



INGENIERÍA EN MECATRÓNICA

**ESTRATEGIAS DE CONTROL DE
MOVIMIENTO EN ROBOT PENDULAR**

**Tesis profesional para obtener el grado de
INGENIERO EN MECATRÓNICA**

Presenta:

JACOBO TORRES FIGUEROA

Asesor de tesis:

Dr. Hiram Netzahualcóyotl García Lozano

Loma Bonita, Oax.

Enero 2015

Dedicatorias

A mis padres y hermanos, por el apoyo incondicional, comprensión, y amor que me han brindado toda mi vida.

Al desequilibrio en la red, que trae movilidad, aunque no parezca y no lo quiera.

A Fátima por creer en mi y verme, cuando todo estaba «más oculto y yo no me veía».

Agradecimientos

A Dios (Hashém) por darme el ser trascendente necesario, para venir a existir y poder realizar las cosas en el mundo físico, en este espacio y tiempo concretos, aun cuando «sea de noche».

A todas las personas que me han transmitido conocimiento, que ha tenido repercusión en este trabajo, pero de forma particular al Dr. Hiram N. García Lozano, por fungir como mi asesor de tesis, brindarme apoyo y motivación siempre durante toda la carrera.

A Iván, Temo, Gaspar y Erick, por su amistad.

Al factor «k», por ayudarme en momentos difíciles.

Resumen

En este trabajo se aborda el diseño, simulación y construcción de un prototipo funcional, basado en un robot pendular, controlado en posición bajo las leyes de control Proporcional Derivativo (PD) y Proporcional Derivativo con compensación precalculada de gravedad. Es necesaria la abstracción de los diferentes modelos matemáticos que describen la mecánica del movimiento del robot (modelo cinemático directo e inverso y modelo dinámico), así como del modelo dinámico del actuador, para estudiar el comportamiento en lazo cerrado del sistema, bajo la acción de las leyes de control.

Se realiza una comparación en la compensación del error de los controladores PD y PD con compensación precalculada de gravedad, a partir de simulaciones numéricas bajo condiciones iniciales específicas, involucrando parámetros mecánicos y eléctricos del robot. Los parámetros mecánicos del robot parten directamente del eslabón, el cual es diseñado de forma que cuente con menor masa y presente menor inercia.

La implementación en el prototipo de las leyes de control, se lleva a cabo mediante un sistema electrónico basado en el microcontrolador PIC18F2431 donde son embebidos los controles diseñados, además gestiona los diversos componentes electrónicos y procesa las señales propias del robot.

Abstract

In this paper the design, simulation and building a working prototype based on a pendulum robot, controlled in position under the laws of Control Proportional Derivative (PD) and Proportional Derivative with precalculated gravity compensation is discussed. Abstraction of different mathematical models describing the mechanics of robot motion (direct and inverse kinematic model and dynamic model) is required as well as the dynamic model of the actuator, to study the behavior of the closed loop system under the action of control laws.

A comparison is made in the error compensation PD and PD controllers with precalculated gravity compensation, from numerical simulations under specific initial conditions, involving mechanical and electrical parameters of the robot. The mechanical parameters of the robot start directly the link, which is designed so that has less mass and lower inertia present.

Implementing control laws in the prototype, is performed by an electronic system based on PIC18F2431 microcontroller, embedded controls which are designed, also manages the various electronic components and processes signals themselves robot.

Índice

1. Introducción	11
1.1. Historia de la Robótica	11
1.2. Aplicaciones de la Robótica	14
1.3. Clasificación de los Robots	16
2. Modelos matemáticos de robot pendular	18
2.1. Introducción	18
2.2. Robot Pendular RP1	20
2.3. Modelos cinemáticos	21
2.3.1. Modelo cinemático directo de robot pendular	21
2.3.2. Modelo cinemático inverso de robot pendular	24
2.4. Modelos dinámicos	25
2.4.1. Ecuaciones de movimiento de Lagrange	25
2.4.2. Modelo dinámico general de robots manipuladores	26
2.4.3. Modelo dinámico lineal del actuador	28
2.4.4. Modelo dinámico general de robots manipuladores con actuadores	31
2.4.5. Modelo dinámico de robot pendular sin actuador	32
2.4.6. Modelo dinámico de robot pendular con actuador lineal	35
3. Estrategias de control de posición para robots	36
3.1. Introducción	36
3.2. Estabilidad en el sentido de Lyapunov	38
3.2.1. Equilibrio	38
3.2.2. Estabilidad	40
3.2.3. Estabilidad asintótica	41
3.2.4. Estabilidad asintótica global	41
3.2.5. Estabilidad exponencial global	41
3.2.6. Inestabilidad	42
3.3. Método directo de Lyapunov	42
3.3.1. Función de Lyapunov	42
3.3.2. Estabilidad	43
3.3.3. Estabilidad y acotamiento de las soluciones	43

3.3.4.	Estabilidad asintótica global	44
3.4.	Control Proporcional Derivativo (PD)	44
3.4.1.	Equilibrio único	47
3.4.2.	Diseño de control PD para robot pendular	47
3.5.	Control PD con compensación precalculada de gravedad	50
3.5.1.	Equilibrio único	53
3.5.2.	Diseño de control PD con compensación precalculada de gravedad para robot pendular	53
4.	Simulaciones	58
4.1.	Introducción	58
4.2.	Simulación del actuador	58
4.3.	Simulación de modelo dinámico robot con actuador sin controlador	65
4.4.	Simulación de Control PD	67
4.5.	Simulación de Control PD con compensacion precalculada de gra- vedad	73
5.	Implementación	78
5.1.	Sistema mecánico	78
5.1.1.	Introducción	78
5.1.2.	Diseño de eslabón rotacional para robot pendular RP1	79
5.2.	Sistema eléctrico y electrónico	86
5.2.1.	Introducción	86
5.2.2.	Motor de CD.	87
5.2.3.	Microcontrolador PIC18F2431	89
5.2.4.	Modulo de retroalimentacion de movimiento	91
5.2.5.	Configuración del módulo QEI	93
5.2.6.	Convertidor Digital/Analógico DAC0800	95
5.2.7.	Etapas de potencia	97
5.2.8.	Comunicación serial mediante USART	99
5.2.9.	Tarjeta electrónica de control	102
5.3.	Implementación del algoritmo de control	106
5.3.1.	Introducción	106
5.3.2.	Estimación de la velocidad articular \dot{q}	111
6.	Conclusiones	112

A. Modelo de sólidos del robot pendular RP1	114
B. Algoritmo de control en lenguaje C	121

Índice de figuras

1. Mecanismos internos del Pato mecánico diseñado y construido por Jacques de Vaucanson, el cual podía realizar movimientos complejos emulando el comportamiento del ave.	11
2. Robot Unimate de aplicación industrial, puesto en marcha por General Motors en 1961.	12
3. Robot IRB6 controlado por microprocesador Intel de 8 bits desarrollado por ABB.	13
4. Robot PUMA desarrollado por Victor Scheinman en 1975.	13
5. Robot ASIMO, uno de los robots mas avanzados en la actualidad, dotado inteligencia artificial y sistemas sensoriales sofisticados. . .	14
6. Proceso de soldado de partes del chasis de automóviles, en una linea de producción automatizada mediante robots.	14
7. Robot cuya función ha sido programada para el paletizado automático en una linea de producción.	15
8. Robot de armado de tarjetas electrónicas, sitúa los diferentes componentes electrónicos en la tarjeta de circuitos impresa a una gran velocidad.	15
9. Sistema quirúrgico Da Vinci, permite operaciones menos invasivas, que se traduce en operaciones con menos dolor para los pacientes y menor riesgo de infecciones.. . . .	16
10. Robot terrestre de locomoción por ruedas Curiosity construido por NASA y puesto en operación en 2012.	17
11. Modelo cinemático directo, de robot de dos grados de libertad. . .	19
12. Múltiples soluciones en el modelo cinemático inverso.	20
13. Robot pendular RP1 de 1 g.d.l.	21
14. Esquema de robot pendular RP1, para obtención de cinemática directa.	22
15. Tabla de parámetros de vinculo del robot RP1.	22
16. Coordenadas del efector final del robot RP1.	24

17.	Esquema electromecánico de motor de C.D. controlado por corriente de armadura con caja de reductora de engranajes acoplado. . .	28
18.	Robot Pendular de 1 GDL, provisto de motor de C.D. como actuador.	32
19.	Diagrama de robot pendular, sin considerar fricción en sus soportes.	33
20.	Ejemplo de puntos en el espacio de trabajo del robot, para control de movimiento.	37
21.	Control por retroalimentación negativa de un robot, nótese la presencia del controlador que genera los pares τ (a partir del error) que son introducidos al robot.	38
22.	Ejemplo de equilibrio cuando $x(0) = x_e$ en \mathbb{R}^2	39
23.	Ejemplo de estabilidad en el sentido de Lyapunov en el espacio vectorial \mathbb{R}^2	40
24.	Ejemplo de estabilidad asintótica para el caso de $x(0) \in \mathbb{R}^2$	41
25.	Diagrama a bloques del control de posición con retroalimentación de velocidad.	44
26.	Diagrama a bloques del control proporcional derivativo (PD). . .	45
27.	Diagrama a bloques del control PD con compensación de gravedad.	50
28.	Diagrama de bloques del control PD con compensación precalculada de gravedad.	52
29.	Diagrama esquemático simplificado del robot pendular RP1. . . .	59
30.	Diagrama a bloques en Simulink, de motor de CD.	61
31.	Gráfica de velocidad en el tiempo del motor de CD.	62
32.	Gráfica de la corriente en el tiempo del motor de CD.	63
33.	Mecanismo de reducción de engranes, con relación de transmisión de 1:64, implementado mediante el bloque de ganancia 1/64.	64
34.	Gráfica de la velocidad angular \dot{q} , del motor después del juego de engranes.	64
35.	Diagrama a bloques del modelo dinámico, considerando el actuador en lazo abierto y sin controlador.	66
36.	Gráfica de la posición articular q del robot pendular RP1.	67
37.	Diagrama en Simulink de implementación de control PD.	70
39.	Gráfica de la velocidad articular \dot{q} , bajo la ley de control PD. . . .	71
38.	Gráfica de la posición articular q , bajo la ley de control PD. . . .	71
40.	Gráfica del error de posición \tilde{q} , respecto del tiempo, bajo la ley de control PD.	72

41.	Gráfica del error de velocidad $\dot{\tilde{q}}$, respecto del tiempo, bajo la ley de control PD.	72
42.	Gráfica del error de posición q cuando $q_a = 90$, bajo la ley de control PD.	73
43.	Diagrama a bloques en Simulink del control PD con compensación precalculada de gravedad.	75
44.	Gráfica de la posición articular q , del robot pendular, bajo la ley de control PD con compensación precalculada de gravedad.	76
45.	Gráfica de la velocidad articular \dot{q} , del robot pendular, bajo la ley de control PD con compensación precalculada de gravedad.	77
46.	Gráfica del error de posición \tilde{q} , del robot pendular, bajo la ley de control PD con compensación precalculada de gravedad.	77
47.	Gráfica del error de velocidad $\dot{\tilde{q}}$, del robot pendular, bajo la ley de control PD con compensación precalculada de gravedad.	78
48.	Diagrama esquemático de robot pendular construido.	79
49.	Parámetros críticos en el diseño del eslabón rotacional.	80
50.	Perfil para eslabón rotacional no simétrico, considerando las restricciones de diseño.	81
51.	Perfil del eslabón rotacional para robot pendular.	82
52.	Diagrama del sistema de control, del robot pendular RP1.	87
53.	Motor de CD. con escobillas de la marca Faulhaber.	88
54.	Diagrama del circuito controlador del codificador óptico incremental, acoplado al motor de CD.	89
55.	Diagrama de conexiones del microcontrolador PIC18F2431.	90
56.	Diagrama simplificado de bloques del modulo QEI, del PIC18F2431.	93
57.	Diagrama de conexiones del DAC0800.	96
58.	Configuración del convertidor digital/analógico, DAC0800 en modo bipolar.	97
59.	Etapa de potencia Push-Pull, basada en dos transistores de potencia NPN y PNP.	98
60.	Forma de onda resultante V_{SAL} , del arreglo de transistores Push-Pull, al circular un voltaje V_{ENT} de forma senoidal.	98
61.	Esquema de comunicación serial sincrona y asincrona mediante USART.	101

62.	Trama para envío de datos, del protocolo estándar RS-232, utilizado en la comunicación serial mediante USART.	102
63.	Tarjeta electrónica de control, para robot pendular.	103
64.	Diagrama de bloques del sistema electrónico de control, para robot pendular, tomando en consideración la naturaleza de las señales involucradas.	107
65.	Inconsistencia aparente, de señales en el lazo de retroalimentación, de los controladores de posición PD y PD con compensación precalculada de gravedad.	108
66.	Bloque de conversión par-voltaje, añadido al lazo de los controles de posición PD y PD con compensación precalculada de gravedad.	109
67.	Diagrama de flujo del algoritmo del sistema electrónico de control.	111
68.	Vista explotada del robot pendular RP1.	114
69.	Plano de pieza Base.	115
70.	Plano de pieza Tapa Inferior.	115
71.	Plano de pieza Soporte.	116
72.	Plano de pieza Tapa Superior.	116
73.	Plano de pieza Inferior.	117
74.	Plano de pieza Superior.	117
75.	Plano de pieza Lateral 1 (equivalente a pieza Lateral 2).	118
76.	Plano de pieza Eslabón.	119
77.	Plano de pieza Frontal 1.	120
78.	Plano de pieza Frontal 2.	120
79.	Plano de pieza Frontal 3.	121

1. Introducción

1.1. Historia de la Robótica

Una idea presente en la historia de la humanidad es la de simplificar los trabajos que más esfuerzo demandan. A esto se debe el desarrollo paulatino de herramientas, mecanismos y máquinas. La complejidad de éstas creció a medida que los siglos trascurrieron. Estas herramientas principalmente mecánicas, se tornaron cada vez más complejas y con diversas funciones, evolucionando de herramientas manuales hasta sistemas autónomos. Esta idea se ve reflejada en los sistemas mecánicos con el propósito de automatizar procesos simples, como se relata en los escritos de Herón de Alejandría hacia el siglo I AC. en su obra: Tratado de Neumática.

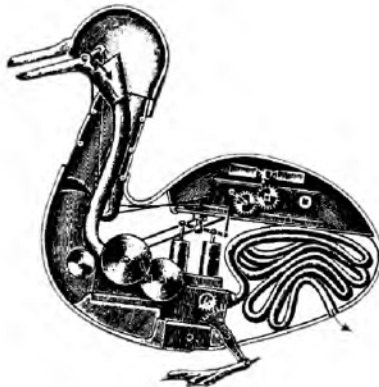


Figura 1: Mecanismos internos del Pato mecánico diseñado y construido por Jacques de Vaucanson, el cual podía realizar movimientos complejos emulando el comportamiento del ave.

Dichos autómatas solo eran diseñados para una tarea específica y normalmente desarrollados como medio de entretenimiento. Por ejemplo, en el año 1206 Al-jazari construyó el primer autómatas humanoide, que además era programable mediante un tambor giratorio. Posteriormente en 1352 se construyó un mecanismo complejo que simulaba el comportamiento de un gallo, denominado Gallo de Estrasburgo, que cantaba y agitaba las alas. Otro ejemplo, es el pato de Jacques de Vaucanson, el cual realizaba múltiples movimientos e inclusive digestión de manera simulada emulando el comportamiento de un pato real, ver figura 1. El Caballero mecánico diseñado por Leonardo Da Vinci en 1495 es otro ejemplo de autómatas humanoides de gran complejidad. Estos autómatas vislumbraban ya el

concepto de Robot, sin embargo carecían de las características de los Robots que se desarrollaron en el siglo XX.

El termino Robot, utilizado hasta el año 1920, fue introducido por primera vez por el escritor Karel Čapek en su novela *Russum's Universal Robots*.

La palabra Robot tiene origen en el vocablo Checo: *Robota*, que significa trabajos forzados. En las décadas de los 40's comenzó el desarrollo de la Robótica. Múltiples empresas y centros de investigación han diseñado y construido robots programables [1]. Uno de los primeros fue construido por George Devol y Joseph Engelberger en 1954, denominado *Unimate*, fue puesto en operación por la empresa General Motors en 1961 como se aprecia en la figura 2.



Figura 2: Robot *Unimate* de aplicación industrial, puesto en marcha por General Motors en 1961.

La introducción de articulaciones electromecánicas en los robots por la empresa Kuka Robotics con el robot *FUMALUS* de 6 ejes, se produjo en 1973. Con el desarrollo de la electrónica y la tecnología de semiconductores, fue posible en 1974 la construcción del primer Robot controlado por un microprocesador de Intel de 8 bits, por la empresa ABB (Asea Brown Boveri) denominado *IRB6*, el cual se muestra en la figura 3.



Figura 3: Robot IRB6 controlado por microprocesador Intel de 8 bits desarrollado por ABB.

En 1975 se construye el primer brazo manipulador industrial por Victor Scheinman denominado PUMA (Programmable Universal Manipulation Arm) ver figura 4, posteriormente en 1981, se diseñó y construyó en la Universidad de Carnegie-Mellon el primer Robot de transmisión directa por Haruhiko Asada.



Figura 4: Robot PUMA desarrollado por Victor Scheinman en 1975.

Fue con el avance de la electrónica, la computación y la teoría de control que poco a poco se logró que los sistemas mecánicos robóticos realizaran tareas cada vez más precisas y complejas. Esto dio lugar a máquinas realmente autónomas e inclusive dotadas de inteligencia artificial, haciendo posible la construcción de los Robots modernos como ASIMO desarrollado por la compañía Honda Motor Co. Ltd, ver Figura 5, el cual es capaz de interactuar con las personas y el entorno en tiempo real.



Figura 5: Robot ASIMO, uno de los robots mas avanzados en la actualidad, dotado inteligencia artificial y sistemas sensoriales sofisticados.

1.2. Aplicaciones de la Robótica

Existen diversas áreas en las que la robótica ha incursionado con resultados asombrosos. Por ejemplo, en los procesos de manufactura es cada vez más común y necesario el uso de robots. Actualmente existe una tendencia hacia la automatización en las plantas de diversas índoles, como por ejemplo en las industrias: automotriz, aeroespacial, de productos electrónicos, de alimentos, etc. La utilización de la robótica en estos procesos ha dado lugar a la manufactura flexible, que se caracteriza por una alta capacidad de reprogramación de las líneas de producción para generar una gran variedad de partes. En la figura 6 se aprecia una línea de producción automotriz automatizada en el proceso de soldadura.



Figura 6: Proceso de soldado de partes del chasis de automóviles, en una línea de producción automatizada mediante robots.

Otra de las actividades en las cuales los robots han sido involucrados es el paletizado. Este trabajo puede automatizarse de tal manera que el tiempo de estibar las tarimas se reduce considerablemente en comparación que si se realizara con maquinaria convencional (montacargas) manipulada por operadores humanos, ver figura 7.



Figura 7: Robot cuya función ha sido programada para el paletizado automático en una línea de producción.

Los robots se caracterizan por la alta precisión y repetitividad de sus movimientos, gracias a sofisticados controladores de movimiento integrados. Esta característica los hace especialmente indispensables en procesos de ensamblaje de tabletas electrónicas, además de que pueden realizar dichos movimientos con mucha velocidad, ver figura 8.



Figura 8: Robot de armado de tarjetas electrónicas, sitúa los diferentes componentes electrónicos en la tarjeta de circuitos impresa a una gran velocidad.

Las aplicaciones prácticas de la robótica no solo se limitan al campo de manufactura automatizada de productos, sino también en áreas tan distintas como la

medicina, la exploración espacial, aplicaciones militares, educación, etc. En medicina, la robótica ha aportado sistemas que permiten operaciones menos invasivas. Un ejemplo de este tipo es el sistema quirúrgico Da Vinci, desarrollado por la empresa norteamericana Intuitive Surgical en el año 2007. Este sistema proporciona a los cirujanos una visión superior del área a operar, mayor precisión y confort ergonómico, ver figura 9.



Figura 9: Sistema quirúrgico Da Vinci, permite operaciones menos invasivas, que se traduce en operaciones con menos dolor para los pacientes y menor riesgo de infecciones..

1.3. Clasificación de los Robots

La gran variedad de formas y medios donde realizan tareas los robots, dio lugar a la necesidad de clarificarlos. Existen básicamente dos clasificaciones para los robots de acuerdo a su espacio de trabajo.

- Robots móviles
- Robots manipuladores

Un robot móvil es aquel que es capaz de desplazarse de un punto a otro sin intervención de un conductor. Dependiendo del medio donde se desplacen pueden ser: robots terrestres, robots aéreos, robots marinos y submarinos. El sistema de locomoción de los robots terrestres puede variar según las condiciones del terreno, lo más comúnmente empleado son: ruedas, orugas y patas. El robot Curiosity, que sirve de robot científico y de exploración desde 2012 en marte, es un ejemplo de robot terrestre cuyo medio de locomoción son las ruedas, ver figura 10.



Figura 10: Robot terrestre de locomoción por ruedas Curiosity construido por NASA y puesto en operación en 2012.

Los robots manipuladores son amplia mente utilizados en la industria. La definición que propone la Federación Internacional de Robótica para robot manipulador es la siguiente:

«Un robot manipulador industrial es una maquina manipuladora con varios grados de libertad controlada automáticamente, reprograble y de múltiples usos, pudiendo estar en un lugar fijo o móvil para su empleo en aplicaciones industriales».

Estas características permitieron a los robots manipuladores lograr un gran impacto en los procesos de manufactura industriales desde mediados del siglo XX [2].

2. Modelos matemáticos de robot pendular

2.1. Introducción

Existen básicamente dos modos de estudiar el movimiento de un robot; tomando en cuenta las fuerzas que hacen posible sus movimientos o prescindir de ellas, que permiten la abstracción de dos formas de modelos matemáticos distintos, la primera forma se denomina modelo cinemático, que involucra por lo general funciones trigonométricas simples. La segunda forma de estudiar el movimiento de un robot es considerando las fuerzas que intervienen para que cada eslabón se mueva como lo hace y a su vez en armonía con los demás, a esta forma de abstracción matemática del robot se le conoce como modelo dinámico del robot que es una caracterización matemática del sistema mecánico y puede ser obtenido haciendo uso de las leyes de la mecánica de Newton. Sin embargo este método resulta muy complicado y laborioso cuando el número de articulaciones de un robot aumenta, por lo que con el fin de simplificar la obtención del modelo se puede hacer uso del método de las ecuaciones de movimiento de Lagrange [2].

Por su parte el modelo cinemático describe la posición y orientación del efector final de un robot (herramienta, pinza o elemento final del robot), respecto a un sistema de coordenadas fijo.

Se puede estudiar dicho modelo de dos formas, una de ellas es suponiendo conocidas las posiciones articulares q de cada eslabón, y así determinar una expresión matemática, donde sustituyendo dichos valores se obtenga la posición y orientación en el espacio del efector final, a esta forma de estudiar la cinemática del robot se le conoce como: modelo cinemático directo. Para encontrar el modelo cinemático directo del robot, se puede hacer uso del método creado por Denavit y Hartenberg (aplicable a los robots) [4], el cual dicta una serie de pasos para determinar una matriz de transformación homogénea que describe la posición y orientación del efector final del robot (ver figura 11).

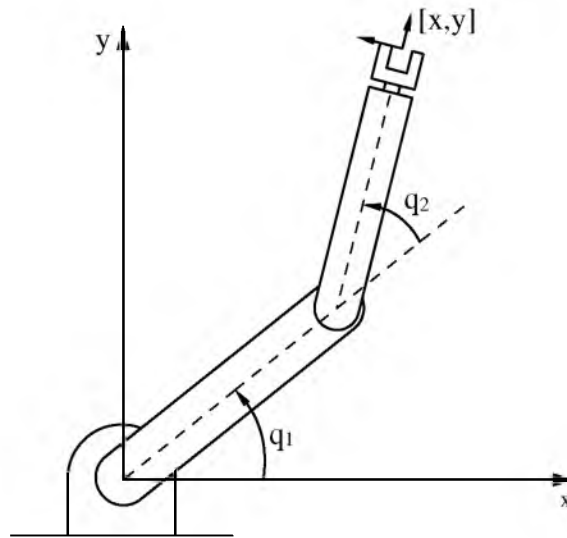


Figura 11: Modelo cinemático directo, de robot de dos grados de libertad.

Por otra parte se puede ver el mismo problema cinemático, pero esta vez conociendo la posición del efector final y no las posiciones articulares q del robot, y de esta manera encontrar una expresión matemática mediante la cual se encuentren las posiciones articulares q , con las cuales el robot hace que el efector final esté en ese punto. A esta abstracción matemática de la cinemática del robot, se le conoce como modelo cinemático inverso del robot [3]. Cabe mencionar que en el modelo cinemático inverso pueden existir múltiples soluciones en las posiciones articulares q para obtener la misma ubicación del efector final como se observa en la figura 12.

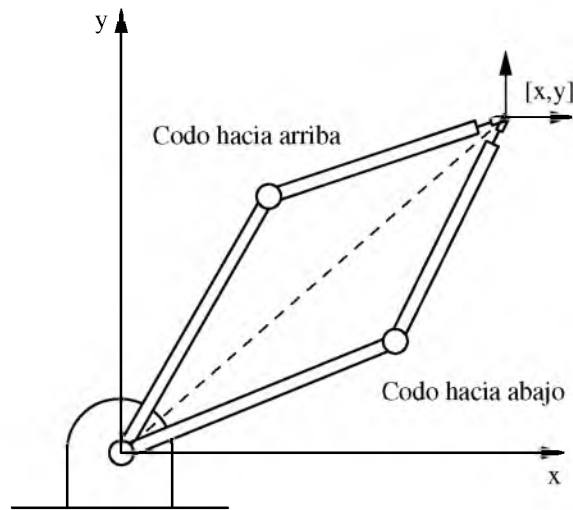


Figura 12: Múltiples soluciones en el modelo cinemático inverso.

2.2. Robot Pendular RP1

El termino robot pendular, se refiere a un robot con una sola articulación rotacional ubicada en un plano vertical, cuyo eje de giro es perpendicular al mismo bajo la acción de la fuerza de la gravedad, teniendo por tanto un solo grado de libertad. Dicho robot puede incluirse en la clasificación de robot manipulador. En la figura 13 se muestra un modelo del robot pendular RP1.

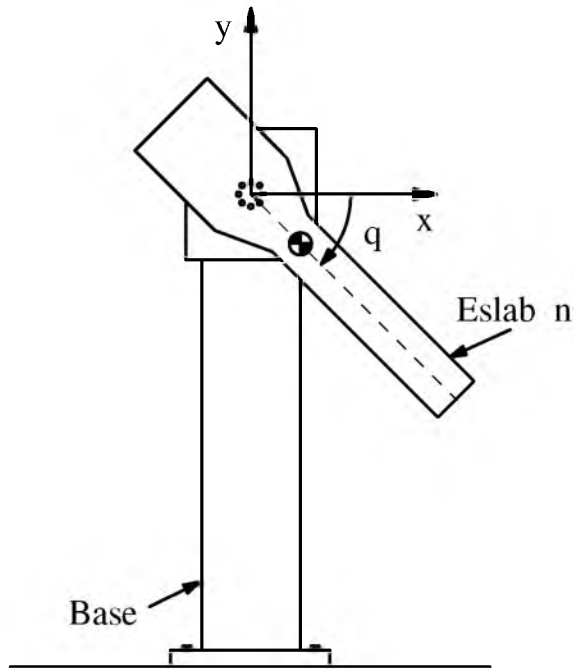


Figura 13: Robot pendular RP1 de 1 g.d.l.

En la figura 13 se observa que la forma del eslabón no es uniforme en anchura a lo largo del mismo, esto es así para mover su centro de masa a una posición más próxima al eje de giro, con el fin de disminuir la inercia del eslabón respecto a dicho eje, de manera que la fuerza requerida por el actuador para mover el eslabón sea menor. En las siguientes secciones de este capítulo se describe la obtención de los modelos cinemáticos y el modelo dinámico con y sin actuador del robot pendular RP1.

2.3. Modelos cinemáticos

2.3.1. Modelo cinemático directo de robot pendular

En este apartado se describe la obtención del modelo cinemático directo del robot pendular mostrado en la figura 13, sin embargo para propósito de mayor claridad en el procedimiento, en la figura 14 se presenta un esquema del mismo robot de manera simplificada. Las tramas asociadas a cada articulación del robot se ubicaron mediante el método propuesto por Denavit y Hartenberg.

Para encontrar la matriz de transformación 0_2T que relaciona la posición del efector final con la base del robot, es preciso antes conocer las matrices de trans-

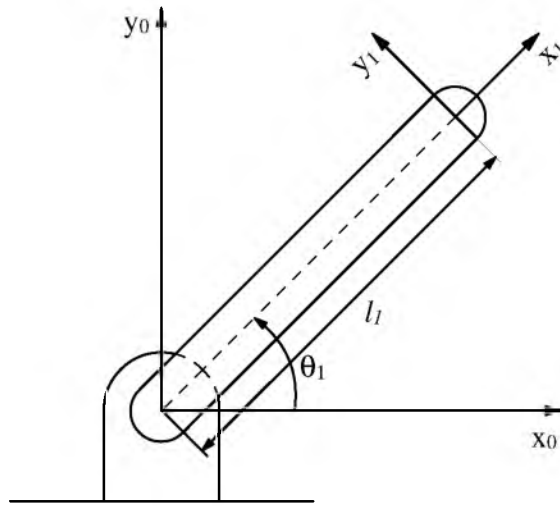


Figura 14: Esquema de robot pendular RP1, para obtención de cinemática directa.

formación 0T_1 y 1T_2 de modo que:

$${}^0T_2 = {}^0T_1 {}^1T_2 \quad (1)$$

Para este propósito se utilizan los parámetros de vínculo que se muestran en la figura 15:

i	$\alpha_i - 1$	$a_i - 1$	d_i	θ_i
1	0	0	0	θ_1
2	0	l_1	0	0

Figura 15: Tabla de parámetros de vínculo del robot RP1.

donde:

a_i = la distancia de \hat{Z}_i a $\hat{Z}_i + 1$ medida sobre \hat{X}_i .

α_i = el ángulo de \hat{Z}_i a $\hat{Z}_i + 1$ medido sobre \hat{X}_i .

d_i = la distancia de \hat{X}_{i-1} a \hat{X}_i medida sobre \hat{Z}_i .

θ_i = el ángulo de \hat{X}_{i-1} a \hat{X}_i medido sobre \hat{Z}_i .

Haciendo uso de la formula general para encontrar la matriz de transformación entre tramas de eslabones (2) [4], y de los parámetros de eslabón mostrados en la figura 15, se procede a obtener las matrices de transformación 0_1T y 1_2T .

$${}^{i-1}_i T = \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -s\alpha_{i-1} d_i \\ s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & c\alpha_{i-1} d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Al sustituir los valores de los parámetros de los eslabones en la formula general, para el eslabón 2, se obtiene la matriz de transformación 1_2T :

$${}^1_2 T = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Del mismo modo se obtiene la matriz de transformación 0_1T , utilizando los parámetros para el eslabón 1:

$${}^0_1 T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Teniendo ya, calculadas las matrices de transformación 0_1T y 1_2T , se obtiene 0_2T haciendo uso de la ecuación (1):

$${}^0_2 T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & c\theta_1 l_1 \\ s\theta_1 & c\theta_1 & 0 & s\theta_1 l_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Es en el componente vectorial de posición de la matriz de transformación 0_2T (las primeras tres posiciones de la columna cuatro) donde se expresa en vector de posición del efector final en relación con la base del robot, y en su parte rotacional se expresa la orientación del efector final. Por consiguiente las coordenadas del efector final quedan expresadas como:

$$x = l_1 \cos(\theta_1) \quad (4)$$

$$y = l_1 \sin(\theta_1); \quad (5)$$

Siendo las ecuaciones 4 y 5 el modelo cinemático directo del robot RP1.

2.3.2. Modelo cinemático inverso de robot pendular

El modelo cinemático inverso supone conocidas las coordenadas del efector final, tal como se muestra en la figura 16, de modo que la relación matemática que proporciona una expresión con la cual obtener el valor de θ_1 que hace al efector final ubicarse en dicha posición en el espacio es simple.

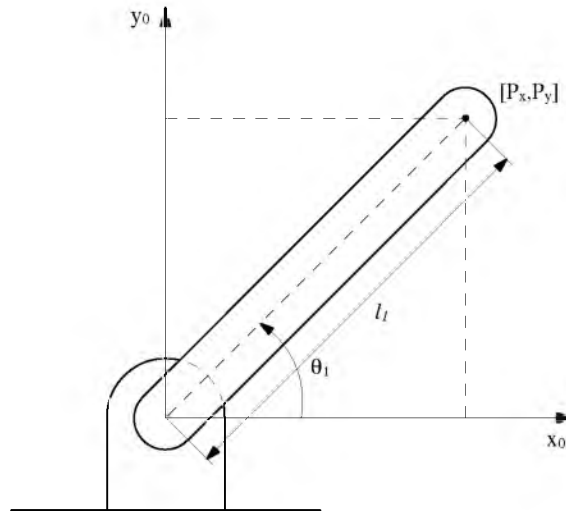


Figura 16: Coordenadas del efector final del robot RP1.

De la figura 16 puede deducirse que por simple inspección que:

$$\tan(\theta_1) = \frac{P_x}{P_y}$$

Despejando θ_1 se obtiene el modelo cinemático inverso del robot, resultando en la ecuación (6):

$$\theta_1 = \tan^{-1} \left(\frac{P_x}{P_y} \right) \quad (6)$$

2.4. Modelos dinámicos

2.4.1. Ecuaciones de movimiento de Lagrange

El modelo matemático dinámico de un robot puede ser obtenido mediante las ecuaciones de movimiento de Lagrange. En este método es preciso definir la energía cinética k y potencial u del sistema, en este caso un robot.

El Lagrangiano de un robot manipulador se define como la diferencia entre su energía cinética y su energía potencial.

$$\mathcal{L}(q(t), \dot{q}(t)) = k(q(t), \dot{q}(t)) - u(q(t)) \quad (7)$$

Las ecuaciones de movimiento de Lagrange para un manipulador de n g.d.l. viene dada por la siguiente ecuación:

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial L(q, \dot{q})}{\partial q} = \tau \quad (8)$$

En forma general, para cada eslabón:

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}_i} \right] - \frac{\partial L(q, \dot{q})}{\partial q_i} = \tau_i \quad (9)$$

donde τ_i son los pares ejercidos por los actuadores en cada una de las articulaciones, incluyendo sus fuerzas no conservativas. Cabe mencionar que se tendrán tantas ecuaciones dinámicas como eslabones tenga el robot.

La metodología para obtener el modelo dinámico de un robot manipulador por el método de Lagrange puede reducirse a cuatro fases [2]:

1. Cálculo de la energía cinética.
2. Cálculo de la energía potencial.
3. Cálculo de Lagrangiano.
4. Desarrollo de las ecuaciones de Lagrange.

Una vez desarrolladas las ecuaciones de movimiento de Lagrange, éstas representan el modelo dinámico del robot. Este modelo puede ser empleado en simulaciones numéricas para su estudio.

2.4.2. Modelo dinámico general de robots manipuladores

Haciendo uso de las ecuaciones de movimiento de Lagrange, es posible obtener una ecuación diferencial compacta como modelo dinámico general para robots manipuladores. Este modelo puede o no considerar los efectos de la fuerza de fricción en sus articulaciones (en este trabajo no se toman en cuenta estas fuerzas).

Considerando la energía cinética del robot manipulador:

$$k(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q}$$

Donde $M(q)$ es una matriz simétrica definida positiva que contiene los términos de masa del manipulador de dimensión $n \times n$, conocida como matriz de inercias. La energía potencial del sistema $u(q)$ depende solo de la ubicación en el espacio de las articulaciones.

La ecuación del Lagrangiano del robot manipulador queda expresada de la siguiente manera:

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} - u(q)$$

A partir de la ecuación anterior se determina la ecuación de movimiento de Lagrange:

$$\frac{d}{dt} \left[\frac{\partial}{\partial \dot{q}} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] \right] - \frac{\partial}{\partial q} \left[\frac{1}{2} \dot{q}^T M(q) \dot{q} \right] + \frac{\partial u(q)}{\partial q} = \tau \quad (10)$$

Realizando las operaciones en (10), se obtiene la siguiente expresión:

$$M(q)\ddot{q} + \dot{M}(q)\dot{q} - \frac{1}{2} \frac{\partial}{\partial \dot{q}} \left[\dot{q}^T M(q) \dot{q} \right] + \frac{\partial u(q)}{\partial q} = \tau$$

En forma general esta ecuación puede presentarse de la siguiente manera:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (11)$$

donde

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

$$g(q) = \frac{\partial u(q)}{\partial q}$$

El modelo de ecuación dinámica de un manipulador de n g.d.l queda expresada en (11). Aquí $C(q, \dot{q})\dot{q}$ es un vector de $n \times 1$ denominado vector de fuerzas centrífugas o de Coriolis, y $g(q)$ es el vector de fuerzas o pares gravitacionales, de dimensión $n \times 1$. Éste puede ser obtenido con relativa sencillez unicamente conociendo la energía potencial del robot. El vector τ contiene las fuerzas externas al robot que generalmente son las producidas por los actuadores [2].

La ecuación (11) no toma en cuenta un fenómeno presente en todos los sistemas mecánicos donde existe contacto entre materiales, es decir la fricción. Dicho fenómeno se presenta en los robots principalmente en las uniones o articulaciones. Éste puede ser modelado de manera estática y local mediante el vector $f(\dot{q})$:

$$f(\dot{q}) = \begin{bmatrix} f_1(\dot{q}_1) \\ f_2(\dot{q}_2) \\ \vdots \\ f_n(\dot{q}_n) \end{bmatrix}$$

El modelo clásico de fricción estática combina los fenómenos de fricción viscosa y de Coulomb de modo que el vector $f(\dot{q})$ presenta necesariamente los componentes:

$$f(\dot{q}) = F_a \dot{q} + F_b \text{sign}(\dot{q})$$

donde F_a y F_b representan matrices de dimensión $n \times n$ diagonales definidas positivas cuyos elementos son los parámetros de fricción viscosa y de Coulomb respectivamente. El termino $\text{sign}(\dot{q})$ es un vector cuyos elementos se pueden representar como:

$$\text{sign}(\dot{q}) = \begin{bmatrix} \text{sign}(\dot{q}_1) \\ \text{sign}(\dot{q}_2) \\ \vdots \\ \text{sign}(\dot{q}_n) \end{bmatrix}$$

donde $\text{sign}()$ es la función signo.

El modelo general para robots de n grados de libertad considerando fricción [2], (modelando de forma estática y de manera que estas fuerzas contrarrestan el torque suministrado por el actuador) es el siguiente:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) + f(\dot{q}) = \tau \quad (12)$$

2.4.3. Modelo dinámico lineal del actuador

En general los actuadores que suministran las fuerzas o pares para que cada eslabón se mueva, son de tipo electromecánico, pudiendo ser motores de CD. A continuación se describe la obtención del modelo matemático de un motor de CD. Para tal propósito en la figura 17 se presenta el esquema eléctrico y mecánico de un motor de CD. Éste además tiene acoplado en su eje un juego de engranes reductores de velocidad.

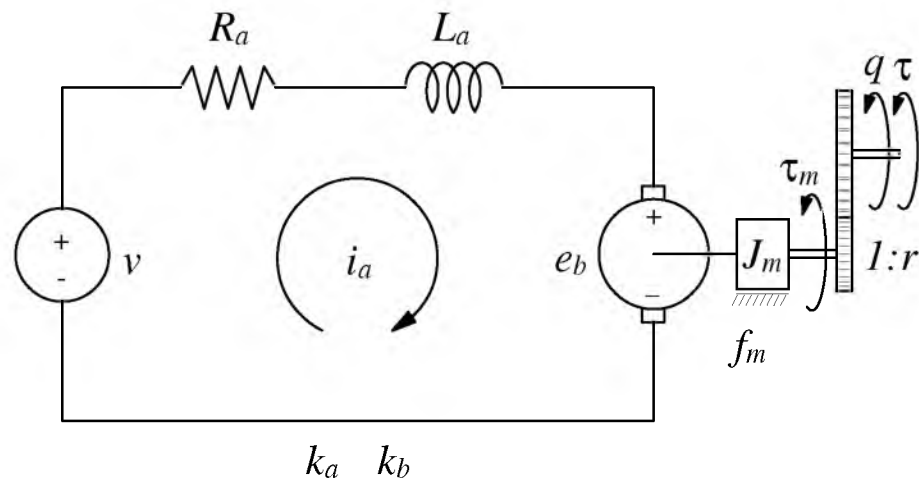


Figura 17: Esquema electromecánico de motor de C.D. controlado por corriente de armadura con caja reductora de engranajes acoplado.

Los parámetros eléctricos característicos de un motor de C.D. controlado por corriente de armadura, se enlistan a continuación:

- i_a : corriente de armadura.
- v : voltaje de armadura.
- e_b : voltaje de fuerza contraelectromotriz.
- L_a : inductancia de armadura.
- R_a : resistencia de armadura.

- K_a : constante par-motor.
- K_b : constante de contrarreacción electromotriz.

Aplicando la ley de voltajes de Kirchoff al circuito eléctrico del motor de C.D. se obtiene la siguiente ecuación:

$$v = R_a i_a + L_a \frac{di_a}{dt} + e_b \quad (13)$$

Por otra parte se sabe que el torque τ_m que suministra el motor de C.D. cumple con la ecuación:

$$\tau_m = K_a i_a \quad (14)$$

Así mismo, el voltaje de fuerza contra electromotriz relaciona directamente la velocidad del eje de la carga del motor, con la constante de contrarreacción electromotriz K_b :

$$e_b = K_b \dot{q}_m \quad (15)$$

En este caso se tiene acoplado un sistema de reducción de engranajes al eje del motor, la relación de reducción r satisface:

$$q_m = r q \quad (16)$$

donde q_m es la velocidad del eje del motor antes de la carga.

El modelo dinámico de un motor de CD. no sólo involucra los parámetros eléctricos descritos. Dado que es a la vez un sistema mecánico, se deben considerar parámetros mecánicos tales como el par neto aplicado después del juego de engranajes τ , la inercia del rotor J_m y el coeficiente de fricción debido a los soportes del rotor f_m . La ecuación que describe el movimiento involucrando de los parámetros mecánicos del motor de CD. es:

$$J_m \ddot{q}_m = \tau_m - f_m(\dot{q}_m) - \frac{\tau}{r} \quad (17)$$

Las ecuaciones diferenciales (13) y (17) describen la dinámica del motor de CD. Dichas ecuaciones pueden relacionarse al observar que la entrada en voltaje v

produce un determinado torque τ en la salida del sistema [2]. Una expresión que relacione ambas variables puede ser obtenida al utilizar las expresiones para i_a y e_b dadas en las ecuaciones (14) y (15) respectivamente en la ecuación (13) :

$$v = \frac{R_a}{K_a} \tau_m + L_a \frac{di_a}{dt} + K_b \dot{q}_m \quad (18)$$

Sustituyendo la derivada de la corriente de la ecuación (18) tomado la derivada respecto del tiempo de la ecuación (14), de modo que se reescribe como:

$$v = \frac{R_a}{K_a} \tau_m + \frac{L_a}{K_a} \dot{\tau}_m + K_b \dot{q}_m \quad (19)$$

Para sustituir τ_m de la ecuación (19) se recurre a la ecuación de movimiento mecánico (17) donde al despejar τ_m se obtiene:

$$\tau_m = J_m \ddot{q}_m + f_m(\dot{q}_m) + \frac{\tau}{r}$$

Sin embargo, aun es necesario encontrar $\dot{\tau}_m$ para poder reemplazarla en la ecuación (19) de modo que:

$$\dot{\tau}_m = J_m \frac{d}{dt} \ddot{q}_m + \frac{\partial f_m(\dot{q}_m)}{\partial \dot{q}_m} + \frac{\dot{\tau}}{r}$$

Al sustituir τ_m y $\dot{\tau}_m$ en la ecuación (19) y teniendo en cuenta que $q_m = r q$ se obtiene:

$$\frac{K_a}{r R_a} v = \frac{L_a J_m}{R_a} \frac{d}{dt} \ddot{q} + \left[J_m + \frac{L_a}{R_a} \frac{\partial f_m(r \dot{q})}{\partial \dot{q}_m} \right] \ddot{q} + \frac{1}{r} f_m(r \dot{q}) + \frac{K_a K_b}{R_a} \dot{q} + \frac{\tau}{r^2} + \frac{L_a}{r^2 R_a} \dot{\tau} \quad (20)$$

La ecuación (20) describe el comportamiento dinámico de un motor de C.D. relacionando el voltaje de entrada del motor con el torque obtenido en el eje despues del juego de engranaje. Sin embargo hacer uso de dicha ecuación no es práctico debido a su complejidad. Teniendo en cuenta que en la mayoría de los casos la inductancia de armadura L_a es (a menudo) cercana a cero, ésta puede ser despreciada, de modo que la ecuación (20) se reescribe como:

$$J_m \ddot{q} + \frac{1}{r} f_m(r \dot{q}) + \frac{K_a K_b}{R_a} \dot{q} + \frac{\tau}{r^2} = \frac{K_a}{r R_a} v \quad (21)$$

Esta expresión puede simplificarse más en la ecuación si se toma en cuenta

que el término de fricción $\frac{1}{r}f_m(r\dot{q})$ puede modelarse como:

$$\frac{1}{r}f_m(r\dot{q}) = \frac{1}{r}f_m r\dot{q} = f_m\dot{q}$$

donde f_m es una constante positiva que generalmente es proporcionada por el fabricante del actuador. Reescribiendo la ecuación (21) con la consideración de fricción se obtiene:

$$J_m\ddot{q} + \left[f_m + \frac{K_a K_b}{R_a} \right] \dot{q} + \frac{\tau}{r^2} = \frac{K_a}{r R_a} v \quad (22)$$

La ecuación (22) es un modelo dinámico lineal simplificado y de mayor utilidad práctica (debido a su simplicidad) de un motor de C.D. con sistema reductor de engranaje acoplado.

2.4.4. Modelo dinámico general de robots manipuladores con actuadores

Teniendo en consideración que el modelo lineal presentado en la ecuación (22) es la idealización matemática para un solo actuador, además que normalmente los robots están constituidos por mas de un eslabón y un actuador, este modelo puede expresarse de forma general para n actuadores presentes en las n articulaciones de un robot como:

$$J\ddot{q} + B\dot{q} + R\tau = Kv \quad (23)$$

donde J , B , R y K son matrices diagonales cuyos elementos son:

$$J = \text{diag} \{ J_{m_i} \}$$

$$B = \text{diag} \left\{ f_{m_i} + \left(\frac{K_a K_b}{R_a} \right)_i \right\}$$

$$R = \text{diag} \left\{ \frac{1}{r_i^2} \right\}$$

$$K = \text{diag} \left\{ \left(\frac{K_a}{R_a} \right)_i \frac{1}{r_i} \right\}$$

El subíndice i indica que a cada elemento de la diagonal le corresponde un

actuador utilizado en el robot desde $i = 1$ hasta $i = n$. Teniendo una expresión que incluya todos los parámetros y dinámicas de los actuadores del robot, se obtiene el modelo dinámico general para robots con motores en sus articulaciones y considerando fricción en ellas [2]. Sustituyendo τ de la ecuación (23) en (12):

$$[RM(q) + J]\ddot{q} + RC(q, \dot{q})\dot{q} + Rg(q) + Rf(\dot{q}) + B\dot{q} = Kv \quad (24)$$

2.4.5. Modelo dinámico de robot pendular sin actuador

El modelo dinámico del robot pendular, sin considerar la dinámica aportada por el actuador, se obtiene utilizando las ecuaciones de movimiento de Euler-Lagrange. Para ello es necesario realizar un diagrama simplificado del robot como el que se muestra en la figura 19. Aquí se presenta el sistema ubicado en el cuarto cuadrante del sistema de coordenadas cartesiano.

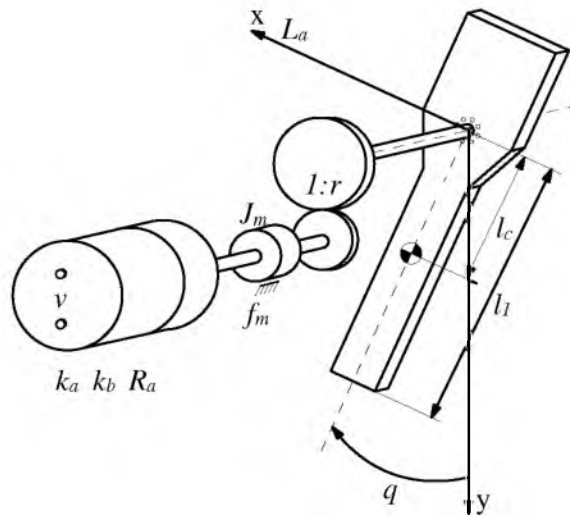


Figura 18: Robot Pendular de 1 GDL, provisto de motor de C.D. como actuador.

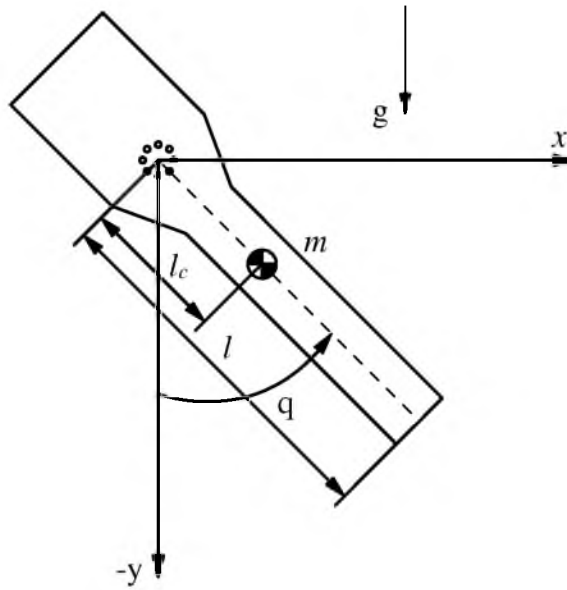


Figura 19: Diagrama de robot pendular, sin considerar fricción en sus soportes. provisto de motor de C.D. como actuador.

El Lagrangiano viene dado por la ecuación:

$$\mathcal{L}(q, \dot{q}) = K(q, \dot{q}) - U(q)$$

donde $K(q, \dot{q})$ es la energía cinética del robot y $U(q)$ la energía potencial, de modo que es preciso conocer ambas.

Las coordenadas al centro de masa son:

$$x_1 = l_c \text{sen}(\theta)$$

$$y_1 = -l_c \text{cos}(\theta)$$

Debido a que la energía cinética se obtiene a partir de la expresión $K = \frac{1}{2}mv^2$, su equivalente vectorial $K = \frac{1}{2}mv^T v$ puede obtenerse derivando las coordenadas del centro de masa para construir el vector velocidad:

$$v = \begin{bmatrix} l_c \text{cos}(\theta) \dot{\theta} \\ l_c \text{sen}(\theta) \dot{\theta} \end{bmatrix}$$

Para calcular el cuadrado de la rapidez en la ecuación de energía cinética, se realiza el siguiente producto:

$$v^T v = \begin{bmatrix} l_c \cos(\theta) \dot{\theta} & l_c \sin(\theta) \dot{\theta} \end{bmatrix} \begin{bmatrix} l_c \cos(\theta) \dot{\theta} \\ l_c \sin(\theta) \dot{\theta} \end{bmatrix}$$

$$v^T v = \left[l_c^2 \cos^2(\theta) \dot{\theta}^2 + l_c^2 \sin^2(\theta) \dot{\theta}^2 \right]$$

$$v^T v = \left[l_c^2 \dot{\theta}^2 (\cos^2(\theta) + \sin^2(\theta)) \right]$$

$$v^T v = l_c^2 \dot{\theta}^2$$

de modo que la energía cinética del robot está dada por:

$$K = \frac{1}{2} m_c l_c^2 \dot{\theta}^2 \quad (25)$$

La energía potencial U del robot se expresa de la forma:

$$U = m_c g h$$

donde m_c es la masa en el centro de masa, g es la aceleración de la gravedad y h la longitud al centro de masa menos la componente vertical al centro de masa, de modo que la energía potencial del robot es:

$$U = m_c g l_c [1 - \cos(\theta)] \quad (26)$$

haciendo uso de las ecuaciones (25) y (26) el Lagrangiano del robot pendular se expresa como:

$$\mathcal{L}(q, \dot{q}) = \frac{1}{2} m_c l_c^2 \dot{\theta}^2 - m_c g l_c [1 - \cos(\theta)] \quad (27)$$

La ecuación de movimiento de Euler-Lagrange es:

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right] - \frac{\partial L(q, \dot{q})}{\partial q} = \tau$$

Sustituyendo el Lagrangiano obtenido en la ecuación de movimiento de Euler-Lagrange se obtiene:

$$\frac{d}{dt} \left[\frac{\partial \frac{1}{2} m_c l_c^2 \dot{\theta}^2 + m_c g l_c \cos(\theta)}{\partial \dot{\theta}} \right] - \frac{\partial \frac{1}{2} m_c l_c^2 \dot{\theta}^2 - m_c g l_c [1 - \cos(\theta)]}{\partial \theta} = \tau$$

$$\frac{\partial L(q, \dot{q})}{\partial q} = -m_c g l_c \sin(\theta)$$

$$\frac{\partial L(q, \dot{q})}{\partial \dot{q}} = m_c l_c^2 \dot{\theta}$$

$$\frac{d}{dt} \left[\frac{\partial L(q, \dot{q})}{\partial \dot{q}} \right] = m_c l_c^2 \ddot{\theta}.$$

Finalmente el modelo dinámico del robot pendular es:

$$\tau = m_c l_c^2 \ddot{\theta} + m_c g l_c \sin(\theta).$$

Este modelo puede ser expresado en términos de la posición articular q :

$$\tau = m_c l_c^2 \ddot{q} + m_c g l_c \sin(q) \quad (28)$$

donde $M(q) = m_c l_c^2$, $C(q, \dot{q}) = 0$ y $g(q) = m_c g l_c \sin(q)$.

2.4.6. Modelo dinámico de robot pendular con actuador lineal

El modelo dinámico del robot pendular, no solo incluye la parte mecánica del mismo, ecuación (28), sino que además al estar provisto por un actuador del tipo electromecánico, (un motor de CD.) ésta dinámica debe incluirse en el modelo dinámico del robot.

Este modelo mostrado en la ecuación (28) puede escribirse en forma compacta como:

$$J_L \ddot{q} + K_L \sin(q) = \tau$$

donde

$$J_L = m_c l_c^2$$

$$K_L = m_c g l_c$$

Utilizando el modelo dinámico lineal del actuador, ecuación (23), se obtiene el modelo dinámico considerando ambas dinámicas. Sustituyendo τ en dicho modelo:

$$\left[J_m + \frac{J_L}{r^2} \right] \ddot{q} + \left[f_m + \frac{K_a K_b}{R_a} \right] \dot{q} + \frac{K_L}{r^2} \text{sen}(q) = \frac{K_a}{r R_a} v \quad (29)$$

Este modelo puede expresarse en forma compacta con relación a la ecuación (24), donde se aprecia que:

$$M(q) = J_L \quad C(q, \dot{q}) = 0 \quad f(\dot{q}) = 0$$

$$R = \frac{1}{r^2} \quad J = J_m \quad g(q) = K_L \text{sen}(q)$$

$$B = f_m + \frac{K_a K_b}{R_a} K = \frac{K_a}{r R_a}$$

Finalmente cabe mencionar que, el termino de fricción estática $f(\dot{q})$ es igual a cero en este modelo debido a que el eslabón esta directamente acoplado al eje de salida del motor después del juego de engranajes como se muestra en la figura 18, evitando así la fricción en los rodamientos de soporte si estos estuvieran presentes.

3. Estrategias de control de posición para robots

3.1. Introducción

En un brazo manipulador se pueden considerar dos problemas principalmente: hacer que el efector final del robot se ubique en una posición deseada en su entorno de trabajo y que se mueva por una trayectoria deseada para llegar a dicho punto. La manera más sencilla de controlar el movimiento de un robot es el denominado punto a punto, que consiste en dictarle al robot una serie de marcas en su espacio de trabajo hasta que él se posicione en el punto donde se desea. Ver figura 20.

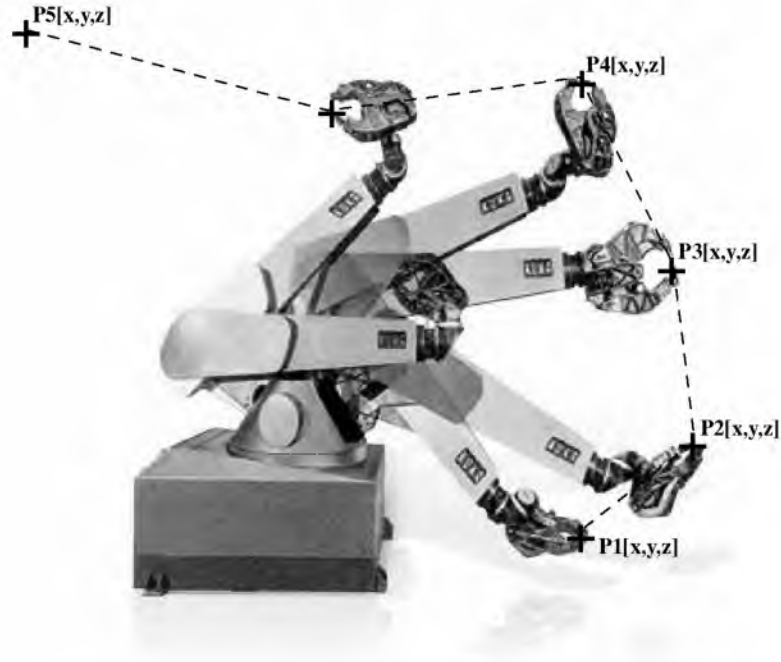


Figura 20: Ejemplo de puntos en el espacio de trabajo del robot, para control de movimiento.

El control de posición puro es un caso particular del control de movimiento de robots manipuladores. Este tipo de control consiste en dado un vector de posición articulares deseado q_d , encontrar una función vectorial de pares τ , tales que lleven las posiciones articulares q asociadas a cada articulación del robot a las posiciones deseadas en q_d de manera asintótica. Con base en lo anterior, para poder controlar el movimiento de un robot, es preciso tener a nuestra disposición sus posiciones q y velocidades articulares \dot{q} , éstas pueden ser estimadas por medio de un observador de estados u obteniéndolas a partir de sensores para el control del robot [2].

El método de retroalimentación negativa consiste, en dado el vector de posición articulares deseadas q_d restarle los valores medidos. La resta de $q_d - q$ representa la desviación o error en la posición actual de la posición articular del robot respecto a la posición deseada. Éste error es compensado por un controlador, que proporciona valores de fuerza necesarios para compensar el error mediante pares τ adecuados en cada actuador a fin de lograr que $q_d = q$. En la figura 21 se presenta un diagrama a bloques del sistema de lazo cerrado de un robot con controlador por

retroalimentación negativa.

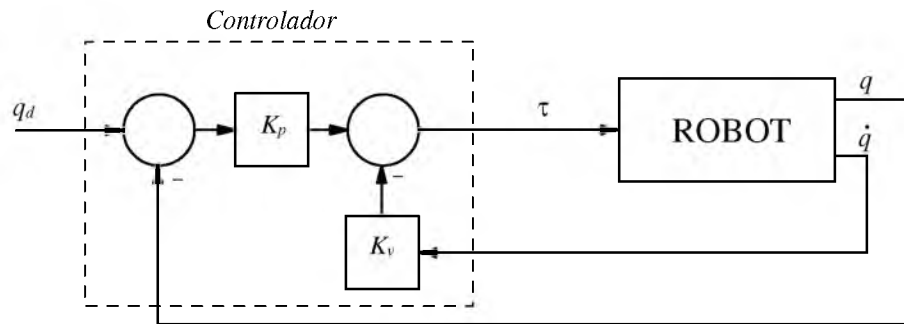


Figura 21: Control por retroalimentación negativa de un robot, nótese la presencia del controlador que genera los pares τ (a partir del error) que son introducidos al robot.

Existen dos tipos de controladores de acuerdo a las necesidades del manipulador. Para el caso del robot pendular, las estrategias de control implementadas son:

- Control Proporcional Derivativo (PD)
- Control Proporcional Derivativo con compensación precalculada de gravedad.

3.2. Estabilidad en el sentido de Lyapunov

Los modelos dinámicos que describen el comportamiento de los robots, son por lo general ecuaciones diferenciales no lineales, por lo que estudiar su estabilidad va más allá de los métodos empleados en el control clásico. Sin embargo, existe una teoría propuesta por Aleksander Lyapunov; la cual permite analizar el comportamiento de ecuaciones diferenciales no lineales en términos de estabilidad.

Para revisar el método directo de Lyapunov, es preciso conocer los conceptos en los que se basa; los cuales son: equilibrio, estabilidad, estabilidad asintótica y estabilidad exponencial [2].

3.2.1. Equilibrio

Un punto de equilibrio es aquel en el cual la dinámica del sistema es igual a cero. Una ecuación diferencial puede tener uno o más puntos de equilibrios,

pudiendo inclusive tener infinitos puntos, pero también puede ocurrir el caso donde no se tenga puntos de equilibrio.

Dado un vector constante $x_e \in \mathbb{R}^n$ que se evalúa en una función vectorial tal que:

$$f(t, x_e) = 0 \quad \forall t \geq 0 \quad (30)$$

Entonces dicho vector es un punto de equilibrio. Por otra parte si la condición inicial $x(0) \in \mathbb{R}^n$ coincide con ser un punto de equilibrio $x(0) = x_e \in \mathbb{R}^n$ se cumple que:

- $x(t) = x_e \quad \forall t \geq 0$
- $\dot{x}(t) = 0 \quad \forall t \geq 0$

Es decir; que la evolución en el tiempo de la función solución es el mismo punto de equilibrio (constante) lo cual implica que la derivada de la solución es cero. Un ejemplo ilustrativo en \mathbb{R}^2 donde $x(0)$ es justamente x_e se muestra en la figura 22.

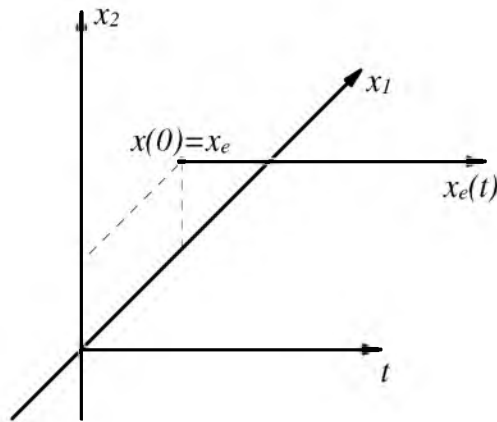


Figura 22: Ejemplo de equilibrio cuando $x(0) = x_e$ en \mathbb{R}^2 .

El origen del espacio de estados \mathbb{R}^n es decir $x = 0$, se supone normalmente como un equilibrio de la ecuación (30), de no ser el caso, puede aplicarse un cambio en las coordenadas, de modo que todo equilibrio puede ser trasladado al origen [2].

3.2.2. Estabilidad

En consecuencia al estudio de los puntos de equilibrio de las ecuaciones diferenciales, es preciso conocer también la estabilidad de los mismos, es decir; la forma en que la solución evoluciona con la condición inicial considerada como equilibrio, teniendo el supuesto de que el origen del campo vectorial es un punto de equilibrio $x = 0 \in \mathbb{R}^n$.

El origen de la ecuación (30) $x = 0 \in \mathbb{R}^n$ es estable en el sentido de Lyapunov si existe un número $\varepsilon > 0$ para el cual es posible encontrar un número $\delta > 0$ tal que:

$$\|x(0)\| < \delta \implies \|x(t)\| < \varepsilon \quad \forall t \geq 0$$

Es decir, que la solución que parte de una condición inicial acotada, es de igual modo acotada. Es importante considerar que la discrepancia entre el origen y las condiciones iniciales deben ser pequeñas. En la figura 23 se ilustra el caso para $x(0) \in \mathbb{R}^2$ considerando al origen $x = 0 \in \mathbb{R}^2$ como un equilibrio estable.

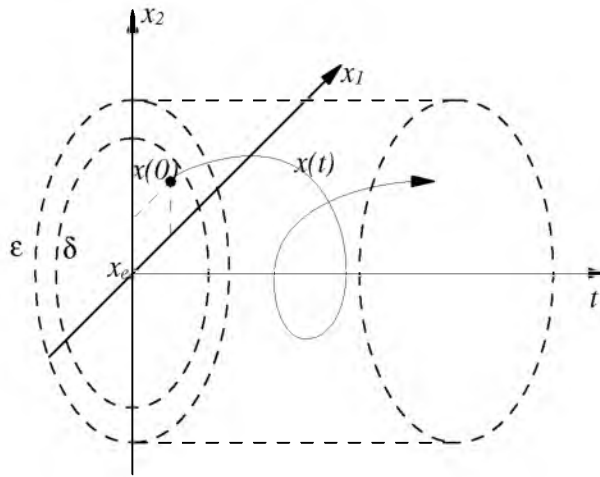


Figura 23: Ejemplo de estabilidad en el sentido de Lyapunov en el espacio vectorial \mathbb{R}^2 .

En la figura 23 se observa que la trayectoria de la solución $x(t)$ esta acotada por ε debido a que la condición inicial lo esta por δ .

3.2.3. Estabilidad asintótica

El origen de la ecuación (30) $x = 0 \in \mathbb{R}^n$ es asintóticamente estable si el origen es estable y además es atractivo, pues existe un número $\delta' > 0$ de modo que:

$$\|x(0)\| < \delta' \implies \|x(t)\| \rightarrow 0 \quad \text{cuando} \quad t \rightarrow \infty$$

es decir que la solución $x(t)$ acotada tiende a cero cuando ha transcurrido un tiempo considerable. En la figura 24 se ilustra el caso para $x(0) \in \mathbb{R}^2$.

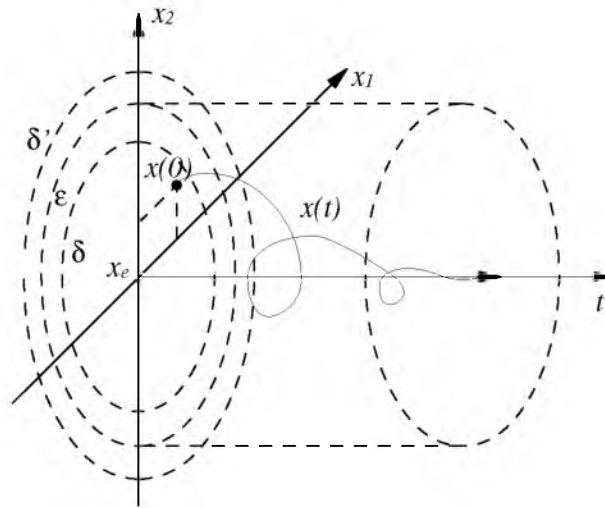


Figura 24: Ejemplo de estabilidad asintótica para el caso de $x(0) \in \mathbb{R}^2$.

3.2.4. Estabilidad asintótica global

El origen de la ecuación (30) $x = 0 \in \mathbb{R}^n$ es asintóticamente estable en forma global si el origen es estable y además es atractivo:

$$\|x(t)\| \rightarrow 0 \quad \text{cuando} \quad t \rightarrow \infty, \quad \forall x(0) \in \mathbb{R}^n$$

Lo cual implica que el origen no está acotado por la constante δ' .

3.2.5. Estabilidad exponencial global

La estabilidad exponencial global implica que la solución $x(t)$ tiende a cero de forma acotada por una función exponencial, si existen constantes α y β tales que:

$$\|x(t)\| < \alpha \|x(0)\| e^{-\beta t}, \quad \forall t \geq 0, \quad \forall x(0) \in \mathbb{R}^n$$

por lo que el origen de la ecuación (30) $x = 0 \in \mathbb{R}^n$ es un equilibrio exponencialmente estable en forma global.

3.2.6. Inestabilidad

El origen de la ecuación (30) $x = 0 \in \mathbb{R}^n$ es inestable si no existe al menos un número $\varepsilon > 0$ para el cual no es posible encontrar un $\delta > 0$ tal que:

$$\|x(0)\| < \delta \implies \|x(t)\| < \varepsilon \quad \forall t \geq 0$$

3.3. Método directo de Lyapunov

Para comprender mejor la aplicación del método directo de Lyapunov para estudiar la estabilidad de equilibrios, es preciso antes conocer qué es una función de Lyapunov y algunas de sus propiedades, las cuales se describen a continuación.

3.3.1. Función de Lyapunov

La derivada de la función de Lyapunov juega un papel importante en el estudio de la estabilidad del equilibrio en el origen. Por tal motivo se debe establecer con anterioridad una función candidata de Lyapunov $V(t, x)$ para el equilibrio $x = 0 \in \mathbb{R}^n$ de la ecuación $\dot{x} = f(t, x)$ la cual debe cumplir que [2]:

- $V(t, x)$ es una función definida positiva localmente.
- $\frac{\partial V(t, x)}{\partial t}$ es una función continua con respecto a t y x .
- $\frac{\partial V(t, x)}{\partial x}$ es una función continua con respecto a t y x globalmente.

Por otra parte la derivada $\dot{V}(t, x)$ de la función de Lyapunov puede expresarse como:

$$\dot{V}(t, x) = \frac{d}{dt} V(t, x) = \frac{\partial V(t, x)}{\partial t} + \frac{\partial V(t, x)^T}{\partial x} f(t, x)$$

cuando $V(t, x)$ no depende explícitamente del tiempo, entonces su derivada se escribe como:

$$\dot{V}(t, x) = \frac{\partial V(t, x)^T}{\partial x} f(t, x)$$

Finalmente puede establecerse que una función candidata de Lyapunov $V(t, x)$ para $\dot{x} = f(t, x)$ es una función de Lyapunov si su derivada a lo largo de las trayectorias de $\dot{x} = f(t, x)$ cumple que:

$$\dot{V}(t, x) \leq 0 \quad \forall t \geq 0 \quad \text{al menos para } \|x\| \text{ pequeña}$$

El método directo de Lyapunov se basa en una serie de teoremas que lo fundamentan, a continuación se presenta una descripción no rigurosa matemáticamente, para cada uno de ellos.

3.3.2. Estabilidad

Si para una función candidata de Lyapunov $V(t, x)$ su derivada temporal globalmente cumple que:

$$\dot{V}(t, x) \leq 0 \quad \forall t \geq 0 \quad \text{al menos para } \|x\| \text{ pequeña}$$

el origen $x = 0 \in \mathbb{R}^n$ es un estado de equilibrio estable de $\dot{x} = f(t, x)$. Cabe mencionar que al establecerse que «al menos para valores de x pequeños» se entiende que la función de Lyapunov es definida positiva localmente, puede ser válido el teorema también si fuese definida positiva globalmente.

3.3.3. Estabilidad y acotamiento de las soluciones

El origen de la ecuación (30) $x = 0 \in \mathbb{R}^n$ es un equilibrio estable, y sus soluciones $x(t)$ son acotadas para todas las posibles condiciones iniciales $x(0) \in \mathbb{R}^n$, cuando existe una función de Lyapunov $V(t, x)$ tal que:

$$\dot{V}(t, x) \leq 0 \quad \forall t \geq 0 \quad \forall x \in \mathbb{R}^n$$

se asume por tanto que $V(t, x)$ es definida positiva globalmente, radialmente des-
acotada.

3.3.4. Estabilidad asintótica global

Cuando la derivada de una función de Lyapunov $V(t, x)$ definida positiva globalmente, radialmente desacotada y menguante satisface las siguientes condiciones:

- $\dot{V}(t, 0) = 0 \quad \forall t \geq 0$
- $\dot{V}(t, x) < 0 \quad \forall t \geq 0, \quad \forall x \neq 0 \in \mathbb{R}^n$

el origen $x = 0 \in \mathbb{R}^n$ es un estado de equilibrio asintóticamente estable en forma global de $\dot{x} = f(t, x)$. Lo cual implica que $\dot{V}(t, x)$ sea definida negativa globalmente puesto que no puede valer cero.

3.4. Control Proporcional Derivativo (PD)

El control proporcional derivativo (PD), está basado en el control de posición con retroalimentación de velocidad. Éste es el controlador más simple para control de posición pura de robots. El control de posición con retroalimentación de velocidad, únicamente utiliza el error en posición. El diagrama a bloques del control de posición con retroalimentación se muestra en la figura 25.

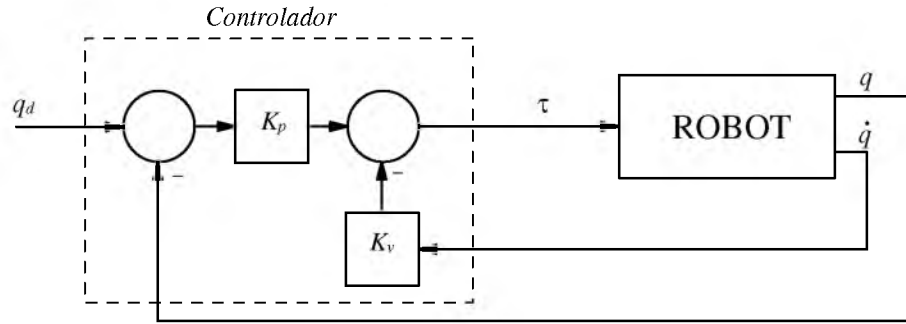


Figura 25: Diagrama a bloques del control de posición con retroalimentación de velocidad.

La ley de control que rige el esquema de control de posición con retroalimentación de velocidad mostrado en la figura 25 es la que sigue:

$$\tau = K_p \tilde{q} - K_v \dot{q} \quad (31)$$

donde K_p , K_v son matrices simétricas definidas positivas de dimensión $n \times n$, cuyos valores son determinados por el diseñador, y son conocidas como ganancias

proporcional y derivativa o de velocidad respectivamente. El vector \tilde{q} es el vector del error de posición y se expresa como:

$$\tilde{q} = q_d - q \quad \in \mathbb{R}^n.$$

El vector q_d contiene las posiciones articulares deseadas, y el vector q las posiciones articulares reales del robot, en consecuencia el vector \dot{q} contiene las velocidades angulares articulares reales del robot. La aplicación típica de este tipo de controladores es en el control de posición angular de motores de CD.

Una mejora sustantiva a este controlador se presenta en el control Proporcional Derivativo cuya ley de control no solo incluye el error de posición \tilde{q} , sino además el error en velocidad articular del robot [2]. La ley de control para el control PD esta dada por:

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} \quad (32)$$

donde, de igual modo, K_p , K_v son matrices simétricas definidas positivas de dimensión $n \times n$, determinados por el diseñador. El diagrama a bloques de este controlador PD se muestra en la figura 26.

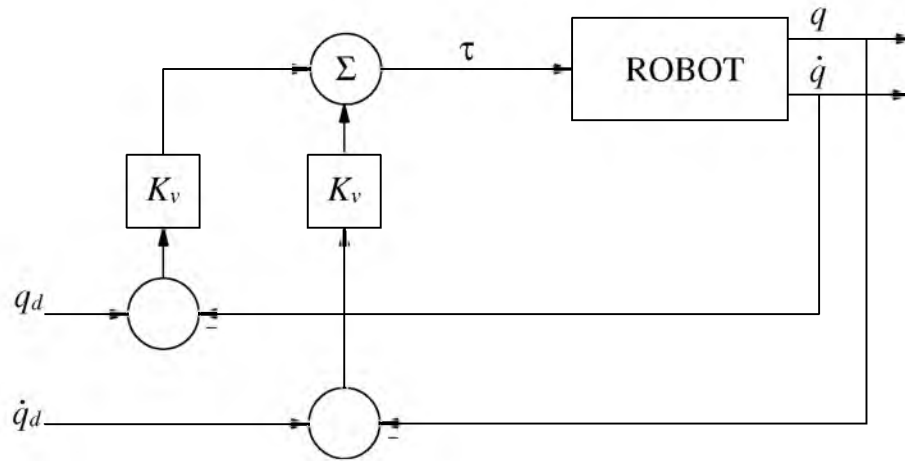


Figura 26: Diagrama a bloques del control proporcional derivativo (PD).

El vector de error en velocidad $\dot{\tilde{q}}$ está determinado por:

$$\dot{\tilde{q}} = \dot{q}_d - \dot{q} \quad \in \mathbb{R}^n$$

Cuando el vector de posiciones deseadas \dot{q}_d es constante, el vector de error de velocidad se expresa como $\dot{\tilde{q}} = -\dot{q}$ que al sustituirse en la ecuación (32) se obtiene:

$$\tau = K_p \tilde{q} + K_v (-\dot{q})$$

$$\tau = K_p \tilde{q} - K_v \dot{q}$$

Para tal caso, las leyes de control proporcional con retroalimentación de velocidad (31) y proporcional derivativo PD (32) son iguales, de modo que solo es necesario especificar el vector de posiciones deseadas q_d y ambas leyes de control pueden utilizarse genéricamente.

El vector τ en el modelo del control PD, contiene los pares requeridos en cada actuador del robot, para llevar los eslabones a las posiciones articulares deseadas. Por tal motivo el modelo matemático para robots con n articulaciones rotacionales se obtiene al sustituir el vector de pares τ en (11):

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = K_p \tilde{q} - K_v \dot{q} \quad (33)$$

El estudio del controlador se lleva a cabo en el supuesto de que los actuadores son fuentes ideales de torque, despreciando así su dinámica, y además sin considerar fricción en las articulaciones, lo cual es suficiente para realizar el análisis del comportamiento del sistema en lazo cerrado en su conjunto bajo la ley de control, siendo el motivo por el cual el término $f(\dot{q})$ no se incluye en la ecuación (33).

La ecuación (33) puede reescribirse en términos del vector de estado $\begin{bmatrix} \tilde{q}^T & \dot{\tilde{q}}^T \end{bmatrix}^T$ como:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ \ddot{q}_d - M(q)^{-1} [K_p \tilde{q} - K_v \dot{q} - C(q, \dot{q})\dot{q} - g(q)] \end{bmatrix}$$

considerando que el vector de posiciones articulares deseadas q_d es constante, el vector de estado resulta ser $\begin{bmatrix} \tilde{q}^T & \dot{\tilde{q}}^T \end{bmatrix}^T$ y por tanto la ecuación de lazo cerrado es:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(q)^{-1} [K_p \tilde{q} - K_v \dot{q} - C(q, \dot{q})\dot{q} - g(q)] \end{bmatrix} \quad (34)$$

dato que q_d es constante la ecuación diferencial (34) es por tanto no lineal pero autónoma, y además tiene múltiples equilibrios, que están dados por

$\begin{bmatrix} \tilde{q}^T & \dot{q}^T \end{bmatrix}^T = \begin{bmatrix} s^T & 0^T \end{bmatrix}^T$ donde $s \in \mathbb{R}^n$ es solución de:

$$K_p s - g(q_d - s) = 0 \quad (35)$$

3.4.1. Equilibrio único

Como se puede observar en (35), si consideramos robots con sólo articulaciones rotacionales y que además el vector de pares gravitacionales $g(q)$ está presente en el modelo dinámico, el origen del modelo dinámico del robot (34) no es un punto de equilibrio. Sin embargo este modelo sí posee puntos de equilibrio; los cuales están dados por la ecuación (35).

Para lograr obtener que la ecuación de lazo cerrado posea equilibrio único, basta con seleccionar una ganancia K_p tal que:

$$\lambda_{\min} \{K_p\} > K_g$$

donde K_g es una constante que surge de una propiedad del vector de pares gravitacionales $g(q)$, que establece que existe una constante $K_g > 0$ tal que:

$$\|g(x) - g(y)\| \leq K_g \|x - y\|$$

para todo $x, y \in \mathbb{R}^n$. La cual puede ser calculada mediante:

$$K_g = n \left[\max_{i,j,q} \left| \frac{\partial g_i(q)}{\partial q_j} \right| \right] \quad (36)$$

donde n es el número de vectores de pares gravitacionales presentes en el modelo dinámico del robot.

3.4.2. Diseño de control PD para robot pendular

El controlador proporcional derivativo (PD), no hace uso explícitamente del modelo dinámico del robot, sin embargo para el cálculo de las ganancias proporcional y derivativa es posible pero no necesario considerar una parte del modelo, como se describe a continuación.

La ley de control proporcional derivativo (PD) está dada por:

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}}$$

Por otra parte el modelo dinámico del robot pendular RP1 considera la dinámica del actuador, y se escribe como:

$$\left[J_m + \frac{J_L}{r^2} \right] \ddot{q} + \left[f_m + \frac{K_a K_b}{R_a} \right] \dot{q} + K_L \text{sen}(q) = \frac{r K_a}{R_a} v \quad (37)$$

El modelo puede expresarse en forma compacta, en la forma de (24) donde:

$$M(q) = J_L \quad C(q, \dot{q}) = 0 \quad f(\dot{q}) = 0,$$

$$R = \frac{1}{r^2} \quad J = J_m \quad g(q) = K_L \text{sen}(q),$$

$$B = f_m + \frac{K_a K_b}{R_a} K = \frac{K_a}{r R_a},$$

y:

$$J_L = m_c l_c^2,$$

$$K_L = m_c g l_c.$$

Para la ecuación (37) es factible el uso del método de diseño utilizado en (11) para robots de n grados de libertad, ya que la diferencia no es suficientemente grande como para que no se pueda aplicar el mismo método de diseño. Para este tipo de control sólo es necesario seleccionar las matrices de ganancias K_p y K_v definidas positivas [2], pudiendo utilizar únicamente del modelo del robot el vector de pares gravitacionales $g(q)$, de modo que:

$$\lambda_{\min} \{K_p\} > K_g.$$

La constante positiva K_g se calcula mediante la ecuación (36), en la cual se puede apreciar que únicamente se utiliza el vector de pares gravitacionales del modelo del robot tal como se mencionó antes, de manera que; la expresión para calcular la constante K_g en el caso del robot pendular RP1 queda expresada como:

$$K_g = 1 \left[\max_{i,j,q} \left| \frac{\partial(m_c g l_c \sin(q))}{\partial q} \right| \right],$$

realizando la derivada parcial se tiene que

$$\frac{\partial m_c g l_c \sin(q)}{\partial q} = m_c g l_c \cos(q),$$

así una cota superior que se puede utilizar es

$$\max \left| \frac{\partial(m_c g l_c \sin(q))}{\partial q} \right| = m_c g l_c,$$

de modo que K_g se expresa como:

$$K_g = m_c g l_c.$$

La matriz de ganancia proporcional K_p debe cumplir con $\lambda_{\min} \{K_p\} > K_g$. Sin embargo, al tratarse solamente de un único eslabón rotacional, la matriz K_p es en realidad una constante escalar, siendo así que se debe seleccionar un valor tal que:

$$K_p > m_c g l_c$$

donde m_c es la masa del eslabón, g es la aceleración de la gravedad, y l_c es la distancia al centro de masa del eslabón. A continuación se en listan los valores numéricos para dichos parámetros:

- $m_c = 0.301kg$
- $g = 9.81m/s^2$
- $l_c = 0.0516m$

Considerando estos valores, la constante $K_g = 0.1528$. Tomando en cuenta que $K_p > 0.1528$, se propone un valor para $K_p = 20$, y un valor para K_v del 20 % de K_p , siendo esta proporción bastante aceptada entre los diseñadores de control de robots. Por lo tanto $K_v = 4$.

La ley de control PD para el robot pendular queda expresada finalmente como:

$$\tau = 20\tilde{q} + 4\dot{\tilde{q}} \tag{38}$$

donde \tilde{q} es el error de posición articular del eslabón, y $\dot{\tilde{q}}$ es el error en velocidad articular.

Si bien se puede calcular las ganancias proporcional y derivativas utilizando como referencia al vector de pares gravitacionales $g(q)$, con el fin de cumplir la condición $K_p > K_g$ cabe señalar que para el diseño de control PD solo basta con la selección de las ganancias como matrices simétricas definidas positivas. Esto puede apreciarse fácilmente si se piensa en robots cuyo modelo dinámico no incluye explícitamente el vector de pares gravitacionales.

3.5. Control PD con compensación precalculada de gravedad

El control proporcional derivativo con compensación precalculada de gravedad, surge como una mejora en la aplicación del controlador proporcional derivativo con compensación en gravedad. Es por ello que antes debe entenderse dicho controlador predecesor.

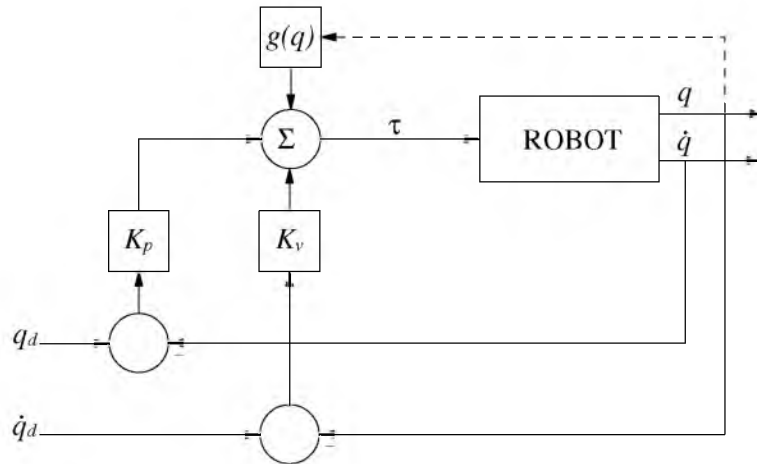


Figura 27: Diagrama a bloques del control PD con compensación de gravedad.

El diagrama a bloques del control PD con compensación de gravedad se muestra en la figura 27, y la ley de control se expresa como:

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + g(q) \quad (39)$$

De la ecuación (39) se observa que el controlador utiliza de manera explícita parte del modelo dinámico del robot, (el vector de pares gravitacionales $g(q)$) en

el cual se evalúa en tiempo real la posición articular real del eslabón del robot, es decir que, en una aplicación digital para tal controlador, cada vez que se muestrean las posiciones y velocidades articulares, se realiza el cálculo del vector $g(q)$ para completar la acción de control. Normalmente la frecuencia de muestreo suele ser alta, y dado que el vector de pares gravitacionales normalmente incluye funciones trigonométricas, resulta ser difícil calcularlo, lo que puede llegar a contribuir con la inclusión de tiempos de retardo para la ejecución del algoritmo de control, lo que puede desestabilizar el sistema [12][13][14][15].

Sin embargo es posible evitar este problema en la aplicación del controlador, si se toma en cuenta que se puede calcular de ante mano la contribución de par gravitacional por parte del vector de pares gravitacionales. Si se conoce la posición articular deseada q_d previamente a la ejecución del algoritmo de control dentro del sistema digital; se puede calcular dicha contribución solo una vez fuera de línea antes de computar la parte PD del controlador (de ahí el nombre de «precalculada de gravedad»), evitando así el problema con la alta frecuencia de muestreo al momento de calcular la contribución de $g(q)$ como en el control PD con compensación de gravedad [2].

Por tanto la ley de control proporcional derivativo con compensación precalculada de gravedad se expresa como:

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + g(q_d) \quad (40)$$

Obsérvese que a diferencia de la ley de control mostrada en la ecuación (39) (control PD con compensación de gravedad), en el control PD con compensación pre calculada de gravedad el vector de pares gravitacionales es evaluado en el vector de posiciones articulares deseadas q_d . El diagrama a bloques del control PD con compensación pre calculada de gravedad se muestra en la figura 28.

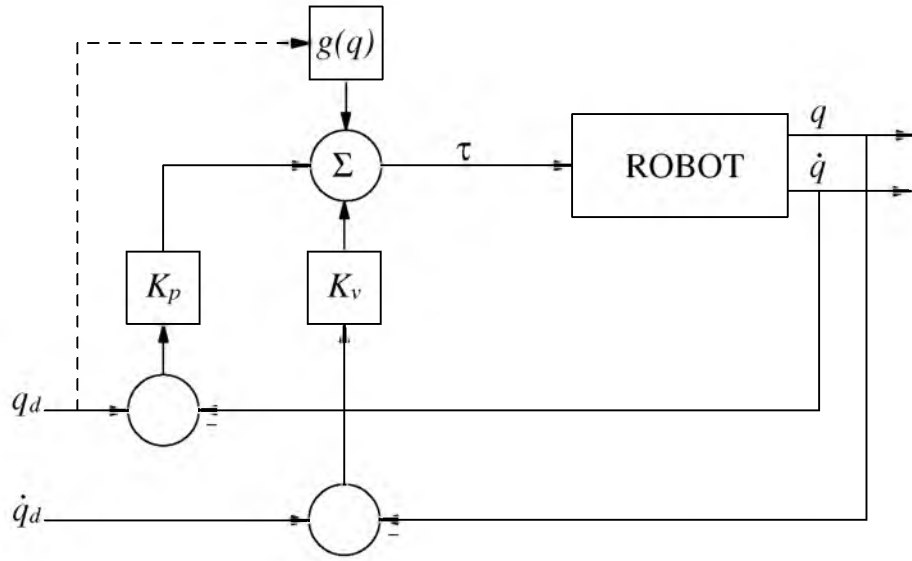


Figura 28: Diagrama de bloques del control PD con compensación precalculada de gravedad.

El vector τ en el modelo del control PD con compensación pre calculada de gravedad, contiene los pares requeridos de cada actuador del robot para llevar los eslabones a las posiciones articulares deseadas. Por tal motivo el modelo matemático para robots con n articulaciones rotaciones considerando el controlador (ecuación de lazo cerrado) se obtiene al sustituir el vector de pares τ en (11):

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = K_p\tilde{q} + K_v\dot{\tilde{q}} + \dot{g}(q_d) \quad (41)$$

De la misma forma que en el caso anterior se supondrá que los actuadores son fuentes ideales de torque, y además no se consideran las fuerzas de fricción en las articulaciones, lo cual es suficiente para realizar el análisis del comportamiento del sistema en lazo cerrado en su conjunto bajo la ley de control.

La ecuación (41) puede escribirse en términos del vector de estados $\begin{bmatrix} \tilde{q}^T & \dot{\tilde{q}}^T \end{bmatrix}^T$ como:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} \dot{\tilde{q}} \\ \ddot{q}_d - M(q)^{-1} \left[K_p\tilde{q} - K_v\dot{\tilde{q}} - C(q, \dot{q})q + \dot{g}(q_d) - g(q) \right] \end{bmatrix}$$

Cuando q_d es constante entonces el vector de estados es $\begin{bmatrix} \tilde{q}^T & \dot{\tilde{q}}^T \end{bmatrix}^T$ y la ecuación de lazo cerrado es:

$$\frac{d}{dt} \begin{bmatrix} \tilde{q} \\ \dot{\tilde{q}} \end{bmatrix} = \begin{bmatrix} -\dot{q} \\ M(q)^{-1} [K_p \tilde{q} - K_v \dot{q} - C(q, \dot{q}) \dot{q} + g(q_d) - g(q)] \end{bmatrix} \quad (42)$$

El origen $\begin{bmatrix} \tilde{q}^T & \dot{\tilde{q}}^T \end{bmatrix}^T = 0 \in \mathbb{R}^n$ es un equilibrio de la ecuación diferencial no lineal y autónoma (42), el cual no es único. Dichos equilibrios se encuentran en las raíces de la expresión:

$$K_p \tilde{q} = g(q_d - \tilde{q}) - g(q_d) \quad (43)$$

3.5.1. Equilibrio único

Como se menciona en la sección anterior el modelo dinámico del robot (41), cuenta con mas de un punto de equilibrio y ninguno de estos se encuentra en el origen, ecuación (43).

Para lograr obtener que la ecuación de lazo cerrado posea equilibrio único, basta con seleccionar una ganancia K_p tal que:

$$\lambda_{min} \{K_p\} > K_g$$

donde K_g se define como en (36).

Si se selecciona K_p de modo que $\lambda_{min} \{K_p\} > K_g$ se asegura que la ecuación (42) posea un único equilibrio $\begin{bmatrix} \tilde{q}^T & \dot{\tilde{q}}^T \end{bmatrix}^T = \begin{bmatrix} 0^T & 0^T \end{bmatrix}^T$, asintóticamente estable en forma global.

3.5.2. Diseño de control PD con compensación precalculada de gravedad para robot pendular

El control PD con compensación precalculada de gravedad, incluye explícitamente en su ley de control el vector de pares gravitacionales $g(q)$. Dicha ley de control se expresa como:

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + g(q_d)$$

Por otra parte, el modelo dinámico del robot pendular RP1 considera la dinámica del actuador y se escribe como:

$$\left[J_m + \frac{J_L}{r^2} \right] \ddot{q} + \left[f_m + \frac{K_a K_b}{R_a} \right] \dot{q} + K_L \text{sen}(q) = \frac{r K_a}{R_a} v$$

El modelo puede expresarse en forma compacta, en la forma de (24) donde:

$$M(q) = J_L \quad C(q, \dot{q}) = 0 \quad f(\dot{q}) = 0$$

$$R = \frac{1}{r^2} \quad J = J_m \quad g(q) = K_L \text{sen}(q)$$

$$B = f_m + \frac{K_a K_b}{R_a} K = \frac{K_a}{r R_a}$$

y:

$$J_L = m_c l_c^2$$

$$K_L = m_c g l_c$$

el modelo se reduce pasando R al lado derecho de la ecuación, por tanto:

$$\left[J_m + \frac{J_L}{r^2} \right] \ddot{q} + \left[f_m + \frac{K_a K_b}{R_a} \right] \dot{q} + K_L \text{sen}(q) = \frac{r K_a}{R_a} v$$

Si se considera al robot en una posición estática, es decir; donde las derivadas se son cero, el modelo dinámico del robot se expresa como:

$$K_L \text{sen}(q) = \frac{r K_a}{R_a} v$$

Tomando en cuenta el modelo dinámico para robots con n articulaciones rotacionales considerando a los actuadores como fuente ideal de torque se obtiene:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau$$

en comparación con 37, en la cual se aprecia que $\tau = \frac{r K_a}{R_a} v$, donde el torque esta

expresado en relación al voltaje de entrada del actuador, de manera que el par gravitacional τ_g a compensar esta dado por:

$$\tau_g = K_L \text{sen}(q)$$

Sin embargo en la ley de control PD con compensación pre calculada de gravedad el vector $g(q)$ es evaluado en la posición deseada q_d . Por tanto, el par gravitacional a compensar se expresa como:

$$\tau_g = m_c g l_c \text{sen}(q_d) \quad (44)$$

está definido en parámetros concretos de robot, donde m_c es la masa del eslabón, g es la aceleración de la gravedad, y l_c es la distancia al centro de masa del eslabón. A continuación se en listan los valores numéricos para dichos parámetros:

- $m_c = 0.301kg$
- $g = 9.81m/s^2$
- $l_c = 0.0516m$

Sustituyendo los valores para m_c , g y l_c , en la ecuación (44) se obtiene el par gravitacional para la ley de control:

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + 0.1528 \text{sen}(q_d)$$

Es importante observar que una vez calculado el par gravitacional a compensar, la selección de las ganancias es equivalente al caso del control PD; por lo que el procedimiento resulta ser el mismo. Por tanto, la selección de las ganancias proporcional y derivativas (K_p y K_v respectivamente) se reduce a las siguientes consideraciones:

- K_p y K_v sean matrices simétricas y definidas positivas.
- Seleccionar K_p tal que $\lambda_{min} \{K_p\} > K_g$.

La constante positiva K_g se calculada mediante la ecuación (36) de manera que la expresión para calcular la constante K_g , en el caso del robot pendular RP1 queda expresada como:

$$K_g = 1 \left[\max_{i,j,q} \left| \frac{\partial(m_c g l_c \text{sen}(q))}{\partial q} \right| \right]$$

realizando la derivada parcial se tiene que

$$\frac{\partial m_c g l_c \text{sen}(q)}{\partial q} = m_c g l_c \cos(q)$$

$$\max \left| \frac{\partial(m_c g l_c \text{sen}(q))}{\partial q} \right| = m_c g l_c$$

de modo que K_g se expresa como:

$$K_g = m_c g l_c.$$

La matriz de ganancia proporcional K_p debe cumplir con $\lambda_{\min} \{K_p\} > K_g$. Sin embargo, al tratarse solamente de un único eslabón rotacional, la matriz K_p es en realidad una constante escalar; siendo así que se debe seleccionar un valor tal que:

$$K_p > m_c g l_c,$$

donde m_c es la masa del eslabón, g es la aceleración de la gravedad, y l_c es la distancia al centro de masa del eslabón. A continuación se en listan los valores numéricos para dichos parámetros:

- $m_c = 0.301kg$
- $g = 9.81m/s^2$
- $l_c = 0.0516m$

Considerando los valores numéricos de los parámetros involucrados del robot, la constante $K_g = 0.1528$. Tomando en cuenta que K_p debe satisfacer $K_p > 0.1528$ se propone un valor para $K_p = 20$, y un valor para K_v del 20 % de K_p , por lo tanto $K_v = 4$. La ley de control PD con compensación pre calculada de gravedad para el robot pendular queda expresada finalmente como:

$$\tau = 20\tilde{q} + 4\dot{\tilde{q}} + 0.1528\text{sen}(q_d) \quad (45)$$

Debe hacerse notar que las ecuaciones 38 y 45 son iguales en cuanto a las ganancias K_p y K_v . La diferencia radica obviamente en el término de compensación de pares gravitacionales. Por tal motivo, las ganancias proporcional y derivativa se consideraron iguales para propósitos ilustrativos de reducción del error en las simulaciones numéricas.

4. Simulaciones

4.1. Introducción

La simulación numérica es en muchos casos efectuada como medio de verificación y validación de los diseños antes de la implementación. En ese sentido aquí se aborda la validación de los controladores diseñados con base a los modelos dinámicos aproximados involucrados del robot pendular RP1; los cuales se en listan a continuación:

- Modelo dinámico del actuador.
- Modelo dinámico del robot con actuador sin controlador.
- Control PD.
- Control PD con compensación pre calculada de gravedad.

La simulación numérica es efectuada en el software MATLAB, con la herramienta de simulación de sistemas dinámicos Simulink; el cuál se basa en un entorno gráfico de diagramas a bloques y además cuenta con la capacidad de auto generación de código para su implementación [9]. Sin embargo, el estudio de simulación en este caso solo se limita en la graficación de resultados y su análisis con el objetivo de validación para su posterior implementación.

4.2. Simulación del actuador

El modelo dinámico lineal del actuador se basa en parámetros específicos propios del motor empleado (actuador). Normalmente los motores comerciales cuentan con una hoja de datos del fabricante, donde se presentan la mayoría de los valores numéricos para cada parámetro importante. El diseño del control y el modelo dinámico mismo del robot utilizan dichos parámetros, por ello es conveniente hacer una validación del modelo del actuador, para evitar una discrepancia significativa entre el modelo y el robot real mismo.

- v voltaje de armadura [V].
- q posición angular del eje de carga [rad].

Sin embargo cabe mencionar que este modelo es idealizado por razones específicas descritas anteriormente (con el fin de evitar complejidad innecesaria en el modelo dinámico completo del robot), pero para propósitos de verificación de que los parámetros del motor son correctos, se utiliza el modelo del motor sin considerar los parámetros de reducción de engranajes y considerando esta vez la inductancia L_a cuyo valor es: $L_a = 65mH$. Además, para mayor simplicidad en la construcción del diagrama a bloques y claridad en la simulación, es necesario desacoplar las dinámicas eléctricas y mecánicas del motor de CD.

La contribución mecánica que se expresa a partir de la aplicación de las Leyes de Newton

$$J_m \ddot{q}_m = \tau_m - f_m \dot{q}_m$$

La contribución eléctrica en el modelo dinámico del motor viene dada por:

$$v = R_a i_a + L_a \frac{di_a}{dt} + e_b$$

Para la construcción del diagrama a bloques en Simulink se despeja la derivada de mayor grado de ambas ecuaciones, de modo que pueden reescribirse como:

$$\frac{d^2 q}{dt^2} = \frac{1}{J_m} (\tau_m - f_m \frac{dq_m}{dt}) \quad (46)$$

$$\frac{di_a}{dt} = \frac{1}{L_a} (-R_a i_a + v - e_b) \quad (47)$$

tomando en cuenta que:

$$\tau_m = K_a i_a e_b = K_b \dot{q}_m$$

las ecuaciones 46 y 47 se expresan finalmente como:

$$\frac{d^2 q}{dt^2} = \frac{1}{J_m} (K_a i_a - f_m \frac{dq_m}{dt}) \quad (48)$$

$$\frac{di_a}{dt} = \frac{1}{L_a}(-R_a i_a + v - K_b \frac{dq_m}{dt}) \quad (49)$$

Obsérvese que las ecuaciones 48 y 49 están acopladas, en el sentido de que ambas dependen de la corriente de armadura i_a . Esta relación se muestra en el diagrama a bloques mostrado en la figura 30.

La simulación numérica es resuelta mediante el método numérico Runge-Kutta de cuarto orden (integrado en Simulink), con un paso de integración de $1e-6$ s y un tiempo de simulación de $50e-3$ s [9]. Todas las simulaciones son ejecutadas considerando el voltaje nominal del motor de CD. $v = 12V$ aplicado en sus terminales, debido a que los datos del fabricante son medidos y proporcionados sobre este criterio.

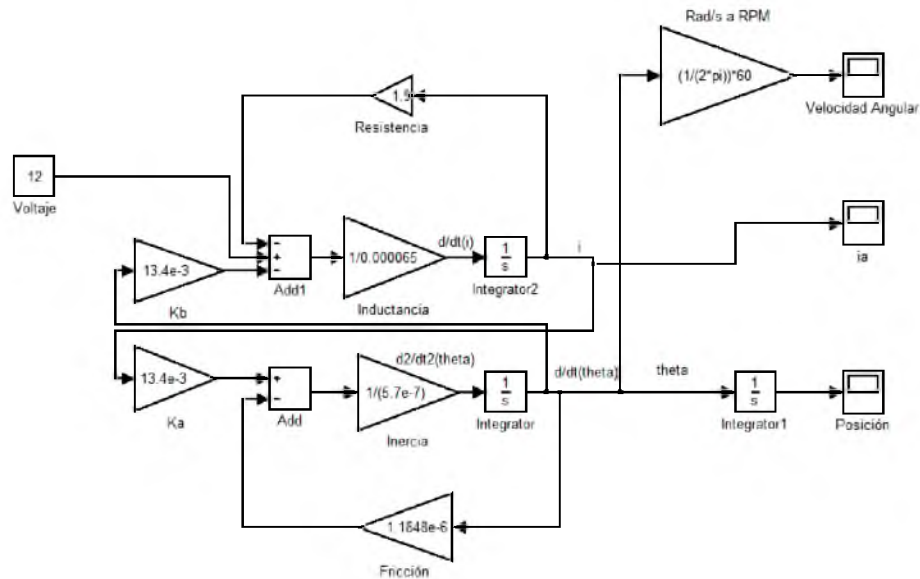


Figura 30: Diagrama a bloques en Simulink, de motor de CD.

Los parámetros de validación para el modelo del motor de CD se en listan a continuación junto con sus unidades y magnitudes proporcionadas por el fabricante:

- Corriente sin carga en el eje del motor y sin juego de engranes acoplado $|A| = 75$ mA.
- Corriente de arranque sin juego de engranes acoplado $|A| = 6$ A.

- Velocidad angular sin carga sin juego de engranes acoplado [RPM] = 8500 RPM.
- Tiempo mecánico de establecimiento sin juego de engranes acoplado [s] = 6 ms.
- Velocidad angular después con juego de engranes acoplado [RPM] = 130 RPM.

Los parámetros de validación del modelo son: la velocidad angular \dot{q} y la posición angular q del eje del motor, medidos antes y después del juego de engranes, además de la corriente de armadura i_a .

La gráfica de velocidad angular \dot{q} respecto del tiempo, (sin considerar el juego de engranes acoplado al eje del motor) se presenta en la figura 31, donde se observa la curva en el tiempo de la velocidad angular medida en RPM.

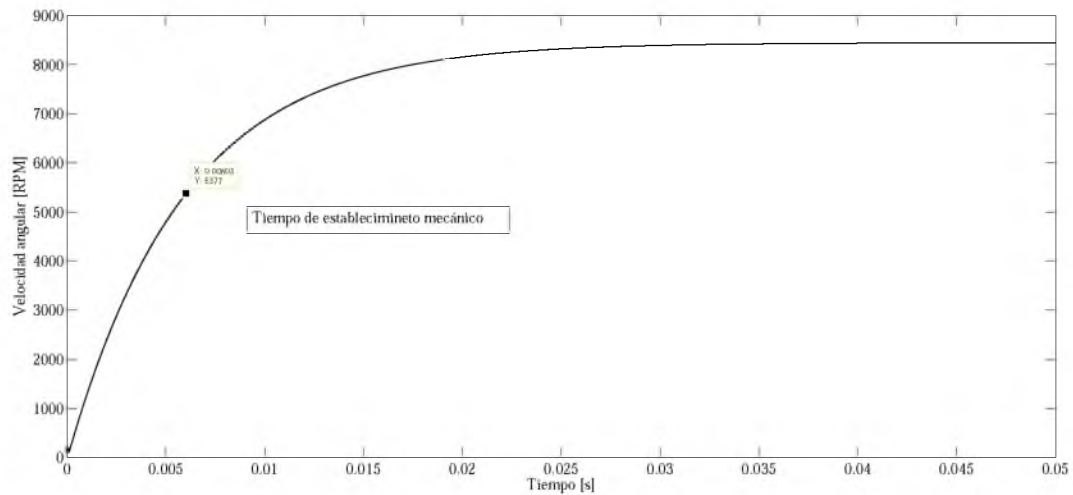


Figura 31: Gráfica de velocidad en el tiempo del motor de CD.

En la figura 31 se observa que el máximo valor de la velocidad angular es menor a 8450 RPM, lo que es suficientemente aproximado al valor proporcionado por el fabricante de 8500 RPM.

La gráfica de corriente i_a respecto del tiempo se muestra en la figura 32 donde se observa la curva que describe la corriente en el tiempo cuando se aplican 12V en las terminales del motor y el eje no tiene carga.

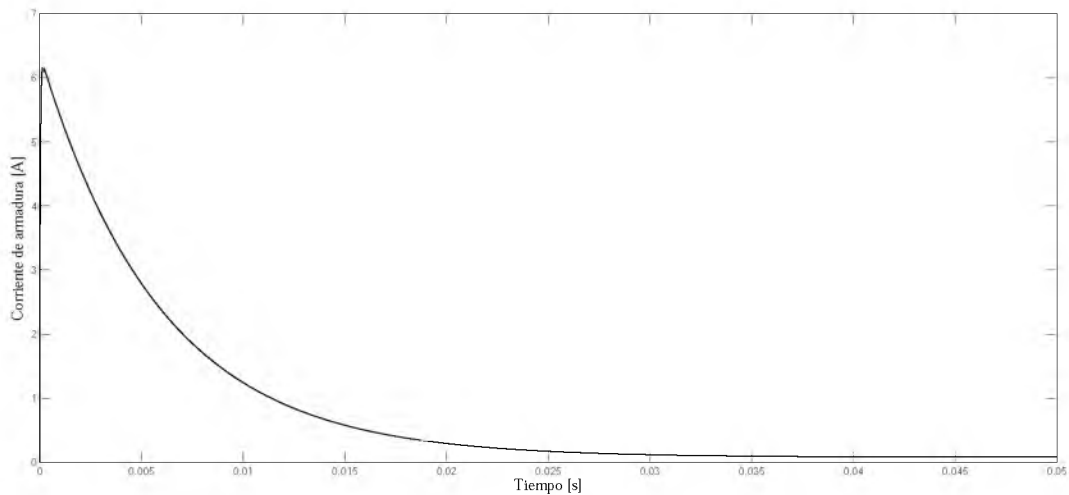


Figura 32: Gráfica de la corriente en el tiempo del motor de CD.

En la figura 32 se muestra que la curva de corriente presenta un pico al inicio de la simulación, esto se debe a que se debe vencer la inercia del rotor del motor para poder empezar a girar; el pico de corriente alcanza los 6 A, que es justamente lo establecido por el fabricante.

La velocidad máxima alcanzada por el motor, de la figura 31 es de 8440 RPM, haciendo uso de $t_s = 63.3\%$ del valor máximo alcanzado, para el cálculo del tiempo de establecimiento t_s , se observa que $t_s = 6ms$, lo cual concuerda con el parámetro establecido por el fabricante.

Por otra parte haciendo uso de la relación de transmisión de engranajes $1 : r$ donde $r = 64$ se indica que la velocidad angular \dot{q} es reducida 64 veces. Para propósitos de simulación es equivalente a añadir un bloques cuyas entrada sea \dot{q} , y multiplicarlo con una ganancia de $1/64$ para efectuar dicha reducción. En la figura 33 se observa el diagrama de Simulink utilizado.

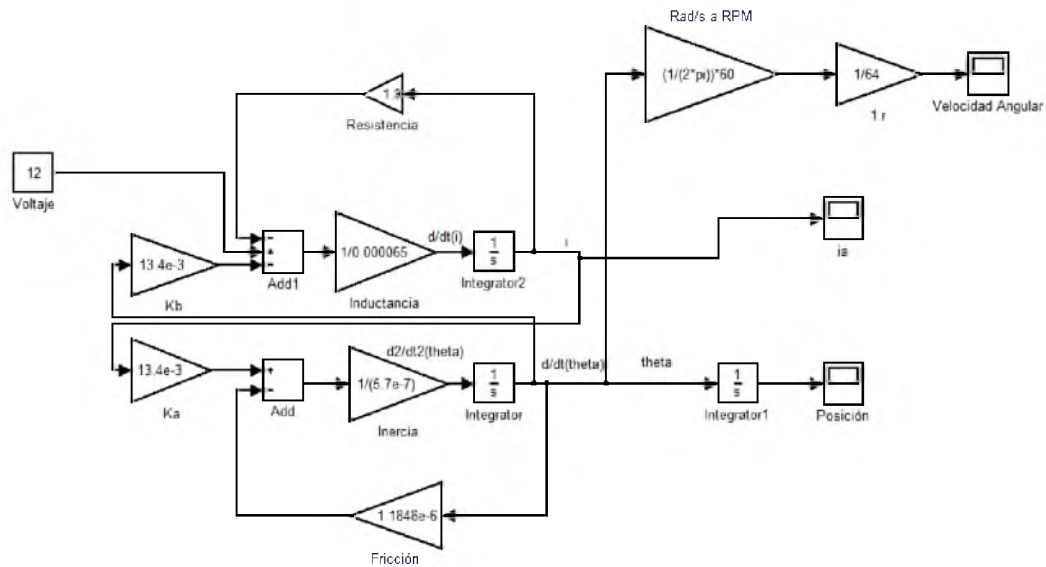


Figura 33: Mecanismo de reducción de engranes, con relación de trasmisión de 1:64, implementado mediante el bloque de ganancia 1/64.

La gráfica de velocidad angular \dot{q} después del juego de engranes se observa en la figura 34, donde se puede apreciar además, que el valor máximo alcanzado es cercano a las 130 RPM, con lo cual queda validado el parámetro establecido por el fabricante.

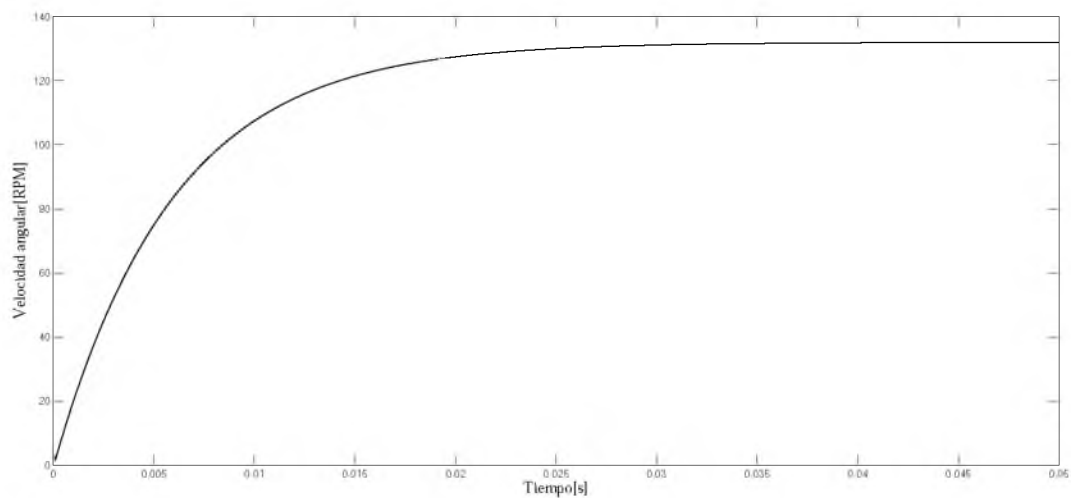


Figura 34: Gráfica de la velocidad angular \dot{q} , del motor después del juego de engranes.

Finalmente el modelo del motor de CD, es validado en los parámetros reque-

ridos, con lo cual dichos valores son confiables para su uso en el diseño de control y en modelo dinámico del robot pendular RP1.

4.3. Simulación de modelo dinámico robot con actuador sin controlador

El propósito de esta sección es el estudio del comportamiento del modelo del robot considerando el actuador, en lazo abierto, sin la presencia del controlador. Para ello se hace uso del modelo dinámico completo del robot pendular:

$$\left[J_m + \frac{J_L}{r^2} \right] \ddot{q} + \left[f_m + \frac{K_a K_b}{R_a} \right] \dot{q} + \frac{K_L}{r^2} \text{sen}(q) = \frac{K_a}{r R_a} v$$

donde

$$J_L = m_c l_c^2$$

$$K_L = m_c g l_c$$

Todos los parámetros tanto eléctricos como mecánicos propios del sistema son conocidos, de modo que para mayor simplicidad puede reescribirse como:

$$A\ddot{q} + B\dot{q} + C\text{sen}(q) = Dv \quad (50)$$

donde

$$A = \left[J_m + \frac{J_L}{r^2} \right], B = \left[f_m + \frac{K_a K_b}{R_a} \right]$$

$$C = \frac{K_L}{r^2}, D = \frac{K_a}{r R_a}$$

despejando la derivada de mayor grado de (50) se tiene:

$$\ddot{q} = \frac{1}{A} (Dv - B\dot{q} - C\text{sen}(q)) \quad (51)$$

A continuación se en listan los parámetros del motor de CD. y del eslabón, así

como sus unidades y valores numéricos:

- J_m inercia del motor [kgm^2] = $5.7\text{e-}7$.
- f_m coeficiente de fricción del rotor [Nm] = $1.1848\text{e-}6$.
- K_a constante motor-par [Nm/A] = $13.4\text{e-}3$.
- K_b constante de fuerza contraelectromotriz [Vs/rad] = $13.4\text{e-}3$.
- R_a resistencia de armadura [Ω] = 1.9.
- τ par aplicado después del juego de engranes [Nm].
- r relación de reducción de engranes [1:r] = 64.
- v voltaje de armadura [V].
- q posición angular del eje de carga [rad].
- m_c masa del eslabón [Kg] = 0.301.
- l_c distancia al centro de masa del eslabón [m] = $5.16\text{ e-}2$.
- g aceleración de la gravedad [m/s^2] = 9.81.

El diagrama a bloques mostrado en la figura 35 se basa en la ecuación (51). El parámetro de interés del modelo es la posición articular q , la cual es medida a través del bloque de Simulink denominado Scope, como se observa en la figura 35.

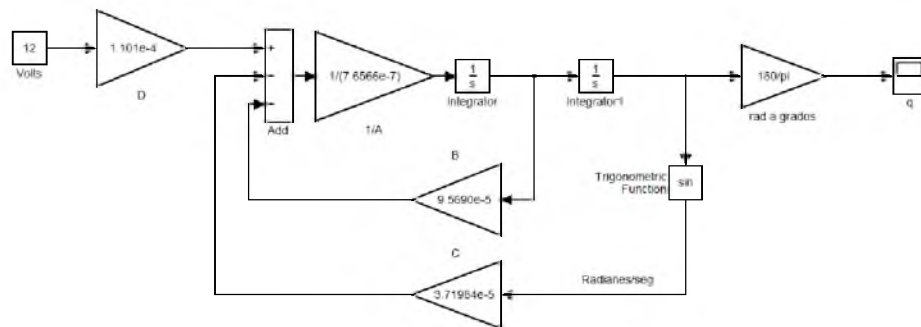


Figura 35: Diagrama a bloques del modelo dinámico, considerando el actuador en lazo abierto y sin controlador.

La simulación numérica es resuelta mediante el método numérico Runge-Kutta de cuarto orden (integrado en Simulink), con un paso de integración de $1e-6s$ y un tiempo de simulación de $6s$. La simulación es ejecutada considerando el voltaje nominal del motor de CD. $v = 12V$ aplicado en sus terminales.

Se utiliza el método numérico Runge-Kutta de cuarto orden (integrado en Simulink), para resolver la simulación, con un paso de integración de $1e-6s$. El tiempo de simulación es de $6s$ y se ejecuta considerando el voltaje nominal del motor de CD. $v = 12V$ aplicado en sus terminales.

En la figura 36 se observa la gráfica de la posición articular q del robot, cuando se aplican $12V$ en las terminales del actuador.

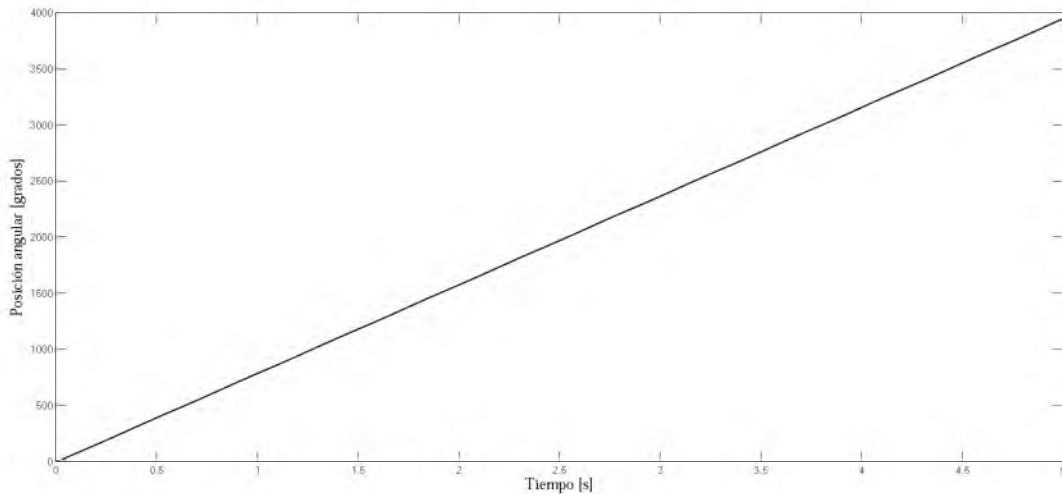


Figura 36: Gráfica de la posición articular q del robot pendular RP1.

La tendencia de la curva descrita en el tiempo por la posición articular q , mostrada en la figura 36 indica que el eslabón permanece girando indefinidamente hasta que se deje de aplicar los $12V$ a las terminales del actuador; siendo practica mente imposible llevar al eslabón a alguna posición deseada q_d ; por lo que es obviamente necesario el uso del controlador para tal propósito.

4.4. Simulación de Control PD

La adhesión del controlador PD al lazo cerrado del sistema, crea condiciones suficientes para lograr que se cumpla que $q = q_d$ cuando $t \rightarrow \infty$, al menos para el caso de robots que no incluyen el vector de pares gravitacional $g(q)$ en sus modelos dinámicos. Sin embargo, el robot pendular RP1 incluye en su modelo dinámico el

vector de pares gravitacionales $g(q)$, por lo que es de esperarse que para ciertas posiciones articulares, no se cumpla que $q = q_d$. A pesar de ello en general, es suficiente para reducir el error a niveles aceptables cercanos a cero.

Para la simulación del control PD diseñado, se hace uso de la ley de control 38 expresada como:

$$\tau = 20\tilde{q} + 4\dot{\tilde{q}}$$

donde \tilde{q} es el error de posición y $\dot{\tilde{q}}$ es el error en velocidad; pero tomando en consideración que el vector de posiciones deseadas $q_d = 0$, entonces la ley de control se reescribe como:

$$\tau = 20\tilde{q} - 4\dot{q}$$

de modo que las ganancias proporcional y de velocidad son: $K_p = 20, K_v = 4$.

Por otra parte del modelo dinámico del robot pendular considerando el actuador se tiene:

$$\left[J_m + \frac{J_L}{r^2} \right] \ddot{q} + \left[f_m + \frac{K_a K_b}{R_a} \right] \dot{q} + \frac{K_L}{r^2} \text{sen}(q) = \frac{K_a}{r R_a} v$$

donde

$$J_L = m_c l_c^2$$

$$K_L = m_c g l_c$$

Cada uno de los parámetros tanto eléctricos como mecánicos del sistema son conocidos. Nuevamente reescribiremos el modelo para mayor simplicidad en la transformación del modelo a diagrama de bloques, puede compactarse reescribiendo la ecuación como:

$$A\ddot{q} + B\dot{q} + C\text{sen}(q) = Dv$$

donde

$$A = \left[J_m + \frac{J_L}{r^2} \right], B = \left[f_m + \frac{K_a K_b}{R_a} \right]$$

$$C = \frac{K_L}{r^2}, D = \frac{K_a}{r R_a}$$

despejando la derivada de mayor grado de $A\ddot{q} + B\dot{q} + C\text{sen}(q) = Dv$ se tiene:

$$\ddot{q} = \frac{1}{A}(Dv - B\dot{q} - C\text{sen}(q))$$

A continuación se en listan los parámetros del motor de CD. y del eslabón, así como sus unidades y valores numéricos:

- J_m inercia del motor [kgm^2] = 5.7e-7.
- f_m coeficiente de fricción del rotor [Nm] = 1.1848e-6.
- K_a constante motor-par [Nm/A] = 13.4e-3.
- K_b constante de fuerza contraelectromotriz [Vs/rad] = 13.4e-3.
- R_a resistencia de armadura [Ω] = 1.9.
- τ par aplicado después del juego de engranes [Nm].
- r relación de reducción de engranes [1:r] = 64.
- v voltaje de armadura [V].
- q posición angular del eje de carga [rad].
- m_c masa del eslabón [Kg] = 0.301.
- l_c distancia al centro de masa del eslabón [m] = 5.16 e-2.
- g aceleración de la gravedad [m/s^2] = 9.81.

El modelo dinámico del robot pendular utilizado en la simulación se expresa en la ecuación (51). El diagrama a bloques construido en Simulink de este sistema, es mostrado en la figura 37. Aquí se aprecia la disposición del controlador, para

que se cumpla que $\tau = 20\ddot{q} - 4\dot{q}$. Este torque producido por el controlador pasa directamente al modelo del robot, proporcionándole así movimiento, hasta que se cumpla que $q_d = q$. El modelo del robot es representado por el subsistema ROBOT PENDULAR, en el diagrama en Simulink.

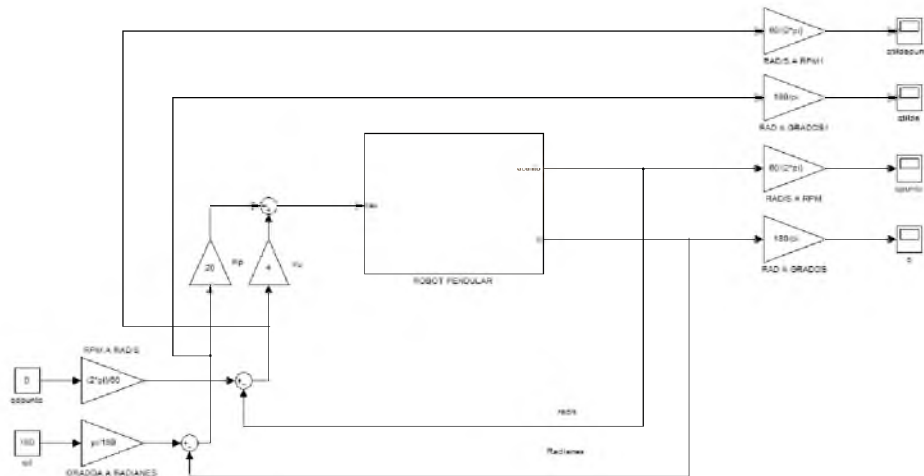


Figura 37: Diagrama en Simulink de implementación de control PD.

Con el objetivo de verificación de compensación del error del controlador, en la simulación se toman en cuenta los siguientes condiciones:

- $q_d = 180$
- $q(0) = 0$
- $\dot{q}(0) = 0/s$
- $\dot{q}_d = 0/s$

Los parámetros necesarios para verificación del controlador son la posición q y velocidad \dot{q} articulares, así como los errores de posición \tilde{q} y velocidad $\dot{\tilde{q}}$, dichos parámetros son mostrados mediante los bloques de Simulink denominados Scope. Por otra parte la simulación numérica es resuelta mediante el método numérico Runge-Kutta de cuarto orden (integrado en Simulink), con un paso de integración de $1e-6s$ y un tiempo de simulación de $2.5s$. A continuación se presentan los resultados de simulación.

La gráfica de la posición articular q , se muestra en la figura 38, en donde se aprecia que la curva respecto del tiempo parte de la condición inicial $q(0) = 0$,

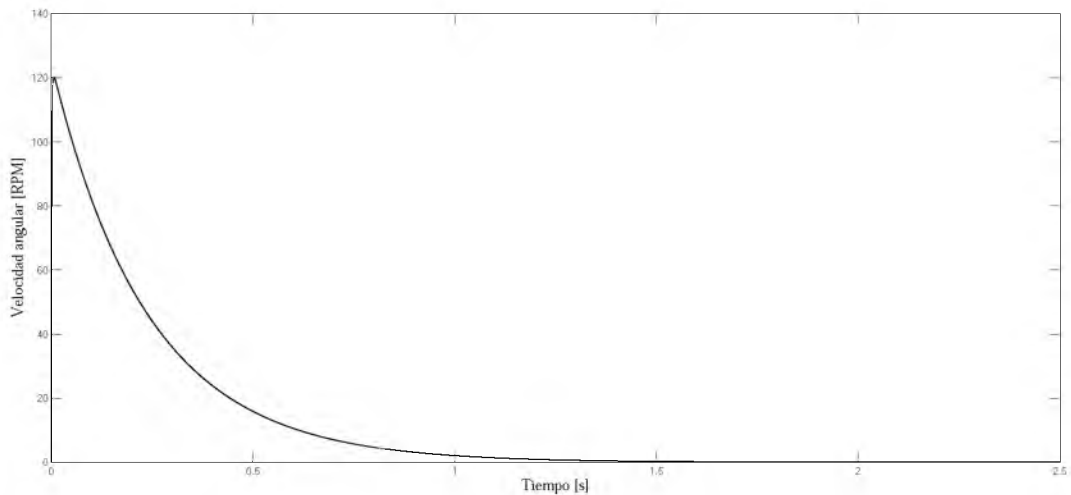


Figura 39: Gráfica de la velocidad articular \dot{q} , bajo la ley de control PD.

hasta la posición deseada $q_d = 180$ en aproximadamente. Cabe mencionar que la velocidad con la que llega a ese punto, puede ser lenta o rápida, debido a que la velocidad deseada es simplemente $\dot{q}_d = 0/s$, lo que no condiciona al controlador en ese aspecto.

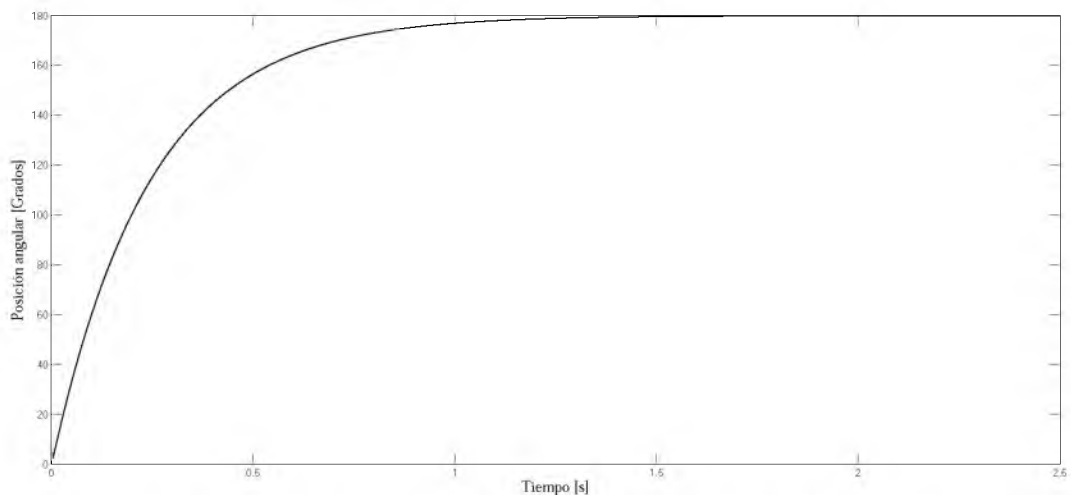


Figura 38: Gráfica de la posición articular q , bajo la ley de control PD.

En el caso de la velocidad articular \dot{q} , figura 39, se observa que parte de cero, generando un pico en la curva, para después volver a cero cuando se ha alcanzado que $q = q_d$.

El parámetro de verificación mas importante en este caso, es el error en posición \tilde{q} . Su representación se observa en la figura 40. Aquí se observa como el error es

llevado a un valor cercano a cero.

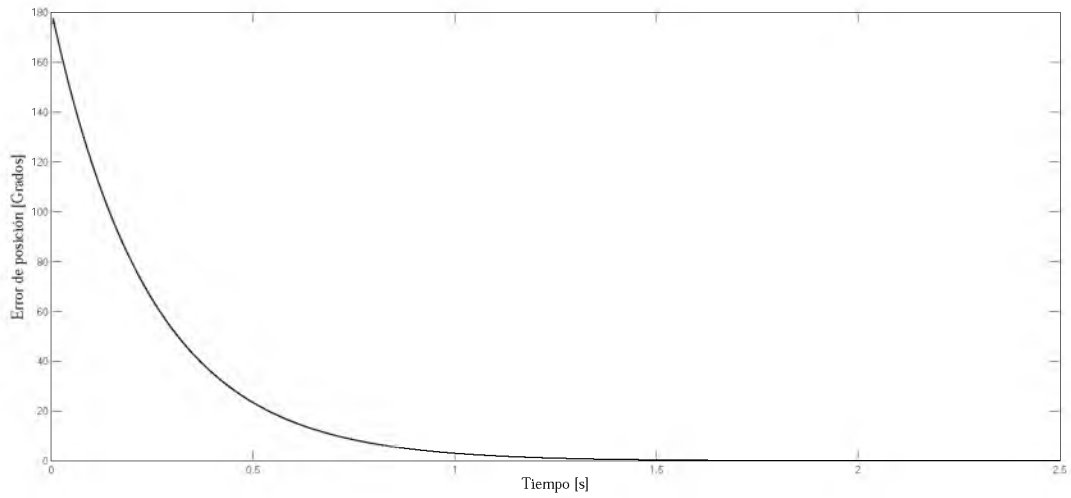


Figura 40: Gráfica del error de posición \tilde{q} , respecto del tiempo, bajo la ley de control PD.

El error de velocidad $\dot{\tilde{q}}$ se observa en la figura 41. En este gráfico se observa el efecto de freno mecánico producido por la compensación de velocidad, produciendo así que el error tienda a cero.

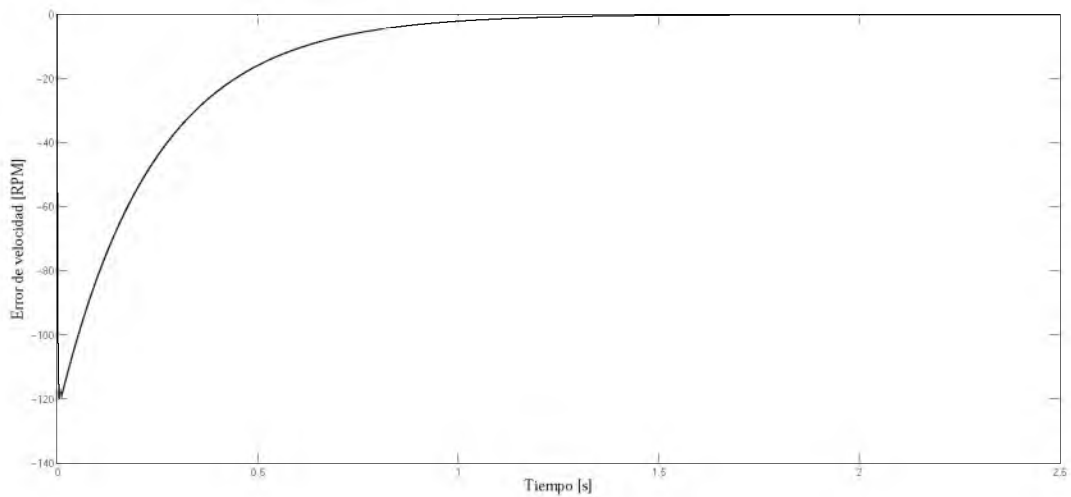


Figura 41: Gráfica del error de velocidad $\dot{\tilde{q}}$, respecto del tiempo, bajo la ley de control PD.

En general la aplicación de la ley de control PD resulta suficiente para llevar el error de posición a valores muy cercanos a cero, en el caso del robot pendular. Sin embargo cuando $q_d = 90$ (cuando el eslabón se encuentra completamente

horizontal, presentándose el máximo par generado); el control PD no es suficiente para corregir el error de posición a valores cercanos a cero [2]. En la figura 42 se presenta el resultado de una simulación para $q_d = 90$ y $q(0) = 0$.

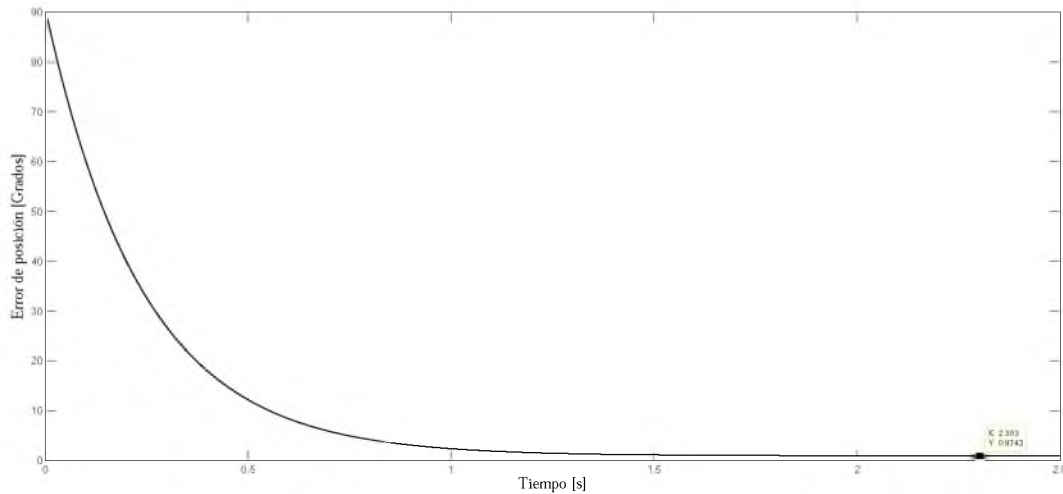


Figura 42: Gráfica del error de posición q cuando $q_d = 90$, bajo la ley de control PD.

4.5. Simulación de Control PD con compensación precalculada de gravedad

Cuando el vector de pares gravitacionales $g(q)$ está presente en el modelo dinámico del robot, existen pares mecánicos que, cuando el robot se encuentra en equilibrio estático en alguna posición deseada q_d , generan fuerzas que deben ser compensadas más allá de las ganancias proporcional K_p y derivativa K_v . Este es el caso del robot pendular, y el efecto de estas fuerzas se aprecia con claridad en la figura 42, donde se puede observar que el controlador PD no es suficiente para llevar el error de posición \tilde{q} a cero.

El control PD con compensación precalculada de gravedad, es suficiente para compensar los pares mecánicos producidos por el eslabón, en cualquier posición deseada q_d , para el robot pendular.

La simulación del control PD con compensación pre calculada de gravedad, hace uso de la ley de control diseñada expresada en la ecuación (45) como:

$$\tau = 20\ddot{\tilde{q}} + 4\dot{\tilde{q}} + 0.1528\text{sen}(q_d)$$

donde

$$K_p = 20K_v = 0.4$$

El modelo dinámico del robot considerando el actuador:

$$\left[J_m + \frac{J_L}{r^2} \right] \ddot{q} + \left[f_m + \frac{K_a K_b}{R_a} \right] \dot{q} + \frac{K_L}{r^2} \text{sen}(q) = \frac{K_a}{r R_a} v$$

donde

$$J_L = m_c l_c^2$$

$$K_L = m_c g l_c$$

Cada uno de los parámetros tanto eléctricos como mecánicos del sistema son conocidos. Para mayor simplicidad en la transformación del modelo a diagrama de bloques, la ecuación es compactada como:

$$A\ddot{q} + B\dot{q} + C\text{sen}(q) = Dv$$

donde

$$A = \left[J_m + \frac{J_L}{r^2} \right], B = \left[f_m + \frac{K_a K_b}{R_a} \right]$$

$$C = \frac{K_L}{r^2}, D = \frac{K_a}{r R_a}$$

despejando la derivada de mayor grado de $A\ddot{q} + B\dot{q} + C\text{sen}(q) = Dv$ se tiene:

$$\ddot{q} = \frac{1}{A}(Dv - B\dot{q} - C\text{sen}(q))$$

A continuación se en listan los parámetros del motor de CD. y del eslabón, así como sus unidades y valores numéricos:

- J_m inercia del motor $[\text{kgm}^2] = 5.7\text{e-}7$.

- f_m coeficiente de fricción del rotor [Nm] = 1.1848e-6.
- K_a constante motor-par [Nm/A] = 13.4e-3.
- K_b constante de fuerza contraelectromotriz [Vs/rad] = 13.4e-3.
- R_a resistencia de armadura [Ω] = 1.9.
- τ par aplicado después del juego de engranes [Nm].
- r relación de reducción de engranes [1:r] = 64.
- v voltaje de armadura [V].
- q posición angular del eje de carga [rad].
- m_c masa del eslabón [Kg] = 0.301.
- l_c distancia al centro de masa del eslabón [m] = 5.16 e-2.
- g aceleración de la gravedad [m/s²] = 9.81.

El modelo dinámico del robot pendular (51), se muestra en la figura 43. El diagrama a bloques del modelo dinámico del robot es condensado en el subsistema denominado ROBOT PENDULAR. La ley de control está construida de modo que se cumpla con $\tau = 20\ddot{q} + 4\dot{q} + 0.1528\text{sen}(q_d)$.

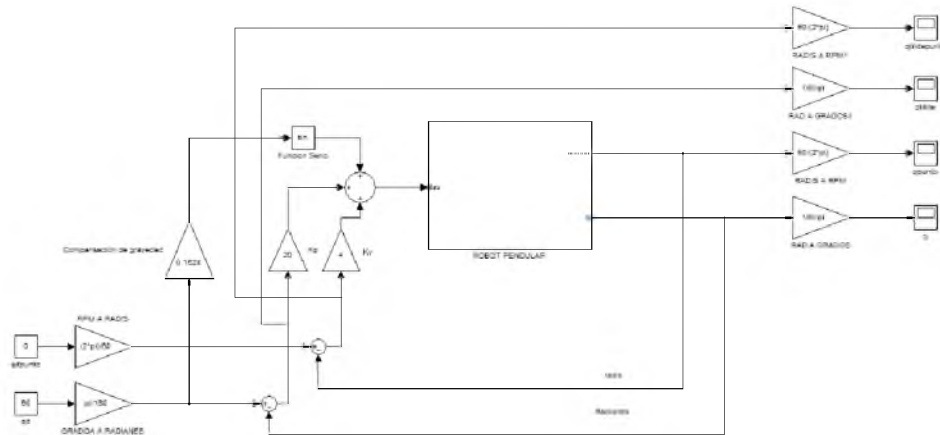


Figura 43: Diagrama a bloques en Simulink del control PD con compensación precalculada de gravedad.

La simulación es resuelta mediante el método numérico de Runge-Kutta de cuarto orden, con un paso de integración de $1e-6s$, y un tiempo de simulación de $3s$. Las condiciones iniciales, posición y velocidad deseadas se muestran a continuación:

- $q_d = 90$
- $\dot{q}_d = 0/s$
- $q(0) = 0$
- $\dot{q}(0) = 0/s$

Se hace uso de $q_d = 90$ debido a que en esta posición articular se presenta el máximo par mecánico generado por el eslabón. De este modo se comprueba la compensación en gravedad realizada por el controlador.

Los parámetros necesarios para verificación del controlador son la posición q y velocidad \dot{q} , así como los errores de posición \tilde{q} y de velocidad $\dot{\tilde{q}}$.

En la figura 44 se observa la gráfica de la posición articular q respecto del tiempo, la cual parte de la condición inicial $q(0) = 0$, hasta alcanzar el valor de q_d en aproximadamente $2s$.

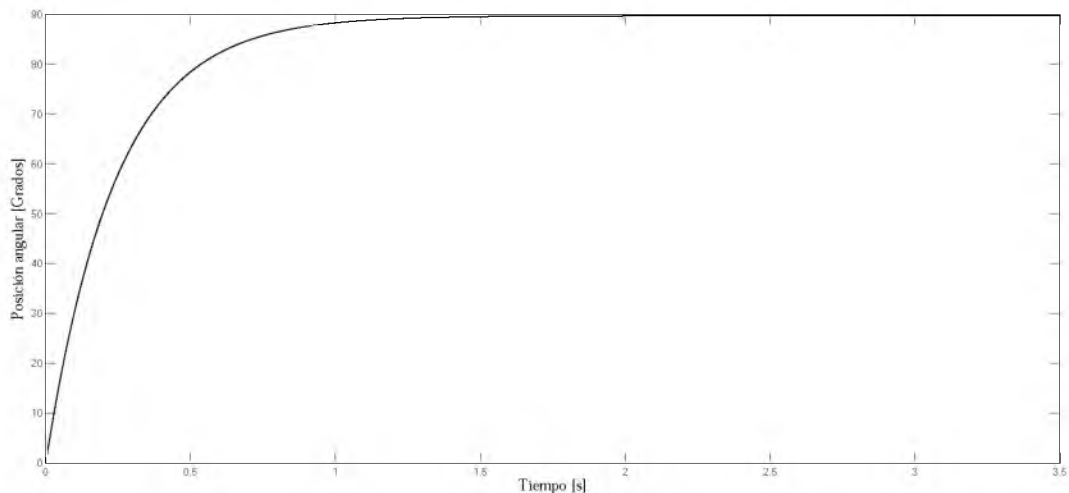


Figura 44: Gráfica de la posición articular q , del robot pendular, bajo la ley de control PD con compensación precalculada de gravedad.

De la figura 44 se observa que sin importar que la posición deseada $q_d = 90$, el control lo compensa de tal manera que $q_d = q$.

La gráfica de la velocidad articular \dot{q} respecto del tiempo, se muestra en la figura 45. La velocidad angular parte de la condición inicial $\dot{q}(0) = 0$, levanta un pico a modo de freno mecánico, para posteriormente volver a cero, cuando $q_d = q$.

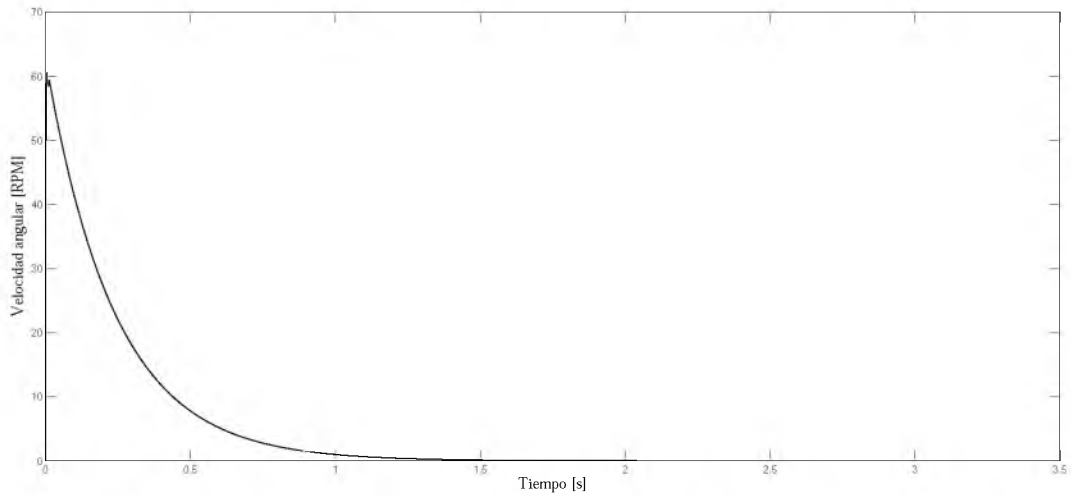


Figura 45: Gráfica de la velocidad articular \dot{q} , del robot pendular, bajo la ley de control PD con compensación precalculada de gravedad.

A continuación se presentan gráficas del comportamiento del error en posición \tilde{q} y velocidad $\dot{\tilde{q}}$. En la figura 46, se observa que a pesar de que la posición deseada q_d genera el máximo torque mecánico por el eslabón, el control es capaz de llevar el error a cero, a diferencia evidente del control PD sin compensación de gravedad.

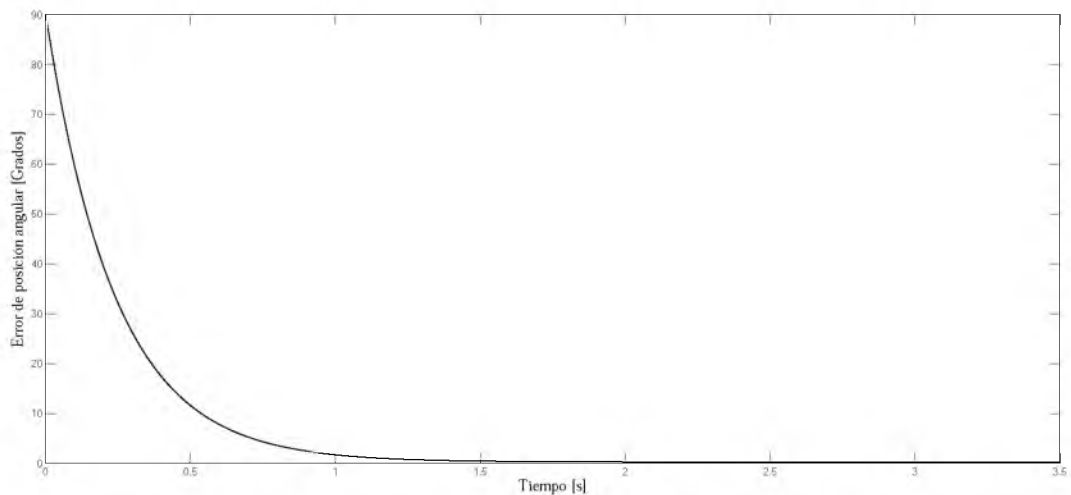


Figura 46: Gráfica del error de posición \tilde{q} , del robot pendular, bajo la ley de control PD con compensación precalculada de gravedad.

La gráfica del error de velocidad \dot{q} se muestra en la figura 47. La curva describe el efecto de freno mecánico producido por la compensación de velocidad produciendo así que el error tienda a cero.

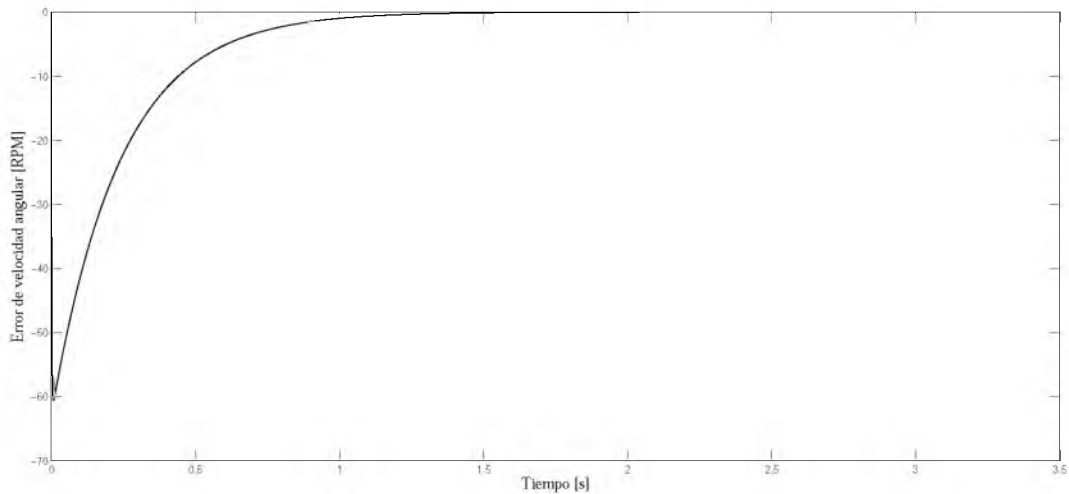


Figura 47: Gráfica del error de velocidad \dot{q} , del robot pendular, bajo la ley de control PD con compensación precalculada de gravedad.

Con base en los resultados de simulación, es evidente que el control PD con compensación precalculada de gravedad es suficiente para reducir el error de posición a valores cercanos a cero e incluso a cero; a pesar de que el modelo dinámico del robot cuenta con el vector de pares gravitacionales $g(q)$.

5. Implementación

5.1. Sistema mecánico

5.1.1. Introducción

El robot pendular diseñado para este trabajo es un prototipo funcional, a modo de plataforma experimental. El diseño conceptual se muestra en la figura 48. Su arquitectura abierta, le da la capacidad de ser reprogramable para la aplicación de diversos algoritmos de control ya sea de posición pura o de velocidad.

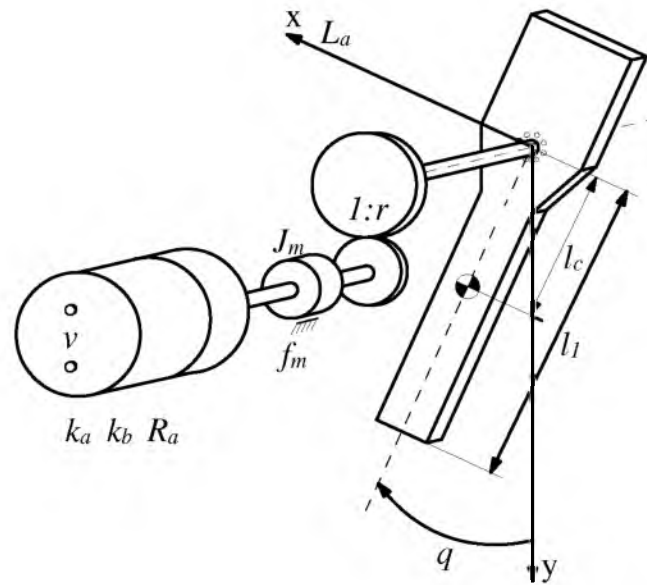


Figura 48: Diagrama esquemático de robot pendular construido.

Debido a que el robot pendular es un prototipo enfocado en la aplicación de diversos algoritmos de control, el diseño y análisis mecánico de la estructura no es pertinente. Por tanto, el estudio mecánico se redujo únicamente al diseño del eslabón rotacional en función de disminuir el momento de inercia a fin de reducir el torque requerido, para lograr el movimiento del eslabón sobre su eje.

Para el diseño del eslabón, fue construido un modelo de sólidos completo del robot en el software SolidWorks con el propósito de generar los planos que sirven para la construcción del mismo.

El material seleccionado para el eslabón rotacional es aluminio, por ser un material liviano y relativamente más fácil de maquinarse en comparación con otros materiales más densos como el acero. Sin embargo, la estructura de soporte del eslabón y el motor de CD, está constituida por un perfil tubular rectangular (PTR) de acero.

5.1.2. Diseño de eslabón rotacional para robot pendular RP1

El principal parámetro en el diseño del eslabón rotacional es la distancia del centro de masa al eje de rotación del mismo. De este parámetro depende la magnitud del momento de inercia generado en el eje de rotación y del torque máximo cuando el eslabón rotacional se encuentra a 90° , figura 49.

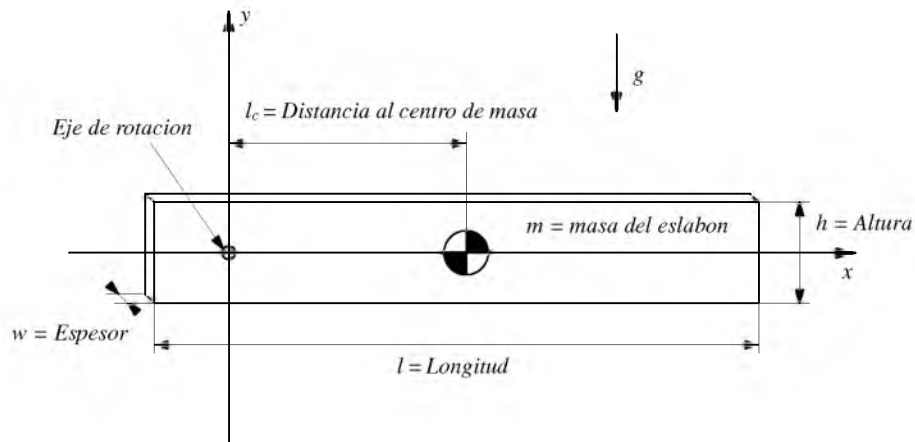


Figura 49: Parámetros críticos en el diseño del eslabón rotacional.

La selección del perfil para el eslabón rotacional está condicionado por las siguientes restricciones de diseño, las cuales tienen base en el material utilizado para su construcción; y en las capacidades mecánicas del actuador.

- Longitud máxima [cm]: 35.
- Anchura [cm]: 7.
- Espesor máximo [cm]: 1.
- Máximo torque generado, medido desde el eje de giro al centro de masa [Nm]: < 1.7 .
- Distancia al centro de masa al eje de giro [cm]: ≤ 6 .

Con base en las restricciones, el eslabón puede tener un perfil rectangular mostrado en la figura 49, donde se indica además, la ubicación del centro de masa del perfil. El eje de rotación no puede ser colocado en el mismo lugar donde se encuentra el centro de masa puesto que, la dinámica del robot sería equivalente al caso de un motor de CD. con una masa acoplada a su eje, lo cual no es el propósito de este trabajo.

Sin embargo es evidente que, mientras más cercano se encuentre el eje de rotación al centro de masa, el momento de inercia y por tanto el máximo torque son menores.

El inconveniente de la ubicación del eje de rotación radica en que el eslabón es completamente simétrico sobre su centro de masa, es decir; existe la misma cantidad de masa a cada lado del centro de masa. Por tanto, resulta más conveniente considerar un perfil de eslabón no simétrico como el que se muestra en la figura 50.

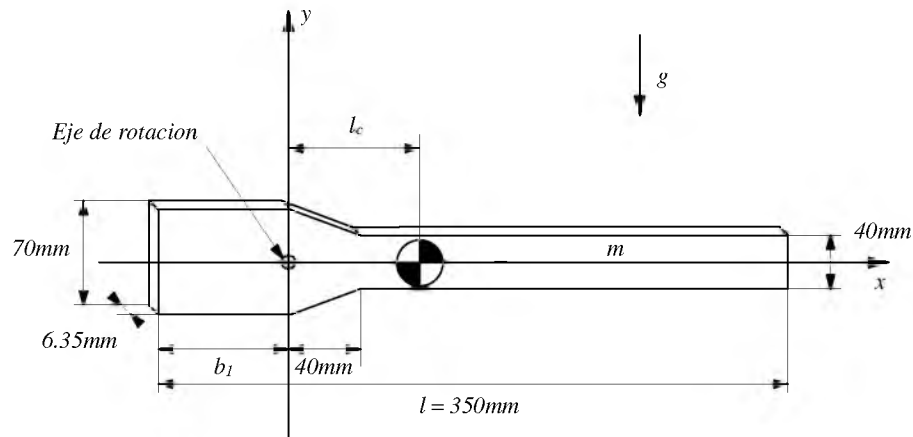


Figura 50: Perfil para eslabón rotacional no simétrico, considerando las restricciones de diseño.

Por otra parte, el perfil mostrado en la figura 50 en comparación con el anterior (ver figura 49), tiene menor masa además de que no es simétrico alrededor del centro de masa; de modo que éste se ubica cercano al eje de giro del eslabón reduciendo el torque máximo generado y el momento de inercia medido desde el eje de giro.

Para el estudio de la ubicación del centro de masa del perfil de eslabón, se divide el volumen en entidades más simples. Se establece un sistema de coordenadas en una posición apropiada, en este caso; el origen de dicho sistema está ubicado en el eje de rotación del eslabón. Además, se aprecia que el perfil es simétrico sobre el eje x , por tanto; el cálculo de las coordenadas al centro de masa C_x y C_y , se reduce al estudio de la mitad del eslabón como se observa en la figura 51.

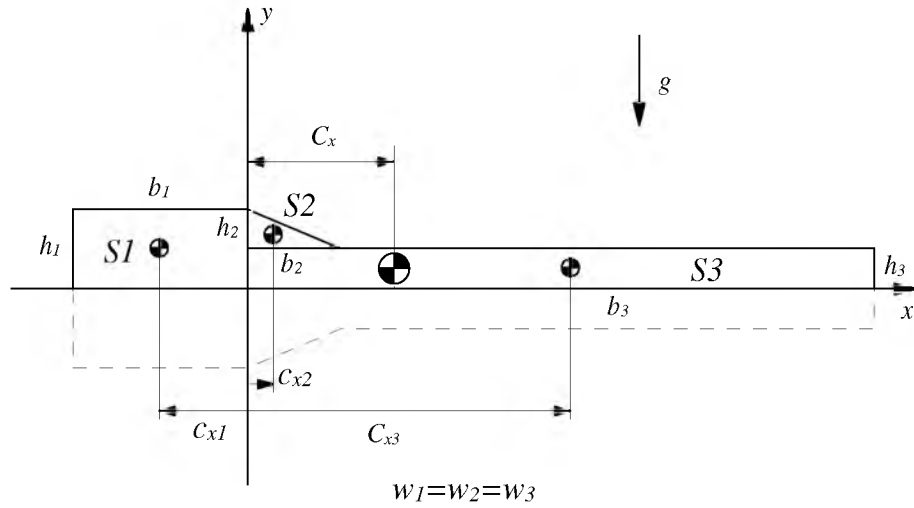


Figura 51: Perfil del eslabón rotacional para robot pendular.

Basado en la figura 51, se tiene entonces que las coordenadas al centro de masa C_x y C_y están dadas por las ecuaciones [11]:

$$C_x = \frac{c_{x1}m_1 + c_{x2}m_2 + c_{x3}m_3}{m_1 + m_2 + m_3} \quad (52)$$

$$C_y = \frac{c_{y1}m_1 + c_{y2}m_2 + c_{y3}m_3}{m_1 + m_2 + m_3}$$

donde

- C_x = coordenada al centro de masa sobre el eje x.
- C_y = coordenada al centro de masa sobre el eje y.
- c_{x1} = coordenada al centro de masa sobre el eje x del solido S1.
- c_{x2} = coordenada al centro de masa sobre el eje x del solido S2.
- c_{x3} = coordenada al centro de masa sobre el eje x del solido S3.
- m_1 = masa del solido S1.
- m_2 = masa del solido S2.
- m_3 = masa del solido S3.
- c_{y1} = coordenada al centro de masa sobre el eje y del solido S1.

- c_{y_2} = coordenada al centro de masa sobre el eje y del sólido S2.
- c_{y_3} = coordenada al centro de masa sobre el eje y del sólido S3.

Sin embargo, debido a que el perfil del eslabón es simétrico sobre el eje x y además el origen del sistema de coordenadas coincide con el eje de giro; la coordenada $C_y = 0$. Como se puede observar en la figura 51 todas las cotas son conocidas en el perfil del eslabón, a excepción de la base del sólido S1. Ésto es debido a que se propone que la distancia al centro de masa $C_x = 5e - 2m$; de este modo se fuerza al eslabón a un balanceo de masas respecto al eje de giro con el objetivo de cumplir con las restricciones de diseño.

Por tanto la coordenada al centro de masa sobre el eje x , es:

$$C_x = \frac{c_{x_1}m_1 + c_{x_2}m_2 + c_{x_3}m_3}{m_1 + m_2 + m_3}$$

Considerando que la masa se expresa en términos de volumen como:

$$m = \rho V$$

donde

$$\rho = \text{Densidad del material} \quad V = \text{Volumen del sólido.}$$

Debido a que todo el eslabón está hecho del mismo material, y suponiendo que la densidad es constante a lo largo de los mismos; la ecuación (52) que expresa la coordenada C_x al centro de masa en términos de masa del eslabón se reescribe como:

$$C_x = \frac{\rho(c_{x_1}V_1 + c_{x_2}V_2 + c_{x_3}V_3)}{\rho(V_1 + V_2 + V_3)}$$

$$C_x = \frac{c_{x_1}V_1 + c_{x_2}V_2 + c_{x_3}V_3}{V_1 + V_2 + V_3} \quad (53)$$

despreciando por tanto la densidad del material y expresando únicamente la coordenada C_x en términos de volumen; la ecuación (53) se reescribe como:

$$C_x = \frac{c_{x_1}(b_1h_1w_1) + c_{x_2}\left(\frac{b_2h_2}{2}w_2\right) + c_{x_3}(b_3h_3w_3)}{(b_1h_1w_1) + \left(\frac{b_2h_2}{2}w_2\right) + (b_3h_3w_3)}$$

donde: b_1 , b_2 y b_3 son las bases de los sólidos S1, S2 y S3 respectivamente; h_1 , h_2 y h_3 las alturas de los sólidos S1, S2 y S3; w_1 , w_2 y w_3 representan el espesor de los sólidos S1, S2, S3. Dado que el espesor del eslabón es uniforme se tiene que: $w_1 = w_2 = w_3$. Por tanto, la ecuación para la coordenada C_x queda expresada en términos de área como:

$$C_x = \frac{w(c_{x_1}(b_1h_1) + c_{x_2}(\frac{b_2h_2}{2}) + c_{x_3}(b_3h_3))}{w((b_1h_1) + (\frac{b_2h_2}{2}) + (b_3h_3))}$$

$$C_x = \frac{c_{x_1}(b_1h_1) + c_{x_2}(\frac{b_2h_2}{2}) + c_{x_3}(b_3h_3)}{(b_1h_1) + (\frac{b_2h_2}{2}) + (b_3h_3)} \quad (54)$$

Sustituyendo los valores numéricos (en milímetros), considerando que c_{x_1} puede expresarse en términos de la base b_1 como $c_{x_1} = b_1/2$, se tiene:

$$C_x = \frac{-\frac{b_1}{2}(35b_1) + 13.3333(300) + 125(5000)}{(35b_1) + (300) + (5000)}$$

$$C_x = \frac{-17.5b_1^2 + 628999.99}{35b_1 + 5300} = 50mm \quad (55)$$

Ahora es posible encontrar una expresión para la medida de la base b_1 del sólido S1:

$$-17.5b_1^2 - 1750b_1 + 363999.99 = 0 \quad (56)$$

Las raíces de la ecuación (56), determinan la longitud de la base del sólido S1 siendo:

$$b_{1x} = 102.6433mm$$

$$b_{1y} = -202.6433mm$$

Tomando la raíz positiva b_{1x} , se tiene que el centro de masa del eslabón se ubica a 50 mm o 5 cm del eje de giro. Sin embargo, para efectos prácticos; se selecciona un valor de $b_{1x} = 100mm$ puesto que resulta mas fácil en la manufactura. Por tanto, al sustituir el valor de $b_1 = 100mm$ en la ecuación (55), la coordenada C_x al centro de masa del eslabón resulta ser:

$$C_x = 51.6mm = 5.16cm$$

De modo que se cumple con la restricción de diseño $C_x \leq 6cm$.

Por otra parte, el máximo torque a 90° generado por el eslabón, no debe superar los 1.72 Nm; puesto que es el torque de parada del motor de CD utilizado. Para conocer el torque máximo del eslabón, se tiene que la masa del mismo está dada por el doble de la suma de las masas de los sólidos primitivos S1, S2 y S3:

$$m_T = 2 [m_1 + m_2 + m_3]$$

$$m_T = 2 \left[\rho(b_1h_1w_1) + \rho\left(\frac{b_2h_2}{2}w_2\right) + \rho(b_3h_3w_3) \right]$$

donde la densidad del aluminio $\rho = 2698,4kg/m^3$. Por tanto, al sustituir los volúmenes de los sólidos S1, S2 y S3 expresados en m^3 , se tiene:

$$m_T = 2 [2698.4kg/m^3((2.2225e - 5m^3) + (1.905e - 6m^3) + (3.175e - 5m^3))]$$

$$m_T = 0.301kg.$$

El torque máximo viene dado por:

$$T = Fd$$

donde F es el peso en Newtons del eslabón y d es la distancia al centro de masa medido desde el eje de rotación del eslabón, debido a que la restricción supone al eslabón en posición horizontal. Por tanto, $F = (0.301kg)(9.81kgm/s^2) = 2.95281N$, y $d = Cx = 0.0516m$. De este modo, se tiene que el torque máximo producido es:

$$T_m = (2.95281N)(0.0516m) = 0.15236Nm$$

Entonces, el par máximo T_m generado por el eslabón a 90° cumple con $T_m < \text{Torque de parada del motor de CD}$. que es de 1.72Nm. Con ello se valida la restricción de máximo torque, indicando que el motor es capaz de hacer girar el eslabón acoplado

a su eje sin problemas en cualquier posición angular deseada.

5.2. Sistema eléctrico y electrónico

5.2.1. Introducción

El sistema eléctrico y electrónico del robot pendular, esta enfocado en la naturaleza electromecánica del actuador (un motor de CD), por tal motivo es necesario dividir el sistema como se en lista a continuación:

- Electrónica de control.
- Electrónica de potencia.

La electrónica de control es el lugar físico donde es embebido el algoritmo de control. Este sistema esta basado en el microcontrolador PIC18F2431 por sus capacidades y facilidades en la implementación de sistemas de control con retroalimentación de movimiento y velocidad. Dicho microcontrolador cuenta con un modulo especializado denominado «modulo de retroalimentacion de movimiento» (MRM). Este modulo es utilizado debido a que la retroalimentación en posición se realiza por medio de un codificador incremental de cuadratura de dos canales.

La señal de referencia, en este caso la posición articular q_d , es provista mediante un teclado matricial de 16 teclas, conectado al modulo de Conversión Analógica Digital del microcontrolador, quien a su vez, despliega la información pertinente en una pantalla LCD.

Por otra parte, es evidente que la electrónica de control no puede ser alimentada a niveles de voltajes superiores a 5V, y esta a su ves no es capaz, de suministrar ni el voltaje, ni la corriente necesaria para accionar el motor con el eslabón acoplado a su eje, por tal motivo es necesaria una etapa de potencia, alimentada con niveles pertinentes de voltaje y corriente. La etapa de potencia esta basada en el arreglo de transistores Push-Pull. Este diseño opera tanto para voltajes positivos como negativos y ademas de su fácil implementación.

Tomando en cuenta que la señal de control proporcionada por el microcontrolador, en este caso es un voltaje representado en código binario de 8 bits; ésta no puede ser directamente aplicada a la etapa de potencia que es de naturaleza analógica. Por tanto, se hace uso de un Convertidor Digital Analógico (CAD) [5], cuya señal convertida es aplicada directamente a la etapa de potencia.

En la figura 52 ,se muestra el diagrama general basado en el microcontrolador PIC18F2431 del sistema de control implementado.

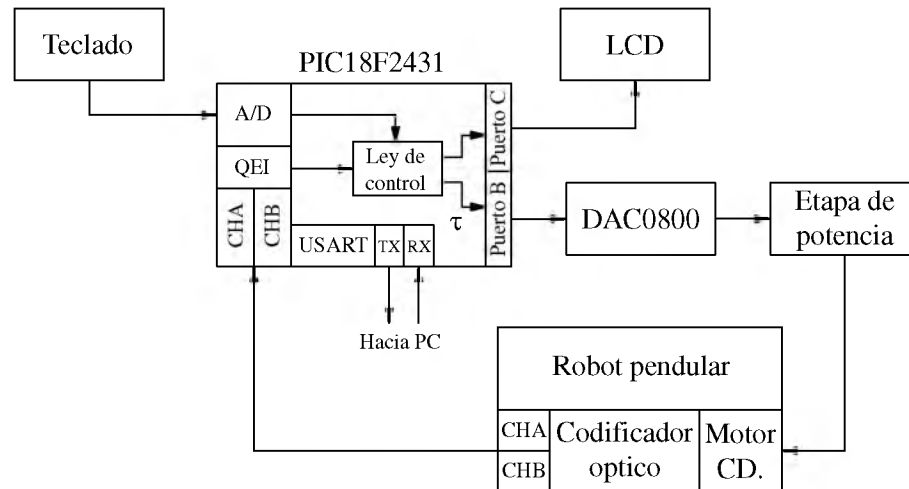


Figura 52: Diagrama del sistema de control, del robot pendular RP1.

Para propósitos de captura y graficación de los datos generados por el microcontrolador, se hace uso del módulo de comunicación serial para la transmisión de los datos a una computadora; como se observa en la figura 52.

A continuación, se presenta la descripción de los bloques mas significativos del sistema de control mostrados en la figura 52.

5.2.2. Motor de CD.

El actuador del robot pendular es un motor de CD. con escobillas e imán permanente modelo 23421012CR, fabricado por la empresa FAULHABER; el cual se muestra en la figura 53.

A continuación, se en listan sus principales parámetros.

- V_n voltaje nominal [V]: 12.
- P_s potencia de salida [W]: 17.
- ω_o velocidad sin carga [rpm]: 8500.
- I_o corriente sin carga [mA]: 75.
- T_s torque después del juego de engranes [Nm]: 1.72.

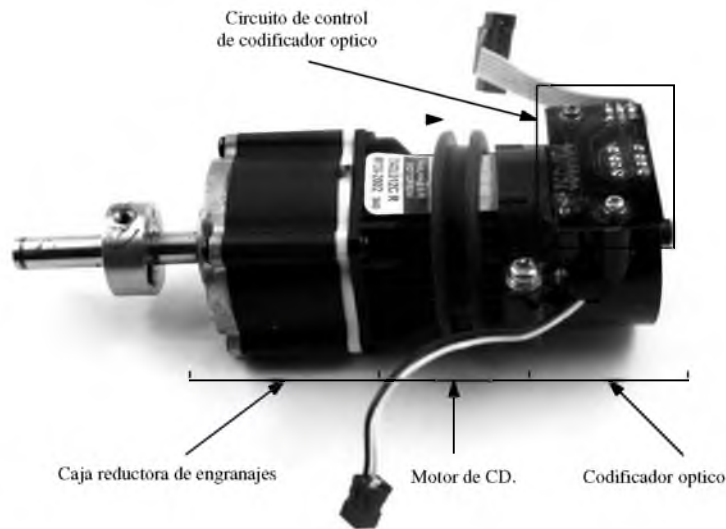


Figura 53: Motor de CD. con escobillas de la marca Faulhaber.

- T_p torque de parada sin juego de engranes [mNm]: 80.
- J_m inercia del motor [kgm^2] = $5.7\text{e-}7$.
- f_m coeficiente de fricción del rotor [Nm] = $1.1848\text{e-}6$.
- K_a constante motor-par [Nm/A] = $13.4\text{e-}3$.
- K_b constante de fuerza contraelectromotriz [Vs/rad] = $13.4\text{e-}3$.
- R_a resistencia de armadura [Ω] = 1.9.
- L_a inductancia de armadura [μH]: 65.
- I_a corriente de arranque [A]: 6.

El motor cuenta con una caja reductora de engrandes tipo planetario con relación de reducción: 1:64 acoplado a su eje y un sistema de censado de velocidad o posición angulares basado en un codificador incremental en cuadratura, con las siguientes características técnicas:

- Resolución [ppr] 12.
- Canales de codificación: dos canales, A y B.

- Tecnología de operación: Óptico de barrera.
- Voltaje de alimentación [V]: 5.

El codificador óptico incremental esta acoplado en la parte anterior del motor, como se muestra en la figura 53. El diagrama del circuito de control del codificador incremental, se presenta en la figura 54.

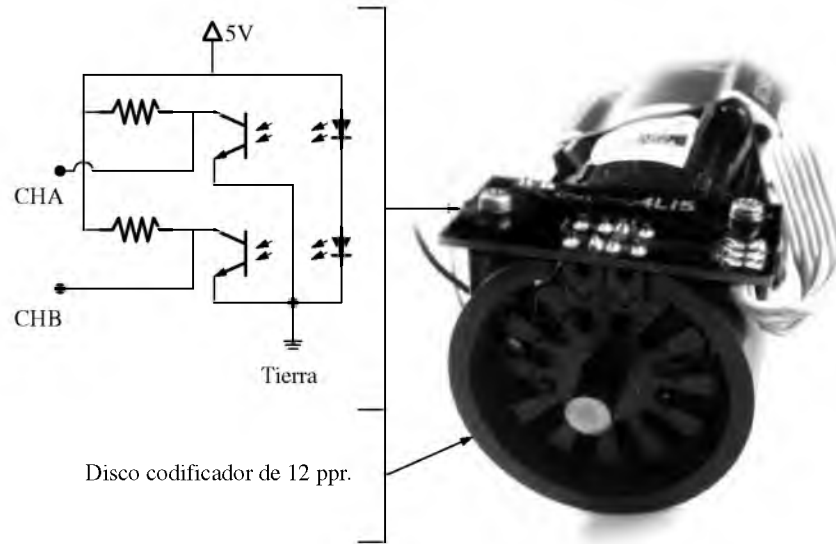


Figura 54: Diagrama del circuito controlador del codificador óptico incremental, acoplado al motor de CD.

Cabe mencionar que es necesario la conexión de dos resistencias externas (mayores a $1K\Omega$), en las entradas CHA y CHB, para poder generar las señales correspondientes a los canales; como se muestra en el diagrama del circuito de la figura 54.

5.2.3. Microcontrolador PIC18F2431

La razón principal de la utilización del PIC18F2431, radica en el módulo integrado de retroalimentación de movimiento (MRM) con el que cuenta, el cual es muy útil para simplificar y agilizar la tarea de interpretar la señal proveniente del codificador incremental acoplado al eje del motor de CD. Por tal motivo, a continuación se presenta información de configuración precisa, específicamente para la interface de decodificador en cuadratura; así como características técnicas del microcontrolador.

El PIC18F2431 forma parte de la familia PIC18F2331/2431/4331/4431, que ofrece las ventajas de todos los microcontroladores PIC18 de alto rendimiento computacional, ya que ofrece la adición de memoria de programa Flash mejorada y una alta velocidad en el convertidor analógico/digital de 10 bits. Además de estas características, la familia PIC18F2331/2431/4331/4431 introduce mejoras de diseño que hacen de estos microcontroladores una elección lógica para muchas aplicaciones de alto rendimiento, como son: control de potencia y control de motores.

Entre los periféricos especiales incluye:

- Módulo de control de potencia PWM, de 14 Bits de resolución.
- Módulo de retroalimentación de movimiento (MRM), que incluye tres canales de módulo de captura y de interface decodificador en cuadratura.
- Convertidor analógico/digital de alta velocidad de 10 Bits de resolución.

La interface decodificador en cuadratura, proporciona una precisa retroalimentación de posición y velocidad. Ésta característica es importante en la implementación del sistema de control para el robot pendular, ya que las señales de posición q y velocidad \dot{q} articulares son indispensables en las leyes de control utilizadas. El diagrama de conexiones del PIC18F2431 se muestra en la figura 55.

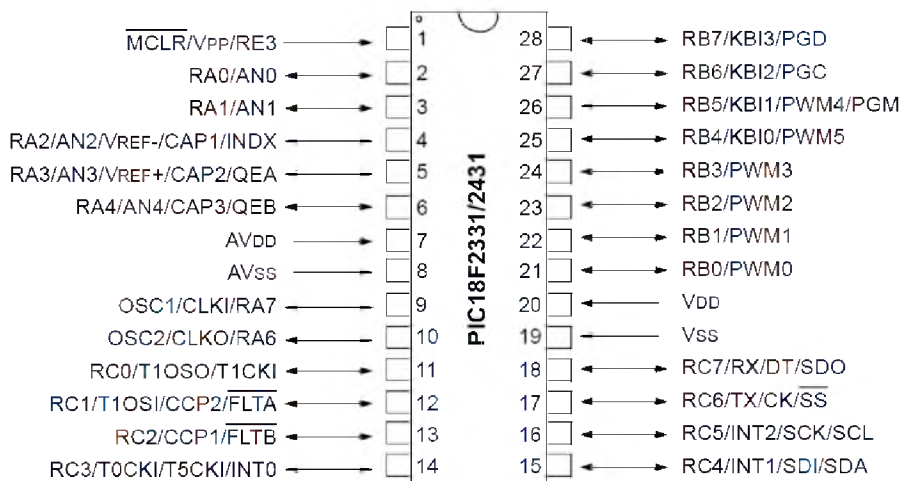


Figura 55: Diagrama de conexiones del microcontrolador PIC18F2431.

Las características técnicas del PIC18F2431 se enlistan a continuación:

- Tipo de memoria de programa: Flash.
- Memoria de programa [KB]: 16.
- Velocidad del CPU [MIPS]: 10.
- RAM [Bytes]: 768.
- Memoria de datos EEPROM [bytes]: 256.
- Periféricos de comunicación digital: 1-UART, 1-A/E/USART, 1-SPI, 1-I2C, 1-SSP(SPI/I2C).
- periféricos de Comparación/Captura/PWM : 2 CCP.
- Contadores: 1x8 bits, 3x16 bits.
- ADC: 5 canales de 10 bits de resolución, multiplexados.
- Rango de temperatura [C]: de -40 hasta 125.
- Rango de voltaje de operación [V]: de 2 hasta 5.5.
- Numero de pines: 28.

5.2.4. Modulo de retroalimentacion de movimiento

El módulo cuenta con una interface decodificador en cuadratura QEI, por sus siglas en ingles de: Quadrature Encoder Interface; y un módulo de captura de entrada IC (siglas en Ingles de: Interface Capture). El módulo QEI acepta dos entradas de fase: QEA y QEB, además de una entrada de vuelta completa INDX (índice), provenientes de un codificador incremental. Puede configurarse en precisión alta o baja en la medición de posición, detección de cambio de dirección y medición de velocidad.

La interface decodificador en cuadratura (QEI) en general, decodifica información de posición y de velocidad; de modo que se puede utilizar en cualquier aplicación que utilice un codificador en cuadratura para retroalimentación de dichas señales. La interfaz tiene las siguientes características:

- Tres entradas QEI: dos señales de fase QEA y QEB, una señal de índice (INDX).

- Detección de dirección de movimiento por interrupción (IC3DRIF).
- Contador de posición ascendente/descendente de 16 bits.
- Modos de seguimiento de posición de precisión estándar y de alta precisión.
- Dos modos de actualización de posición: 2X y 4X.
- Medición de velocidad con un postscaler programable para una rápida medición de velocidad.
- Interrupción en contador de posición (IC2QEIF en el registro PIR3).
- Interrupción en control de velocidad (IC1IF en el registro PIR3).

El módulo QEI tiene tres componentes principales:

- El bloque lógico de control QEI
- El contador de posición
- Y postscaler de velocidad.

El bloque lógico de control del módulo QEI detecta los flancos de las señales en los pines de las entradas de fase QEA o QEB del microcontrolador, y genera el conteo de los pulsos que son enviados al bloque lógico del contador de posición, También se encarga de muestrear la señal de INDX (índice), generar la señal de dirección de rotación (adelante/reversa) y las señales de eventos de velocidad.

El contador de posición actúa como un integrador para el seguimiento de la distancia recorrida. Los flancos de entrada de QEA y QEB sirven como estímulos para crear la entrada de reloj, que avanza las posiciones en el registro de conteo de 16 bits (POSCNT). El registro es incrementado dependiendo del modo de operación para ambos casos, ya sea con los flancos de entrada de QEA, o con los flancos de entrada de QEA y QEB. El contador se restablece ya sea por una coincidencia con un valor cargado en el registro de período MAXCNT, o con una señal externa aplicada al pin de entrada para índice (INDX). Se genera una interrupción cuando se restablece el registro de contador POSCNT si la interrupción de posición del contador se habilita. El diagrama simplificado del módulo QEI se muestra en la figura 56.

El postscaler de velocidad es utilizado para programar el periodo de conteo de los pulsos de velocidad a una frecuencia constante, es decir; se cuenta los pulsos de velocidad en determinado tiempo a una frecuencia constante haciendo uso del TIMER5.

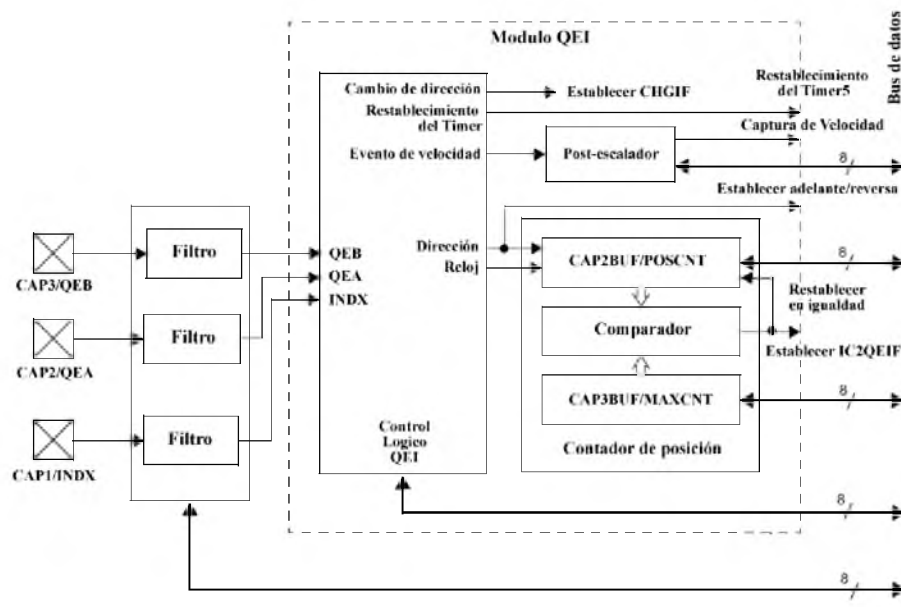


Figura 56: Diagrama simplificado de bloques del módulo QEI, del PIC18F2431.

5.2.5. Configuración del módulo QEI

El módulo QEI comparte sus pines de entrada con el módulo de Captura (IC). Las entradas son mutuamente exclusivas, sólo el módulo de IC o el módulo QEI pero no los dos se pueden activar al mismo tiempo; debido a que los módulos IC y QEI están multiplexados a los mismos pines de entrada del microcontrolador. Los filtros de ruido programables pueden ser dedicados solamente a un módulo. La operación del módulo QEI es controlada por el registro de configuración QIECON de 8 bits, el cual se describe a continuación:

\overline{VELM}	QERR	$\overline{UP/DOWN}$	QEIM2	QEIM1	QEIM0	PDEC1	PDEC0
Bit 7							Bit 0

Bit 7 \overline{VELM} : Bit de modo velocidad.

- 1 = Modo Velocidad desactivado.
- 0 = Modo Velocidad activado.

Bit 6 QERR: Bit de error de QEI.

- 1 = Desbordamiento o subdesbordamiento, del contador de posición.
- 0 = No hay desbordamiento, ni subdesbordamiento.

Bit 5 $\overline{UN/DOWN}$: Bit de estatus de dirección de rotación.

- 1 = Adelante.
- 0 = Reversa.

Bit 4-2 QEIM2:QEIM0: Bits para configuración de modos del módulo QEI.

- 111 = No usado.
- 110 = QEI activado en modo 4x; restablecimiento del contador de posición cuando POSCNT = MAXCNT.
- 101 = QEI activado en modo 4x; restablecimiento del contador de posición por señal de INDX.
- 100 = No usado.
- 010 = QEI activado en modo 2x; restablecimiento del contador de posición cuando POSCNT = MAXCNT.
- 001 = QEI activado en modo 2x; restablecimiento del contador de posición por señal de INDX.
- 000 = QEI inactivo.

Bit 1-0 PDEC1:PDEC0: Bits de relación de reducción de velocidad de pulso.

- 11 = 1:64.

- $10 = 1:16$.
- $01 = 1:4$.
- $00 = 1:1$.

Con el módulo QEI configurado correctamente, es posible acceder a la medición de la posición leyendo el contenido del registro POSCNT; el cual se actualiza con cada tren de pulsos provenientes de las entradas QEA y QEB del codificador incremental, incrementando o decrementando el conteo según el sentido de giro.

5.2.6. Convertidor Digital/Analógico DAC0800

Debido a que los controles diseñados para el robot pendular (PD y PD con compensación precalculada de gravedad), son programados en el microcontrolador, y este a su vez entrega la compensación como una señal digital representada en un número binario de 8 bits (el cual expresa el voltaje necesario para generar determinado torque en el motor), es evidente que la señal es incompatible con el motor de CD, debido a que el motor es de naturaleza analógica. Por tal motivo, es necesario utilizar un convertidor digital/analógico (CDA); el cual convierte el número binario a una señal continua de voltaje equivalente [5]. El CDA utilizado para tal propósito, es el integrado DAC0800 de Texas Instruments. A continuación se describen las características técnicas y capacidades del mismo.

El DAC0800 es un convertidor digital/analógico encapsulado en un circuito integrado monolítico de 8 bits de resolución y de alta velocidad de conversión (alrededor de 100 ns). Es utilizable en modo unipolar o bipolar, es decir; configurado para trabajar con representaciones positivas de voltaje o representaciones positivas y negativas de hasta ± 20 Vpp. Cuenta con entradas inmunes al ruido, por lo que es compatible con múltiples niveles lógicos de tecnologías distintas. Los voltajes de los niveles lógicos es independiente del voltaje de alimentación. En la figura 57 se muestra el diagrama de conexiones del DAC0800.

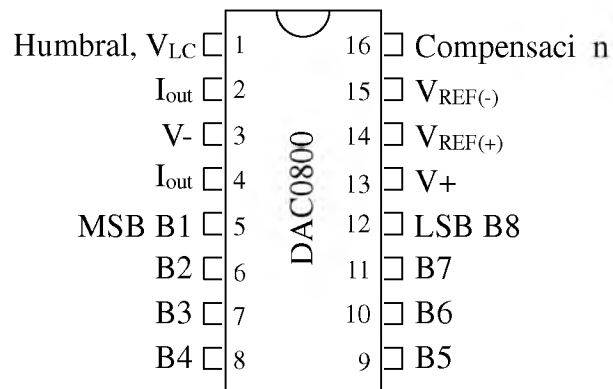


Figura 57: Diagrama de conexiones del DAC0800.

Las características y capacidades técnicas, se en listan a continuación:

- Rápido tiempo de conversión [ns]: 100.
- Error a escala completa: ± 0.1 LSB.
- Alto voltaje diferencial de salida [V]: -10 a 18.
- Interface directa con: TTL, CMOS y PMOS.
- Rango de voltaje de alimentación [V]: ± 4.5 a ± 18 .
- Bajo consumo de potencia [mW]: 33 en $\pm 5V$.
- Numero de pines: 16.

El torque generado es bidireccional debido a que la señal de control representa un voltaje bipolar y utiliza la configuración del convertidor digital/analógico de igual forma, en modo bipolar. El diagrama de configuración del DAC0800 en modo bipolar, se muestra en la figura 58.

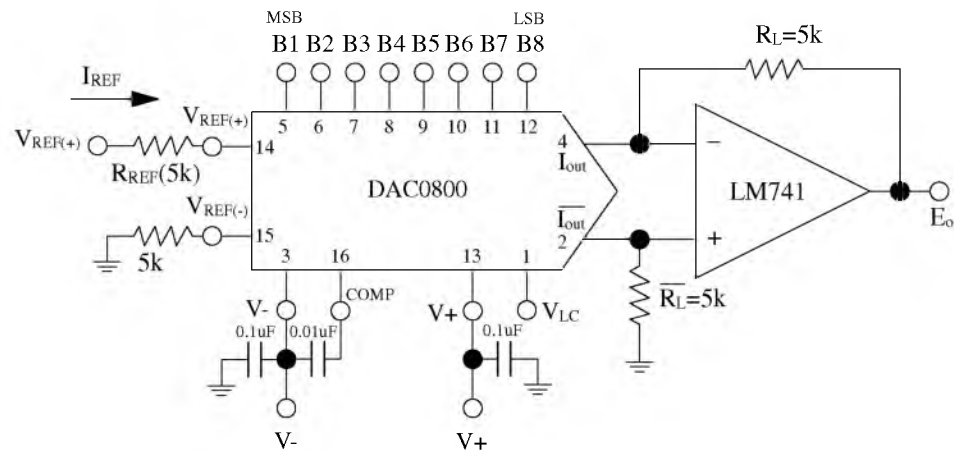


Figura 58: Configuración del convertidor digital/analógico, DAC0800 en modo bipolar.

Como se observa en la figura 58, es necesario conectar las salidas del DAC0800 a un amplificador operacional configurado como conversor de corriente a tensión, de modo que el voltaje a la salida del amplificador operacional cumpla con:

$$E_{SAL} = V_{REF} \left(-\frac{255}{256} + \frac{2x}{256} \right)$$

Donde x es el código digital de entrada. Además se debe cumplir que: $R_L = \overline{R_L} = R_{REF}$.

Teniendo finalmente en E_{SAL} , una representación equivalente analógica del voltaje en código binario con tamaño de palabra de 8 bits procesada por el microcontrolador, con base en las leyes de control en él programadas.

5.2.7. Etapa de potencia

La señal de voltaje E_{SAL} producida por el convertidor digital/analógico como equivalencia de la señal digital entregada por el microcontrolador, no tiene la suficiente corriente para generar el par requerido en el motor de CD. Para efectuar el movimiento deseado (debido a que la corriente proporcionada por el amplificador operacional está restringida a un nivel determinado), es necesario que esta señal pase por una etapa que le provea de la potencia necesaria, pero con el mismo nivel de voltaje. Es decir, que la señal sea equivalente en voltaje pero mayor en corriente a la salida de la etapa de potencia.

La etapa de potencia utilizada esta basada en el arreglo de transistores, denominado Push-Pull, el cual consiste en un par de transistores de potencia, del tipo NPN y PNP, de las mismas características [7], interconectados como se observa en la figura 59.

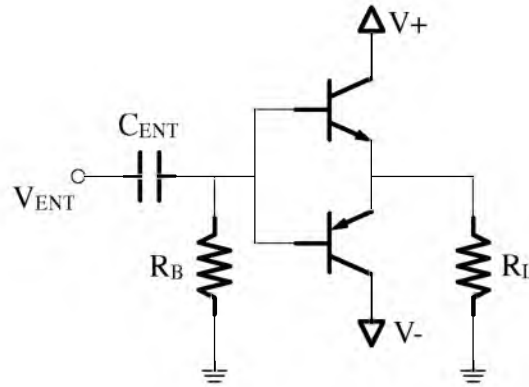


Figura 59: Etapa de potencia Push-Pull, basada en dos transistores de potencia NPN y PNP.

Si una señal senoidal de voltaje V_{ENT} es proporcionado al arreglo Push-Pull, se produce entonces un voltaje V_{SAL} proporcional a V_{ENT} con las características mostradas en la figura 60.

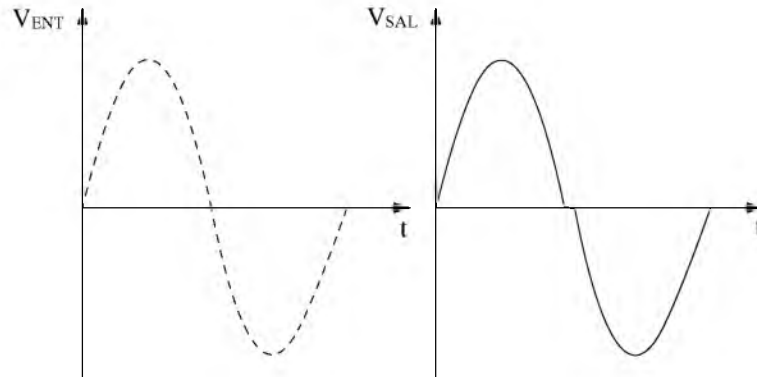


Figura 60: Forma de onda resultante V_{SAL} , del arreglo de transistores Push-Pull, al circular un voltaje V_{ENT} de forma senoidal.

De la figura 60 se observa que cuando la señal V_{ENT} se encuentra en el semiciclo positivo de la forma de onda, el transistor Q1 se encuentra en polarización directa; por lo que circula la corriente hasta la resistencia de carga R_L , y el transistor Q2 se encuentra en corte. Cuando el semiciclo de onda es negativo sucede lo contrario,

es decir; el transistor Q1 se encuentra en corte y el transistor Q2 en polarización directa, lo que permite el paso de la corriente hacia la resistencia de carga R_L generando a la salida un voltaje V_{SAL} proporcional al voltaje de entrada. Sin embargo, puede apreciarse en la figura 60, el efecto en la forma de onda V_{SAL} cuando ocurre el cruce por cero de V_{ENT} ; produciéndose una distorsión típica debido al voltaje de 0.6V que existe entre emisor y colector. Esta distorsión puede corregirse añadiendo una fuente de voltaje entre las bases de los transistores. Sin embargo, para el propósito requerido en el robot pendular, este efecto no es significativo; puesto que dicha distorsión ocurre cuando el voltaje de salida V_{SAL} es cercano a cero tanto de forma negativa como positiva. Esto se debe a que voltajes tan pequeños no provocan cambios en el actuador, ya que requiere mas de 0.6V aplicado en sus terminales para generar un movimiento [7].

Es importante mencionar, que la fuente de alimentación para la etapa de potencia es independiente de la fuente de alimentación para el microcontrolador y el convertidor digital/analógico, por tanto dependiendo de la carga del circuito de potencia, que en este caso es el motor de CD., la fuente puede proporcionar la corriente necesaria, teniendo así finalmente la señal de voltaje V_{SAL} con la suficiente potencia para producir el torque que el microcontrolador computo, para corregir el error.

5.2.8. Comunicación serial mediante USART

Debido a que el propósito de comunicación con un dispositivo externo de la tarjeta de control del robot pendular, es el de graficación de los datos de posición q ; es necesario solo el envío de datos sin efectuar ninguna recepción de datos transmitidos por el dispositivo receptor hacia el microcontrolador. Para tal fin, es utilizado el tipo de comunicación serial denominada USART (Siglas en ingles de: Universal Synchronous Asynchronous Reciver Transmitter); principalmente por que el PIC18F2431 tiene implementado hardware dedicado para tal propósito y además, porque este tipo de comunicación cumple con las necesidades de trasmisión requeridos.

El modulo USART, es una de las formas que el PIC18F2431 posee para la transmisión/recepción de datos de forma serial. El USART es capaz de transmitir y recibir información de las siguientes formas:

- Asíncrona (full-duplex).

- Sincronía (Half-duplex).

Cuando la transmisión es realizada en modo síncrono, es necesario contar con la estructura: maestro-esclavo, donde el maestro dicta la señal de sincronización o de reloj. Por tal motivo, solo es posible la transmisión de los datos hacia una sola dirección al mismo tiempo (Half-duplex). Por otra parte, cuando se realiza una transmisión de datos de forma asíncrona, ya no es necesario contar con una señal de reloj que controle la transmisión, por lo tanto; se trata de comunicación bidireccional (Full-duplex) [6].

Los pines que sirven para establecer la comunicación mediante el USART en el PIC18f2431 son:

- Pin 17: RC6/TX/CK como pin de salida a línea de transmisión en modo asíncrono o línea de señal de reloj en modo síncrono.
- Pin 18: RC7/RX/DT como pin de entrada o receptor en modo asíncrono o línea de señal de datos en modo síncrono.

En la figura 61, se presenta el esquema de establecimiento de comunicación de forma asíncrona y síncrona, entre el PIC18F2431 y algún dispositivo externo [10].

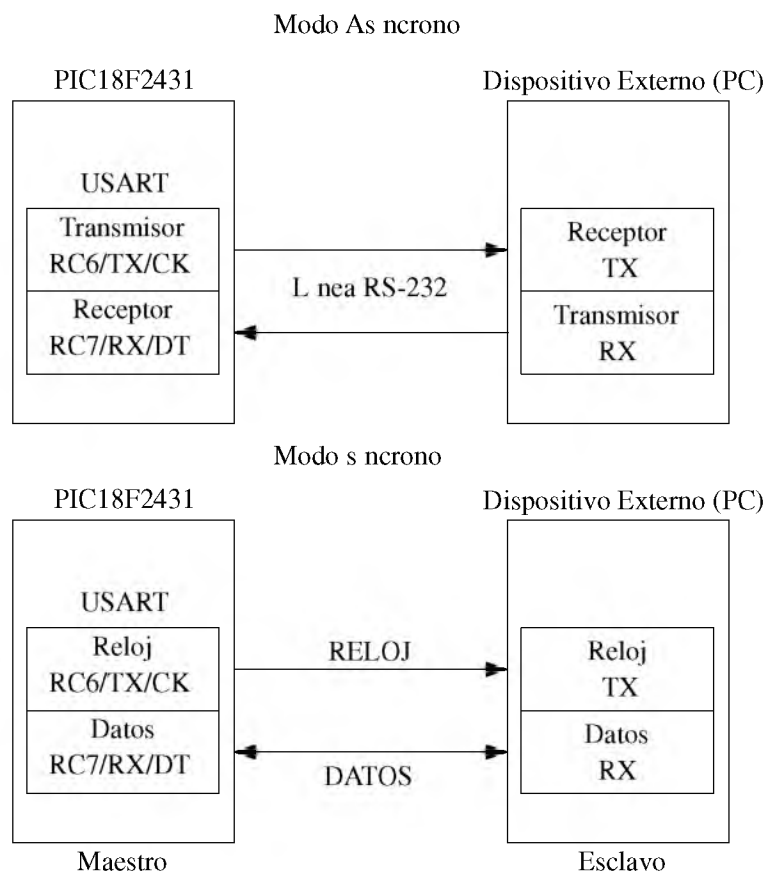


Figura 61: Esquema de comunicación serial sincrona y asincrona mediante USART.

De la figura 61 se observa que para la comunicación asíncrona, la transmisión de los datos se realiza sobre las dos líneas que salen de RC6 y RC7 del microcontrolador; se deduce entonces, que las líneas se conectan en forma entrelazada, es decir; TX del microcontrolador se conecta con RX del dispositivo externo, y a su vez TX del dispositivo externo con RX del microcontrolador. La frecuencia del envío de los datos, es determinada por el módulo USART de manera interna.

En el esquema de comunicación síncrona que se observa en la figura 61, se tiene que el microcontrolador es el maestro y el dispositivo externo es el esclavo, por tanto; la línea de reloj que controla el ritmo del envío de datos se establece entre la salida de CK del microcontrolador (pin RC6) y el puerto CK del dispositivo externo. La línea de datos por tanto, se establece entre el pin RC7 de microcontrolador y el puerto DT del dispositivo externo.

El envío de datos está establecido bajo la norma de comunicación serial RS-

232, donde las palabras de información están empaquetadas en cadenas de bits (de generalmente 8 o 9 bits de longitud), y son independientes unas de otras.

La trama de palabra en el protocolo estándar RS-232 se muestra en la figura 62, donde se aprecia que la línea de datos está normalmente en un estado lógico alto.

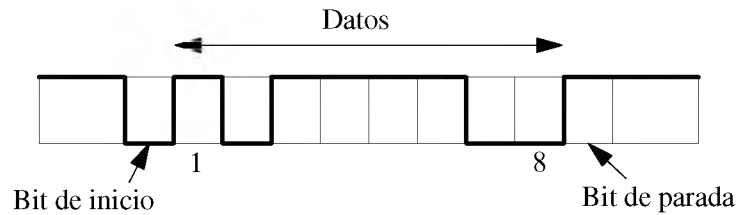


Figura 62: Trama para envío de datos, del protocolo estándar RS-232, utilizado en la comunicación serial mediante USART.

De la trama de datos del protocolo RS-232 que se observa en la figura 62, se aprecia que está compuesta principalmente por 8 bits para el tamaño de palabra de datos, precedido por un bit de inicio y finalizando con un bit de parada. También puede extenderse el tamaño de palabra de datos a 9 bits.

En el protocolo RS-232 para la transmisión serial de datos de forma asíncrona, la frecuencia en bits por segundo (baudios) a la que se realiza la transferencia de datos; se hace a un valor normalizado [10]. Alguno de estos valores son: 330, 600, 1200, 2400, 4800, 9600, 19200 y 38400.

Para la implementación de la transmisión de datos de forma asíncrona, en el PIC18F2431 se hace uso de una frecuencia de envío de datos de 9600 baudios. A pesar de que la transmisión es asíncrona (Full-duplex), en la práctica se ha optado por desactivar la línea de recepción de datos hacia el microcontrolador puesto que el propósito es simplemente el envío de paquetes de información hacia la unidad receptora.

5.2.9. Tarjeta electrónica de control

Con base en la descripción de bloques del sistema de control para el robot pendular mostrado en la figura 52, se construyó una tarjeta electrónica de circuito impreso basada en el microcontrolador PIC18F2431 para el control del robot. En la figura 63 se muestra la tarjeta electrónica de control construida, indicando sus diferentes bloques y componentes relevantes.

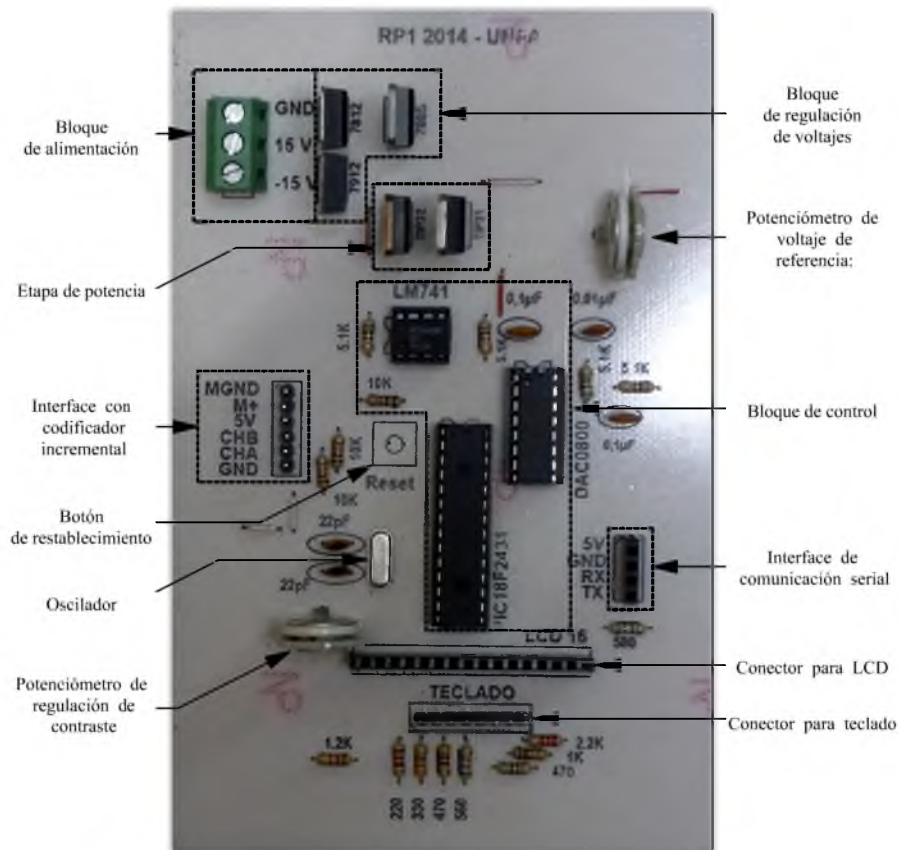


Figura 63: Tarjeta electrónica de control, para robot pendular.

Las características técnicas de la tarjeta de control diseñada y construida se en listan a continuación:

- Voltaje de alimentación [V]: ± 15 .
- Dimensiones [cm]: 15x10.
- Interface con codificador en cuadratura incremental de dos canales.
- Interface de comunicación serial, por protocolo USART.
- Etapa de potencia.
- Visualización de interface de configuración por medio de LCD.
- Entrada de datos por teclado matricial de 4x4.

A continuación, se presenta la descripción de los diferentes bloques que integran la tarjeta de control.

Bloque de Alimentación: La tarjeta es alimentada únicamente con $\pm 15V$, los cuales son enviados directamente a la etapa de potencia basada en dos transistores: TIP31 y TIP32, en arreglo Push-Pull.

Bloque de Regulación: Los voltajes de alimentación son regulados a $\pm 12V$ para alimentar al amplificador operacional LM741 y al integrado DAC0800, por medio de los reguladores LM7812 y LM7912. La tensión requerida de 5V para la electrónica de control, es regulada de 12V a 5V por medio del regulador de voltaje LM7805. Dicho voltaje es suministrado directamente al microcontrolador PIC18F2431, así como a los demás periféricos que requieren de la misma tensión de voltaje tale como: LCD, teclado matricial, interface con codificador en cuadratura y la interface de trasmisión de datos serial.

Bloque de Control: El circuito de control está basado en los integrados PIC18F2431, el convertidor digital/análogo DAC0800 y el amplificador operacional LM741. El microcontrolador PIC18F2431 entre otras tareas, se encarga de la gestión de los dispositivos de entrada: teclado matricial y de salida: LCD, así como la trasmisión de los datos a través del puerto serial integrado.

Botón de reset: La tarjeta de control, cuenta con un botón de restablecimiento denominado Reset; mismo que al ser presionado, regresa el microcontrolador a su estado inicial.

Potenciómetro de regulación de contraste: Se utiliza para ajustar el contraste del LCD para una óptima visualización de la información presentada.

Potenciómetro de voltaje de referencia: Se utiliza para ajustar el voltaje de referencia del conversor digital/análogo. Normalmente está ajustado para proporcionar una tensión de voltaje de 2.5V.

Interface con codificador incremental: La interface dispone de seis pines provistos por un conector macho donde se conectan el codificador incremental y el motor de CD. La descripción de las funciones de cada uno de los pines se presenta a continuación, en el orden de aparición en la tarjeta:

- Pin 1: MGND: Corresponde a la tierra que debe conectarse a una terminal del motor de CD.
- Pin 2: M+: A través del pin es suministrada la señal de control hacia el motor de CD.
- Pin 3: 5V: Suministra 5V al circuito electrónico del codificador incremental.
- Pin 4: CHB: Recibe la señal proveniente del canal A del codificador incremental.
- Pin 5: CHA: Recibe la señal proveniente del canal B del codificador incremental.
- Pin 6: GND: Señal de masa o tierra para el circuito electrónico del codificador incremental.

Interface de comunicación serial: Por medio de esta interface, se transmite la posición q articular a una tasa de transferencia de 9600 baudios; ya sea hacia una computadora o algún otro dispositivo capaz de efectuar la lectura. La interface cuenta con un conector hembra de cuatro pines o entradas, los cuales se describen a continuación en el orden de aparición en la tarjeta de control:

- Pin 1: 5V: Debido a que la transmisión es unidireccional, este pin debe puentearse con el pin RX para evitar que el microcontrolador mal interprete la señal si este estuviera al aire considerándolo como entrada de datos.
- Pin 2: GND: Sirve de tierra común para el dispositivo que esté conectado y se encarga de la transmisión de los datos.
- Pin 3: Rx: Este pin no es utilizado debido a que se efectúa una transmisión de datos de forma unidireccional asíncrona simulada, por ello; es conveniente enlazarlo con el pin de 5V para evitar mal funcionamiento, ya que el módulo del microcontrolador está configurado en modo serial asíncrono; por lo que se espera una recepción de datos a través de esta línea.
- Pin 4: Tx: Mediante este pin, se realiza la transmisión de los datos hacia la unidad receptora.

Conector para LCD: Este conector hembra de 16 pines, está rodeado por un rectángulo, el cual tiene marcado uno de sus vértices con un triángulo. El LCD debe ser conectado haciendo coincidir el pin 1 del mismo con dicho triángulo.

Conector para teclado: El conector macho de 8 pines, está rodeado por un rectángulo, el cual tiene marcado uno de sus vértices con un triángulo. El teclado matricial debe ser conectado haciendo coincidir el pin 1 del mismo con dicho triángulo.

Oscilador: La tarjeta de control cuenta con un oscilador de cristal de 4 Mhz de frecuencia, el cual provee la señal de reloj para el microcontrolador PIC18F2431.

5.3. Implementación del algoritmo de control

5.3.1. Introducción

Debido a las necesidades requeridas para implementar todo el conjunto de partes que conforman el sistema de control de posición del robot pendular como son: dispositivos de entrada/salida, gestión de señales de retroalimentación, interface con el usuario y procesamiento de datos; se requiere de un lenguaje de alto nivel para evitar complejidad innecesaria en el desarrollo e implementación del algoritmo que rige el sistema electrónico de control; el cual está basado en el microcontrolador PIC18F2431.

Por tanto, se hace uso del lenguaje C de alto nivel para la programación del algoritmo de control. Dicho lenguaje está desarrollado en el software PIC C Compiler, el cual ofrece funciones especiales para lograr un manejo fluido de los módulos dedicados del microcontrolador; que en este caso son: Módulo de retroalimentación de movimiento, específicamente el submódulo de interface decodificador de cuadratura QEI, así como el módulo de transmisión de datos serial sincrónica/asincrónica USART.

Por otra parte, los algoritmos de control de posición diseñados, e implementados en lenguaje C, se en listan a continuación:

- Control de PD.
- Control PD con compensación precalculada de gravedad.

Para implementar los algoritmos basados en las leyes de control de posición PD y PD con compensación precalculada de gravedad, es preciso identificar la naturaleza de las diferentes señales involucradas, y como estas pueden ser procesadas por el microcontrolador. Para ello en la figura 64, se observa el diagrama a bloques del sistema de control, haciendo énfasis en tipo de señal que interconectan a cada parte.

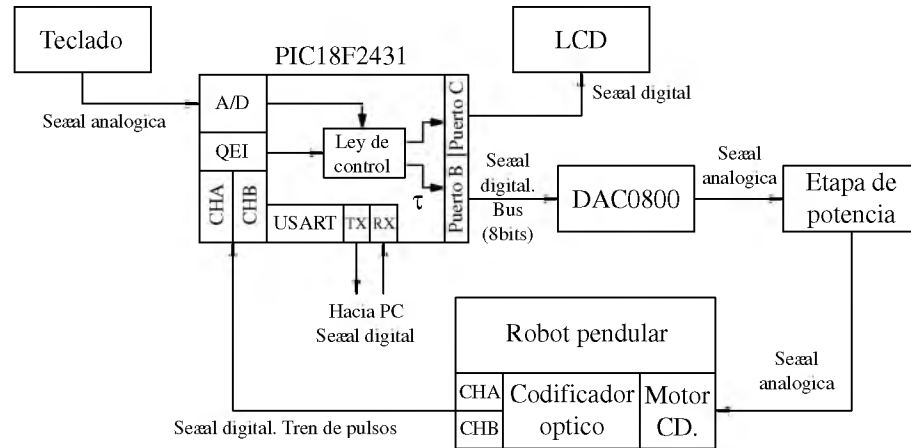
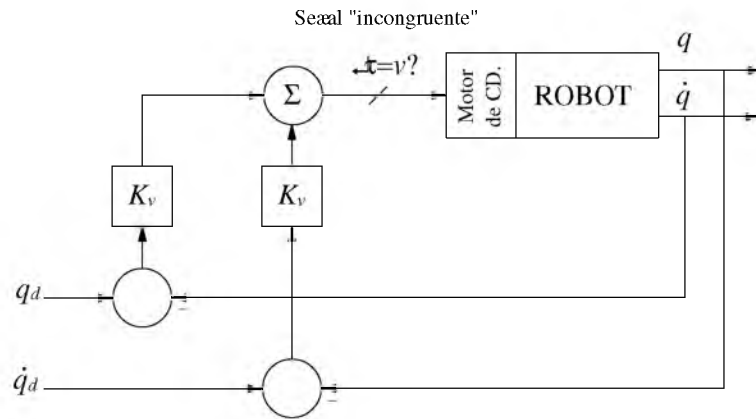


Figura 64: Diagrama de bloques del sistema electrónico de control, para robot pendular, tomando en consideración la naturaleza de las señales involucradas.

El objetivo de ambos controladores (PD y PD con compensación precalculada de gravedad) es calcular la compensación del error en términos de par, y las leyes de control son procesadas dentro del microcontrolador. Para la implementación de estos controles tal cual fueron expresados hasta el momento, existe un inconveniente, y éste radica en un punto específico del lazo de control en ambos tipos de control; lo cual se ilustra en la figura 65.

Control PD (Proporcional Derivativo)



Control PD con compensacion precalculada de gravedad

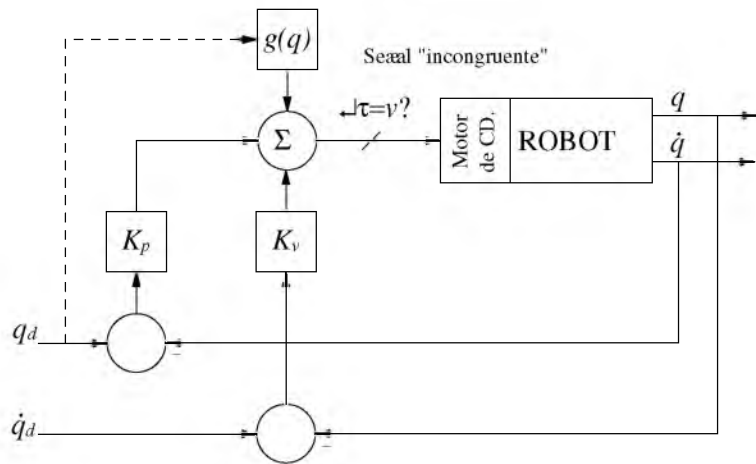


Figura 65: Inconsistencia aparente, de señales en el lazo de retroalimentación, de los controladores de posición PD y PD con compensación precalculada de gravedad.

De la figura 65 se observa que, a pesar de que el microcontrolador calcule satisfactoriamente el par necesario basado en alguna de las dos leyes de control implementadas, éste no tiene forma de comunicarle dicho par calculado al motor de CD. Esto se debe a que el microcontrolador está expresado en variables mecánicas, y el motor de CD sólo puede recibir en sus terminales, un voltaje eléctrico para efectuar movimiento. Esta incongruencia de señales es corregida considerando que:

$$\tau = k_s v$$

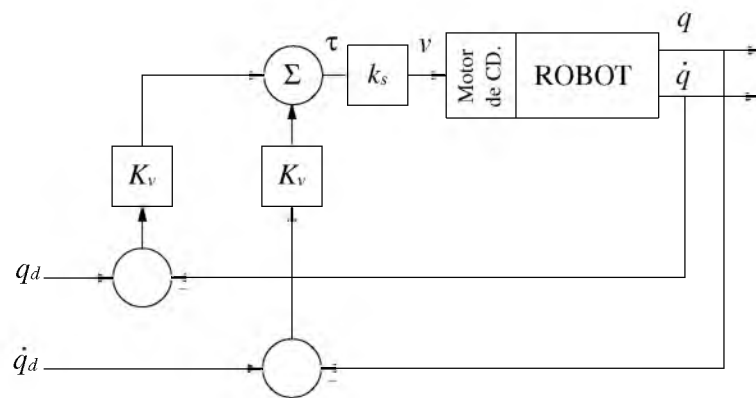
donde k_s es una constante de proporcionalidad que generalmente se conoce

como ganancia de servoamplificador y v es el voltaje aplicado a las terminales del motor para generar el par τ [8]. De esta manera se obtiene una expresión que relaciona el torque con el voltaje como:

$$v = \frac{\tau}{k_s}$$

de modo que es preciso entonces, conocer la magnitud de la constante k_s . Esta constante sera adicionada como un bloque mas al lazo de control de ambas leyes tal y como se ilustra en la figura 66, con el objetivo de hacer la conversión par-voltaje y de esta manera entregar una señal en voltaje proporcional al par calculado, congruente (en términos de señal) al motor de CD.

Control PD (Proporcional Derivativo)



Control PD con compensacion precalculada de gravedad

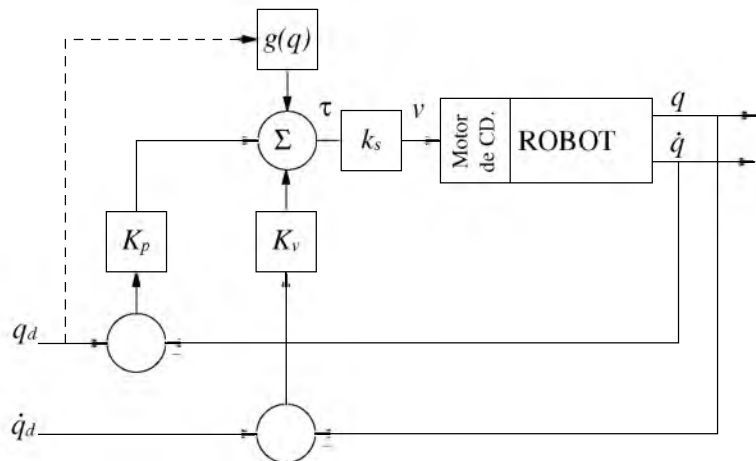


Figura 66: Bloque de conversión par-voltaje, añadido al lazo de los controles de posición PD y PD con compensación precalculada de gravedad.

La constante k_s para el caso del motor seleccionado como actuador del robot pendular, toma en cuenta los siguientes valores:

- V_n voltaje nominal [V]: 12.
- T_o torque de salida sin carga [Nm]: 1.72.

Tomando en cuenta el voltaje nominal y el torque de salida del motor de CD, la constante k_s se expresa como:

$$k_s = \frac{\tau}{v} = \frac{1.72Nm}{12V}$$

$$k_s = 0.1433$$

Finalmente, en la implementación del algoritmo para el sistema de control digital que gestiona la ejecución de ambas leyes de control; es preciso establecer qué parámetros pueda configurar de forma interactiva, el usuario final del robot. A continuación, se en listan los parámetros configurables por el usuario del programa final del sistema electrónico de control.

- Selección del tipo de control a ejecutar (PD y PD con compensación precalculada de gravedad).
- Configuración de ganancias K_p y K_v , siempre positivas.
- Configuración inicial de la posición deseada q_d .
- Configuración sobre la marcha (cuando esté en ejecución alguna de las dos leyes de control) de la posición deseada q_d .
- Retorno a cero automático del eslabón sobre la marcha.

Con base en lo anterior, en la figura 67 se muestra el diagrama de flujo del sistema electrónico de control completo: el cual es implementado en lenguaje C en el microcontrolador PIC18F2431.

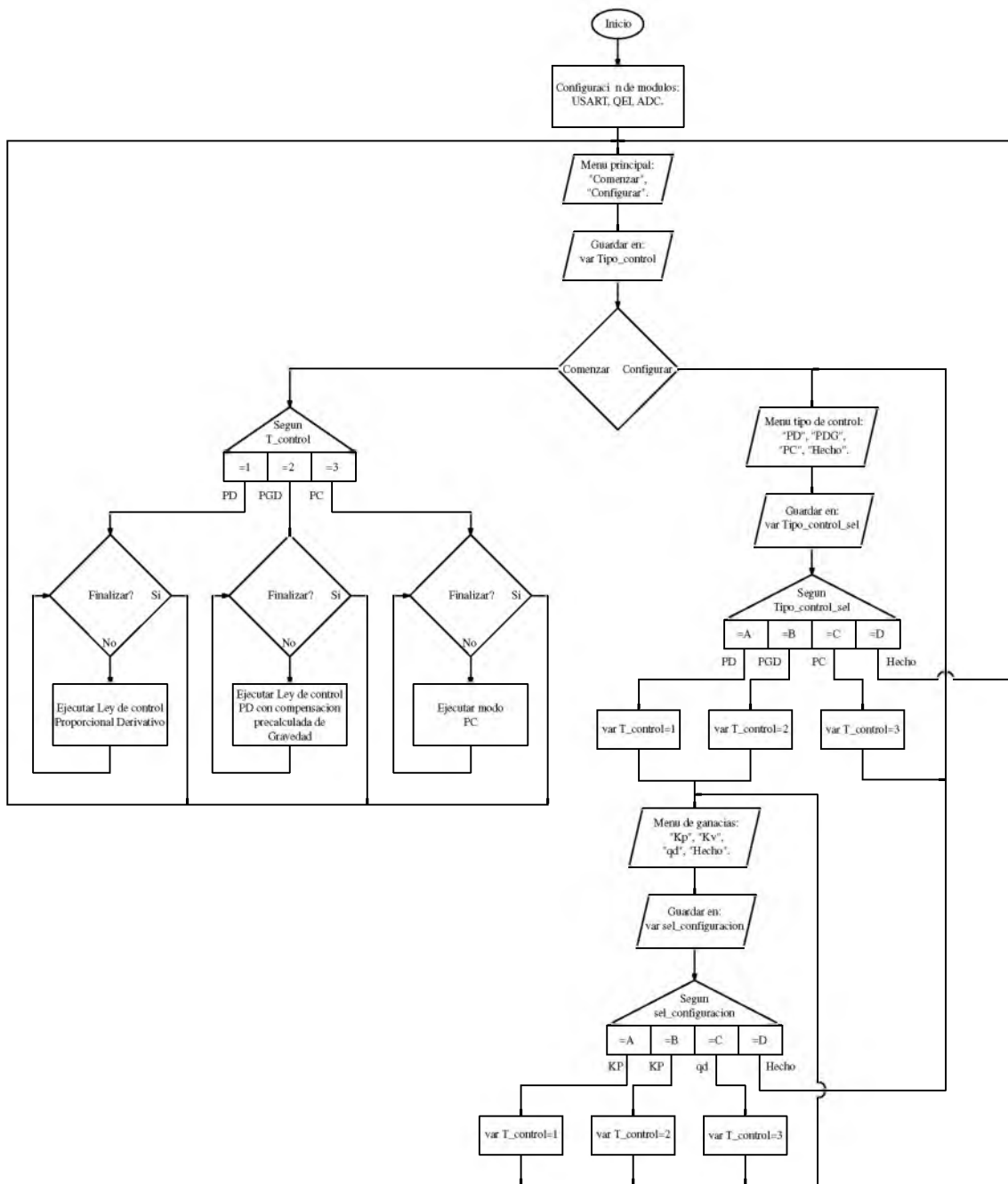


Figura 67: Diagrama de flujo del algoritmo del sistema electrónico de control.

5.3.2. Estimación de la velocidad articular \dot{q}

Un parámetro importante involucrado en las leyes de control de posición PD y PD con compensación precalculada de gravedad, es la retroalimentación de la velocidad articular \dot{q} . En este caso, dicho parámetro no es directamente medido

desde el motor mediante algún sensor específico; sino que es estimado mediante derivación numérica. Este consiste en el cálculo de la razón de la diferencial de posición entre la diferencial de tiempo; lo cual se expresa como:

$$\dot{q} = \frac{q(T) - q(T - 1)}{h}$$

donde $q(T)$ es la posición articular actual tomada del codificador en cuadratura, $q(T - 1)$ es la posición articular q obtenida un muestreo antes, y h es el tiempo de muestreo; el cual debe ser lo suficientemente pequeño para reducir el error de aproximación [8].

6. Conclusiones

El diseño y construcción del prototipo experimental abordado en esta tesis, está basado en un robot pendular y surge del estudio de los modelos cinemáticos y dinámicos del mismo, sirviendo como plataforma de pruebas de las leyes de control de posición PD y PD con compensación precalculada de gravedad. El modelo dinámico del robot está estrechamente relacionado con el diseño de las leyes de control de posición. Por otra parte, es necesario no sólo tomar en cuenta la contribución mecánica en la dinámica del robot, sino también la del actuador; que en el caso del robot pendular es un motor de CD.

Durante el desarrollo del prototipo, surgieron en general dos inconvenientes. El primero referente a la validación del modelo dinámico del actuador que fue realizada en Simulink, el cual, debido al parámetro de inductancia muy pequeño; generaba errores de simulación. Dicho problema finalmente se corrigió disminuyendo el paso de integración.

El segundo inconveniente presentado, se relacionó con la incongruencia aparente entre las señal generada por el control para producir movimiento en el motor; puesto que es de naturaleza mecánica y el motor tiene como entrada una señal de voltaje. El error fue corregido agregando un bloque mas al lazo de control entre el par calculado por el control y el motor de CD. Dicho bloque contiene una ganancia, que relaciona las variables par-voltaje.

Se construyó satisfactoriamente un circuito electrónico de control montado en una tarjeta de circuito impresa y basado en el microcontrolador PIC18F2431. Dicho microcontrolador debido a sus características en hardware, resultó adecuado

en la implementación. Sin embargo, dado que tiene un número limitado de puertos de entrada y salida, dio lugar a problemas con el manejo de periféricos de entrada y salida en términos de agotamiento de recursos. Esto propició que se optimizara el uso del teclado matricial, no mediante muestreo, sino utilizando un arreglo de resistencias ponderadas con el fin de ser utilizado mediante uno de los módulos de conversión analógica/digital del microcontrolador.

Finalmente y con base en los resultados de simulación para ambas leyes de control de posición PD y PD con compensación precalculada de gravedad, se concluye que, para robots cuyo modelo dinámico contiene explícitamente el vector de pares gravitacionales $g(q)$; la ley de control de posición PD con compensación precalculada de gravedad es mejor en términos de disminución del error comparada con la ley de control de posición PD.

A. Modelo de sólidos del robot pendular RP1

A continuación se presentan los planos de diseño del robot pendular RP1 (medidas en cm). El modelo de sólidos es construido en el software SolidWorks. En la figura 68 se aprecia una vista explotada del robot, indicándose las diferentes partes que lo integran.

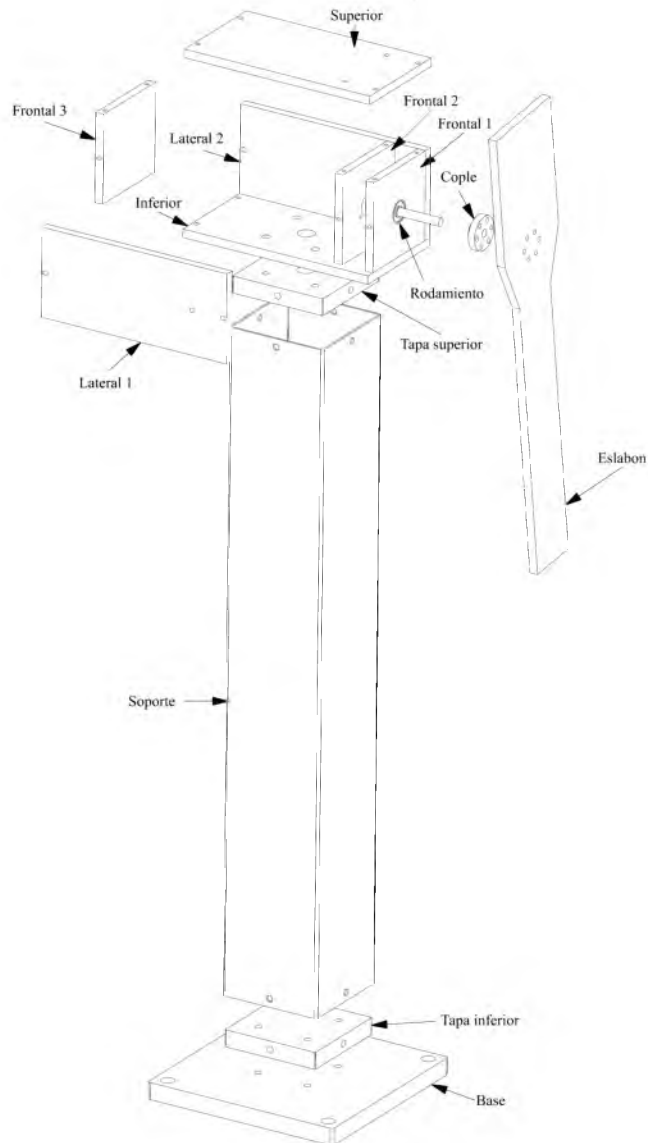


Figura 68: Vista explotada del robot pendular RP1.

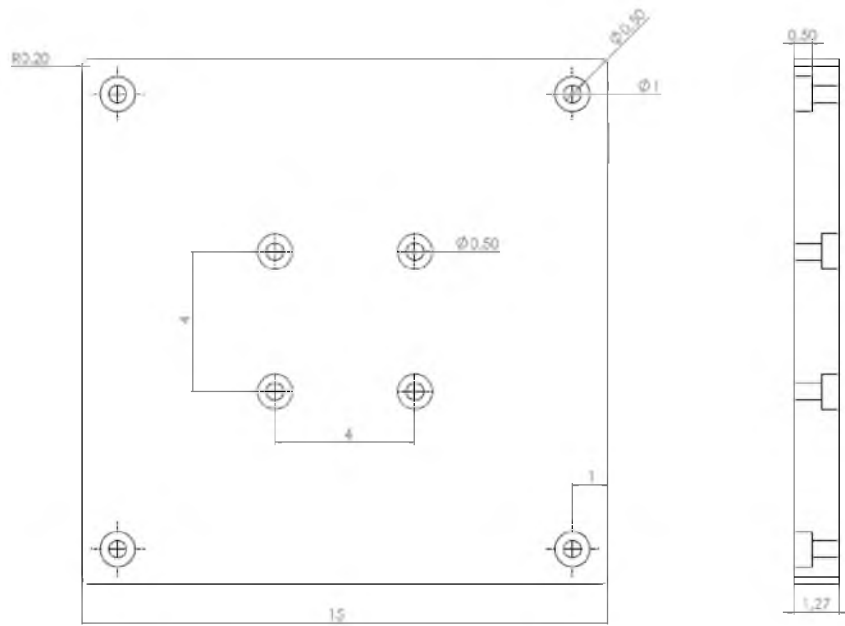


Figura 69: Plano de pieza Base.

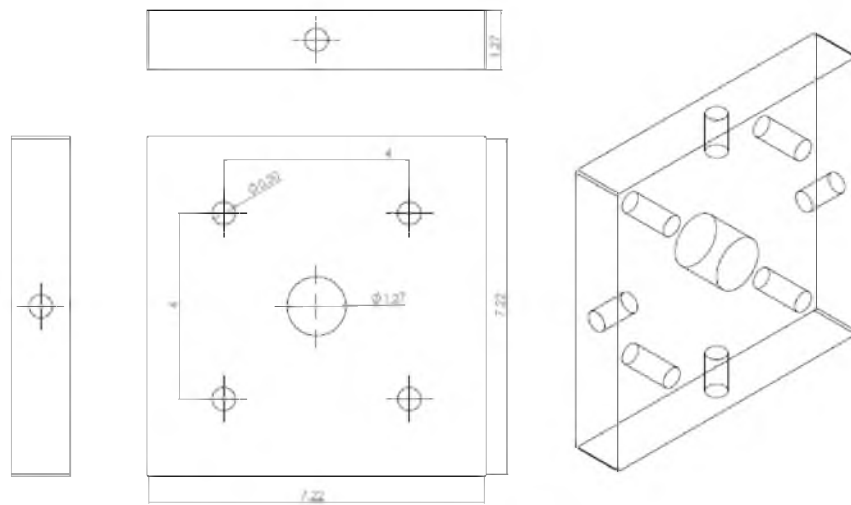


Figura 70: Plano de pieza Tapa Inferior.

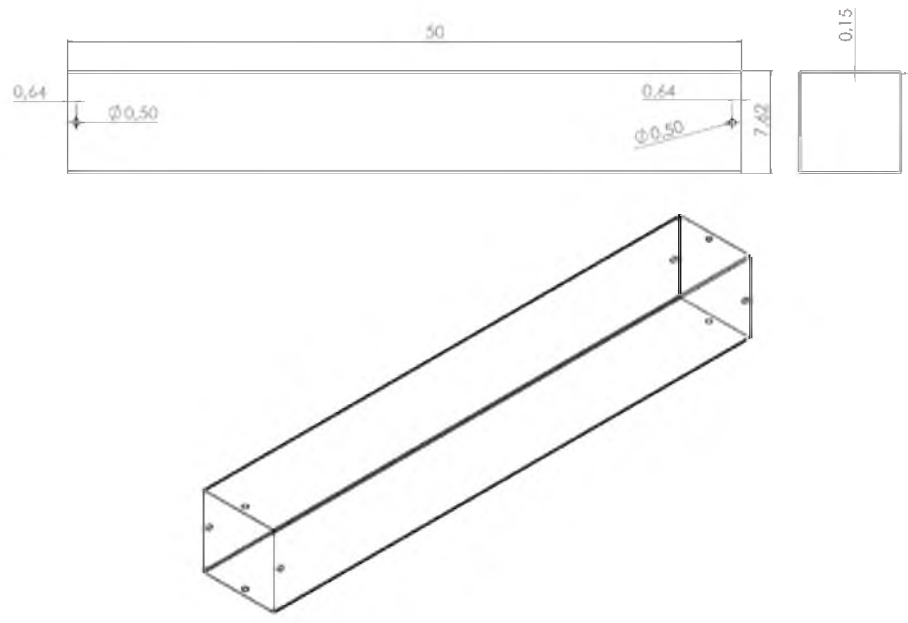


Figura 71: Plano de pieza Soporte.

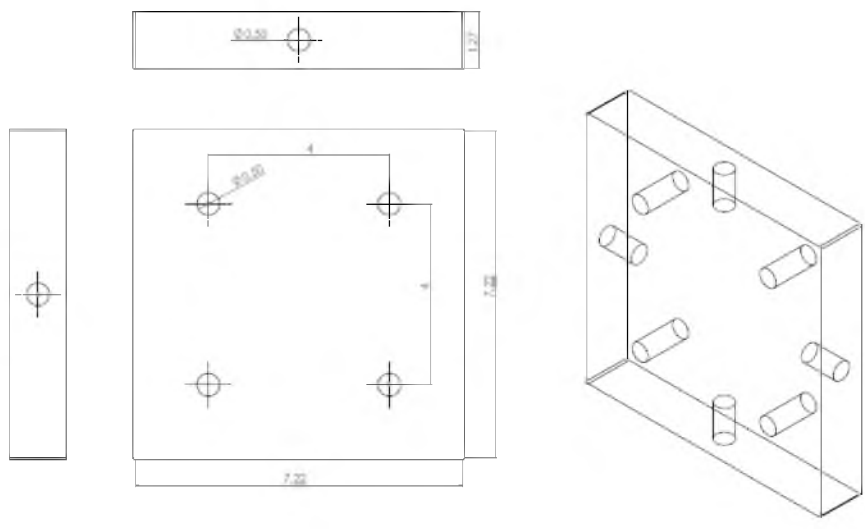


Figura 72: Plano de pieza Tapa Superior.

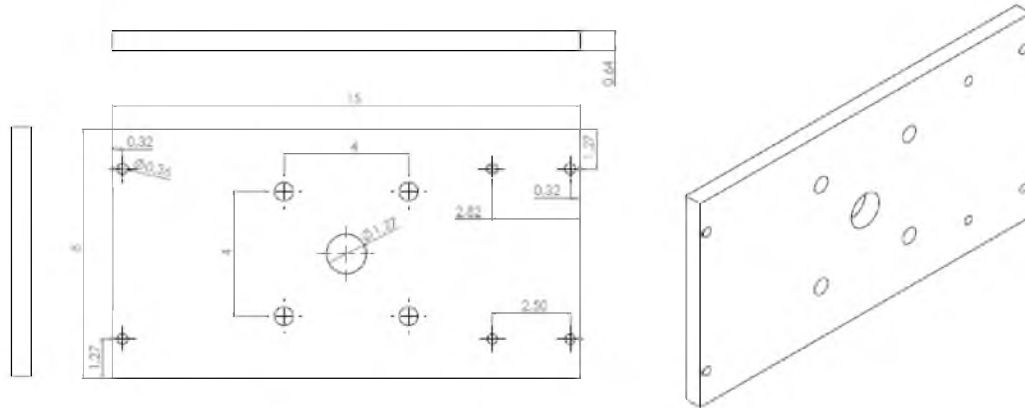


Figura 73: Plano de pieza Inferior.

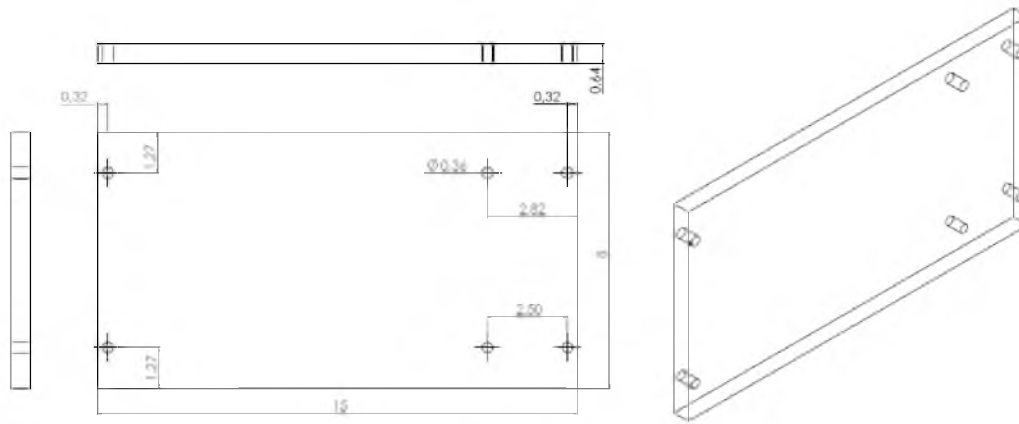


Figura 74: Plano de pieza Superior.

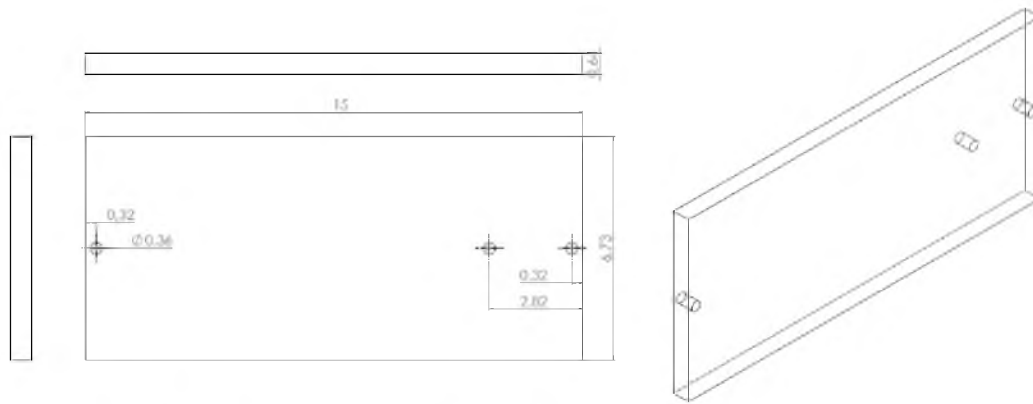


Figura 75: Plano de pieza Lateral 1 (equivalente a pieza Lateral 2).

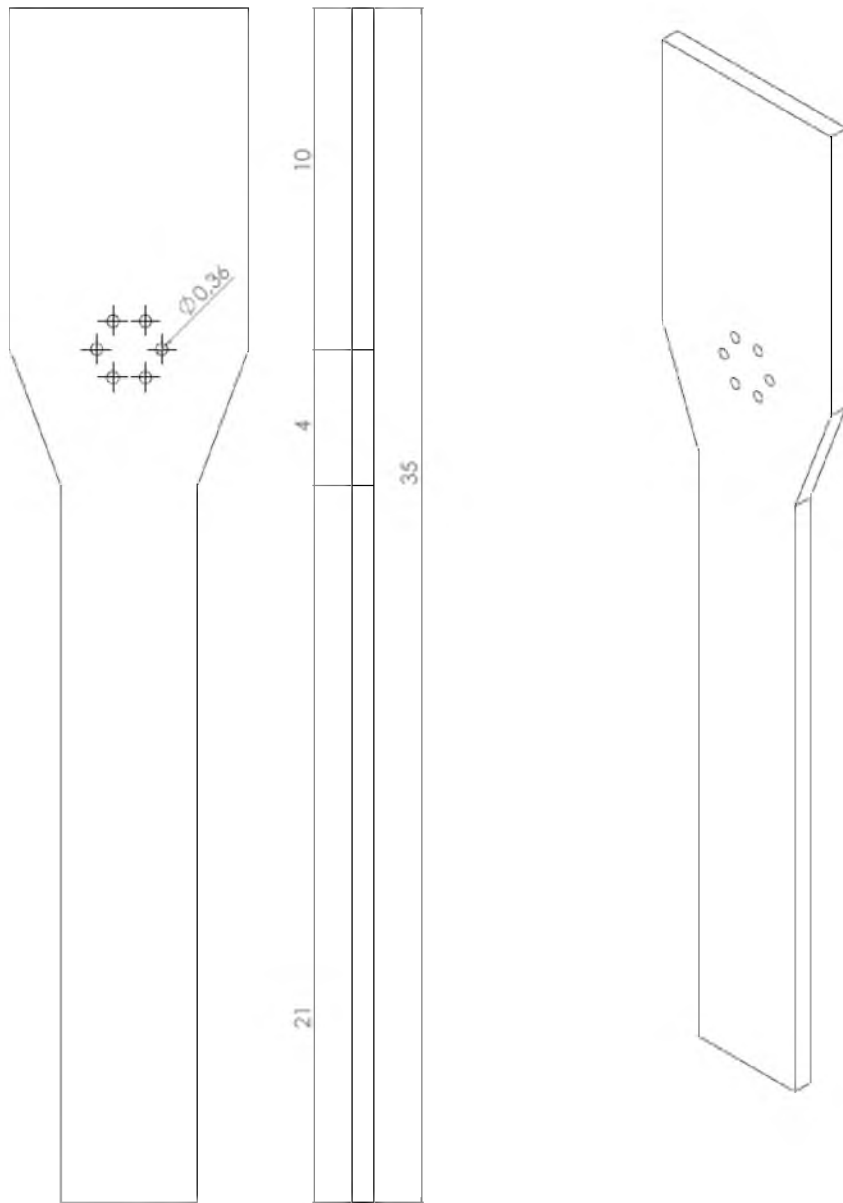


Figura 76: Plano de pieza Eslabón.

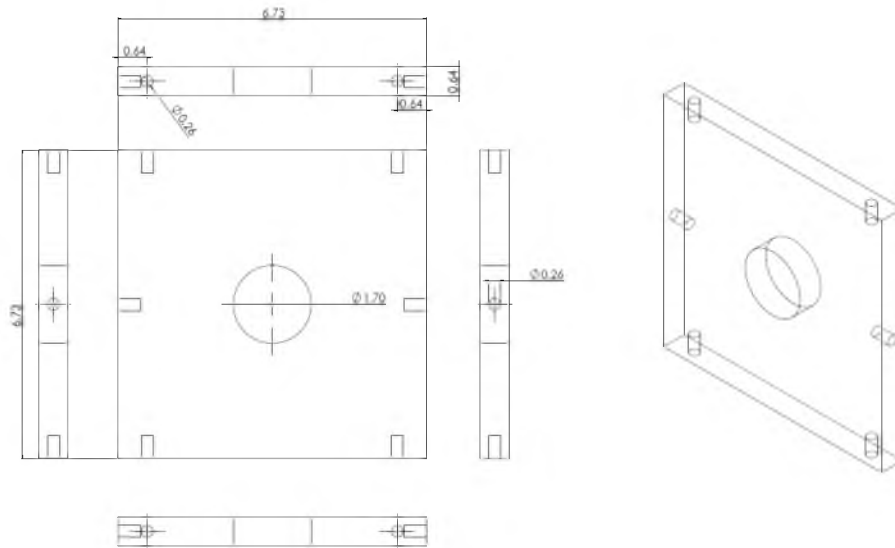


Figura 77: Plano de pieza Frontal 1.

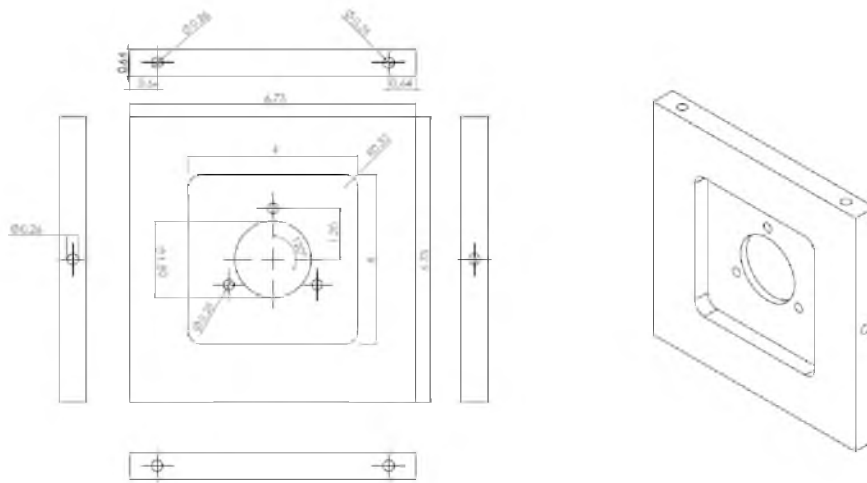


Figura 78: Plano de pieza Frontal 2.

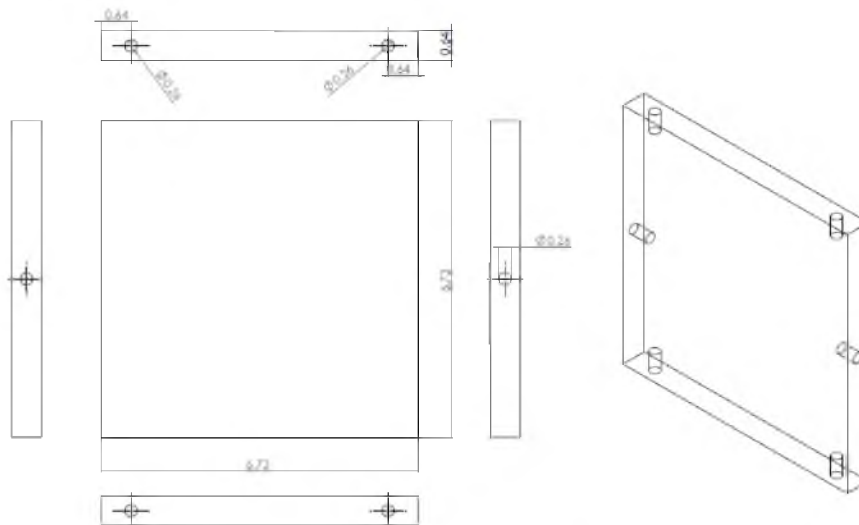


Figura 79: Plano de pieza Frontal 3.

B. Algoritmo de control en lenguaje C

Con base en el diagrama de flujo mostrado en la figura 67, a continuación; se presenta el código fuente en lenguaje C para el microcontrolador PIC18F2431 del sistema electrónico de control.

```

#include <18F2431.h> //Carga de libreria del PIC18F2431
#define adc=10      //Establece la resolucio del convertidor analogico
                    digital en 10 bits
#define FUSES NOWDT //Desactiva el Watch Dog Timer
#define FUSES WDT128 //Post escalador del Watch Dog Timer con relacion
                    1:128
#define FUSES XT //Se establece un cristal como reloj
#define FUSES NOBROWNOUT
#define FUSES NOLVP
#define delay(clock=4000000) //Frecuencia de reloj de 4Mhz
//Se establece la relacion de conexio entre los pines del LCD y el
PIC
#define LCD_ENABLE_PIN PIN_C2
#define LCD_RS_PIN PIN_C0
#define LCD_RW_PIN PIN_C1
#define LCD_DATA4 PIN_C3
#define LCD_DATA5 PIN_A0

```

```

#define LCD_DATA6      PIN_C4
#define LCD_DATA7      PIN_C5
//Cargado de librerias utilizadas
#include <lcd.c> //Libreria para el manejo del LCD
#include <stdlib.h>
#INCLUDE <math.h>
#use rs232(baud=9600, UART1)
//Establecimiento de la frecuencia de transmision de datos a 9600
    baudios
int tauDAC=0; // Variable de señal de control
//Interrupcion de modulo USART, cuando ocurre una recepcion de datos
    #INT_RDA
void RDA_isr(void)
{
    tauDAC=getc(); //Recepcion de la señal de control procesada
        por un dispositivo externo
    output_b(tauDAC); //Aplicacion de la señal recibida, por el
        puerto B del PIC
}
//Declaracion de cabecera de las funciones utilizadas
int configuracion();
float adquieredato();
int dac(float val);
//Declaracion de las variables utilizadas
long tipo_control=0;
float Kp=20, Kv=4, qd=90; //Variables de las ganancias de control
long sel_configuracion=0;
int hecho=0;
int hecho2=0;
int FLAG_interrup=0;
long tipo_control_sel=0;
long T_control=0;//Para modo de pruebas en proteus, debe T_control
    debe valer: 335 para PD o 325 para PDG
float posT0=0.0;
long posT0_l=0;
int alta=0;
int baja=0;
float posT1=0.0;
float vel=0.0;
//Funcion principal del programa
void main()
{

```

```

setup_adc_ports(sAN1); //Selección del canal analógico en A1
setup_adc(ADC_CLOCK_DIV_2|ADC_TAD_MUL_0|ADC_WHEN_INT0|
    ADC_INT_EVERY_OTHER); //Configuración interna del ADC
setup_timer_5(T5_DISABLED | T5_DIV_BY_1); //Configuración
    interna del Timer 5
setup_qei(QEI_MODE_X2_RESET_WITH_INDX); //Modulo QEI
    configurado en 2X y reestablecimiento por señal de index
set_tris_b(0); //Puerto B como salida
output_b(128); //Se carga 128 en el puerto B, ya que este
    valor representa 0 en la escala del DAC externo
lcd_init(); //Inicialización del LCD
set_adc_channel(1); //Activación del ADC en A1
float tau=0.0;
long ad=0;
int FLAG_salir=0;
//Impresión en en LCD de mensaje de inicio
lcd_gotoxy(7,1);
lcd_putc("UNPA");
lcd_gotoxy(1,2);
lcd_putc("Robot_Pendular_I");
delay_ms(2000); //Pausa de 2s
while(TRUE) //Ciclo infinito principal
{
    configuracion(); //Para simulación en proteus la
        función debe ser deshabilitada
    qei_set_count(100); //Se carga el valor de 100 en el
        contador del modulo QEI
    // Dependiendo del tipo de control seleccionado en la
        función configuracion(), se entra a solo una
        condicional
    if(T_control>= 333 && T_control<=339)
    {
        T_control=1;
        lcd_putc("\fPD._[0]H_[A]S");
    }
    if(T_control>= 322 && T_control<=328)
    {
        T_control=2;
        lcd_putc("\fPDG._[0]H_[A]S");
    }
    if(T_control>= 312 && T_control<=316)
    {

```

```

        T_control=3;
    }
    while(T_control==1) //Ciclo infinito de ejecucion del
        control PD
    {
        ad=read_adc(); //Lectura del ADC, en la
            espera de datos del teclado
        //Comienza el muestreo de la posicion angular
        posT0=qei_get_count(0); //Lectura del
            contador del QEI
        posT0=(posT0-100); //Resta efectuada para
            comenzar la precarga del contador
        delay_ms(250); //Tiempo de muestreo de 0.25s
        posT1=qei_get_count(0); //Lectura del
            contador del QEI
        posT1=(posT1-100); //Resta efectuada para
            comenzar la precarga del contador
        //Termina muestreo de la posicion angular
        //Calculo de la velocidad angular en RPM, ((
            Pulsos en 1s)/(resolucion del encoder por
            2))*60s                                vel
            =(((posT1-posT0)*4)/1536)*60;//Para
            simulacion en Proteus, el valor 1536 debe
            ser cambiado por: 1440
        //si se elige un motor con encoder de 180
            ppr, ya que QEI esta a modo 2X, por
            tanto 180*4=1440
        tau=(Kp*(qd-(posT1*0.25)))-(Kv*vel); //
            Calculo de la ley de control PD
        tau=(tau/0.1433); //Conversion par en voltaje
            mediante la constante Ks
        lcd_gotoxy(1,2);
        printf(lcd_putc, "P:%ft:%f_", (posT1*0.25),
            tau); //Impresion en LCD de la posicion
            angular y el torque
        //Saturacion de torque, 12 max, -12 min
        if(tau > 12)
        {
            tau=11.9;
        }
        if(tau < -12)
        {

```

```

        tau=-12;
    }
    tauDAC=dac(tau); //Adecuacion de la señal de
        control a un valor que pueda ser
        entregado al DAC externo mediante la
        funcion dac()
    output_b(tauDAC); //Aplicacion de la señal de
        control por el puerto B
    //Si se preciona la tecla A se suna 20 grados
        a qd
    if(ad>=322 && ad<=328)
    {
        qd=qd+20;
    }
    //Se se presiona la tecla B se resta 20
        grados a qd
    if((ad>=312 && ad<=316) && qd> 0.0)
    {
        qd=qd-20;
    }
    if(ad>540 && ad<546) //Si se preciona la
        tecla 0, se recorre el eslabon a la
        posicion 0
    {
        qd=0.0;
        FLAG_salir=1; //Activacion de la
            bandera para salir del ciclo
    }
    //Si se presiona la tecla D, se sale del
        ciclo de control
    if((ad>=333 && ad <=339) && FLAG_salir==1)
    {
        FLAG_salir=0; //Reestablecimiento de
            bandera
        T_control=0; //Para no cumplir
            condicion de ciclo y salir
        lcd_putc("\f"); //Limpia LCD
    }
}
while(T_control==2) //Ciclo infinito de ejecucion del
    control PD con compensacion precalculada de
    gravedad

```

{

```
ad=read_adc(); //Lectura del ADC, en la
    espera de datos del teclado
//Comienza el muestreo de la posicion angular
posT0=qei_get_count(0); //Lectura del
    contador del QEI
posT0=(posT0-100); //Resta efectuada para
    comenzar la precarga del contador
delay_ms(250); //Tiempo de muestreo de 0.25s
posT1=qei_get_count(0); //Lectura del
    contador del QEI
posT1=(posT1-100); //Resta efectuada para
    comenzar la precarga del contador
//Termina muestreo de la posicion angular
//Calculo de la velocidad angular en RPM, ((
    Pulsos en 1s)/(resolucion del encoder por
    2))*60s vel
=((posT1-posT0)*4)/1536*60;//Para
    simulacion en Proteus, el valor 1536 debe
    ser cambiado por: 1440
//si se elige un motor con encoder de 180
    ppr, ya que QEI esta a modo 2X, por
    tanto 180*4=1440
tau=(Kp*(qd-(posT1*0.25)))-(Kv*vel)
+(0.152729*sin(((qd*pi)/180))); //Calculo
    de la ley de control PD con compensacion
    precalculada de gravedad
tau=(tau/0.1433); //Conversion par-voltaje
    mediante la constante Ks
lcd_gotoxy(1,2);
printf(lcd_putc,"P:%ft:%f_", (posT1*0.25),
    tau); //Impresion en LCD de la posicion
    angular y el torque
//Saturacion de torque, 12 max, -12 min
if(tau > 12)
{
    tau=11.9;
}
if(tau < -12)
{
    tau=-12;
}
```

```

tauDAC=dac(tau); //Adecuacion de la señal de
                control a un valor que pueda ser
                entregado al DAC externo mediante la
                funcion dac()
output_b(tauDAC); //Aplicacion de la señal de
                control por el puerto B
//Si se presiona la tecla A se suma 20 grados
                a qd
if(ad>=322 && ad<=328)
{
    qd=qd+20;
}
//Se se presiona la tecla B se resta 20
                grados a qd
if((ad>=312 && ad<=316) && qd> 0.0)
{
    qd=qd-20;
}
if(ad>540 && ad<546) //Si se presiona la
                tecla 0, se recorre el eslabon a la
                posicion 0
{
    qd=0.0;
    FLAG_salir=1; //Activacion de la
                bandera para salir del ciclo
}
//Si se presiona la tecla D, se sale del
                ciclo de control
if((ad>=333 && ad <=339) && FLAG_salir==1)
{
    FLAG_salir=0; //Reestablecimiento de
                bandera
    T_control=0; //Para no cumplir
                condicion de ciclo y salir
    lcd_putc("\f"); //Limpia LCD
}
}
while(T_control==3) //Ciclo infinito de comunicacion
                entre el PIC y un dispositivo externo mediante
                USART
{

```

```

    if(FLAG_interrup==0) //Activacion de las
        interrupciones por USART y globales
    {
        enable_interrupts(INI_RDA);
        enable_interrupts(GLOBAL);
        FLAG_interrup=1;
    }
    ad=read_adc(); //Lectura del teclado
    //Se efectura lectura del contador de QEI
    posT0_l=qe_i_get_count(0);
    printf(lcd_putc, "\fTx:_%u", posT0_l); //
        Impresion de la posicion angular actual
        en LCD
    //Comienza envio de datos al dispositivo
        externo
    printf("[");
    alta=make8(posT0_l,1);
    putc(alta);
    printf(",");
    baja=make8(posT0_l,0);
    putc(baja);
    printf("]");
    //Termina envio de datos
    lcd_gotoxy(1,2);
    printf(lcd_putc, "Rx:%i_[A]S", tauDAC); //Se
        muestra en LCD la señal de control
        calculada por el dispositivo externo
    delay_ms(250); //Tiempo de muestreo
    if((ad>=333 && ad <=339)) //Presional tecla D
        para salir de ciclo
    {
        lcd_putc("\f");
        T_control=0;
        output_b(128);
    }
}
FLAG_interrup=0;
//Precarga de ganancias de control
Kp=20;
Kv=4;
qd=90;
tauDAC=0; //Restablecimiento de la señal de control

```

```

    }
}
//Funcion que permite convertir un numero flotante (señal de control)
, a un numero entero de 8 bits de entre 0 y 255, para ser
entregado al DAC externo. Establece un rango de valores con
amplitud de 0.09675, que va incrementandose partiendo ya sea
desde 0V a 12V (escala positiva) o de 0V a -12V (escala negativa)
, haciendo una comparacion de la señal de control con cada rango,
llevando siempre un contador de 0 a 128, que se incrementa o
decrementa con cada comparacion, cuando la señal de control cae
dentro del rango, se toma el valor mas cercano de los limites del
rango como nueva señal de control expresado en un numero entre 0
y 128 (contador).
int dac(float val)
{
    int DACCNT=0;
    float deltasup=0.09375; //Tamaño de paso que resulta de 12V
    /128
    float deltainf=0.0;
    float deltainf_b=0.0;
    float deltasup_b=-0.09375;
    if(val>=0) //Si señal de control es positiva o cero
    {
        DACCNT=128;
        while(DACCNT<=255)
        {
            if(val>=deltainf && val<=deltasup)
            {
                deltasup=deltasup-val;
                deltainf=val-deltainf;
                if(deltasup < deltainf)
                {
                    return DACCNT+1;
                }
                if(deltasup > deltainf)
                {
                    return DACCNT;
                }
            }
            DACCNT=DACCNT+1;
            deltasup=deltasup+0.09375;
            deltainf=deltainf+0.09375;
        }
    }
}

```

```

    }
}
if(val<0) //Si señal de control es negativa
{
    DACCNT=128;
    while(DACCNT>=0)
    {
        if(val<=deltainf_b && val>=deltasup_b)
        {
            deltasup_b=deltasup_b-val;
            deltainf_b=val-deltainf_b;
            if(deltasup_b > deltainf_b)
            {
                return DACCNT-1;
            }
            if(deltasup_b < deltainf_b)
            {
                return DACCNT;
            }
        }
        DACCNT=DACCNT-1;
        deltasup_b=deltasup_b -0.09375;
        deltainf_b=deltainf_b -0.09375;
    }
}
}
//Funcion que convierte las lecturas del ADC conectado al teclado, a
su correspondiente valor alfanumerico. Guarda los datos en un
vector de 5 celdas, la funcion retorna el equivalente como
flotante del vector de caracteres
float adquieredato()
{
    char dato[5];
    long adc_dato=0;
    int i=0;
    float suma=0;
    int FLAG_punto=0;
    lcd_gotoxy(1,2);
    lcd_putc("[A]DEL_[B]Hecho");
    lcd_gotoxy(5,1);
    do
    {

```

```

//Comienza la lectura del puerto analogico donde se
    conecta el teclado
adc_dato=read_adc();
delay_ms(200);
//Dependiendo del valor leido , se entra a cada rango
    correspondiente a cada letra del teclado
//Se muestra en LCD la tecla presionada , y se guarda
    el dato en el vector
if(adc_dato>=842 && adc_dato<=848 && i<=4)
{
    dato[i]='1';
    lcd_putc("1");
    i=i+1;
}
if(adc_dato>=634 && adc_dato<=640 && i<=4)
{
    dato[i]='2';
    lcd_putc("2");
    i=i+1;
}
if(adc_dato>=494 && adc_dato<=500 && i<=4)
{
    dato[i]='3';
    lcd_putc("3");
    i=i+1;
}
if(adc_dato>=758 && adc_dato<=791 && i<=4)
{
    dato[i]='4';
    lcd_putc("4");
    i=i+1;
}
if(adc_dato>=602 && adc_dato<=608 && i<=4)
{
    dato[i]='5';
    lcd_putc("5");
    i=i+1;
}
if(adc_dato>=474 && adc_dato<=480 && i<=4)
{
    dato[i]='6';
    lcd_putc("6");
}

```

```

        i=i+1;
    }
    if(adc_dato>=718 && adc_dato<=724 && i<=4)
    {
        dato[i]='7';
        lcd_putc("7");
        i=i+1;
    }
    if(adc_dato>=562 && adc_dato<=568 && i<=4)
    {
        dato[i]='8';
        lcd_putc("8");
        i=i+1;
    }
    if(adc_dato>=450 && adc_dato<=456 && i<=4)
    {
        dato[i]='9';
        lcd_putc("9");
        i=i+1;
    }
    if(adc_dato>=540 && adc_dato<=546 && i<=4)
    {
        dato[i]='0';
        lcd_putc("0");
        i=i+1;
    }
    //Condicional de borrado de dato erroneo
    if(adc_dato>=333 && adc_dato<=339)
    {
        i=0;
        suma=0;
        lcd_gotoxy(5,1);
        lcd_putc("_____");
        lcd_gotoxy(5,1);
    }
    if(adc_dato>=682 && adc_dato<=688 && FLAG_punto==0)
    {
        dato[i]='.';
        lcd_putc(".");
        FLAG_punto=1;
        i=i+1;
    }
}

```

```

        //Conversion y retorno del vector de caracteres como
        //una variable flotante equivalente
        if(adc_dato>=322 && adc_dato<=328)
        {
            suma=atof(dato);
            i=0;
            FLAG_punto=0;
            return suma;
        }
    }while(TRUE);
}
//Funcion utilizada para seleccionar el tipo de control, y configurar
//las ganancias Kp Kv y establecer el valor de qd
int configuracion()
{
    do //Primer ciclo de menu principal
    {
        lcd_gotoxy(1,1);
        lcd_putc("\f[A] Configurar");
        lcd_gotoxy(1,2);
        lcd_putc("[B] Comenzar");
        do //Ciclo de espera de tecla, la opcion del menu se
        //guarda en la variable tipo_control
        {
            delay_ms(200);
            tipo_control=read_adc();
            delay_ms(200);
        }while(tipo_control==0 || tipo_control>=400 ||
        tipo_control<=320);
        do//Ciclo de seleccion de opcion del menu
        {
            if(tipo_control>=322 && tipo_control<=328)//
            //comenzar
            {
                //hecho=tipo_control_sel;
                hecho=0;
                hecho2=0;
                return 0;
            }
            if(tipo_control>=333 && tipo_control<=339)//
            //configurar
            {

```

```

//Submenu de configuracion
lcd_putc("\f[A]PD_[B]PDG");
lcd_gotoxy(1,2);
lcd_putc("[C]PC_[D]Hecho");
do //Espera de seleccion del submenu
  de configuracion
{
    delay_ms(200);
    tipo_control_sel=read_adc();
    delay_ms(200);
}while(tipo_control_sel==0 ||
        tipo_control_sel>=400);
if( (tipo_control_sel>=305 &&
     tipo_control_sel<=310))//Si se
    preciona la opcion Hecho (Tecla D
    )
{
    hecho2=1;
    hecho=1;
}
if((tipo_control_sel>=312 &&
    tipo_control_sel<=316))//Si se
    preciona la opcion de PC (Tecla C
    )
{
    lcd_putc("\fModo_PC_ activo.");
    ;
    delay_ms(1000);
    T_control=tipo_control_sel;
}
while((hecho2 == 0) && (
    tipo_control_sel>316))//Si se
    preciona la opcion de PD o PGD (
    Tecla A o B)
{
    //Submenu de configuracion de
    parametros
    T_control=tipo_control_sel;
    lcd_putc("\f[A]Kp_[B]Kv");
    lcd_gotoxy(1,2);
    lcd_putc("[C]Ref_[D]Hecho");
}

```

```

do //Ciclo de espera de
    seleccoin del submenu
{
    delay_ms(200);
    sel_configuracion=
        read_adc();
    delay_ms(200);
}while(sel_configuracion==0
    || sel_configuracion >400)
;

        if(
sel_configuracion >=333 &&
sel_configuracion <=339)
//opcion tecla A
{
    lcd_gotoxy(1,1);
    lcd_putc("\fKp:");
    Kp=adquiere dato();
}
if(sel_configuracion >=322 &&
sel_configuracion <=328)//
opcion tecla B
{
    lcd_gotoxy(1,1);
    lcd_putc("\fKv:");
    Kv=adquiere dato();
}
if(sel_configuracion >=312 &&
sel_configuracion <=316)//
opcion tecla C
{
    lcd_gotoxy(12,1);
    lcd_putc("\fRef:");
    qd=adquiere dato();
}
if(sel_configuracion >=305 &&
sel_configuracion <=310)//
opcion tecla D
{
    hecho2=1;

```

```
        }
    }
    hecho2=0;
}
} while (hecho==0);
hecho=0;
hecho2=0;
} while (TRUE) ;
}
```

Referencias

- [1] Barrientos, A. Peñin, L. Balaguer, C. Aracil, R. (2007). *Fundamentos de Robótica*. Segunda Edición. España: McGraw-Hill.
- [2] Kelly, R. Santibáñez, V. (2003). *Control de movimiento de robots manipuladores*. España: Pearson Educación.
- [3] Spong, Mark W. Vidyasagar, M. (1989). *Robots dynamics and control*. New York: Wiley.
- [4] Craig, Jhon J.(2006). *Robótica*. Tercera Edición. México: Pearson Educación.
- [5] Tocci, Ronald J. Widmer, Neal S. (2003). *Sistemas digitales: principios y aplicaciones*. México: Pearson Educación.
- [6] Tomasi, Wayne. (2003). *Sistemas de comunicaciones electrónicas*. México: Pearson Educación.
- [7] Boylestad, Robert L. Nashelsky L. (1997). *Electrónica: Teoría de circuitos*. Sexta Edición. México: Pearson Educación.
- [8] Reyes, F. (2011). *Robótica: control de robots manipuladores*. España: Marcombo.
- [9] Reyes, F. (2012). *Matlab aplicado a robótica y mecatrónica*. España: Marcombo.
- [10] Angulo, José M. Angulo, I. Romero, J. (2006). *Microcontroladores pic 2ª parte: PIC16F87X: Diseño practico de aplicaciones*. España: McGraw-Hill.
- [11] Resnick, R. Halliday, D. Krane, Kenneth S. (2001). *Física Vol. 1*. Tercera Edición. México: Wiley.
- [12] Bellman, R y Cooke, KL. (1963). *Differential-Diference equations*. Primera Edición. New York: Academic press.
- [13] Gu, K. Kharitonov, VL y Chen, J. (1957). *Stability of time-delay systems*. Primera Edición. Boston: Birkhauser.

- [14] Kolmanovskii, VB y Nosov, VR. (1986). Stability of functional difference equations. Primera Edición. Londres: Academic press.
- [15] Krasovskii, NN y Brenner, JL. (1963). Stability motion. Primera Edición. Stanford: University press.