L-S.Shu, S-Y.Ho, Optimizing Fuzzy Neural Networks for Tuning PID Controllers Using an Orthogonal Simulated Annealing Algorithm OSA, IEEE Trans. Fuzzy Syst., Vol.14, NO. 3, 421-434, 2006.

[47] Yu D-Li, T. K. Chang, D-W.Yu, "Fault Tolerant Control of Multivariable Processes Using Auto-Tuning PID Controller", IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol.35, No.1, 32-43, 2005.

[48] Karray F. , W. Gueaieb, S. Al-Sharhan, "The Hierarchical Expert Tuning of PID Controllers Using Tools of Soft Computing", IEEE Transactions on Systems, Man, and Cybernetics, Part B, Vol.35, No.6, 1283-1294, 2005.

[49] Sandor Biro, Radu-Emil Precup and Doru Todinca, Double Inverted Pendulum Control by Linear Quadratic Regulator and Reinforcement Learning, IEEE International Joint Conferences on Computational Cybernetics and Technical Informatics (ICCC-CONTI 2010), May 27-29, 2010, Timisora, Romania.

[50] Sudhir Raj, Reinforcement Learning based Controller for Stabilization of Double Inverted Pendulum, 1st IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES-2016).

[51] Zhijun Li, Junqiang Liu, Zhicong Huang, Yan Peng, Huayan Pu, and Liang Ding, Adaptive Impedance Control of Human-Robot Cooperation Using Reinforcement Learning, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS.

[52] Zhicong Huang, Junqiang Liu, Zhijun Li, and Chun-Yi Su, Adaptive Impedance Control of Robotic Exoskeletons Using Reinforcement Learning, 2016 International Conference on Advanced Robotics and Mechatronics (lCARM).

[53] Abdelhamid Tayebi, Adaptive iterative learningcontrol for robot manipulators, Automatica 40 (2004) 1195 – 1203, Elsevier Ltd.

[54] Santanu Kumar Pradhan and Bidyadhar Subudhi, Real-Time Adaptive Control of a Flexible Manipulator Using Reinforcement Learning, IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, VOL. 9, NO. 2, APRIL 2012.

[55] Carlucho I, De Paula M, Sebastian A. Villar, Gerardo G. Acosta, Incremental Q-learning strategy for adaptive PID control of mobile robots, 2017 Elsevier, Expert Systems With Applications 80 (2017) 183–199.

[56] Dong, Zhidong, Zaixing Zhang, and Peifa Jia. A neural-fuzzy BOXES control system with reinforcement learning and its applications to inverted pendulum. Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century., IEEE International Conference. Vol. 2. IEEE, 1995.

[57] Michie, Donald, and Roger A. Chambers. BOXES: An experiment in adaptive control. Machine intelligence 2.2 (1968): 137-152.

[58] Kumar, Abhishek, and Rajneesh Sharma. A stable Lyapunov constrained reinforcement learning based neural controller for non linear systems. Computing, Communication & Automation (ICCCA), 2015 International Conference on. IEEE, 2015.

[59] Mladenov, Valeri. Application of neural networks for control of inverted pendulum. WSEAS Transactions on Circuits and Systems 10.2 (2011): 49-58.

[60] Ishii, Shin, Wako Yoshida, and Junichiro Yoshimoto. Control of exploitation–exploration meta-parameter in reinforcement learning. Neural networks 15.4 (2002): 665-687.

[61] Mahadevan, Sridhar. Average reward reinforcement learning: Foundations, algorithms, and empirical results. Machine learning 22.1 (1996): 159-195.

[62] Zheng, Yu, Si-wei Luo, and Zi-ang Lv. Active exploration planning in reinforcement learning for Inverted Pendulum system control. Machine Learning and Cybernetics, 2006 International Conference on. IEEE, 2006.

[63] Linglin, Wang, Liu Yongxin, and Zhai Xiaoke. Design of reinforce learning control algorithm and verified in inverted pendulum. Control Conference (CCC), 2015 34th Chinese. IEEE, 2015.

[64] Anderson, Charles W. Learning to control an inverted pendulum using neural networks. IEEE Control Systems Magazine 9.3 (1989): 31-37.

[65] Zheng, Yu, and Siwei Luo. The negative effect on the control of inverted pendulum caused by the limit cycle in reinforcement learning. Neural Networks and Brain, 2005. ICNN&B'05. International Conference. Vol. 2. IEEE, 2005.

[66] Hehn, Markus, and Raffaello D'Andrea. A flying inverted pendulum. Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.

[67] Figueroa, Rafael, et al. Reinforcement learning for balancing a flying inverted pendulum. Intelligent Control and Automation (WCICA), 2014 11th World Congress on. IEEE, 2014.

[68] Faust, Aleksandra, et al. Continuous action reinforcement learning for underactuated dynamical system control. Adaptive Motion Planning Research Group Technical Report TR13-002 (2013).

[69] Raj, Sudhir. Reinforcement learning based controller for stabilization of double inverted pendulum. Power Electronics, Intelligent Control and Energy Systems (ICPEICES), IEEE International Conference on. IEEE, 2016.

[70] Yu, Zheng, et al. Control parallel double inverted pendulum by hierarchical reinforcement learning. Signal Processing, 2004. Proceedings. ICSP'04. 2004 7th International Conference on. Vol. 2. IEEE, 2004.

[71] Biro, Sandor, Radu-Emil Precup, and Doru Todinca. Double inverted pendulum control by linear quadratic regulator and reinforcement learning. Computational Cybernetics

and Technical Informatics (ICCC-CONTI), 2010 International Joint Conference on. IEEE, 2010.

[72] Zheng, Yu, Siwei Luo, and Ziang Lv. Control double inverted pendulum by reinforcement learning with double cmac network. Pattern Recognition, 2006. ICPR 2006. 18th International Conference on. Vol. 4. IEEE, 2006.

[73] Li, Z., Liu, J., Huang, Z., Peng, Y., Pu, H., & Ding, L. (2017). Adaptive impedance control of human–robot cooperation using reinforcement learning. IEEE Transactions on Industrial Electronics, 64(10), 8013-8022.

[74] Kober, Jens, and Jan R. Peters. Policy search for motor primitives in robotics. Advances in neural information processing systems. 2009.

[75] Huang, Zhicong, et al. Adaptive impedance control of robotic exoskeletons using reinforcement learning. Advanced Robotics and Mechatronics (ICARM), International Conference on. IEEE, 2016.

[76] Lillicrap, Timothy P., et al. Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015).

[77] Huang, Z., Liu, J., Li, Z., & Su, C. Y. (2016, August). Adaptive impedance control of robotic exoskeletons using reinforcement learning. In Advanced Robotics and Mechatronics (ICARM), International Conference on (pp. 243-248). IEEE.

[78] Deisenroth, M. P., Rasmussen, C. E., & Fox, D. (2011). Learning to control a low-cost manipulator using data-efficient reinforcement learning.

[79] Lin, C. K. (2009). H ∞ reinforcement learning control of robot manipulators using fuzzy wavelet networks. Fuzzy Sets and Systems, 160(12), 1765-1786.

[80] Pradhan, S. K., & Subudhi, B. (2012). Real-time adaptive control of a flexible manipulator using reinforcement learning. IEEE Transactions on Automation Science and Engineering, 9(2), 237-249.

[81] Miljković, Z., Mitić, M., Lazarević, M., & Babić, B. (2013). Neural network reinforcement learning for visual control of robot manipulators. Expert Systems with Applications, 40(5), 1721-1736.

[82] Duguleana, M., Barbuceanu, F. G., Teirelbar, A., & Mogan, G. (2012). Obstacle avoidance of redundant manipulators using neural networks based reinforcement learning. Robotics and Computer-Integrated Manufacturing, 28(2), 132-146.

[83] Althoefer, K., Krekelberg, B., Husmeier, D., & Seneviratne, L. (2001). Reinforcement learning in a rule-based navigator for robotic manipulators. Neurocomputing, 37(1-4), 51-70.

[84] Kormushev, P., Calinon, S., & Caldwell, D. G. (2010, October). Robot motor skill coordination with EM-based reinforcement learning. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on (pp. 3232-3237). IEEE.

[85] Busoniu, L., De Schutter, B., & Babuska, R. (2006, December). Decentralized reinforcement learning control of a robotic manipulator. In Control, Automation, Robotics and Vision, 2006. ICARCV'06. 9th International Conference on (pp. 1-6). IEEE.

[86] Barto, M. T. R. A. G., & Rosenstein, M. T. (2004). J. 4 supervised actor-critic reinforcement learning. Handbook of learning and approximate dynamic programming, 2, 359.

[87] Park, J. J., Kim, J. H., & Song, J. B. (2007). Path planning for a robot manipulator based on probabilistic roadmap and reinforcement learning. International Journal of Control, Automation, and Systems, 5(6), 674-680.

[88] Adam, S., Busoniu, L., & Babuska, R. (2012). Experience replay for real-time reinforcement learning control. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 42(2), 201-212.

[89] Tzafestas, S. G., & Rigatos, G. G. (2002). Fuzzy reinforcement learning control for compliance tasks of robotic manipulators. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 32(1), 107-113.

[90] Stulp, F., Theodorou, E. A., & Schaal, S. (2012). Reinforcement learning with sequences of motion primitives for robust manipulation. IEEE Transactions on robotics, 28(6), 1360-1370.

[91] Tang, L., Liu, Y. J., & Tong, S. (2014). Adaptive neural control using reinforcement learning for a class of robot manipulator. Neural Computing and Applications, 25(1), 135-141.

[92] Lin, C. K. (2003). A reinforcement learning adaptive fuzzy controller for robots. Fuzzy sets and systems, 137(3), 339-352.

[93] Santamaría, J. C., Sutton, R. S., & Ram, A. (1997). Experiments with reinforcement learning in problems with continuous state and action spaces. Adaptive behavior, 6(2), 163-217.

[94] Benbrahim, H., & Franklin, J. A. (1997). Biped dynamic walking using reinforcement learning. Robotics and Autonomous Systems, 22(3-4), 283-302.

[95] Peters, J., & Schaal, S. (2007, June). Reinforcement learning by reward-weighted regression for operational space control. In Proceedings of the 24th international conference on Machine learning (pp. 745-750). ACM.

[96] Zhang, F., Leitner, J., Milford, M., Upcroft, B., & Corke, P. (2015). Towards vision-based deep reinforcement learning for robotic motion control. arXiv preprint arXiv:1511.03791.

[97] Khan, S. G., Herrmann, G., Lewis, F. L., Pipe, T., & Melhuish, C. (2012). Reinforcement learning and optimal adaptive control: An overview and implementation examples. Annual Reviews in Control, 36(1), 42-59.

[98] Gu, S., Holly, E., Lillicrap, T., & Levine, S. (2017, May). Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In Robotics and Automation (ICRA), 2017 IEEE International Conference on (pp. 3389-3396). IEEE.

[99] Tham, C. K., & Prager, R. W. (1993, July). Reinforcement learning methods for multi-linked manipulator obstacle avoidance and control. In Motion Control Proceedings, 1993., Asia-Pacific Workshop on Advances in (pp. 140-145). IEEE.

[100] Tayebi, A. (2004). Adaptive iterative learning control for robot manipulators. Automatica, 40(7), 1195-1203.

[101] Bondi, P., Casalino, G., & Gambardella, L. (1988). On the iterative learning control theory for robotic manipulators. IEEE Journal on Robotics and Automation, 4(1), 14-22.

[102] Morimoto, J., & Doya, K. (2001). Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. Robotics and Autonomous Systems, 36(1), 37-51.

[103] Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. In Advances in neural information processing systems (pp. 1038-1044).

[104] Kelly Rafael, Santibáñez Víctor. Control de Movimiento de Robots Manipuladores. PEARSON, Prentice Hall. 2003. ISBN: 84-205-3831-0.

[105] Kelly R, Santibáñez V. and Loría A. Control of Robot Manipulators in Joint Sapace. Springer-Verlag London Limited 2005. ISBN-10: 1852339942.

[106] Reyes Cortés Fernando. Robótica, Control de Robots Manipuladores. Primera Edición. Alfaomega, México, marzo 2011. ISBN:978-607-707-190-7.