



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**DISEÑO Y CONSTRUCCIÓN DE UN
BRAZO ROBÓTICO PARA LA
EVALUACIÓN DE ALGORITMOS DE
CONTROL NO LINEALES**

TESIS

Que para obtener el título de

Ingeniero Eléctrico Electrónico

P R E S E N T A

Julio César Flores Díaz

DIRECTOR(A) DE TESIS

Dr. Iván Alejandro Gutiérrez Giles



Ciudad Universitaria, Cd. Mx., 2022

Agradecimientos

Para mis papas Álvaro Flores Cortes y Nieves Díaz Montero, por siempre estar ahí para apoyarme en cada momento, de manera incondicional, ser pacientes y enseñarme a ser una persona íntegra y a siempre luchar por mis ideales, sin importar los obstáculos, a ser agradecido con todo y a dar lo mejor de mí.

A mi hermana Mónica Anahí Flores Díaz, siendo mi ejemplo a seguir por el profundo respeto que tengo hacia ella como profesional y científica, compartiendo todos sus conocimientos conmigo, de igual manera agradezco a su familia por brindarme su compañía y su apoyo en todo momento.

A mi querida amiga Zaifen King Loeza por siempre estar ahí para apoyarme y brindarme su compañía en cada momento que estuvimos en la facultad y fuera de ella, por ser el mejor equipo cuando estuvimos como estudiantes en la facultad y actualmente, además por siempre hacerme reflexionar sobre mi vidas y así crecer como persona.

A mis dos grandes amigos Carlos Jesús León Fonseca y Pablo López Avendaño, por siempre estar ahí de manera incondicional para apoyarme, por darme la oportunidad de ser su amigo tantos años y por siempre ayudarme en los momentos difíciles.

A mi asesor de tesis Iván Alejandro Gutiérrez Giles, por apoyarme en el desarrollo de mi tesis, siendo paciente, atento y responsable en sus labores como tutor, además de proporcionarme sus conocimientos en distintas áreas para lograr cumplir mis metas.

A mi amigo y profesor Daniel Martínez Gutiérrez, por ser parte de mi formación

como profesional de la ingeniería y por permitirme formar parte de su grupo de amigos, le agradezco por compartir sus conocimientos y su lugar de trabajo, que fueron parte fundamental para mi desarrollo actual.

A la Universidad Nacional Autónoma de México y a la Facultad de Ingeniería por haberme otorgado la oportunidad de estudiar en esta gran institución, de formarme como profesional y de brindarme todas las oportunidades para lograr cumplir mis metas.

Finalmente agradezco al Programa de Becas Postdoctorales DGAPA–UNAM número IN117820, por las oportunidades que me brindo para desarrollar este proyecto.

Índice general

Índice de figuras	6
Índice de tablas	9
1. Introducción	10
1.1. Introducción	10
1.2. Objetivo, Planteamiento del Problema e Hipótesis	11
1.2.1. Objetivo	11
1.2.2. Planteamiento del Problema	11
1.2.3. Hipótesis	12
1.3. Antecedentes de los Manipuladores	12
1.3.1. Historia de la Robótica	12
1.3.2. Clasificación de los Robots	14
1.3.3. Robots Manipuladores Serie Rígidos	15
1.4. Aplicaciones	16
1.4.1. Ejemplos de Manipuladores	17
2. Diseño y Construcción del Brazo Robótico	20
2.1. Herramientas y Software	20
2.2. Estructura del Manipulador	23
2.2.1. Materiales y Selección de Elementos	23
2.2.2. Plano Estructural	25
2.2.3. Costos de Fabricación	27
2.3. Eléctrica y Electrónica del Manipulador	28
2.3.1. Materiales y Selección de Elementos	28
Etapa de Potencia	28
Puente H con relés	37
Puente H con Transistor MOSFET.	37
Puente H con Circuito Integrado.	38
Fuente de Suministro Eléctrico, Convertidor CA-CD	42
2.3.2. Esquemático Electrónico y Diagrama de Bloques Eléctrico	44
2.3.3. Costos de Fabricación	46
2.4. Módulo de Control y Adquisición	48

2.4.1.	Materiales y Selección de Elementos	48
	Etapa de Adquisición y Generación	48
	Etapa de Comunicación y Transmisión de Datos	56
	Tarjetas Comerciales Dedicadas a la Adquisición y Generación de señales	63
2.4.2.	Esquemático Módulo de Control y Adquisición, Diagrama de Bloques	
	de Comunicación	64
2.4.3.	Costos de Fabricación	66
2.5.	Ensamblaje y Conexiones Eléctricas del Manipulador	67
	2.5.1. Montaje de Articulaciones y Eslabones.	67
	2.5.2. Conexiones y Cableados	71
2.6.	Programación, Calibración y Acondicionamiento	72
	2.6.1. Software de Adquisición y Control	72
	2.6.2. Acondicionamiento y Programación de Algoritmos de Control	73
	2.6.3. Interfaz de Control	78
	2.6.4. Calibración y Test de la Plataforma	79
2.7.	Espacio de Trabajo del Manipulador	83
2.8.	Costos Generales	84
3.	Modelado Matemático y Planeación de Trayectorias	85
3.1.	Modelo Matemático de Manipuladores Robóticos	85
3.2.	Cinemática Directa	86
	3.2.1. Denavit-Hartenberg	87
3.3.	Cinemática Inversa	90
	3.3.1. Método Geométrico	91
3.4.	Jacobiano del Manipulador	93
	3.4.1. Jacobiano Geométrico	94
3.5.	Dinámica del Manipulador	96
3.6.	Planeación y Generación de Trayectoria tanto en Coordenadas Articulares	
	como en Cartesianas	106
	3.6.1. Trayectoria Articular	106
	3.6.2. Trayectoria Cartesiana	107
4.	Algoritmos de Control Lineal, No Lineal y Determinación de Parámetros.	109
4.1.	Control Lineal	109
	4.1.1. Control PID	109
4.2.	Algoritmo de Determinación de Parámetros, Regresor Lineal	110
4.3.	Control No Lineal	112
	4.3.1. Control PD con Compensación de Gravedad	112
	4.3.2. Control Par Calculado	112
	4.3.3. Control Basado en Pasividad	114
	4.3.4. Control Adaptable	115
	4.3.5. Control por Modos Deslizantes	116

5. Resultados Experimentales	118
5.1. Algoritmo de Determinación de Parámetros	118
5.1.1. Experimento para Determinar los Parámetros del Modelo Completo	118
5.1.2. Cálculo de Parámetros con Valores Conocidos	119
5.1.3. Comparación entre Parámetros	120
5.2. Control Regulación	121
5.3. Control de Seguimiento de Trayectoria Articular	125
5.4. Control de Seguimiento de Trayectoria Cartesiana	129
5.5. Caracterización de la Posición del Efector Final	133
6. Conclusión	136
6.1. Construcción del Manipulador	136
6.2. Comparación de Algoritmos	136
6.3. Áreas de Oportunidad	137
A. Código de Microcontroladores PIC	138
B. Código de Visual Studio	144
Bibliografía	159

Índice de figuras

1.1. Clasificación de los Robots	14
1.2. Articulaciones a)Prismática b)Revolución	15
1.3. Robot Cartesiano	17
1.4. Robot Cilíndrico	17
1.5. Robot Esférico	18
1.6. Robot SCARA	18
1.7. Robot Antropomórfico	19
1.8. Muñeca Esférica	19
2.1. Robot Planar de Dos Grados de Libertad	23
2.2. Estructura del Primer Eslabón	25
2.3. Estructura del Segundo Eslabón	26
2.4. Estructura para la Primera Articulación	26
2.5. Estructura para Montar la Segunda Articulación	26
2.6. Eslabón 2	29
2.7. Eslabón 2	30
2.8. Eslabón 1	32
2.9. Representación en el Plano del Motor para la Articulación 2	33
2.10. Diagrama de Fuerzas	34
2.11. Diagrama de Cuerpo Libre	34
2.12. Puente H con Relé	38
2.13. Puente H con Transistor MOSFET Ejemplo	38
2.14. Circuito de Protección	40
2.15. Suministro Eléctrico por Niveles	45
2.16. Esquemático del Módulo de Potencia (Módulo L298N)	46
2.17. CAD Módulo de Potencia (Módulo L298N)	46
2.18. Señales de un Encoder de Cuadratura	50
2.19. Diagrama de Cuentas	50
2.20. Lectura de Encoders (Función Principal)	52
2.21. Lectura de Encoders (Función de Interrupción)	53
2.22. Señal PWM	54
2.23. Generación PWM (Función Principal)	55
2.24. Diagrama de Conexiones SPI	57

2.25. Diagrama de Tiempo de la Comunicación SPI	57
2.26. Placa JT45k50-v1.0	58
2.27. Diagrama de Flujo SPI Función Principal Esclavo	59
2.28. Diagrama de Flujo SPI Función Interrupción Esclavo	60
2.29. Diagrama de Flujo SPI Función Principal Maestro	61
2.30. Diagrama de Flujo SPI Función USB Maestro	62
2.31. Diagrama de Comunicación	64
2.32. Esquemático Módulo de Control y Adquisición	65
2.33. Esquemático Conector con Placa JT45k50-v1.0	65
2.34. CAD Conector con Placa JT45k50-v1.0	66
2.35. Estructura Completa	67
2.36. Articulación 1	68
2.37. Articulación 2 y Eslabón 1	69
2.38. Articulación 2 y Eslabón 2	70
2.39. Diagrama de Conexiones y Cableado	71
2.40. Diagrama de Flujo Adquisición y Control Visual Studio	73
2.41. Diagrama de Flujo, Acondicionamiento de Señales	75
2.42. Diagrama de Flujo, Función de Control 1	76
2.43. Diagrama de Flujo, Función de Control 2	77
2.44. Diagrama de Flujo, Función de Control 3	78
2.45. Sección de Comunicación y Prueba de la Interfaz Gráfica de Usuario	79
2.46. Sección de Elección de Algoritmo de Control	79
2.47. Sección de Prueba de Encoders y de Motores	80
2.48. Posición de Referencia Para Calibración	81
2.49. Sistema de Adquisición de Señales, Control y Comunicación	81
2.50. Manipulador robótico de dos grados de libertad	82
2.51. Espacio de Trabajo del Manipulador Robótico	83
3.1. Robot Planar de 2 Grados de Libertad	85
3.2. Cinemática Directa	89
3.3. Referencia para el Análisis de la Cinemática Inversa	91
3.4. Cinemática Inversa	93
3.5. Centro de Masa Eslabones	101
3.6. Barra Plana	101
5.1. Regulación q_1 y q_2 : Deseada (--), PID (—), PD+G (—), Par Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).	122
5.2. Error de Posición eq_1 y eq_2 : PID (—), PD+G (—), Par Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).	123
5.3. Par τ_1 y τ_2 : PID (—), PD+G (—), Par Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).	124
5.4. Parámetros: π_1 (—), π_2 (—), π_3 (—), π_4 (—), π_5 (—), π_6 (—), π_7 (—), π_8 (--), π_9 (--).	125
5.5. Trayectoria de q_1 y q_2 : Deseada (--), PID (—), PD+G (—), Par Calculado Identificado (—), Par Calculado Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).	126

5.6. Error de Posición eq_1 y eq_2 : PID (—), PD+G (—), Par Calculado Identificado (—), Par Calculado Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).	127
5.7. Par τ_1 y τ_2 : PID (—), PD+G (—), Par Calculado Identificado(—), Par Calculado Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).	128
5.8. Parámetros: π_1 (—), π_2 (—), π_3 (—), π_4 (—), π_5 (—), π_6 (—), π_7 (—), π_8 (—), π_9 (—).	129
5.9. Trayectoria: Deseada (—), PID (—), PD+G (—), Par Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).	130
5.10. Error de Posición ex_1 y ex_2 : PID (—), PD+G (—), Par Calculado(—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).	131
5.11. Par τ_1 y τ_2 : PID (—), PD+G (—), Par Calculado(—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).	132
5.12. Parámetros: π_1 (—), π_2 (—), π_3 (—), π_4 (—), π_5 (—), π_6 (—), π_7 (—), π_8 (—), π_9 (—).	133
5.13. Posición del Efecto Final Control de Regulación	134
5.14. Posición del Efecto Final Control de Seguimiento	135

Índice de tablas

2.1. Programas utilizados	21
2.2. Herramientas de trabajo y hardware	22
2.3. Propuesta de materiales para la estructura	24
2.4. Tipo de acoplamiento para motor	24
2.5. Costos de la estructura del manipulador	27
2.6. Motores propuestos para la articulación 2	31
2.7. Sistema de engranajes propuestos para la articulación 2	31
2.8. Encoder propuestos para la articulación 2	32
2.9. Motores propuestos para la articulación 1	36
2.10. Propuestas para el sistema de engranajes de la articulación 1	36
2.11. Encoder propuestos para la articulación 1	36
2.12. Opciones de puente H en circuito integrados	39
2.13. Modelos de optoacopladores	41
2.14. Modelo de reguladores	42
2.15. Corriente de consumo	43
2.16. Fuente de voltaje	44
2.17. Costos del módulo de potencia	47
2.18. Motores	47
2.19. Microcontroladores	49
2.20. Tarjetas de adquisición y generación de señales	63
2.21. Costos módulo de control, adquisición y comunicación	66
2.22. Datos nominales	74
2.23. Balance de costos generales del prototipo	84
3.1. Tabla de parámetros DH	90
5.1. Valores nominales	119
5.2. Tabala de parámetros identificados y calculados	121
5.3. Posición del efector final control por regulación	134
5.4. Posición del efector final control por seguimiento	135

Capítulo 1

Introducción

1.1. Introducción

En este trabajo se describe a grandes rasgos el proceso que se requiere para desarrollar una plataforma basada en un manipulador robótico de dos grados de libertad, de bajo costo, funcional y que permita la implementación de técnicas de control, de algoritmos ya conocidos en la literatura o de nuevo desarrollo. Se eligió el manipulador de dos grados de libertad planar ya que presenta un comportamiento no lineal, siendo ideal para la implementación de algoritmos lineales y no lineales, con el objetivo de realizar una comparativa entre el desempeño de cada uno de ellos y de esta manera validar la funcionalidad de la plataforma. Este trabajo se divide en seis capítulos: tres de ellos comprenden temas relacionados con la teoría de la robótica y el control, los tres restantes describen el proceso de desarrollo, diseño y experimentación de la plataforma. En el Capítulo 1 se desarrolla la introducción, se plantea el problema a resolver, se definen los objetivos y se proponen los posibles resultados mediante una hipótesis bien definida más una descripción general de lo que es la robótica, donde se ubica y un resumen del desarrollo histórico del mismo. En el Capítulo 2 se define todo el desarrollo de la plataforma desde su construcción, del sistema de adquisición, control y comunicación a la programación de la plataforma y de la interfaz que permite ejecutar las técnicas de control junto con la síntesis de los costos que conlleva su construcción. En el Capítulo 3 se desarrolla de manera teórica el modelo matemático del manipulador, considerando su cinemática y dinámica, junto con el diseño de la trayectoria propuesta para los experimentos propuestos. En el Capítulo 4 se describen las seis técnicas de control utilizadas en los experimentos que se realizaron para verificar el funcionamiento

de la plataforma más un algoritmo de determinación de parámetros. En el Capítulo 5 se reportan los resultados obtenidos de la implementación de los algoritmos propuestos en el Capítulo 4, para experimentos de regulación y seguimiento. Finalmente en el Capítulo 6 se expone la conclusión de los resultados obtenidos de todo el desarrollo de la plataforma principalmente del diseño y construcción del manipulador y de los experimentos realizados, ya que estos demuestran el cumplimiento de los objetivos propuestos y la validación a la proyección de los posibles resultados a obtener.

1.2. Objetivo, Planteamiento del Problema e Hipótesis

1.2.1. Objetivo

Objetivo General.

- Diseñar y construir una plataforma, basada en un manipulador, con la finalidad de implementar algoritmos de control, para hacer experimentos tanto para investigación como para la docencia.

Objetivo Particular.

- Construir un brazo robótico de bajo costo, modular y flexible que permita evaluar y mostrar los efectos de aplicar leyes de control.
- Implementar técnicas de control, a un manipulador, para comparar el desempeño de cada uno de ellos

1.2.2. Planteamiento del Problema

Actualmente los equipos de adquisición y generación de señales digitales para aplicaciones en varios ámbitos, como lo es la robótica, tienen un precio muy elevado o no permiten la modificación de las técnicas de control ya programadas, por lo que es complicado o poco flexible para la enseñanza o investigación el verificar o comparar el efecto de controladores lineales y no lineales sobre un modelo no lineal, como lo es el manipulador robotico de dos grados de libertad.

Los controladores lineales muestran resultados satisfactorios para sistemas lineales, sin embargo, dicho controlador no toma en cuenta los efectos no lineales de la dinámica

del los manipuladores robóticos, por lo que es necesario evaluar algoritmos de control que tomen en cuenta estas consideraciones, es decir, implementar algoritmos de control no lineal y verificar los resultados aplicados al manipulador.

1.2.3. Hipótesis

La plataforma permitirá implementar algoritmos de control, tanto lineal como no lineal, de los cuales se obtienen datos cuantitativos que ayudan a evaluar el desempeño de cada uno de ellos y con la misma información hacer identificación paramétrica de modelos.

La plataforma del manipulador será capaz de expresar algoritmo de control no lineales, utilizando algoritmos automáticos, en comparación con los controles lineales bien sintonizados, siendo éste un prototipo con un sistema de adquisición de señales digitales y un interfaz de control de bajo costo, modular y flexible para el usuario.

1.3. Antecedentes de los Manipuladores

1.3.1. Historia de la Robótica

En la década de 1920 a nuestra mesa todavía llegaban alimentos sin ser procesados o modificados de alguna manera. Sin embargo, la revolución industrial trajo consigo no solo alimentos procesados, enlatados y empaquetados a nuestra mesa, además nos presentó una obra teatral de ciencia ficción en donde se nos definió un concepto y se agregó un nuevo término a nuestro vocabulario. Esa palabra que hoy en día es muy fácil de reconocer e interpretar es la palabra robot. Karel Čapek la definió en su obra “Rossum’s Universal Robots” [Spong et al., 2006], siendo esta una palabra que se deriva del vocablo checo robota, que significa trabajo. Desde este suceso dicha palabra se ha usado para nombrar a una diversidad de dispositivos mecánicos y personajes de ciencia ficción. Los robots como se figuraban en sus inicios, estaban muy alejados de ese esquema de androide con características humanoides. Esto era debido a todas las limitaciones y una gran cantidad de problemas que se presentaban en la época. Aún hoy en día esa idea de robot no se ha logrado construir, aunque existen una gran gama de ejemplos en donde la evolución de la robótica cada día nos deja más asombrados. Pero aún así muchos de los robots convencionales usados en distintas áreas, siguen siendo útiles para las actividades humanas. Por ejemplo, tenemos a los manipuladores robóticos, que no son otra cosa más que la representación de un brazo

o mano humana llevada a un diseño electrónico y mecánico. George Devol, quien fue el pionero en la robótica industrial, patentó el manipulador programable en 1948, en el mismo año Argonne National Laboratory, desarrolló el primer robot tele manipulador basado en un módulo mecánico con la característica de funcionamiento maestro-esclavo, utilizado para la manipulación de desechos radioactivos. En el año de 1954 Goertz usó la tecnología eléctrica electrónica y el servocontrol para sustituir los sistemas mecánicos del robot telemanipulador, siendo el primero en su clase.

Handy Man, es un robot que consiste en dos brazos tele operados mediante el uso de un exoesqueleto creado por General Electric en 1958. Posterior a los inicios de la robótica un selecto grupo se interesó por el uso de estos dispositivos para realizar una variedad de tareas que les fueran útiles, desde la industria comercial hasta la militar y aeroespacial. Pero aún con las contastes evoluciones que ha presentado la robótica siguen conservando esos rasgos que los caracterizan desde su origen. George Devol y Joseph F. Engelberger fundaron la Consolidate Controls Corporation [Antonio B., 2007], llamada después Universal Automation. En el año de 1960 instalaron su primera máquina en la empresa General Motors, lo que fue considerado en su momento la máxima expresión en la forma de fabricar para el futuro. Otros que también incursionaron en este ámbito fueron la empresa AMF, Kawasaki, ASEA quien construyó el primer robot totalmente electrónico, entre otras cosas. Como consecuencia Japón superó a Estados Unidos en el ámbito de la robótica, gracias a Nissan, dando lugar a la creación de la Asociación de Robótica Industrial de Japón (JIRA [Antonio B., 2007]) en el año de 1972. Dos años después, se fundó la Asociación de Industrias Robóticas y en los 80s se fundó la Federación Internacional de Robótica.

Cuando hablamos de la robótica de los manipuladores hacemos referencia a toda esa gama de opciones que se presentan en diferentes configuraciones que son parte importante del desarrollo histórico de la robótica. Por ejemplo, los primeros manipuladores, que están definidos en función del movimiento de sus articulaciones, se conocen como esférico y antropomórfico. Otro ejemplo importante de ellos es el robot SCARA, que consta de 3 a 4 grados de libertad [Antonio B., 2007].

Ya que conocemos un poco la perspectiva histórica de los manipuladores que resulta ser muy amplia y que conlleva una gran cantidad de desarrollos en el área, nos centraremos en hablar un poco de los avances que surgieron para la robótica industrial, ya que es donde podemos introducir la aplicación, que no es única, de los manipuladores robóticos convencionales. Parte fundamental que se debe de considerar cuando se trabaja con los manipuladores

es que no solo se trata de una estructura física, si no que conlleva también a tratar con el “ente” que nos permite darle “vida” al robot e integrarle la capacidad de realizar tareas sin nuestra intervención. Para ello recurrimos al uso de computadoras y electrónica en general. Gracias al desarrollo de la electrónica fuimos capaces de utilizar microcontroladores para integrar sus funciones de manera automática e implementar las técnicas desarrolladas con la matemática adecuada a modelos físicos y poder con ello revolucionar el desarrollo industrial de la robótica de manipuladores. Hoy en día, por ejemplo, muchas de las líneas de producción requieren el uso de robots manipuladores para realizar las tareas que a una persona le resulta difícil o imposible llevar a cabo.

1.3.2. Clasificación de los Robots

Una de las clasificaciones que podemos definir es la que se muestra en la Figura 1.1, que está en función de las características físicas que lo componen, el tipo de movimiento y su utilidad.

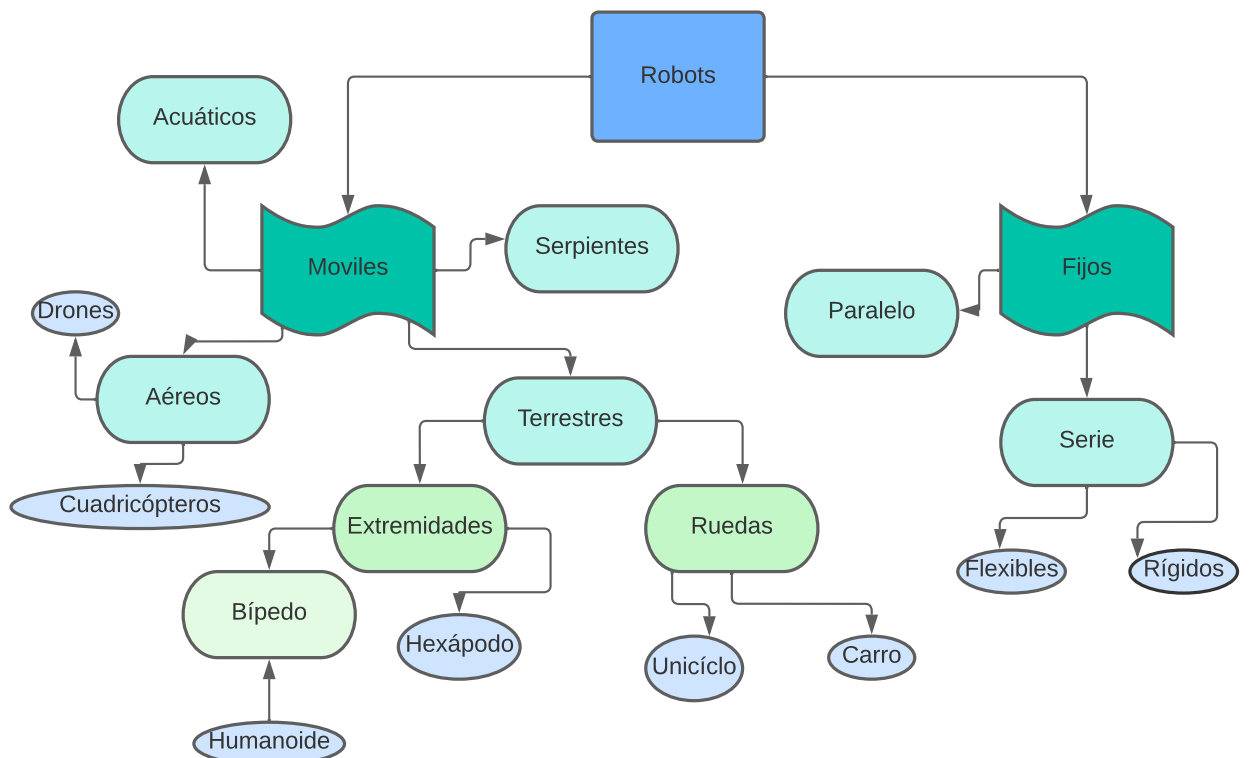


Figura 1.1: Clasificación de los Robots

Esta clasificación sirve de ejemplo para mostrar dónde se desglosa el área de interés, la cual se posiciona como un robot fijo rígido tipo serie.

1.3.3. Robots Manipuladores Serie Rígidos

Primero abordaremos la definición de robot manipulador, “es un manipulador multifuncional reprogramable diseñado para mover materiales, piezas, herramientas o dispositivos especializados a través de movimientos programados variables para la realización de una variedad de tareas” (Robot Institute of America, 1979).

Los manipuladores de tipo serie se componen de n número de eslabones y $n+1$ articulaciones, de lo cual se sabe que los eslabones se enumeran del cero al n -ésimo valor, donde el eslabón cero se considera a la base y el n -ésimo el efector final, mientras que las articulaciones se enumeran del 1 al n -ésimo valor.

De estos manipuladores se conocen dos tipos de articulaciones, las de movimiento lineal y las de rotación, que son mejor conocidas como prismática y de revolución, como se muestra en la Figura 1.2. Ese movimiento lineal para las prismáticas se define como la variable articular d_i , en cambio el movimiento angular para la de revolución se conoce como la variable articular θ_i .

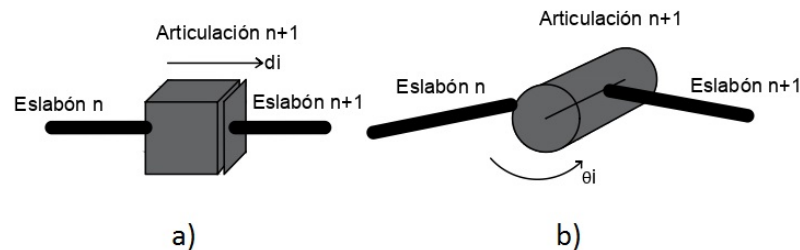


Figura 1.2: Articulaciones a)Prismática b)Revolución

Las articulaciones pueden estar constituidas de actuadores o sistemas de actuación, de tipo eléctrico, hidráulico o neumático. Todo dependerá de las condiciones de operación a las que sea sometidas, por ejemplo, si se requiere de manipular objetos realmente pesados la mejor opción es elegir los actuadores hidráulicos, pero si la aplicación no es muy demandante de fuerza con los actuadores eléctricos, como los motores, es más que suficiente.

El dispositivo final con el que cuentan los manipuladores, el efector final, no es más que la herramienta con la que va a trabajar el manipulador, ya sea una pinza, un imán, un

taladro u otro objeto que permita realizar alguna tarea específica.

Sabemos que los manipuladores tienen que realizar alguna tarea asignada, en un entorno determinado por el tipo de articulaciones que lo conforman, para lo cual contamos con tres conceptos diferentes que más adelante profundizaremos en su utilidad. El primer concepto es el espacio articular que hace referencia al valor que pueden tomar las variables articulares y se representa como un vector de dimensión n como número de articulaciones que tiene el robot. Otro concepto es el espacio de trabajo que hace referencia a los puntos posibles que el efector final del robot puede alcanzar. Por último tenemos el espacio de destreza, que son todos los puntos que puede alcanzar el efector final del robot considerando cualquier orientación deseada y sin singularidades.

1.4. Aplicaciones

Los robots actualmente tienen un impacto significativo en muchas áreas, una de ellas es en el sector de la industria, siendo una de las herramientas más indispensables y que ha innovado con el paso del tiempo. Para dar muestra de ello describiremos las distintas aplicaciones que se le dan a los robots, desde varias perspectivas, entre ellas el área industrial y de servicio. Las aplicaciones de los robots se pueden clasificar de la siguiente manera:

- **Robots Industriales:** soldadura, procesado, corte, montaje y control de calidad.
- **Robots de Servicio:** industria nuclear, agricultura, medicina, construcción y educativos.

Por ejemplo, los robots de soldadura, montaje y corte los podemos encontrar en las ensambladoras automotrices. Los de control de calidad y proceso en alguna línea de manufactura. En cuanto al resto, como la de medicina y agricultura, los podemos encontrar en sus respectivas áreas. Si se desea profundizar más en el tema de aplicaciones se puede consultar [Antonio B., 2007].

En la teoría y práctica de la robótica de los manipuladores, encontramos algunas configuraciones que se toman como referencias para los robots que más se utilizan en las distintas áreas, la industria, la investigación y el desarrollo tecnológico, por mencionar algunas.

1.4.1. Ejemplos de Manipuladores

A continuación se describen algunas configuraciones básicas que podemos encontrar en los manipuladores robóticos.

Robot Cartesiano. Este robot cuenta con tres articulaciones prismáticas, y un ejemplo de aplicación son las impresoras 3D, como se muestra en la Figura 1.3.

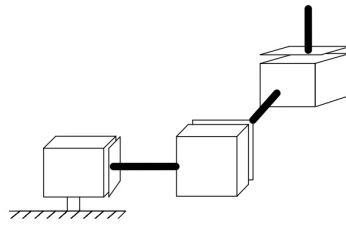


Figura 1.3: Robot Cartesiano

Robot Cilíndrico. El robot cilíndrico cuenta con una articulación de revolución en la base, que permite el movimiento sobre su propio eje, y dos articulaciones prismáticas consecutivas, como se muestra en la Figura 1.4.

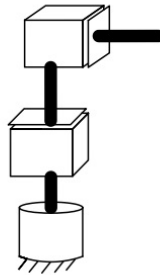


Figura 1.4: Robot Cilíndrico

Robot Esférico. Este robot está constituido por 2 articulaciones de revolución la primera permite la rotación sobre su propio eje y la segunda el desplazamiento de arriba hacia abajo el efector final. Por último, cuenta con una articulación prismática que le da la capacidad de un mayor alcance gracias a su desplazamiento lineal, se puede observar la configuración en la Figura 1.5.

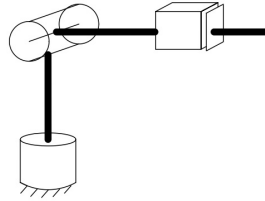


Figura 1.5: Robot Esférico

Robot SCARA. Este robot es similar al robot Esférico, la diferencia radica en que la segunda articulación esta orientada de la misma manera que la primera, provocando la rotación del efector final y también cuenta con una articulación del tipo prismático que le da un mayor alcance en dirección vertical, hacia la base del robot, como se indica en la Figura 1.6

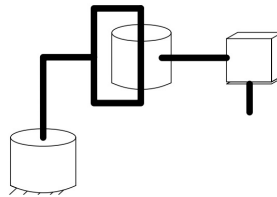


Figura 1.6: Robot SCARA

Robot Antropomórfico. Este robot cuenta con 3 articulaciones de revolución donde la primera permite un giro sobre su propio eje y las otras dos generan un desplazamiento vertical u horizontal depende como este posicionado, y que se puede efectuar de manera independiente una de la otra, esto se puede ver en la Figura 1.7

Muñeca Esférica. Este no es un robot en sí, pero se suele acoplar a alguno de los anteriores, ya que su forma ayuda a tener un mayor control sobre el efector final. Consta de 3 articulaciones de revolución y permite un desplazamiento esférico en un espacio de 360° , la Figura 1.8 es muestra de la configuración.

Estos son los ejemplos más comunes de aplicación de robots manipuladores serie

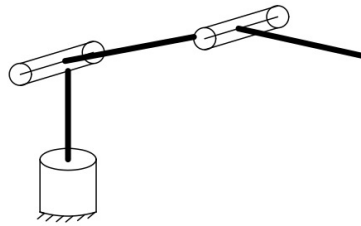


Figura 1.7: Robot Antropomórfico

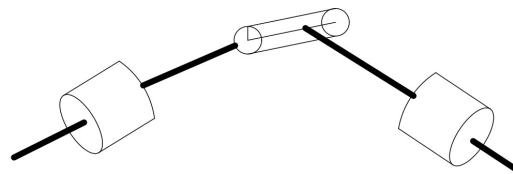


Figura 1.8: Muñeca Esférica

rígidos. El que se presentará en este trabajo es puramente demostrativo y con fines de estudio, ya que cuenta con 2 articulaciones de revolución y por lo tanto un espacio de trabajo muy limitado, pero que da la facilidad de desarrollar los objetivos propuestos y verificar la hipótesis establecida.

Capítulo 2

Diseño y Construcción del Brazo Robótico

2.1. Herramientas y Software

Las herramientas y software necesarios para el diseño y construcción de un manipulador robótico de n grados de libertad, constan de la siguiente lista, divididas en herramientas computacionales y herramientas de trabajo que se utilizaron para la construcción del manipulador. Estas herramientas se enlistan en las Tablas 2.1 y 2.2, considerando que se utilizaron las versiones libres, gratuitas o de prueba por lo tanto no tienen impacto en el costo final.

Esta lista de herramientas, junto con la lista de materiales, sirven de referencia para la descripción del proceso de construcción del manipulador. Especificando su función y su utilidad en la misma de manera general.

Tabla 2.1: Programas utilizados

No.	Nombre	Descripción	Utilidad
1	Visual Studio 2019	Es un entorno de desarrollo integrado para Windows y MacOS, maneja una diversa gama de herramientas de programación.	Esta herramienta es útil para diseñar la interfaz gráfica con la que el usuario aplicará las diferentes técnicas de control.
2	MPLAB X IDE	Es un entorno de desarrollo integrado gratuito, destinado a productos de la marca Microchip, que permite programar diversos modelos de microcontroladores	Este software fue esencial para compilar y depurar el programa de los microcontroladores PIC18F2331 y PIC18F4550, además de proporcionar la interfaz para su respectiva programación
3	KiCAD	Es un programa para el diseño de circuitos electrónicos, que permite pasar de un esquemático a placa de circuito impreso	Esta herramienta que permite diseñar las tarjetas de circuito impreso de la etapa de control y potencia.
4	MATLAB	Es un sistema de cómputo numérico, con un entorno de desarrollo integrado basado en un lenguaje de programación propio	Es la herramienta de computo que fue utilizada en el procesamiento de la información de los resultados obtenidos de cada algoritmo de control.
5	AutoCAD	Es un software de diseño utilizado para dibujos 2D y 3D.	Este software permitió diseñar los planos estructurales de cada uno de los eslabones, junto con la base principal.

Tabla 2.2: Herramientas de trabajo y hardware

No.	Nombre	Descripción
1	PicKit 3	Es un dispositivo programador y depurador de microcontroladores, de la marca Microchip
2	Vernier	Es un instrumento de medición principalmente utilizado para medir espesores, diámetros y tamaños, en unidades de milímetros o pulgadas.
3	Multímetro Digital	Es un instrumento de medición digital, utilizado para determinar magnitudes eléctricas como voltaje, corriente y resistencia.
4	Osciloscopio	Es un instrumento de medición, capaz de obtener de manera gráfica, una señal eléctrica de cualquier tipo.
5	Esmeriladora y disco de corte para metal 4-1/2"	Es una herramienta rotativa de corte mediante el uso de un disco abrasivo que gira a alta velocidad.
6	Taladro y juego de brocas	Es una herramienta rotativa de perforación mediante el uso de brocas que usualmente están construidas de acero
7	Protoboard	Es una tarjeta perforada, con interconexión eléctrica para la implementación de prototipos electrónicos
8	Desarmadores, llaves, pinzas	Herramientas de trabajo para ensamblaje, corte y manipulación de piezas o cableados.
9	Báscula	Instrumento de medición de masa
10	Cautín	Es una herramienta de soldado de componentes electrónicos
11	Computadora de escritorio	Características: Windows 8.1 Pro x64Bits, procesador AMD FX(tm)-6100 six-core, memoria RAM 12Gb, SSD 128Gb, tarjeta gráfica AMD Raizen R7 240

2.2. Estructura del Manipulador

La estructura general consta de dos eslabones, el primero de dos barras paralelas de dimensiones $20 \times 4 \times 0.18$ [cm] y el segundo una sola barra de $30 \times 4 \times 0.18$ [cm], unidos mediante tornillos, rodamientos, coples, conjuntos de engranajes, motores, encoders y una base de soporte a tierra. El manipulador a construir consiste en un robot planar de dos grados de libertad, como se muestra en la Figura 2.1.

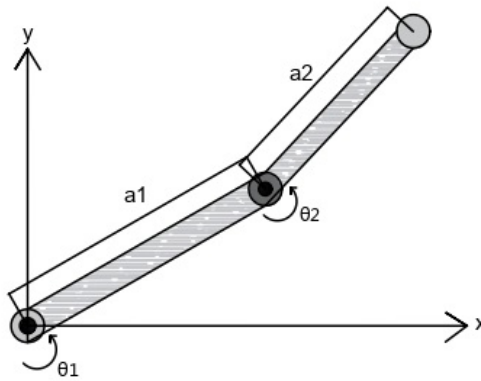


Figura 2.1: Robot Planar de Dos Grados de Libertad

2.2.1. Materiales y Selección de Elementos

Los materiales propuestos para diseño de la estructura de los eslabones del manipulador se especifican en la Tabla 2.3, donde se hace la comparación de los materiales comerciales para estructuras.

En función de la Tabla 2.3, se concluyó que el mejor material para esta aplicación es el aluminio, por su costo, siendo el más barato de los tres. En cuanto a la resistencia mecánica a la torsión y a la elasticidad, aunque es la más pequeña de las tres, es suficiente para la aplicación del manipulador. Su dureza en comparación con los otros dos es menor y todo lo demás se compensa con su ligereza y con la facilidad de penetración a la hora de perforar con broca. El problema de la corrosión se puede corregir con ayuda de algún esmalte acrílico.

Además de la base principal de la estructura que son las placas de aluminio, se requiere del uso de tornillos, tuercas, soportes para motor, abrazaderas de aluminio y acopladores. Los acopladores se eligieron a partir del tamaño de los ejes de los motores y del

Tabla 2.3: Propuesta de materiales para la estructura

Material	Características	lámina	
Aluminio	Cuenta con una gran ligereza alrededor de los 2.7 [g/cm ³], una resistencia mecánica que ronda 130 [N/mm ²], una elasticidad de 110 [N/mm ²] y cuenta con un elevado estado de oxidación	lámina 0.91x2.44	de [m] \$1634
Hierro	Este material lo podemos encontrar en varias gamas según su resistencia mecánica, pero la aleación seleccionada, T-304, tiene un valor de 490 a 686 [N/mm ²], una elasticidad de 205 [N/mm ²], una ligereza de 7.85 [g/cm ³], resistencia a la corrosión mayor a la del aluminio pero menor a la del cobre	lámina 1.22x1	de [m] \$3700
Cobre	El cobre es un material con alta resistencia a la oxidación, con una ligereza de 8.96 [g/cm ³], con una resistencia mecánica de 362 a 402 [N/mm ²] y una elasticidad de 353 [N/mm ²]	lámina 0.91x2.44	de [m] \$2899

ancho de las barras de aluminio, que para ambos casos la dimensión máxima es de 4 [cm].

Tabla 2.4: Tipo de acoplamiento para motor

Motor	Dimensiones eje	Acoplamiento
Motor articulación 1	diámetro 3.6 [mm]	Tipo brida 4 [mm]
Motor articulación 2	diámetro 3 [mm]	Tipo brida 3 [mm]

En la Sección 1.3.3, se menciona que los eslabones se numeran del 0 al $n+1$, considerando al 0 como la base, donde va a estar sujeto el motor de la articulación 1, dicho motor se sujetó utilizando los siguientes materiales. En el plano estructural se detallará su aplicación.

- Soporte de aluminio para motor modelo 37GB.
- Abrazadera de aluminio para tubo 20 [mm]
- Abrazadera de aluminio para tubo 22 [mm]

2.2.2. Plano Estructural

El plano estructural se compone de la siguiente manera. Eslabón 1: dos barras paralelas de 20 [cm] largo por 4 [cm] de ancho y con un espesor de 0.18 [cm], separadas entre ellas 2.7 [cm], ensamblados con 6 tornillos de $\frac{1}{8} \times 1.5$ [in], 4 para fijar el motor de la articulación 2 y 2 para sujetar la base y con un acoplamiento de brida rígida de 4 [mm] para fijar el motor de la articulación 1, al eslabón 1, con tornillos de $\frac{1}{8} \times 0.5$ [in], como se muestra en la Figura 2.2.

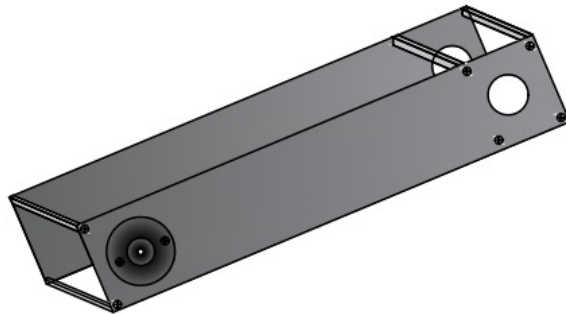


Figura 2.2: Estructura del Primer Eslabón

Eslabón 2: se conforma de una única barra de aluminio de dimensiones 30 [cm] de largo por 4 [cm] de ancho por 0.18 [cm] de espesor, con un acoplamiento de brida rígida de 3 [mm] en la base para fijarlo con el motor de la articulación 2, utilizando tornillos de $\frac{1}{8} \times 0.5$ [in], como se muestra en la Figura 2.3.

Articulación 1: consta del motor 1 sujeto al eslabón 0, es decir, la base donde está sujeto el manipulador completo. Para garantizar que se mantendrá fijo a la base se utilizaron 3 elementos, dos abrazaderas que fijan al motor para evitar que gire con 4 tornillos de $\frac{1}{8} \times 1.5$

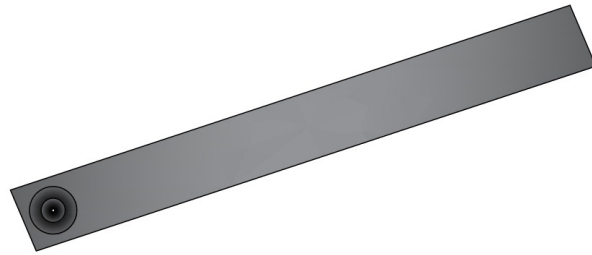


Figura 2.3: Estructura del Segundo Eslabón

[in] y un soporte para motor 37GB con 4 tornillos de $\frac{1}{8} \times 1.5$ [in], que sera la unión entre la base y el manipulador, como se muestra en la Figura 2.4.

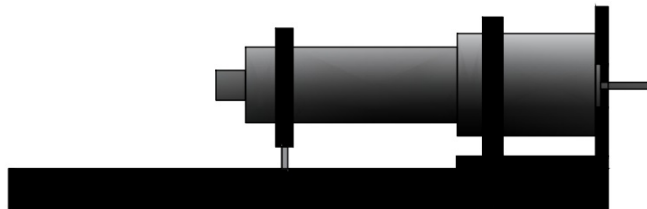


Figura 2.4: Estructura para la Primera Articulación

El soporte de la articulación 2 está constituido por una placa de aluminio de dimensiones 4×3.1 [cm], con un espesor de 0.7 [mm], sujeta al eslabón 1 con 4 tornillos mismos que fijan al motor de la articulación 2. Este soporte cuenta con una perforación en el centro de radio 3.0 [mm] donde embona el motor de la articulación 2 y es sujetado por 3 tornillos de $2.5 \times 1.6 \times 4.5$ [mm], el cual se muestra en al Figura 2.5.

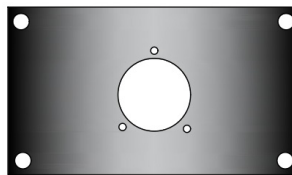


Figura 2.5: Estructura para Montar la Segunda Articulación

2.2.3. Costos de Fabricación

El costo del material está en función de lo descrito en la Tabla 2.5. junto con lo descrito en plano estructural de la Sección 2.2.2. Para determinar los costos de construcción para esta sección solo se considera el material, dado que las herramientas no son parte de la estructura del manipulador.

Tabla 2.5: Costos de la estructura del manipulador

Material	Cantidad	Costo individual	Costo General
Barra de aluminio 30x4x0.18 [cm]	1	\$9 MXN	\$9 MXN
Barras de aluminio 20x4x0.18 [cm]	2	\$6 MXN	\$12 MXN
Acoplamiento de brida 4 [mm]	1	\$40 MXN	\$40 MXN
Acoplamiento de brida 3 [mm]	1	\$30 MXN	\$30 MXN
Tornillo de $\frac{1}{8} \times 1.5$ [in]	14	\$2 MXN	\$28 MXN
Abrazadera de aluminio para tubo 20 [mm]	1	\$50 MXN	\$50 MXN
Abrazadera de aluminio para tubo 20 [mm]	1	\$50 MXN	\$50 MXN
Soporte para motor 37GB	1	\$40 MXN	\$40 MXN
Tornillo de $\frac{1}{8} \times 0.5$ [in]	4	\$1.5 MXN	\$6 MXN
Tornillo de 2.5x1.6x4.5 [mm]	3	\$0.5 MXN	\$1.5 MXN
Placa de aluminio 4x3.1 [cm]	1	\$15 MXN	\$15 MXN

Considerando el costo general por cada material al final tenemos un gasto de \$251.5 MXN, para la parte de la estructura base del manipulador, con esto pasamos a la siguiente

sección donde se describe el diseño de la parte de actuación y suministro de potencia, al igual que los costos de desarrollo.

Nota 1. Los valores nominales de los materiales utilizados se adquirieron de productos comerciales, descritos en la bibliografía.

2.3. Eléctrica y Electrónica del Manipulador

El diseño electrónico, se realizó en diferentes secciones, en función de los niveles de voltaje que implementan cada una de ellas, divididas en los siguientes:

- Etapa de Potencia
- Fuente de Suministro Eléctrico, Convertidor CA-CD
- Etapa de Control y Adquisición

Aunque la tercera etapa se revisará en la Sección 2.4 (Módulo de Control y Adquisición), es importante mencionarla aquí ya que también es parte del diseño electrónico.

2.3.1. Materiales y Selección de Elementos

En esta sección se definen, describen y justifican los componentes eléctricos y electrónicos que se utilizaron para la construcción del manipulador, dividido en las tres etapas ya antes definidas.

Etapa de Potencia

La Etapa de Potencia consta de la electrónica que es capaz de suministrar un determinado valor de potencia a los motores de cada uno los eslabones. Para establecer esto es necesario definir qué motores son los que serán utilizados en función de la carga a la que serán sometidos. Para ello nos basaremos en la Sección 2.2.2 (Plano Estructural), donde se describen las dimensiones, peso y tipo de material que se utilizó para la construcción de cada uno de ellos.

Para determinar el par que deben entregar los motores, es necesario calcular el par que generan los eslabones cuando son sometidos a las fuerzas físicas naturales, por lo que se recurre a un diagrama de cuerpo libre para realizar dicho cálculo, comenzado por el motor de la articulación número 2, que es el que menor par necesita.

Para determinar el modelo, potencia, sistema de engranajes y encoder del motor para el eslabón dos, se analizar la carga correspondiente, para lo que recurriremos al análisis de fuerzas. Pero antes determinaremos el centro de masa del eslabón para posicionar la única fuerza de acción generada por el peso del mismo.

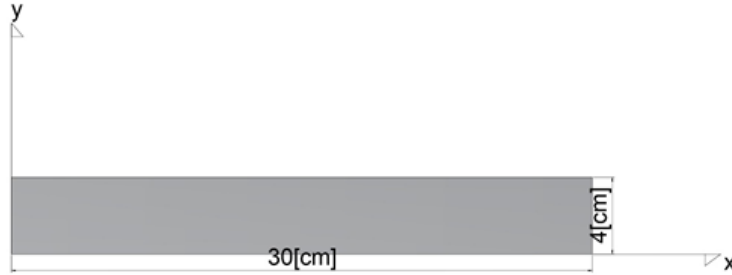


Figura 2.6: Eslabón 2

Para el cálculo del centro de masa utilizaremos las ecuaciones

$$\mathbf{G} = (x, y) = \left(\left(\frac{\int_{x=0}^{x=L} x\rho(x) \cdot dx}{\int_{x=0}^{x=L} \rho(x) \cdot dx} \right), \left(\frac{\int_{y=0}^{y=A} y\rho(y) \cdot dy}{\int_{y=0}^{y=A} \rho(y) \cdot dy} \right) \right).$$

Por las características del material utilizado, se consideró la densidad lineal constante y se define como $\rho = \frac{M}{L}$ y $\rho = \frac{M}{A}$, de tal manera que:

$$x = \frac{\int_{x=0}^{x=L} x\rho \cdot dx}{\int_{x=0}^{x=L} \rho \cdot dx} = \frac{\int_{x=0}^{x=L} x \frac{M}{L} \cdot dx}{\int_{x=0}^{x=L} \frac{M}{L} \cdot dx} = \frac{\frac{M}{2L} x^2 \Big|_0^L}{\frac{M}{L} x \Big|_0^L} = \frac{L}{2},$$

$$y = \frac{\int_{y=0}^{y=A} y\rho \cdot dy}{\int_{y=0}^{y=A} \rho \cdot dy} = \frac{\int_{y=0}^{y=L} y \frac{M}{A} \cdot dx}{\int_{y=0}^{y=A} \frac{M}{A} \cdot dx} = \frac{\frac{M}{2A} y^2 \Big|_0^A}{\frac{M}{A} y \Big|_0^A} = \frac{A}{2}.$$

De esta manera el centro de masa se queda ubicado en:

$$\mathbf{G} = (x, y) = \left(\frac{L}{2}, \frac{A}{2} \right) = (0.15[m], 0.02[m]).$$

Considerando el centro de masa procedemos a realizar un diagrama de cuerpo libre, ubicando la fuerza generada por la acción gravitatoria en dicho punto. Aunado a esto posicionamos al eslabón de manera horizontal, que es cuando se presenta la magnitud máxima de fuerza. El diagrama se presenta en la Figura 2.7.

Para determinar el brazo de palanca, tomaremos en cuenta las siguientes condiciones: constante de aceleración gravitatoria $g=9.82011[m/s^2]$ y masa $m=0.05 [kg]$, por lo tanto el peso del eslabón es:

$$w = mg \tag{2.1}$$

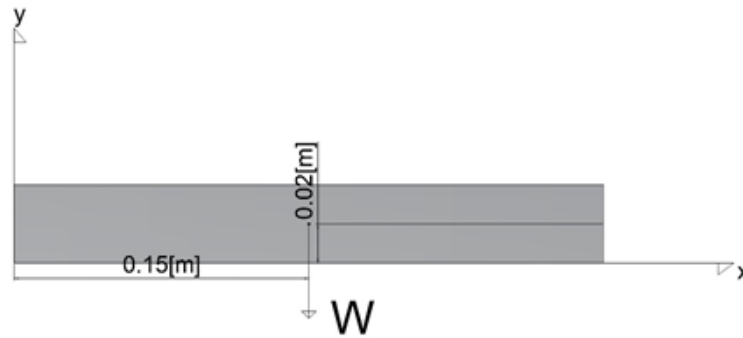


Figura 2.7: Eslabón 2

$$w = (0.05)(9.82011) = 0.5008 \text{ [N]}$$

Con esto el momento generado por el eslabón, con respecto al origen es:

$$M_0 = Fd \tag{2.2}$$

$$M_0 = (0.5008)(0.15) = 0.075 \text{ [Nm]}$$

Considerando cuatro veces la masa $M_0 = 0.30 \text{ [Nm]}$

Con esta información podemos determinar que los motores deben vencer una fuerza contraria al par generado por los efectos de la gravedad sobre la barra de aluminio para ello procedemos a elegir el tipo de motor, sistema de engranajes y encoder, pero además debemos de considerar otros aspectos como el voltaje de operación, la corriente nominal, la corriente de arranque, la relación de reducción del tren de engranajes, la velocidad nominal y máxima, el par máximo y el par de arranque. También se pueden considerar aspectos extra como el tipo de engranajes y el tipo de escobillas. Para considerar lo anterior se necesitan valores de referencia, dichos valores se obtienen del catálogo de juego de motores, encoders y sistemas de engranajes de la marca MAXON y estos son las siguientes:

- Corriente máxima 2[A]
- Tensión 12[V]
- Par final 0.3[Nm]
- Par del motor 2.91[mNm]
- Encoder de cuadratura

- Relación de reducción máxima 103:1

En el cuadro se muestran las opciones propuestas a elección. Estas opciones fueron seleccionadas del catálogo de motores, sistemas de engranajes y encoders de la marca MAXON.

Tabla 2.6: Motores propuestos para la articulación 2

Modelo	Características
DCX 12 L 12 [mm]	par 3.88 [mNm], velocidad 8800 [RPM], voltaje 12 [V], corriente máxima 0.824 [A]
DCX 12 S 12 [mm]	par 1.89 [mNm], velocidad 9030 [RPM], voltaje 12 [V], corriente máxima 0.261 [A]
DCX 14 L 14 [mm]	par 6.38 [mNm], velocidad 7740 [RPM], voltaje 12 [V], corriente máxima 1.35 [A]
DC-max 16 s 16 [mm]	par 4.41 [mNm], velocidad 9960 [RPM], voltaje 12 [V], corriente máxima 1.05 [A]

Tabla 2.7: Sistema de engranajes propuestos para la articulación 2

Sistema de engranajes	Características
GPX12 HP 12 [mm]	Para el DCX 12 L 12 [mm] y DCX 12 S 12 [mm], par máximo 0.3 [Nm], relación de reducción 103:1
GPX14 HP 14 [mm]	Para el motor DCX 14 L 14 [mm], par máximo 0.4 [Nm], Relación de reducción 103:1
GPX16 LZ 16 [mm]	Para el motor DC-max 16 s 16 [mm], par máximo 0.35 [Nm], Relación de reducción 62:1

Comparando las opciones con las limitaciones que se consideraron en un principio, la mejor opción es el motor DCX 12 S 12 mm, el sistema de engranajes GPX12 HP 12mm y el encoder ENX EASY 10.

Para determinar el modelo, potencia, sistema de engranajes y encoder del motor sujeto al eslabón uno, se analizar la carga correspondiente a esta misma, para lo cual recurriremos al análisis de fuerzas presentes en el eslabón uno en conjunto con el dos, con ayuda de

Tabla 2.8: Encoder propuestos para la articulación 2

Encoder	Características
ENX 10	resolución 512 [pulsos], voltaje 5 [v], encoder de cuadratura a 3 canales A, B, I

un diagrama de cuerpo libre y la representación del centro de masa. Para ello se determina el centro de masa de los dos eslabones y del motor de la segunda articulación para posicionar la única fuerza de acción generada por el peso de todo el manipulador en conjunto. Se sabe que el eslabón uno es una barra doble de dimensiones 20 [cm] de largo por 4 [cm] de ancho y 0.18 [cm] de espesor, separada de la otra a 2.7 [cm] y que se considera como una sola pero con el doble de espesor pero que no afecta el centro de masa, con una masa de 55 [g]. El motor de la articulación 2 se considera como una única masa, en forma de cilindro con radio de 6 [mm] y una masa de 44 [g] y por último el segundo eslabón, cuyas dimensiones son 30 [cm] largo, 4 [cm] de ancho y 0.18 [cm] de espesor. Comenzamos el análisis con el eslabón 1, como se muestra en la Figura 2.8.

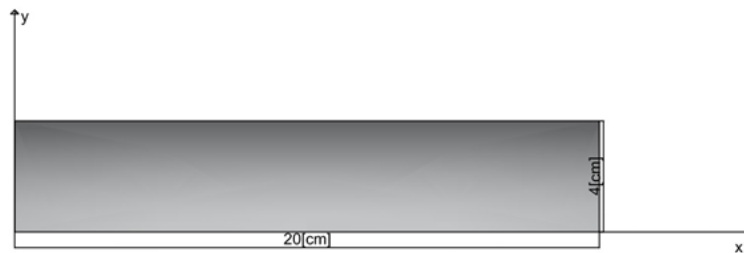


Figura 2.8: Eslabón 1

Para el cálculo del centro de masa utilizaremos las ecuaciones

$$\mathbf{G} = (x, y) = \left(\left(\frac{\int_{x=0}^{x=L} x\rho(x) \cdot dx}{\int_{x=0}^{x=L} \rho(x) \cdot dx} \right), \left(\frac{\int_{y=0}^{y=A} y\rho(y) \cdot dy}{\int_{y=0}^{y=A} \rho(y) \cdot dy} \right) \right).$$

Si suponemos que la masa esta uniformemente distribuida, consideraremos a la densidad lineal constante como $\rho = \frac{M}{L}$ y $\rho = \frac{M}{A}$. de esta forma tenemos que:

$$x = \frac{\int_{x=0}^{x=L} x\rho \cdot dx}{\int_{x=0}^{x=L} \rho \cdot dx} = \frac{\int_{x=0}^{x=L} x \frac{M}{L} \cdot dx}{\int_{x=0}^{x=L} \frac{M}{L} \cdot dx} = \frac{\frac{M}{2L} x^2 \Big|_0^L}{\frac{M}{L} x \Big|_0^L} = \frac{L}{2},$$

$$y = \frac{\int_{y=0}^{y=A} y \rho \cdot dy}{\int_{y=0}^{y=A} \rho \cdot dy} = \frac{\int_{y=0}^{y=L} y \frac{M}{A} \cdot dx}{\int_{y=0}^{y=A} \frac{M}{A} \cdot dx} = \frac{\frac{M}{2A} y^2 \Big|_0^A}{\frac{M}{A} y \Big|_0^A} = \frac{A}{2}.$$

Entonces de esta manera para las condiciones dadas, el centro de masa se encuentra ubicado en.

$$\mathbf{G} = (x, y) = \left(\frac{L}{2}, \frac{A}{2}\right) = (0.1[m], 0.02[m])$$

El segundo análisis es para el motor 2.

Para el cálculo del centro de masa de una circunferencia, nos basamos en tablas, donde se describen el centro de masa de figuras ya definidas, y para este caso tenemos la siguiente representación del motor, como una circunferencia de radio 6 [mm], como se muestra en la Figura 2.9.

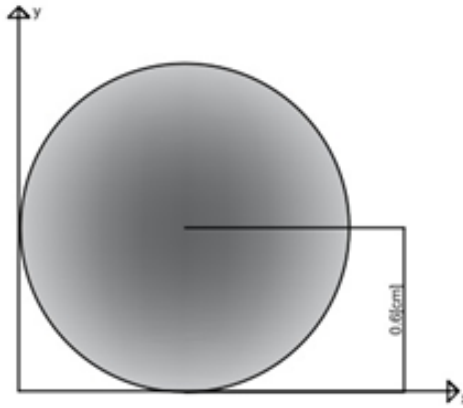


Figura 2.9: Representación en el Plano del Motor para la Articulación 2

El centro de masa de una circunferencia, descrito en tablas, se define de la siguiente manera.

$$\mathbf{G} = (x, y) = (r, r) \tag{2.3}$$

donde:

- r es el radio de la circunferencia.

Por lo tanto, la ubicación del centro de masa queda de la siguiente manera

$$\mathbf{G} = (x, y) = (0.006[m], 0.006[m]).$$

El tercer análisis es para el eslabón 2, el cual ya se calculó en la ecuación (2.4) y tenemos que

$$\mathbf{G} = (x, y) = \left(\frac{L}{2}, \frac{A}{2}\right) = (0.15[\text{m}], 0.02[\text{m}]).$$

Una vez obtenido el centro de masa de cada uno de los cuerpos podemos determinar el centro de masa general de todo el modelo, basándonos en la Figura 2.10

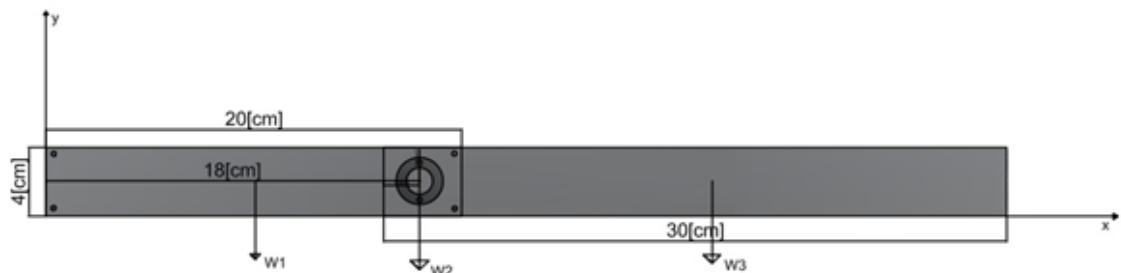


Figura 2.10: Diagrama de Fuerzas

Mediante la siguiente ecuación determinaremos la posición del centro de masa, del modelo

$$\mathbf{G} = (x, y) = \left(\frac{m_1x_1 + m_2x_2 + m_3x_3}{m_1 + m_2 + m_3}, \frac{m_1y_1 + m_2y_2 + m_3y_3}{m_1 + m_2 + m_3}\right) \quad (2.4)$$

Por lo tanto, la posición del centro de masa queda como

$$\mathbf{G} = (x, y) = (0.184[\text{m}], 0.02[\text{m}])$$

Se coloca a la fuerza generada por la acción gravitatoria en el centro de masa y además el máximo par generado por esta fuerza cuando la barra esta en posición horizontal, es decir, perpendicular a los efectos gravitacionales, como se muestra en la Figura 2.11.

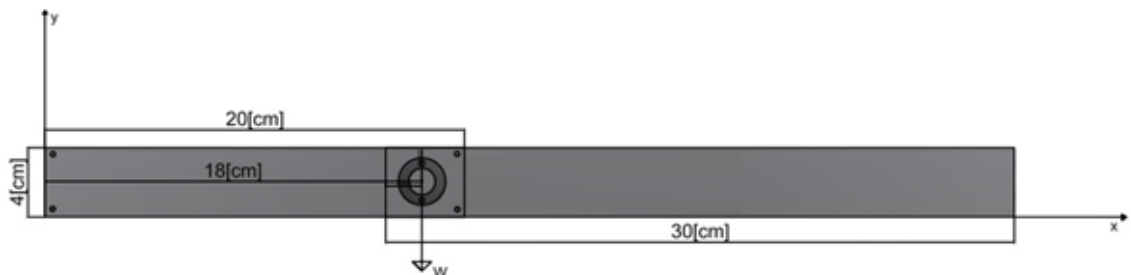


Figura 2.11: Diagrama de Cuerpo Libre

Para determinar el brazo de palanca se consideran las siguientes condiciones, constante de aceleración gravitatoria $g=9.822 \text{ [m/s}^2\text{]}$ y masa $m=0.183 \text{ [Kg]}$, de tal manera que con ayuda de la ecuación (2.5) se determina el peso del robot

$$w = mg \quad (2.5)$$

siendo

$$w = 1.79421 \text{ [N]}. \quad (2.6)$$

El momento generado por esta barra con respecto al origen se da de la siguiente manera

$$M_1 = Fd = 1.79421(0.184) = 0.3301 \text{ [Nm]} \quad (2.7)$$

Considerando cuatro veces la masa $M_1 = 1.79421(0.184)(4) = 1.32 \text{ [Nm]}$

Ya que determinamos el brazo de palanca necesario para mover el manipulador completo, ahora procedemos a la selección del motor y sistema de engranajes adecuado, junto con el sensor, que nos permitirá determinar el ángulo de posición del eslabón correspondiente. Para lo cual nos basamos en el catálogo de motores de MAXON. Tomando en cuenta las siguientes restricciones.

- Corriente máxima 2 [A]
- Tensión 12 [V]
- Par 1.32 [Nm]
- Par de referencia del motor 2.91 [mNm]
- Encoder de cuadratura
- Relación de reducción máxima 103:1

Las propuestas están en función de las especificaciones del catálogo de MAXON, la configuración puede cambiar según las restricciones y especificaciones establecidas por la tienda.

Con base en los datos mostrados en las Tablas 2.9, 2.10 y 2.11 además de las limitaciones preestablecidas deducimos que la mejor opción para elegir es el motor DCX 19 S, el sistema de engranajes GPX22C y el encoder ENX 10.

Tabla 2.9: Motores propuestos para la articulación 1

Modelo	Características
DCX 16 S 16 [mm]	par 5.19 [mNm], velocidad 3320 [RPM], voltaje 12 [V], corriente máxima 0.624 [A]
DCX 19 S 19 [mm]	par 11 [mNm], velocidad 4490[RPM], voltaje 12 [V], corriente máxima 2.13 [A]

Tabla 2.10: Propuestas para el sistema de engranajes de la articulación 1

Sistema de engranajes	Características
GPX16 HP 16 [mm]	Para el motor DCX 16 S 16 [mm], par máximo 0.8 [Nm], relación de reducción 103:1
GPX19 C 16 [mm]	Para el motor DCX 16 S 16 [mm], par máximo 0.86 [Nm], relación de reducción 103:1
GPX19 HP 19 [mm]	Para el motor DCX 19 S 19 [mm], par máximo 1.6 [Nm], relación de reducción 103:1
GPX22 C 22 [mm]	Para el motor DCX 19 S 19 [mm], par máximo 1.6 [Nm], relación de reducción 103:1

Tabla 2.11: Encoder propuestos para la articulación 1

Encoder	Características
ENX 10	resolución 512 [pulsos], voltaje 5 [v], encoder de cuadratura a 3 canales A, B, I

Con las especificaciones eléctricas de cada uno de los motores, empezamos con la elección de la electrónica adecuada para la etapa de potencia. En esta etapa se coloca una sección de aislamiento entre el circuito de mayor voltaje y el circuito de control. Las especificaciones que debe de cumplir son las siguientes:

- Corriente máxima en continuo 2 [A]
- Tensión en continuo 12 [V]

- Voltaje de control lógico de 5 [V]
- Control de velocidad digital por PWM

Para cumplir con las especificaciones antes mencionadas se propone la utilización de 3 tipos de elementos, en los 3 casos son circuitos de control digital e implementan el principio de un puente H.

- Puente H con Relés
- Puente H con transistores MOSFET
- Puente H con Circuito Integrado

Cada uno de ellos cuentan con sus ventajas y desventajas. Explicaremos un poco en qué consiste cada una de ellas, sin llegar a profundizar y nos centraremos en describir más a detalle el que se utilizó.

Puente H con relés

Un puente H con relés, como su mismo nombre lo dice, está constituido por los dispositivos eléctricos llamados relevadores, que funcionan bajo un principio electromecánico, permitiendo que un interruptor conmute en función de la excitación de una bobina, donde esta bobina genera un campo magnético que atrae o repele una lámina que funciona como interruptor, un ejemplo de ello se muestra en la Figura 2.12.

Puente H con Transistor MOSFET.

Este diseño de circuito es similar al del puente con relé. La diferencia radica en que en lugar de utilizar relevadores implementa transistores. En específico los de tecnología MOSFET, que es capaz de soportar mayores niveles de corriente, un variable eléctrica fundamental cuando se requiere un suministro de mayor potencia a un voltaje constante, como en el caso de los controles digitales. Estos transistores funcionan bajo el principio de los transistores de efecto campo, por lo que se polarizan mediante un nivel de voltaje específico y que al igual que los TBJ cuentan con tres regiones de operación. Para el caso de un control digital la región de operación que nos interesa es la de saturación y corte, para crear una señal digital PWM, ejemplo de Un ejemplo de este tipo de puentes lo vemos en la Figura 2.13.

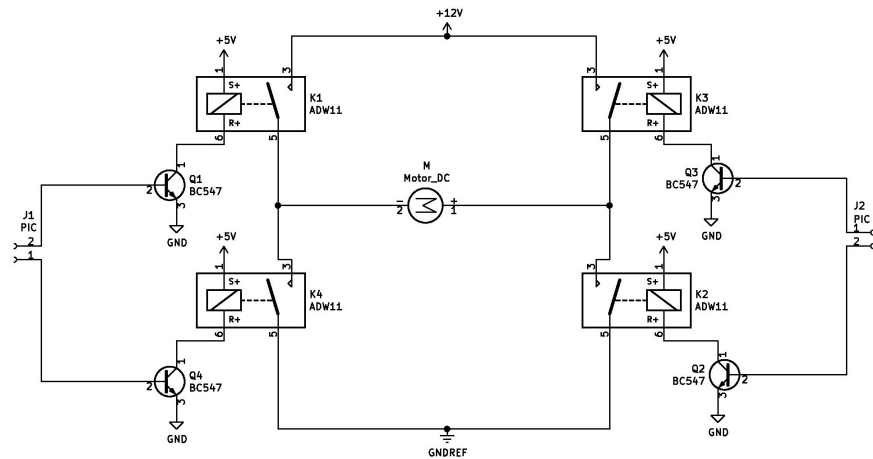


Figura 2.12: Puente H con Relé

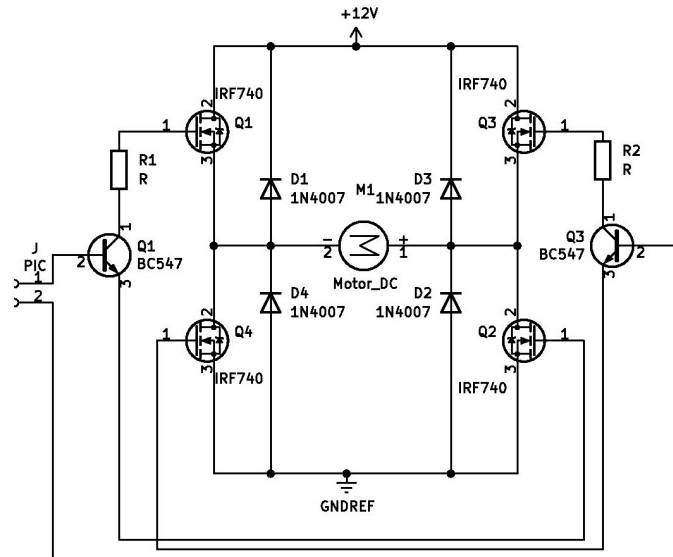


Figura 2.13: Puente H con Transistor MOSFET Ejemplo

Puente H con Circuito Integrado.

Finalmente tenemos al puente H en circuito integrado, es un dispositivo que está conformado por electrónica digital y transistores de potencia, todo integrado en un solo chip. La ventaja es que ya todo viene integrado y solo requiere de elementos adicionales de acondicionamiento para su utilidad, es fácil de conectar y ocupa menos espacio que un puente hecho o construido con los elementos anteriores. La desventaja es que a mayor corriente de

trabajo mayor es el costo del dispositivo, pero al igual que el puente con transistores MOSFET tiene la capacidad de permitir conmutaciones a mayor frecuencia que es la desventaja de un puente con relés ya que estos están limitados a su velocidad de conmutación por sus características mecánicas.

Revisando los tres modelos llegamos a la conclusión de que la mejor opción para esta aplicación es la del puente H en circuito integrado, ya que dadas las condiciones de operación en las que va a trabajar no requiere manejar corrientes muy altas y en el mercado podemos encontrar una gran variedad de opciones a bajo costo y con las especificaciones necesarias. Las opciones candidatas se describen en la Tabla 2.12.

Tabla 2.12: Opciones de puente H en circuito integrados

Modelo	Características
L293B	voltaje lógico 5 [v], voltaje de suministro de 5 [v] a 36 [v], corriente de salida máxima 1 [A]
L298N	voltaje lógico 5 [v], voltaje de suministro de 5 [v] a 36 [v], corriente de salida máxima 2.5 [A], frecuencia de conmutación máxima 40 [kHz]
TB6612FNG	voltaje lógico 5[v], voltaje de suministro de 6 [v] a 15 [v], corriente de salida máxima 1.2 [A], frecuencia de conmutación máxima 100 [kHz]

Estas opciones se basan en los componentes electrónicos más comunes en el mercado, fáciles de conseguir y de costos moderados. Ahora para determinar cuál de los tres modelos es el adecuado para la aplicación, es necesario recurrir a las especificaciones establecidas para los motores de cada una de las articulaciones, considerando el de mayor valor de consumo como referencia base, es decir, al motor de la articulación 1, que cumple con las siguientes características eléctricas.

- Voltaje 12[V]
- Corriente máxima 2.13[A]

Por lo tanto, la mejor opción para esta aplicación es el circuito integrado para motores L298N, el esquemático de la electrónica correspondiente correspondiente se mostrará en la Sección 2.3.2.

Para finalizar esta parte se describirá la forma en que se aislará la parte de alta potencia y baja potencia, esto lo vamos a realizar mediante un circuito básico, que funciona bajo un principio óptico. Los dispositivos que permite transmitir señales sin la necesidad de utilizar cables o conexiones físicas y además de origen digital, es el transmisor y receptor óptico, también conocido como optoacoplador. Aunque existen otros tipos de métodos de transmisor de señales de manera inalámbrica, la ventaja de los dispositivos ópticos es que el costo es bajo y la implementación es bastante sencilla además nos permite una capacidad de aislamiento bastante alta. Dado que la transmisión de la parte de control a la parte de potencia se realiza mediante un dispositivo sin conexión física, el uso de optoacopladores garantiza que la parte de baja potencia no resulte dañada por algún problema que se presente en la parte de alta potencia, como un corto circuito, picos de voltaje o picos de corriente. De esta manera se previene un daño mucho mayor, ejemplo de dicha protección se muestra en la Figura 2.14.

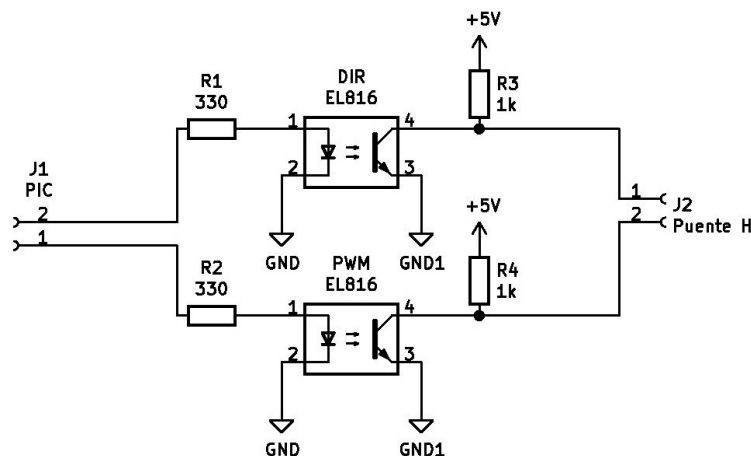


Figura 2.14: Circuito de Protección

Existen diversos tipos de optoacopladores en el mercado, pero el que requiere esta

aplicación debe de contar con ciertas características. Una de las principales es la velocidad de conmutación máxima que puede soportar, ya que una de las señales de control para los motores son del tipo PWM, que cuenta con cierta frecuencia y ciclo de trabajo. Si el dispositivo no es capaz de conmutar a esa velocidad y reproducir el ciclo de trabajo adecuado, entonces no será de utilidad para la etapa de protección, como se muestra en la Tabla 2.13.

- Frecuencia del PWM 4.88[kHz]
- Ciclo de trabajo de 0 % al 100 %

Tabla 2.13: Modelos de optoacopladores

Modelo	Características
EL816	frecuencia máxima 80 [kHz], tiempo de levantamiento 18 [μ s], tiempo de bajada 18 [μ s], salida a transistor
4N25	frecuencia máxima 166 [kHz], tiempo de levantamiento 1.2 [μ s], tiempo de bajada 1.3 [μ s], salida a transistor
PS2561	frecuencia máxima 10 [kHz], tiempo de levantamiento 5 [μ s], tiempo de bajada 3 [μ s], salida a transistor

Como se observa en la Tabla 2.13, cualquiera de los 3 dispositivos cumplen con las condiciones necesarias para la aplicación requerida, la única diferencia es el valor monetario de cada uno, así que el dispositivo a seleccionar es el optoacoplador EL816, por su bajo costo en comparación con los demás.

Para el circuito de protección necesitamos una fuente de voltaje de 5 [V] que polarice la salida a transistor de los optoacopladores y también a los integrados de potencia L298N. Este voltaje se puede obtener si colocamos un regulador de voltaje de 5 [V] a la salida de la fuente de suministro de 12 [V] que alimenta la etapa de potencia de los motores.

Tabla 2.14: Modelo de reguladores

Modelo Regulador	Características
L7805	voltaje de entrada máximo 35 [V], voltaje de salida máximo 5.2 [V], corriente de salida máxima 1.5 [A], regulación 100 [mV]
LM317	voltaje de entrada máximo 40 [V], voltaje de salida 1.2 [V] a 37 [V], corriente de salida máxima 1.5 [A], regulación 1 % para $V_0 \geq 5$ [V]
AN78M05	voltaje de entrada máximo 35 [V], voltaje de salida máximo 5.2 [V], corriente de salida máxima 0.5 [A], regulación 100 [mV]
1N5338A	voltaje de entrada máximo 12 [V], voltaje de salida máximo 5.1 [V], Corriente de salida máxima 1 [A]

También existen diversos tipos de reguladores de voltaje en el mercado. Para elegir entre las distintas opciones, es necesario determinar las condiciones eléctricas para alimentar los optoacopladores y demás dispositivos electrónicos que lo requieran y es aquella que cuenta con un voltaje de salida de 5 [V], una corriente de salida de preferencia lo más grande posible, con una variación de voltaje a la salida muy pequeña y además que soporte 12 [V] de entrada. De los modelos presentes en la Tabla 2.14 la mejor opción es el regulador L7805 ya que cuenta con las características necesarias incluyendo un bajo costo y una configuración de diseño sencilla de implementar, descrita en la hoja de datos del fabricante.

Fuente de Suministro Eléctrico, Convertidor CA-CD

La clave de la selección de una fuente de voltaje adecuada está en función del voltaje y la máxima potencia que suministra. Esto se traduce a la cantidad de corriente que es capaz de entregar a una carga. Para el manipulador se requerirá de una fuente de voltaje de 12 [V] y la corriente de consumo estará dada por cada uno de los motores y los circuitos de potencia. Además se consideró el peor de los casos donde cada elemento consume el máximo valor de corriente. Con base en lo anterior, calculamos lo siguiente, a partir de la Tabla 2.15.

Tabla 2.15: Corriente de consumo

Elemento	Corriente de consumo máxima
Motor articulación 1	2.13[A] en arranque
Motor articulación 2	0.261[A] en arranque
L298N	70[mA] en reposo y 100[μA] activo
EL816	50[mA] activo
L7805	6[mA] en reposo y 1[A] máxima carga

$$I_{necesaria} = 2.13[A] + 0.261[A] + 1[A] = 3.391[A] \quad (2.8)$$

El cálculo se realiza considerando que los dos motores generan su mayor consumo, que es en el arranque, y además el resto de la electrónica está en funcionamiento al mismo tiempo, lo cual solo se presentaría en un caso extraordinario, con esta información procedemos a la selección de la fuente adecuada.

La selección de la fuente adecuada queda en función de la tipología a utilizar, que puede ser mediante el uso de una fuente regulada con transformador de señal senoidal, o mediante una fuente conmutada con transformador de pulsos. La diferencia entre cada una de ellas radica en la potencia máxima de salida, el costo de cada una de ellas y sus dimensiones. Para el manipulador considerando todo lo anterior se deduce que la mejor opción es aquella que nos suministre más de 4 [A] y un voltaje de 12 [V]. Además si necesitamos de una mayor potencia por lo general es recomendable utilizar una fuente conmutada, ya que los transformadores de corriente alterna que entreguen una corriente alta suelen ser de costos muy elevados, demasiado grandes y pesados. Por lo tanto la fuente seleccionada cumple con las especificaciones mostradas en la Tabla 2.16.

Finalmente con toda la información necesaria para la electrónica y eléctrica del manipulador se procede al diseño mediante diagramas de bloques y esquemáticos.

Tabla 2.16: Fuente de voltaje

Modelo	Características
YXD WISH	Voltaje nominal 12 [V], Corriente nominal 10 [A], dimensiones 2.5x15x5 [cm]

2.3.2. Esquemático Electrónico y Diagrama de Bloques Eléctrico

La manera en que se distribuye la energía eléctrica a través de los componentes de la sección de potencia, de adquisición y de control se describe en el esquema de la Figura 2.15. Dicho suministro se divide en 4 secciones. La primera sección, color gris claro, hace referencia a la corriente suministrada por la línea de corriente alterna con un valor RMS de 127 [V]. En la segunda sección, color azul cielo, la señal de corriente alterna se convierte en una señal de corriente directa de 12 [V], que suministra a los integrados de potencia L298N y al regulador de voltaje L7805. En la tercera sección, color verde, el nivel de voltaje, suministrado por la sección de color azul, se reduce en magnitud a 5 [V] para alimenta a la etapa de adquisición, control y protección de la electrónica de bajo voltaje. En la cuarta sección, color rojo, se hace referencia a la señal de control PWM que es suministrada al motor de cada una de las articulaciones, con magnitud de 12 [V] pero con diferente ciclo de trabajo de acuerdo con la señal de control. Una vez que se determinan los niveles de tensión para cada uno de los componentes se procede al diseño electrónico de potencia, para la sección de color morado (L298N) y verde bandera (EL817).

El esquemático de la Figura 2.16 muestra el diseño electrónico del sistema de potencia de los actuadores de cada una de las articulaciones, el cual está constituido principalmente por el integrado L298N(U1), que es alimentado por un regulador de voltaje a nivel lógico de tensión de 5 [V](UF1). Es controlado por las líneas provenientes de la etapa de protección EL817(UD1) y el transistor en configuración compuerta lógica NOT (Qinv1). Ambas señales permiten determinar la dirección de giro del motor, lo que se logra cambiando el valor lógico de UD1. El optoacoplador El817(UPWM1) determina el nivel de potencia suministrado a los motores mediante una señal PWM. La salida al motor está en el arreglo de diodos de protección (D1, D2, D3, D4) y la terminal MOTOR1. El conector POWER_12v1 permite conectar la fuente de voltaje del convertidor AC/DC de la sección de color azul del diagrama

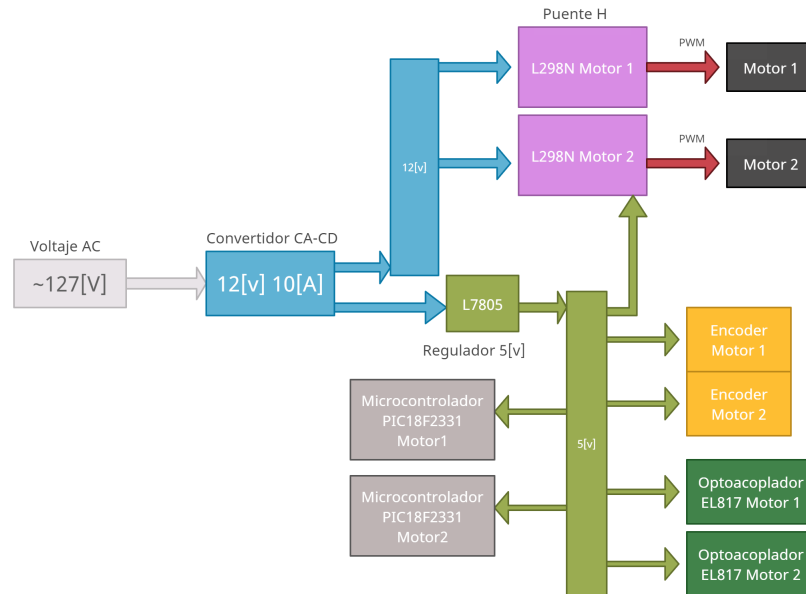


Figura 2.15: Suministro Eléctrico por Niveles

de la Figura 2.15. En la misma zona tenemos la terminal JFM2 que permite tener acceso a una línea proveniente de la fuente del convertidor AC/DC, para alimentar otro módulo de potencia. Por último tenemos las peinetas J2 y J3, conectadas a GND, que permiten conectar el módulo de control sobre el módulo de potencia, ahorrando espacio y permitiendo tener módulos individuales para cada una de las articulaciones. De esta manera se pueden agregar n articulaciones al robot sin la necesidad de modificar el circuito principal de control y adquisición general del manipulador.

EL CAD de la Figura 2.17, muestra el resultado final de dicho diseño. El módulo completo junto con la etapa de control y adquisición se muestra en la Sección 2.5.2.

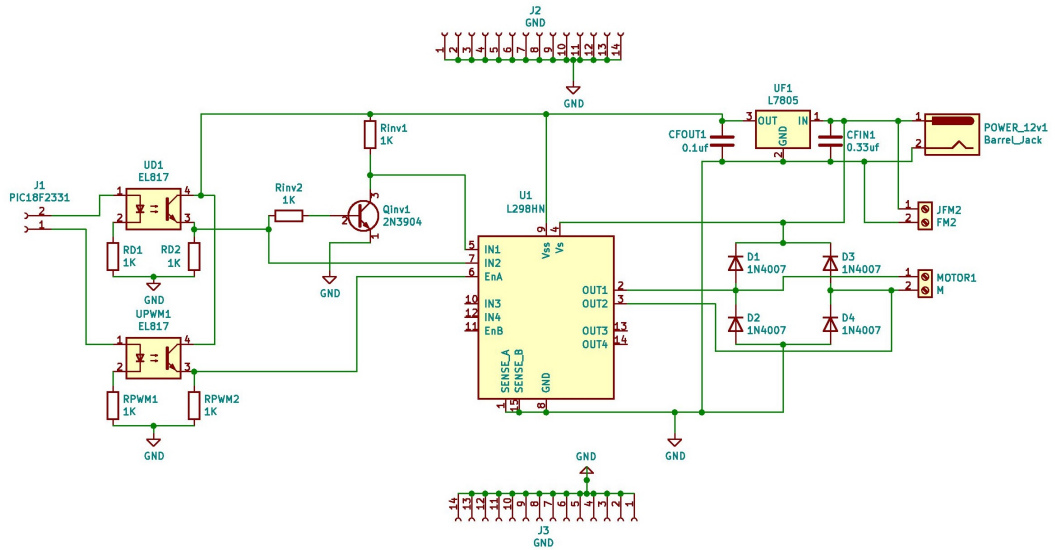


Figura 2.16: Esquemático del Módulo de Potencia (Módulo L298N)

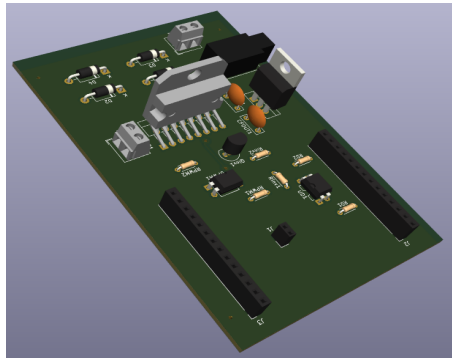


Figura 2.17: CAD Módulo de Potencia (Módulo L298N)

2.3.3. Costos de Fabricación

Para determinar el costo final de los componentes mencionados en la Sección 2.3, se dividen en dos cuadros los materiales y elementos utilizados, el módulo de la electrónica de potencia y los motores. Se realiza esta división ya que la parte fundamental del robot es el sistema de adquisición y control. El sistema de actuación puede ser sustituido por una gran variedad de modelos de motores que cumplan con las condiciones establecidas en la Sección 2.3.1 y que se pueden encontrar a diferentes precios pero funcionar de la misma manera o similar que un sistema con motores de alto costo.

Tabla 2.17: Costos del módulo de potencia

Componente	Cantidad	Costo individual	Costo General
L298N	2	\$51.72 MXN	\$103.45 MXN
EL817	4	\$15 MXN	\$60 MXN
2N3904	2	\$4.5 MXN	\$9 MXN
L7805	2	\$6 MXN	\$12MXN
YXD WISH	1	\$268.70 MXN	\$268.70 MXN
Barrel_jack	1	\$5 MXN	\$5 MXN
Terminales 2x1	4	\$4.31 MXN	\$17.24 MXN
1N4007	8	\$0.86 MXN	\$6.90 MXN
Heder hembra	4	\$8.62 MXN	\$34.48 MXN
Reistencias	12	\$0.5 MXN	\$6 MXN
Capacitores	4	\$2 MXN	\$8 MXN
Placa PCB	2	\$50 MXN	\$100 MXN

Tabla 2.18: Motores

Motor	Cantidad	Costo individual	Costo General	Liga de Referencia del Modelo
B7D1373441A6 A1	1	\$490.8 USD	\$9948 MXN	1
B7E74E5C687B A2	1	\$439.2 USD	\$8903 MXN	2

1 <https://www.maxongroup.com/maxon/view/configurator/?ConfigID=B7D1373441A6>

2 <https://www.maxongroup.es/maxon/view/configurator/?ConfigID=B7E74E5C687B>

El costo final calculado para la sección 2.3 es de \$630.77 MXN sin considerar los elementos de actuación. Agregando ahora el costo de los motores, al tipo de cambio actual (junio del 2022)de \$20.27 pesos, se tendría que \$19,481.87 MXN es el valor para la construcción de este prototipo.

Nota 2. Todas las especificaciones de los elementos propuestos y utilizados para

esta sección se pueden encontrar en las hojas de especificaciones de datos, descritas en la bibliografía.

2.4. Módulo de Control y Adquisición

Esta sección al igual que la anterior consta de dos etapas importantes que constituyen el módulo de Control y Adquisición, los cuales son:

- Etapa de Adquisición y Generación de Señales Digitales.
- Etapa de Comunicación y Transmisión de Datos.

2.4.1. Materiales y Selección de Elementos

Los materiales seleccionados constan de los modelos de microcontroladores, electrónica del acondicionamiento para generar las señales de control y para la adquisición de señales de los encoders, además de la comunicación con la computadora.

Etapa de Adquisición y Generación

Para esta etapa se considera lo siguiente: la velocidad de procesamiento del microcontrolador debe de ser alta, de bajo costo, capaz de generar PWM a una buena resolución, ser capaz de leer encoders de cuadratura para canal A y B, en el peor de los casos contar con mas de 2 interrupciones externas y contar con comunicación SPI. En el mercado existen una gran variedad de microcontroladores, con diversas características, incluidas las que son necesarias para la etapa en cuestión, además de otras funciones que elevan los precios y que para dicho propósito no son útiles. Para cumplir con las características antes mencionadas utilizamos los microcontroladores de 8 bits de la familia Microchip, pero primero se va hacer una comparativa con otros modelos para identificar las diferencias. Estas diferencias se exponen en la Tabla 2.19 para tres modelos de microcontroladores diferentes.

La comparación entre microcontroladores de diferentes marcas, familias y modelos puede ser muy extensa ya que existe una gran variedad de éstos dispositivos digitales, pero solo un número limitado de éstos cumple con las características necesarias y suficientes para satisfacer los objetivos establecidos para el proyecto presente. El microcontrolador conveniente para dicha aplicación es el PIC18LF2331 de la marca Microchip. La ventaja de éste radica en su costo, en el nivel de voltaje con el que trabaja y además cumple con todas

Tabla 2.19: Microcontroladores

Modelo	Características
ATMEL ATME- GA328P	16MHz, 8bits, Comunicación SPI, PWM 16bits, 2 interrupciones, voltaje 5 [V]
Texas Instrumets TM4C1294NCPDT	120 MHz, 32bits, PWM 32bits, Encoder de Cuadratura bufer 32bits, voltaje 3.3 [V]
Microchip PIC18LF2331	20MHz, 8bits, Comunicación SPI, PWM 16bits, Encoder de cuadratura bufer 16bits, voltaje 5 [V]
Microchip PIC24EP128MC202	140MHz, 16bits, Comunicación SPI, PWM 16bits, Encoder de cuadratura búfer 32bits, voltaje 3.3 [V]

las condiciones necesarias establecidas, comparado con el PIC24EP128MC202, que al igual cumple con las condiciones establecidas y es mucho más rápido, pero tiene la desventaja de tener un costo mucho más elevado, trabajar a un nivel de voltaje lógico más bajo y eso conlleva implementar una etapa de conversión de nivel de voltaje extra entre la demás electrónica que trabaja a un nivel de voltaje lógico más alto, que se traduce en un gasto mayor y en posibles problemas entre el microcontrolador y los demás dispositivos causado por el convertidor de nivel de voltaje lógico.

Una vez seleccionado el dispositivo de adquisición y generación de señales digitales, se describe la manera en que se realiza la lectura de los encoders para determinar a partir de ellos la posición angular de las articulaciones y también la manera en que se generan el voltaje basado en una señal PWM para el control de los motores.

Un encoder incremental es un sensor electrónico digital que genera una señal en forma de tren de pulsos, que se utiliza para interpretar cuanto ha girado el rotor de un motor. En particular, un encoder de cuadratura nos permite determinar en qué dirección ha girado y en qué momento se completó una vuelta. El encoder de cuadratura generalmente cuenta con tres salidas físicas, salida A, B e I, donde A y B son dos señales eléctricas en forma de tren de pulsos entre las que existe un desfase de 90° , de aquí viene el término encoder de cuadratura. En otras palabras, el periodo de la señal de tren de pulsos equivale a 360° , por lo tanto, si lo dividimos entre 4 nos da un valor de 90° . La tercera señal es el índice,

que tiene la característica de enviar un solo pulso cuando el motor completó una vuelta. La máxima cantidad de pulsos se conoce como la resolución del encoder y nos indica cuantos pulso se generan en una vuelta. En el diagrama de la Figura 2.18 se ilustra la descripción.

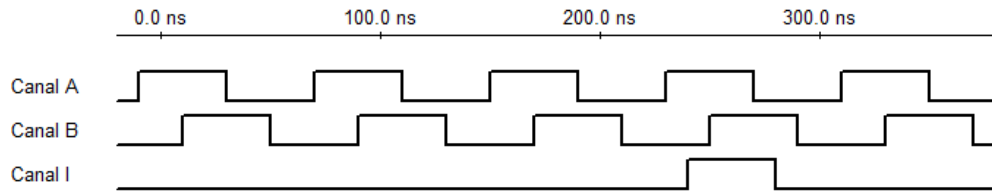


Figura 2.18: Señales de un Encoder de Cuadratura

Ya que la resolución del encoder está determinada desde el momento en que se fabrica, no es posible modificarla, por lo que si queremos una mayor resolución será necesario realizar este incremento desde el microcontrolador, considerando el diagrama de cuentas mostrado en la Figura 2.19.

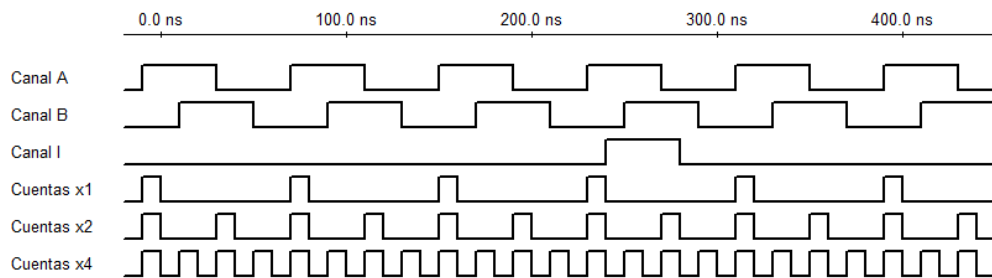


Figura 2.19: Diagrama de Cuentas

Como podemos observar en la Figura 2.19 el conteo se puede realizar de tres maneras diferentes. La primera solo consta en medir el cambio de estado en subida de la señal del canal A, asociando la cuenta solo con la resolución del encoder, donde se puede decir que está se multiplica por 1. En la segunda se miden los cambios de estado de subida y bajada del canal A incrementando al doble la resolución del encoder. En la tercera además de medir el canal A como se indica en la segunda opción se agrega la lectura del cambio de estado en subida y bajada del canal B, generando un par de cuentas adicionales incrementado 4 veces la resolución.

Para implementarlo en el manipulador debemos hacer referencia a tres conceptos

importantes.

- Cuentas por revolución o CPR
- Relación de reducción o RC
- Multiplicador de cuentas x1, x2, x4

El CPR es el número máximo de pulsos que entrega el encoder de fábrica, el RC es la relación de conversión de la caja del reductor y el multiplicador dependerá de la programación o las limitaciones del microcontrolador. Para el caso del microcontrolador seleccionado el PIC18LF2331 tiene como mínimo un multiplicador igual a 2, lo que quiere decir que como mínimo tendremos el doble de las cuentas generadas por el encoder físico. Afortunadamente el mismo cuenta con una interfaz de lectura exclusivamente para encoders de cuadratura incrementales. Este entrega, en un registro de 16 bits, el valor de la posición del encoder en términos de cuentas, la dirección de giro en cada instante y cuando terminó una vuelta. Con todas estas características la lectura del encoder se efectúa de manera sencilla. A continuación se muestra el procedimiento de programación del microcontrolador para efectuar la adquisición del valor de los encoders. Primero se determina la máxima resolución del encoder considerando todo lo anterior, lo cual se calcula de la siguiente manera:

$$MaxCounts = CPR \times RC \times Multiplicador \quad (2.9)$$

Donde el CPR es de 512 pulsos, la relación de conversión para cualquiera de los dos motores establecidos en la Sección 2.3.1 es de 103:1 y si consideramos que el multiplicador es de 2 tenemos que la máxima cuenta es:

$$MaxCounts = 512 \times 103 \times 2 = 105472 \quad (2.10)$$

Si el tamaño de palabra donde se almacena el valor de la lectura del encoder es de 16 bits tenemos que el máximo valor que podemos almacenar es 65536, lo que significa que la máxima resolución del encoder del motor no cabe en 16 bits, por lo que se tendrá que utilizar algún algoritmo que permita crear una variable de por lo menos 24 bits a partir de las cuentas almacenadas en la variable de 16 bits. Con ayuda de las variables que entrega la interfaz de encoder del microcontrolador, se muestra el procedimiento en los diagramas de flujo de las Figuras 2.20 y 2.21.

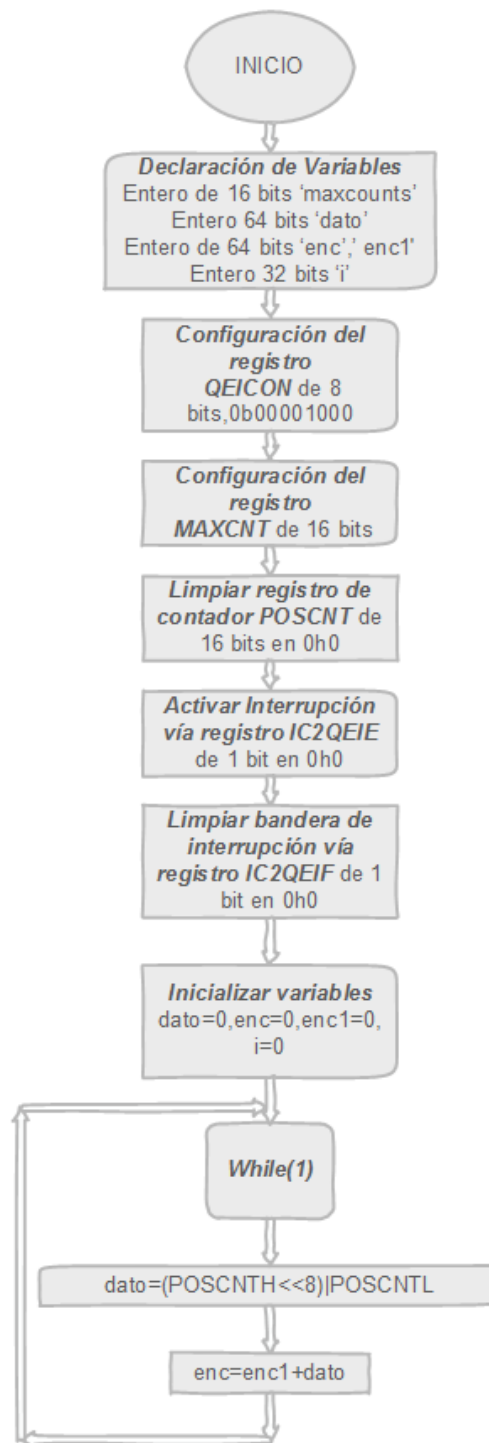


Figura 2.20: Lectura de Encoders (Función Principal)

El PWM tiene la característica de ser una señal de voltaje en forma de tren de pulsos

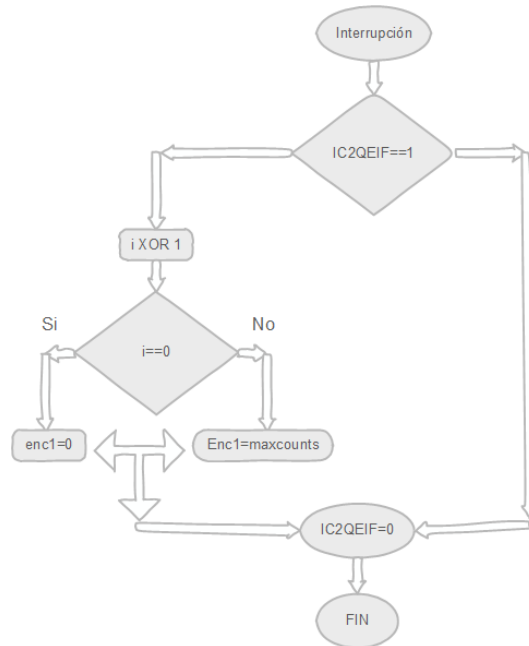


Figura 2.21: Lectura de Encoders (Función de Interrupción)

rectangulares, donde solo podemos tener un valor máximo o uno mínimo. La particularidad de este tipo de señales es el control que se tiene en su ciclo de trabajo, que se define como el porcentaje del tiempo que dura en su máximo valor con respecto al periodo de la señal. El control en el ciclo de trabajo permite al mismo tiempo controlar la potencia promedio que entrega la señal, sin la necesidad de variar el nivel de voltaje de ésta, lo que la hace más sencilla de generar con un microcontrolador, en especial cuando no cuenta con un convertidor digital analógico o en el caso donde no es posible agregar uno de manera externa. En la Figura 2.22 se muestran tres diferentes señales de PWM, a 25 %, 50 % y 75 % de ciclo de trabajo. El ciclo de trabajo se calcula de la siguiente manera.

$$\%D = \frac{TH}{T} \quad (2.11)$$

donde:

- %D es el ciclo de trabajo.
- TH es el tiempo en que la señal tiene el máximo valor.
- T es el periodo de la señal.

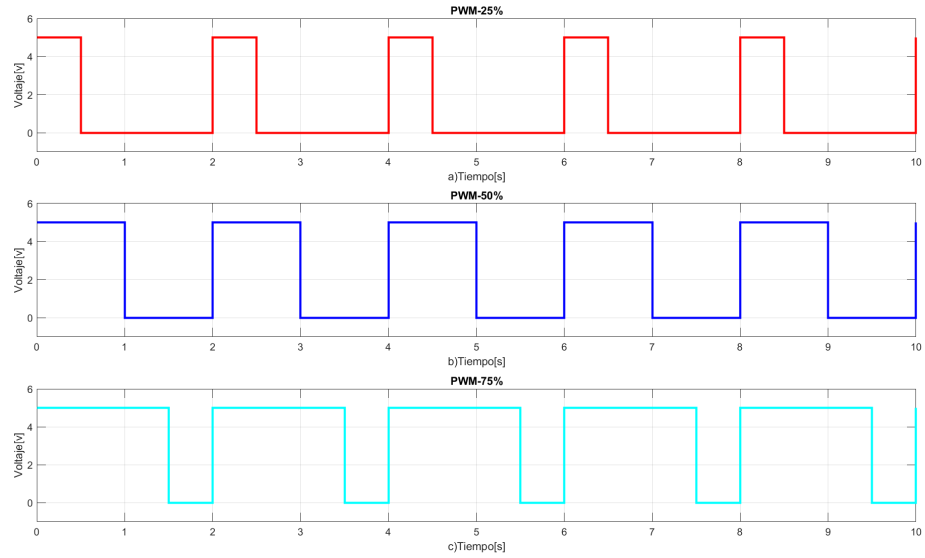


Figura 2.22: Señal PWM

Un microcontrolador tiene la capacidad de generar una señal PWM a diferentes ciclos de trabajo, esto depende de la resolución que proporciona el módulo PWM del microcontrolador, la resolución se refiere a la capacidad de representar un ciclo de trabajo pequeño, es decir, si se tiene una resolución de 8 bits solo se dispone de 2^8 valores para representar del 0 % al 100 % de ciclo de trabajo. Por lo tanto el mínimo ciclo de trabajo representado sería de 0.39 %, valores menores a eso son imposibles de generar con esa resolución. El microcontrolador seleccionado en la Tabla 2.19 tiene una resolución de 16 bits para PWM, suficiente para tener un buen control sobre el ciclo de trabajo con cambios mínimos de 0.0015 %. En la Figura 2.22 se muestra el diagrama de flujo para la programación del PWM en el microcontrolador.

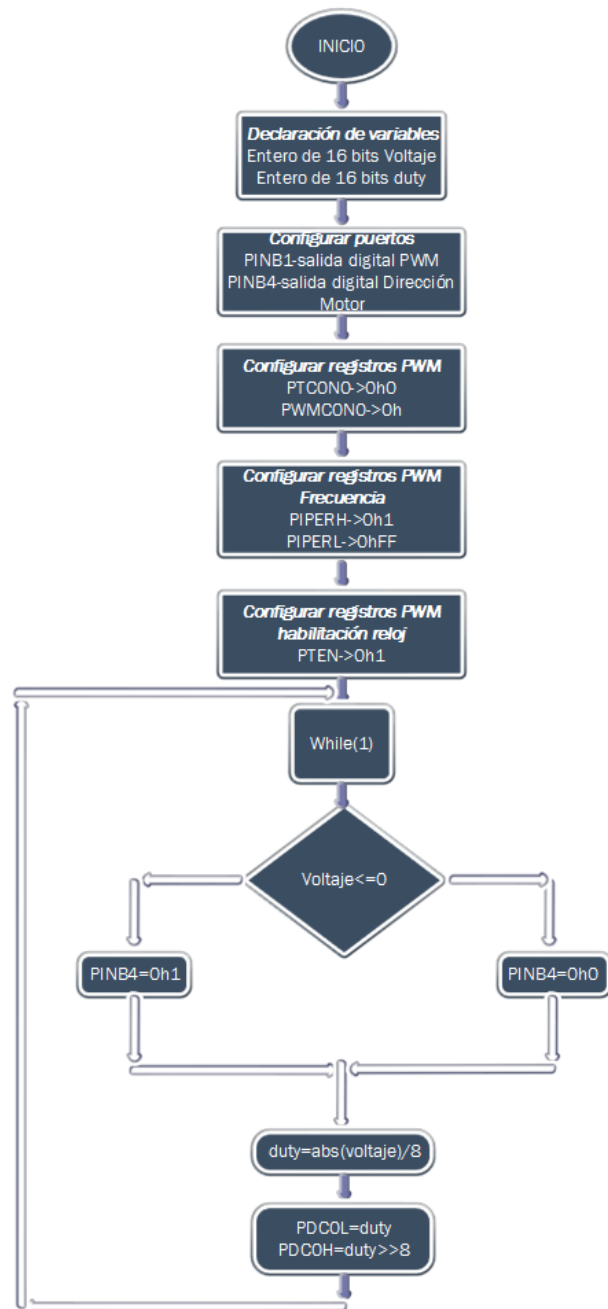


Figura 2.23: Generación PWM (Función Principal)

Etapa de Comunicación y Transmisión de Datos

Para iniciar el desarrollo de este punto, primero se comienza describiendo la manera en que la etapa de adquisición y control recibe y transmite la información mediante comunicación SPI al módulo principal de transmisión de datos entre la computadora y el robot. Dicho esto describiremos a grandes rasgos qué es y cómo funciona la comunicación SPI. Este bus de comunicación es de dos líneas de datos, síncrono, serie y además es maestro-multiesclavo. Tiene ventaja con respecto a la comunicación paralela ya que el número de hilos de comunicación es más reducido y compite contra los demás protocolos de comunicación por su velocidad y manejo.

Descripción de bus SPI. Esta comunicación se establece entre 4 hilos, MISO, MOSI, SCLK y SS, todas son líneas unidireccionales que se caracterizan por lo siguiente.

MOSI: Esta línea transmite los datos que envía el esclavo al maestro, cuando este esclavo no es seleccionado se pone en un estado de alta impedancia para evitar interferencias con los otros dispositivos.

MISO: Esta línea al igual que la otra transmite datos pero ahora lo hace del maestro al esclavo. Lo realiza justo en el momento adecuado para evitar interferencias con los demás esclavos.

SCLK: Es el bus de reloj que los dispositivos usan para sincronizarse y de esta manera transmitir la información de manera correcta. De éste se configuran dos parámetros que permiten trabajar con el bus SPI en 4 modos diferentes. Estos parámetros son la polaridad del reloj y la fase del reloj. Esta configuración debe ser igual para todos los dispositivos conectados en el bus SPI.

SS: También conocido como CS, es la línea de selección del esclavo, a diferencia del I^2C este no requiere de una dirección para determinar al esclavo con el que se va a entablar comunicación si no lo hace mediante una hilo que cambia de estado para determinar el dispositivo con el que se va a comunicar. En la Figura 2.24 se muestra la forma de conectar el bus SPI.

El protocolo de comunicación SPI solo permite la interacción entre el maestro y un esclavo, no se puede realizar comunicaciones múltiples de manera simultánea, por lo tanto, cualquier dispositivo conectado al bus SPI permanece en un estado que no le permita interferir. La manera de seleccionar al esclavo es activar en un nivel de '0' lógico a la línea SS del dispositivo esclavo, mientras las demás se encuentran en nivel lógico '1', con esto

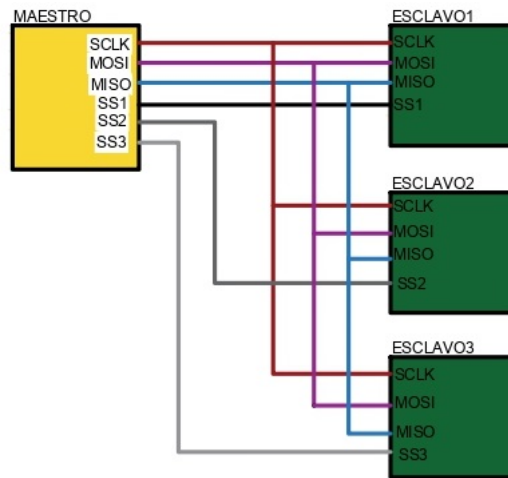


Figura 2.24: Diagrama de Conexiones SPI

aseguramos solo la comunicación con el dispositivo deseado. Cuando ocurre este cambio de nivel inicia la comunicación, transmitiendo y recibiendo datos de manera simultánea, esto ocurre debido a un registro interno en donde al momento de recibir los bits del maestro se transmiten los bits del esclavo. Esta información se transmite a velocidades alrededor de los MHz, para fines prácticos se utilizó una velocidad de 12 [MHz] que es capaz de transmitir 3 [Mbps], es decir, por cada ciclo de reloj se envía un bit.

En la Figura 2.25 se muestra el diagrama de tiempo de la comunicación SPI de 4 hilos.

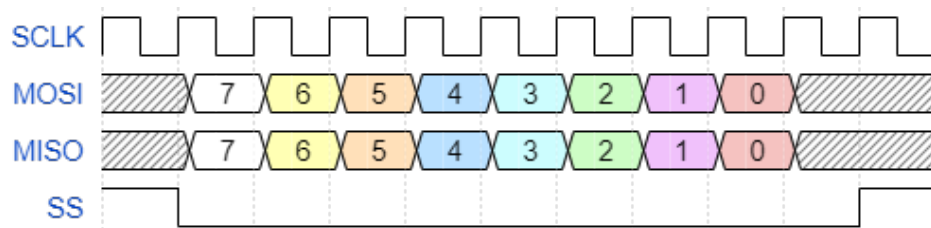


Figura 2.25: Diagrama de Tiempo de la Comunicación SPI

Para transmitir los datos obtenidos de los módulos de control y adquisición, se necesita un intermediario que interprete los datos de bus SPI y los pueda transmitir por medio de comunicación USB. Este dispositivo intermediario es un PIC18F4550 de la marca Microchip, que permite realizar esta acción de transmitir información mediante USB. Esta es la manera convencional, en la actualidad, de recibir y enviar información, por lo que no se requiere de un hardware especial o una computadora con características específicas para su utilización. EL PIC18F4550 trabaja a una velocidad máxima de 48MHz, cuenta con protocolo de comunicación SPI, e interfaz USB, entre otras. Mediante la interfaz USB es capaz de transmitir 64 bytes de datos por cada milisegundo en esta implementación.

El microcontrolador PIC18F4550 se monta sobre una placa base modelo JT45k50-v1.0 de la marca JElectronica. Esta placa cuenta con la electrónica necesaria para tener acceso a sus periféricos, a los pines de programación y a un conector USB tipo B.

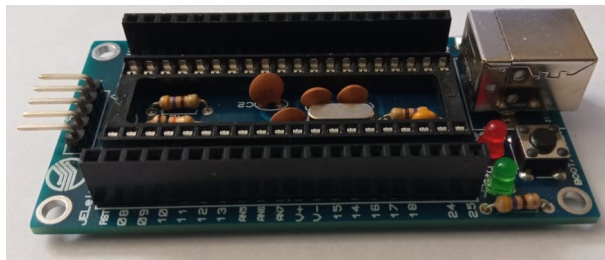


Figura 2.26: Placa JT45k50-v1.0

Con ayuda de los registros de la comunicación SPI de cada uno de los microcontroladores se realizó el diagrama de flujo mostrado en la Figura 2.27 correspondiente a la interacción entre los dos dispositivos. El dispositivo maestro (PIC18F4550) envía datos de voltaje que son recibidos por el esclavo (PIC18F2331) y al mismo tiempo el esclavo (PIC18F2331) envía datos de la posición del encoder de los motores para que el maestro (PIC18F4550) los reciba. Para el PIC18F4550 el envío de datos por USB se realiza en arreglos de 64bytes. Ya que la comunicación SPI en el microcontrolador PIC18F2331 se hace mediante una interrupción, si debe iniciar con la función principal en el diagrama de flujo, en el caso del PIC18F4550 se manda a llamar la función correspondiente a la comunicación USB. Los diagramas de flujo se muestran en las Figuras 2.27, 2.28, 2.29 y 2.30.



Figura 2.27: Diagrama de Flujo SPI Función Principal Esclavo

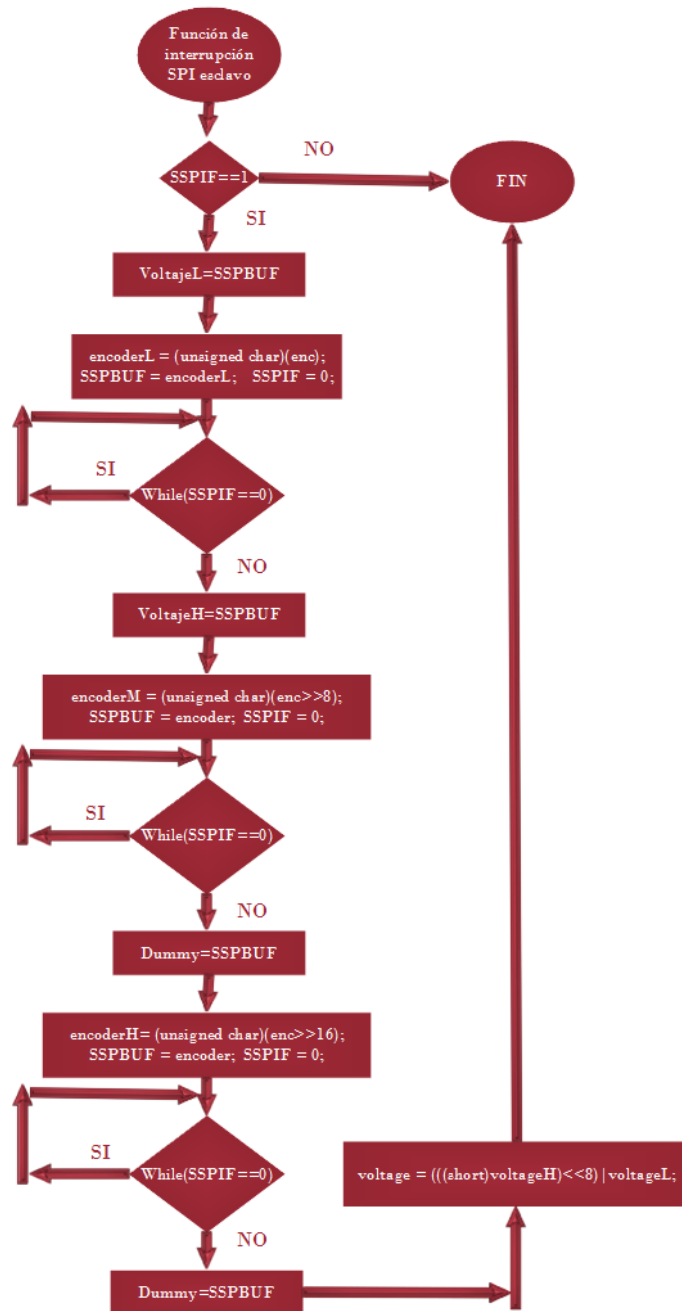


Figura 2.28: Diagrama de Flujo SPI Función Interrupción Esclavo

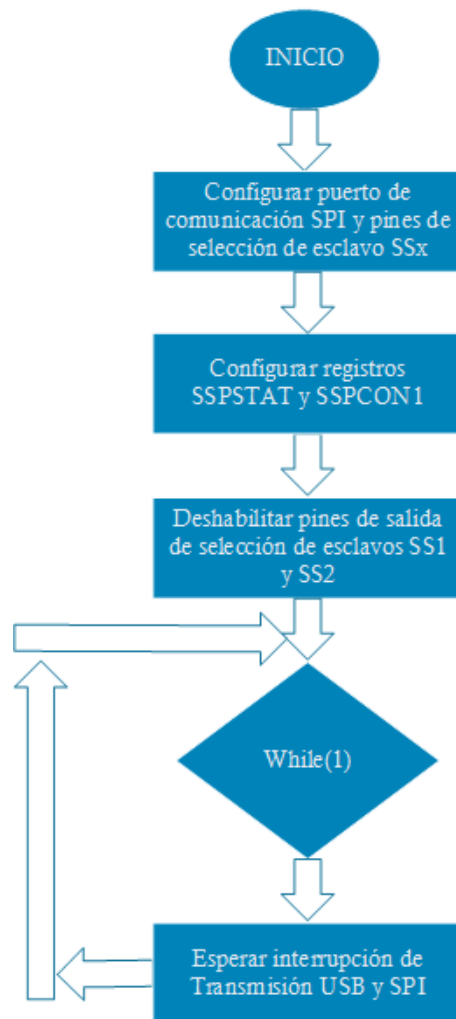


Figura 2.29: Diagrama de Flujo SPI Función Principal Maestro

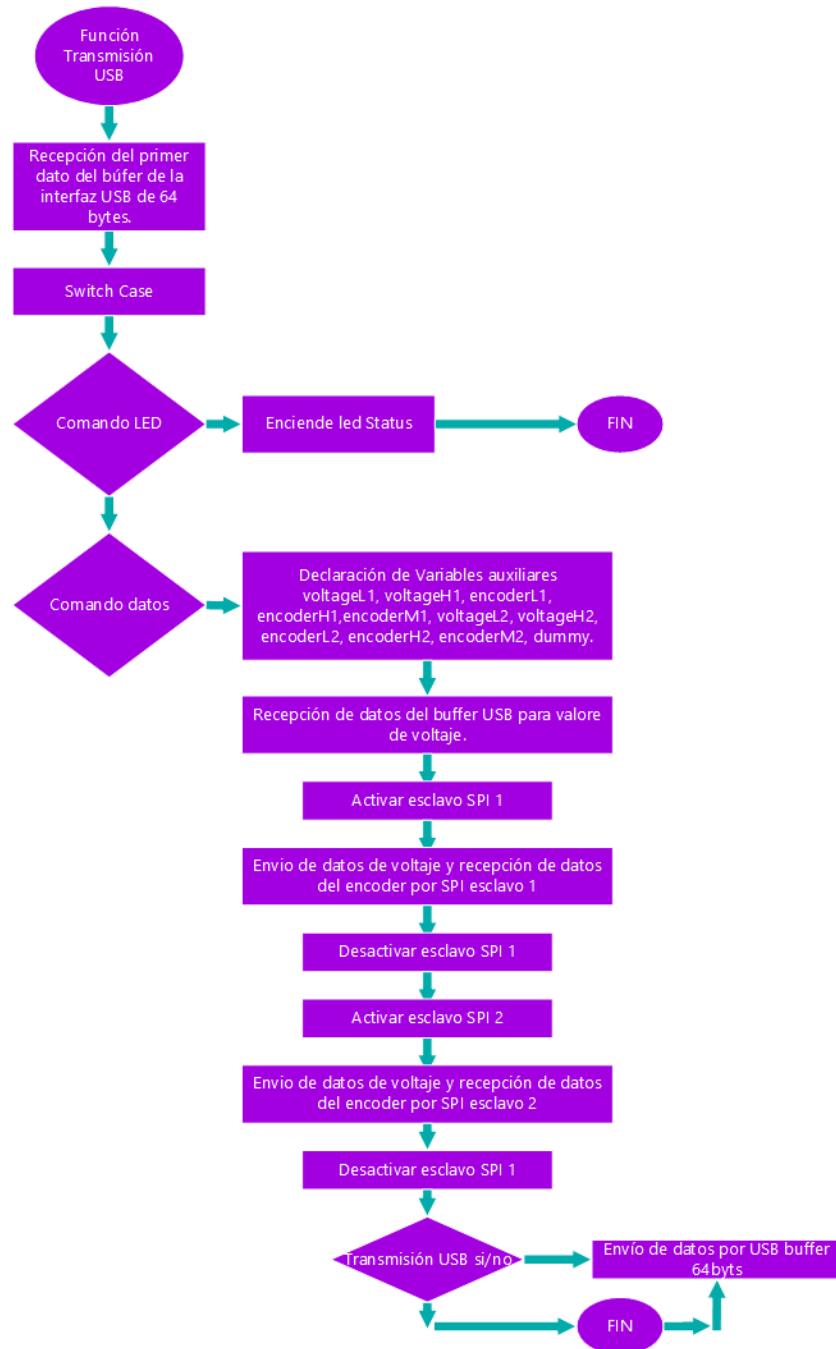


Figura 2.30: Diagrama de Flujo SPI Función USB Maestro

Los bloques correspondientes al envío de información por la interfaz USB, son parte del programa correspondiente a la comunicación USB, el programa utilizado es de la autoría de Microchip, que es citado en la bibliografía de este documento para su consulta, del cual solo se modificaron unas líneas correspondientes a la información transmitida.

El código completo correspondiente a los dos microcontroladores se encuentra en el Anexo A-Código de Microcontroladores PICs.

Tarjetas Comerciales Dedicadas a la Adquisición y Generación de señales

Una manera de mostrar las cualidades que presenta el módulo de control, adquisición y comunicación es justo cuando se compara con otros modelos diseñados por marcas que compiten en el mercado, como National Instruments, Quanser, Maxon Group, Analog Devices, Texas Instruments, Data Translation, entre otras. A continuación se muestra un cuadro comparativo entre los modelos, características y costo, de tarjetas que cumplen el mismo propósito del diseño propuesto en este trabajo, con la finalidad no de competir sino de mostrar una alternativa mas económica, que permita realizar actividades experimentales.

Tabla 2.20: Tarjetas de adquisición y generación de señales

Módulo	Caractrísticas	Costo
NI cRIO-9053	Velocidad 1.33Ghz, Conectividad Ethernet y USB, slots para módulos I/O Analógicos, I/O Digitales.	\$43,575 MXN
Q8-USB	I/O Analógicos, I/O Digitales, 8 Encoders canal A y B, 8 PWM a 2.5[v] con 16bits	\$70,000 MXN
MG Mini-MACS	6 Encoders canal A y B, I/O Analógicas, I/O Digitales, interfaz de comunicación USB, CAN y Ethernet	Costo por pedido
DT9804 USB	Velocidad de muestreo 10kS/s, I/O Analógicos 16 bits, I/O Digitales, interfaz USB	\$26,121 MXN
USBServo v1.0 modelo propuesto	Velocidad 20MHz, Encoder canal A y B, PWM a 12V con 16 bits, comunicación SPI y USB	\$2,408 MXN

2.4.2. Esquemático Módulo de Control y Adquisición, Diagrama de Bloques de Comunicación

El diagrama de comunicaciones muestra mediante bloques en cascada el proceso de comunicación entre todos los dispositivos físicos pertenecientes al manipulador robótico de 2 grados de libertad. En color naranja (PC) tenemos la interfaz de comunicación USB. En color azul marino (PIC18F4550) tenemos el intermediario entre lo que transmiten los módulos de control y adquisición y la computadora. Los bloques en color gris (PIC18F2331) y color morado (L298N) pertenecen a la etapa de control y adquisición. Los bloques de color negro hacen referencia los motores de las articulaciones. En color naranja se tiene a los encoders. En color rosa se hace referencia a la conexión mecánica entre los eslabones y los encoders. Por último, tenemos en color azul cielo al bus de información que se recibe las señales de los encoders y el bus de color guinda que hace referencia a la comunicación SPI. Lo antes descrito se muestra en la Figura 2.31.

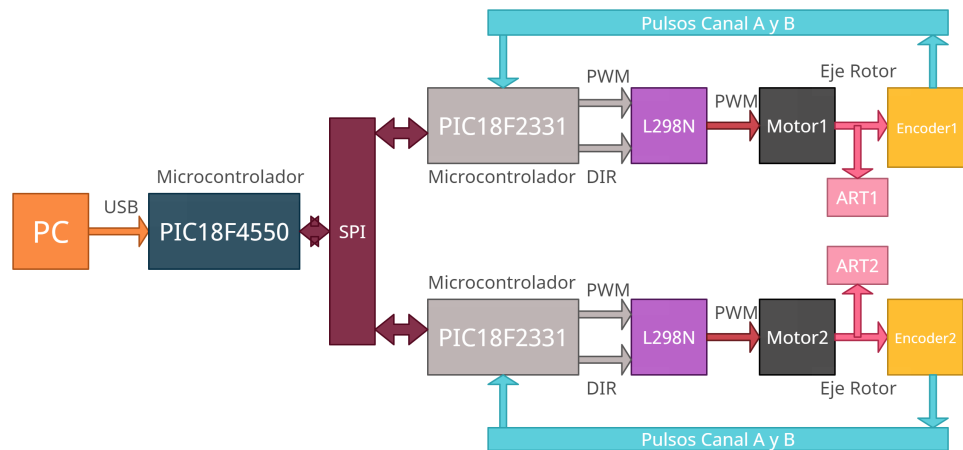


Figura 2.31: Diagrama de Comunicación

El esquemático de conexiones muestra los elementos necesarios para que se pueda utilizar el microcontrolador, así como las entradas y salidas para emitir las señales de control, de adquisición y de comunicación. Estas señales de entrada y salida al módulo son: el bus proveniente de los encoders, las líneas de control para los motores, el bus SPI y la fuente de alimentación del circuito. En el esquema de la Figura 2.32 se muestra lo anterior.

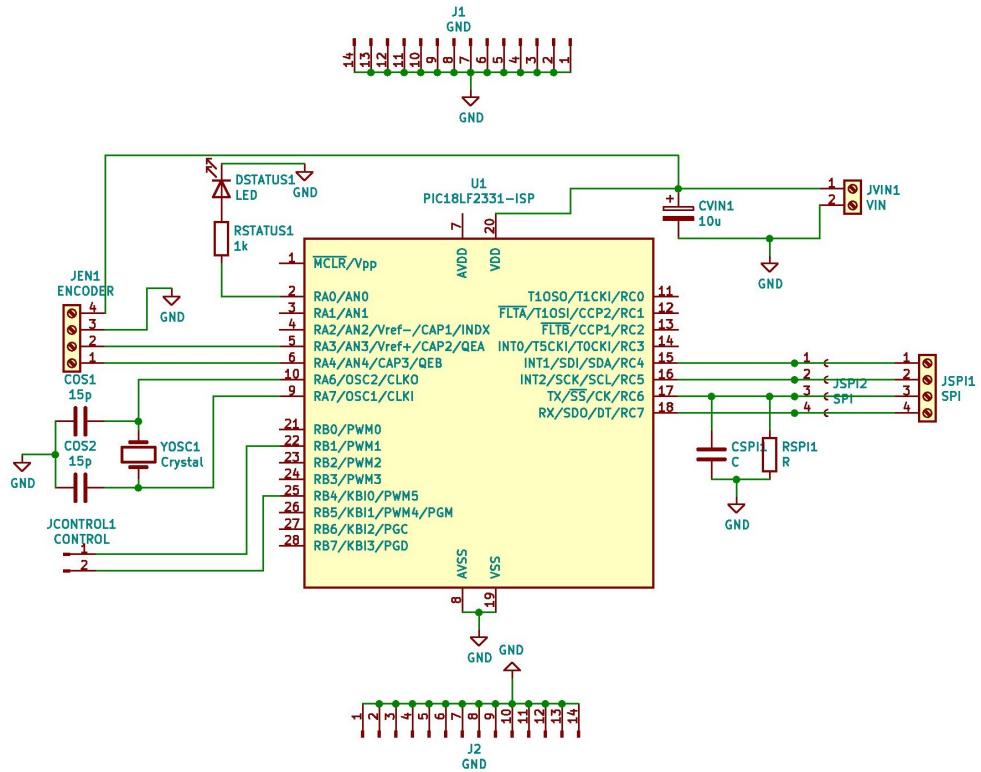


Figura 2.32: Esquemático Módulo de Control y Adquisición

El esquemático de la Figura 2.33 y le CAD de la Figura 2.34 pertenece al conector entre el bus de datos del modulo de control y adquisición y la computadora.

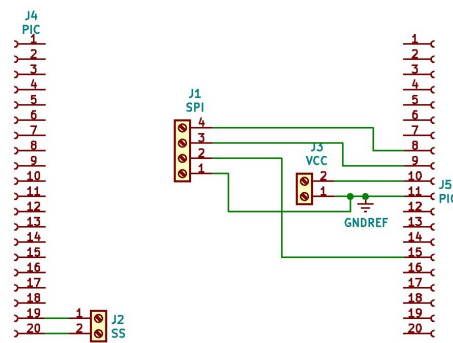


Figura 2.33: Esquemático Conector con Placa JT45k50-v1.0

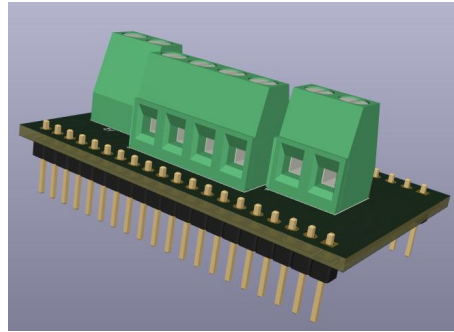


Figura 2.34: CAD Conector con Placa JT45k50-v1.0

2.4.3. Costos de Fabricación

Para definir los costos correspondientes a la etapa de adquisición, control y comunicación, se deben enumerar los elementos que constituyen a esta sección, como lo son microcontroladores, placas PCB y electrónica complementaria. De esta manera los cálculos referentes al costo se determinan en la Tabla 2.21.

Tabla 2.21: Costos módulo de monitrol, adquisición y comunicación

Componente	Cantidad	Costo individual	Costo General
PIC18F2331	2	\$393 MXN	\$789 MXN
PIC18F4550	1	\$410 MXN	\$410 MXN
Cristal 20MHz	2	\$10 MXN	\$20 MXN
Placas PCB	3	\$50 MXN	\$150MXN
Placa JT45k50-v1.0	1	\$254 MXN	\$254 MXN
Cable calibre 20	7.5m	\$5 MXN	\$37.5 MXN
Terminales 2x1	14	\$4.31 MXN	\$60.34 MXN
LEDS	2	\$2 MXN	\$4 MXN
Heder Macho	4	\$8.62 MXN	\$34.48 MXN
Reistencias	4	\$0.5 MXN	\$ 2MXN
Capacitores	8	\$2 MXN	\$ 16MXN

El costo final de la etapa de control, adquisición y comunicación es de \$1,777.32MXN, como se muestra en la Tabla 2.21. Los datos de los costos se obtuvieron de la página del

fabricante o de tiendas de electrónica en línea..

2.5. Ensamblaje y Conexiones Eléctricas del Manipulador

En esta sección se muestra, mediante una imagen y diagramas, cómo se ensamblan los elementos mecánicos y las conexiones eléctricas de cada una de las etapas del manipulador.

2.5.1. Montaje de Articulaciones y Eslabones.

Las articulaciones, motores, se acoplan a los eslabones, barras de aluminio, mediante acoplamientos de brida, y a su vez las articulaciones se sujetan a sus respectivas bases y soportes. En los diagramas siguientes se describe esto mediante imágenes realizadas en programas de dibujo.

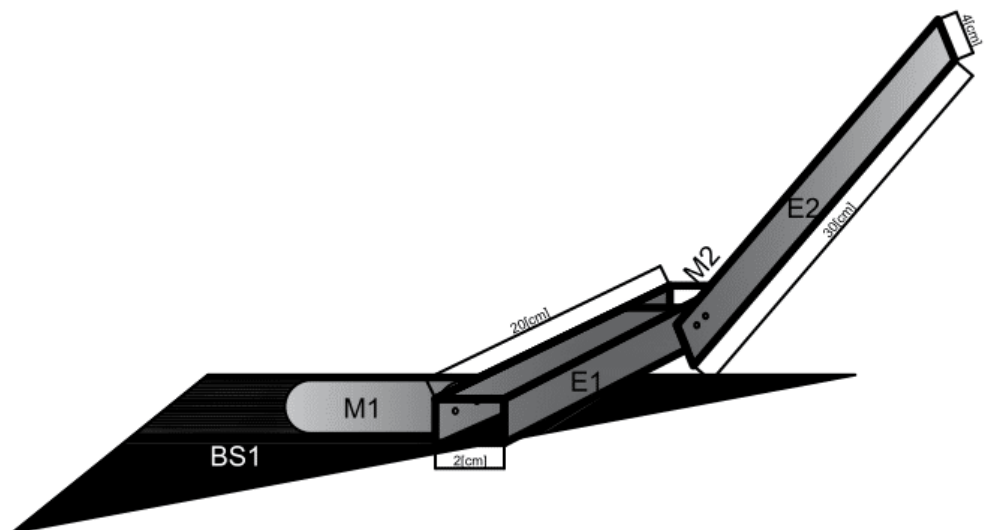


Figura 2.35: Estructura Completa

donde:

- E1 es el eslabón 1.
- E2 es el eslabón 2.
- M1 es la articulación 1.

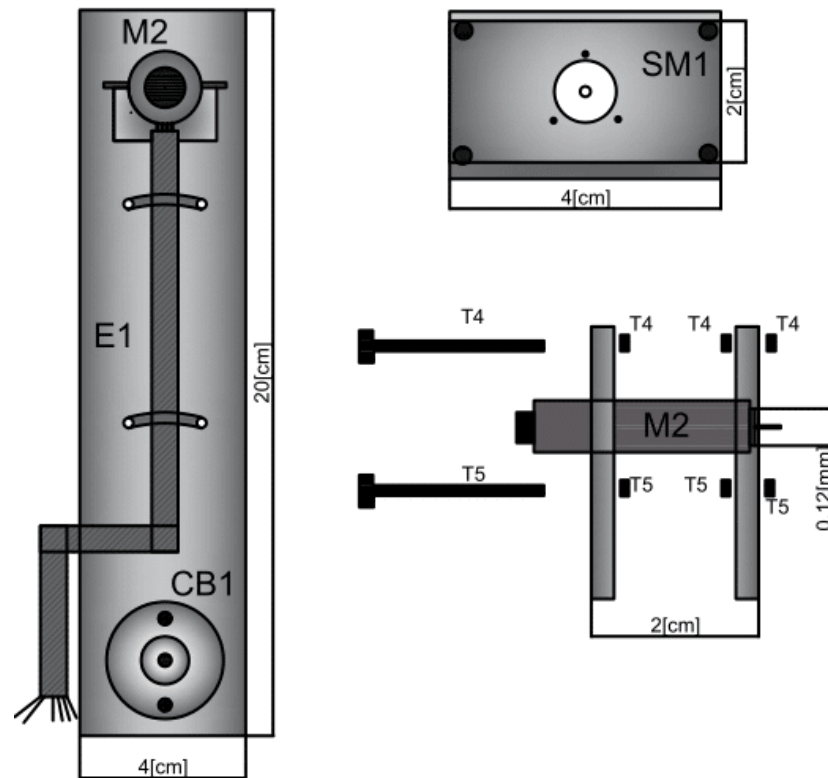


Figura 2.37: Articulación 2 y Eslabón 1

donde:

- E1 es el eslabón 1.
- M2 es la articulación 2.
- SM1 soporte del motor 2.
- CB1 cople de brida del motor 1.
- T4 y T5 son tornillos con tuercas de ajuste.

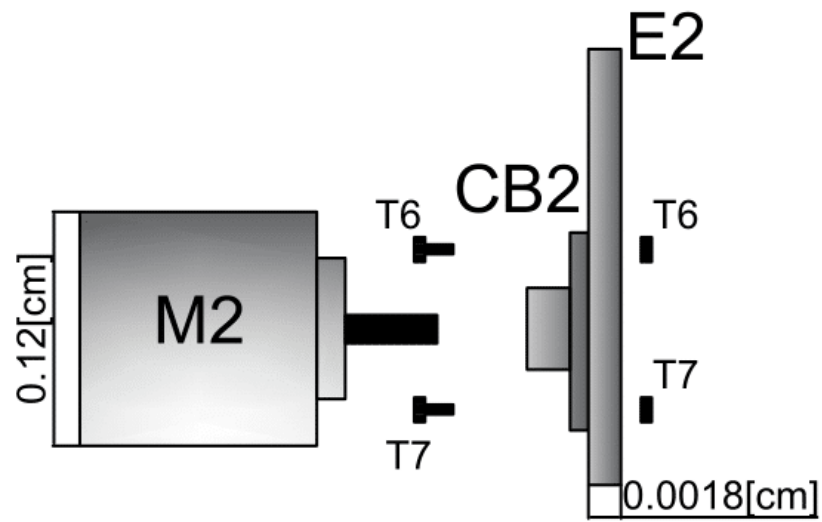


Figura 2.38: Articulación 2 y Eslabón 2

donde:

- E2 es el eslabón 2.
- M2 es la articulación 2.
- CB2 cople de brida del motor 2
- T6 y T7 son tornillos con tuercas de ajuste.

2.5.2. Conexiones y Cableados

En el diagrama de la Figura 2.39, muestran la manera en que se deben realizar las conexiones entre cada una de las secciones de la plataforma. Con distintos colores se diferencian las conexiones y las etiquetas indican la funcionalidad de las mismas. Cada bloque hace referencia mediante una etiqueta a la función que realiza, lo que resulta en una manera práctica de realizar algún ajuste si es necesario, evitando conexiones erróneas, que causarían daños por flujos invertidos de corriente o errores de comunicación.

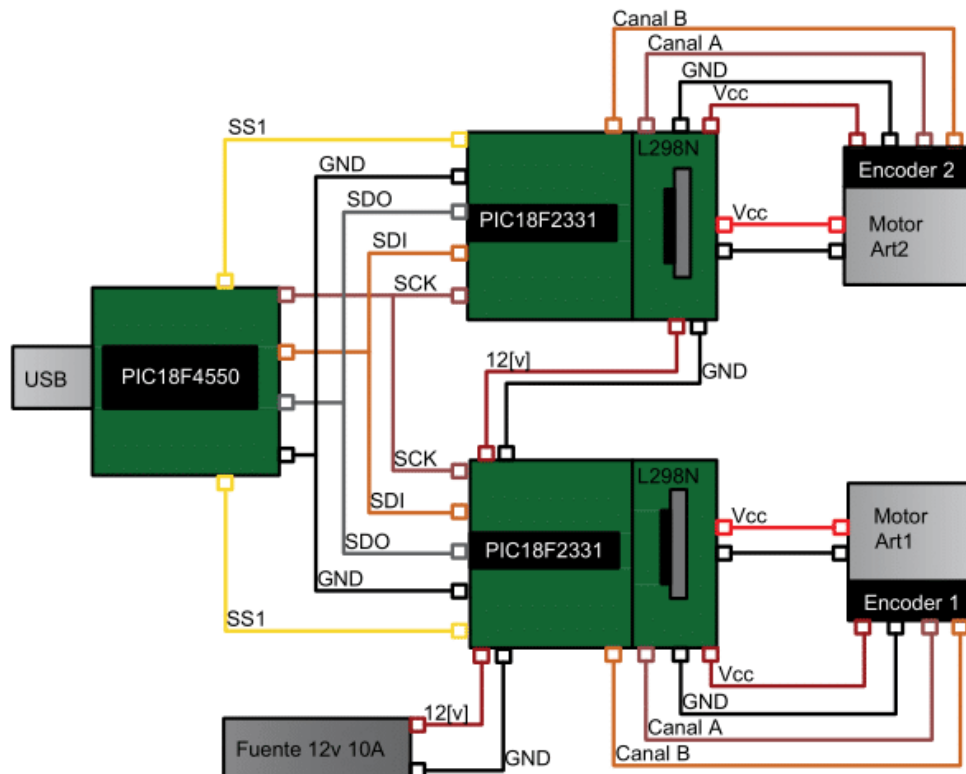


Figura 2.39: Diagrama de Conexiones y Cableado

Donde:

- SS1 y SS2 línea de selección de esclavo.
- SDO línea de datos del esclavo al maestro.
- SDI línea de datos del maestro al esclavo.
- SCK línea de reloj de sincronización.

- Canal A y Canal B líneas de lectura de los encoders.
- 12[V], Vcc y GND líneas de alimentación del circuito.

2.6. Programación, Calibración y Acondicionamiento

El microcontrolador por sí mismo no es capaz de dar una interpretación física a los datos que se generan o que se adquieren, por lo tanto se debe realizar algún acondicionamiento de las señales con las cuales se está trabajando, por ejemplo las cuentas del encoder se deben interpretar como ángulos o radianes, el ciclo de trabajo de la señal PWM se debe de interpretar como el par que entrega el motor, interpretación física que se necesita para implementar las técnicas de control establecidas. Otro aspecto importante es determinar si lo que se está leyendo es correcto o incorrecto. Esto depende de las condiciones iniciales que se le asignan al manipulador antes de iniciar los microcontroladores. También depende de los datos que se reciben están completos, si hay pérdida de información o son datos incoherentes, todo esto se describe a continuación.

2.6.1. Software de Adquisición y Control

En el objetivo del proyecto se plantea que la plataforma tiene que ser flexible al usuario para la utilización de la plataforma y la implementación de las técnicas de control, parte de esa flexibilidad se le atribuye a la interfaz de control que se diseñó para utilizar el manipulador, esta interfaz está diseñada mediante el software Visual Studio, como se describe al inicio de este capítulo, Visual es una plataforma de programación que es utilizada para crear y diseñar aplicaciones o interfaces capaces de ejecutarse en varios sistemas operativos, principalmente Windows. Con este software podemos implementar las diferentes técnicas de control, mediante programación, propuestas en el Capítulo 4. Pero para ello primero se tiene que describir la manera en que se adquiere, se interpreta y cómo se acondicionan los datos de las señales que se adquieren (posición) y las que se generan (control).

Los datos se transmiten a través de la interfaz USB, como se menciono anteriormente. Para recibirlos es necesario crear una función programada en Visual, que lea el búfer de 64 bytes provenientes del microcontrolador PIC18F4550, que son datos de posición de las dos articulaciones. Cada dato de posición, está constituido por tres secciones de 8 bits, que forman un dato de tamaño de palabra de 24 bits, 24 bits para el encoder 1 y 24 bits para

el encoder 2. En la misma función, además de recibir la información del encoder, se envía el valor del ciclo de trabajo del PWM de la señal de control, que es un dato de 16 bits que se divide en dos fragmentos de 8 bits para su envío. En el diagrama de flujo de la Figura 2.40 se muestra este proceso.

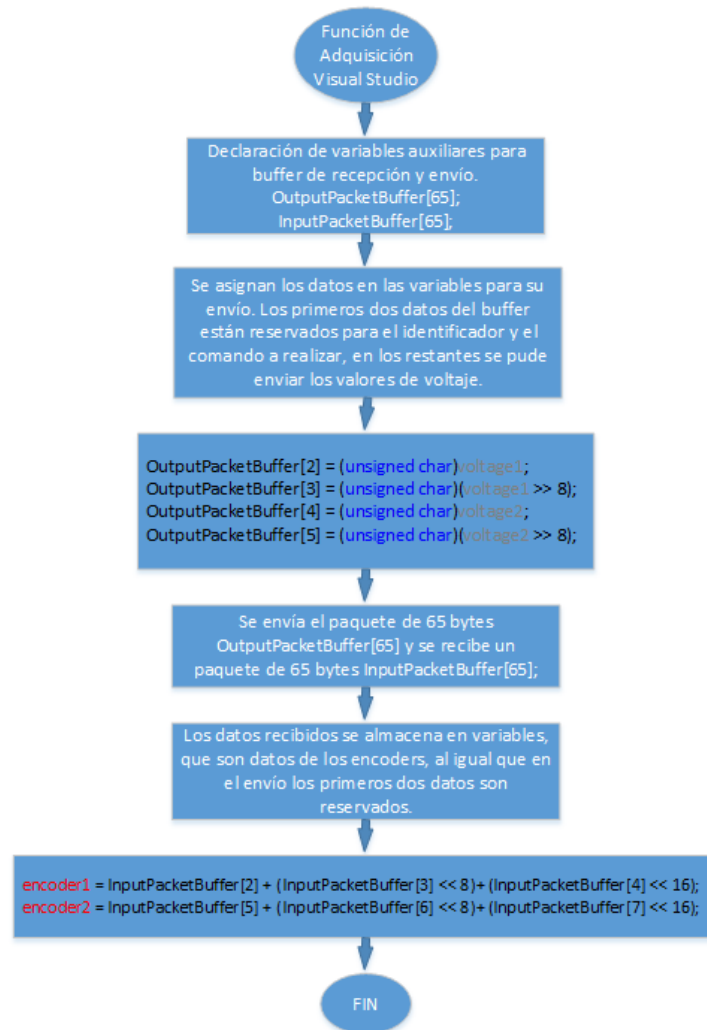


Figura 2.40: Diagrama de Flujo Adquisición y Control Visual Studio

2.6.2. Acondicionamiento y Programación de Algoritmos de Control

Antes de proceder a la programación de algún controlador, es necesario verificar que la interpretación física para posición y par sean correctas, es decir, convertir los datos de cuentas del encoder a radianes o grados y el ciclo de trabajo del PWM a par. Para

realizar la conversión se necesita verificar en las hojas de especificaciones de los encoders y de los motores los valores nominales del máximo número de cuentas y del máximo par que entrega cada motor, así como el valor decimal correspondiente al 100 % del ciclo de trabajo del PWM. En la Tabla 2.22 se desglosa esta información.

Tabla 2.22: Datos nominales

Elemento	valores nominales
PIC18F2331	máximo ciclo de trabajo PWM, numérico entero 17,000
Encoders	número de cuentas por revolución 105,472 MXN
Motor 1	par máximo 1.133 [Nm]
Motor 2	par máximo 0.1947 [Nm]

Con esta información procedemos a realizar la conversión de los datos del encoders a radianes y del ciclo de trabajo a par. Para la conversión basta con realizar una regla de 3 y además considerar que la dirección de giro visto de frente debe de ser en sentido contrario a las manecillas del reloj.

Posición

$$\theta_1 = -\frac{Cuentas1}{105,472} \left(2\pi - \frac{\pi}{2}\right) \quad (2.12)$$

$$\theta_2 = -\frac{Cuentas2}{105,472} (2\pi) \quad (2.13)$$

Par

$$torque1 = -\frac{Torque1}{1.133} (17,000) \quad (2.14)$$

$$torque2 = -\frac{Torque2}{0.1947} (17,000) \quad (2.15)$$

A la posición angular 1 se le restan 90° por la condición de calibración, el signo menos es para cambiar el sentido de giro del eje z. Una condición extra que se agregó es que la posición angular del encoder no se considere en el intervalo entre 0° y 360° sino entre -180° a 0 a 180. De esta manera se define la dirección de movimiento de las articulaciones, dirección

positiva y dirección negativa. Además de las condiciones anteriores hay otro aspectos que deben de considerarse al momento de aplicar el control y eso es la zona muerta de los motores, que no es otra cosa más que el valor mínimo de ciclo de trabajo en el que el motor empieza a moverse, en el sentido contrario también se debe de considerar un valor máximo de ciclo de trabajo para evitar movimientos agresivos que puedan dañar las partes mecánicas del manipulador. Esta acotación de valores entre un máximo y un mínimo se le conoce como saturación y también se debe de programar al momento de acondicionar las señales. El diagrama de flujo de la Figura 2.41 muestra la solución al acondicionamiento de señales que se programó en Visual.

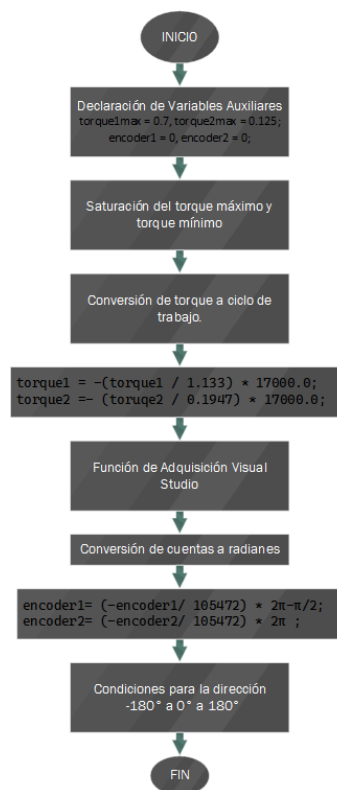


Figura 2.41: Diagrama de Flujo, Acondicionamiento de Señales

Una vez definidas las unidades físicas correspondientes se procede a programar los algoritmos de control, que se definen en el Capítulo 4. Para ello se programa una función donde se integran todos los controladores, cada controlador necesita el registro del tiempo, el valor de la posición angular de los motores, la velocidad y los valores deseados de posición, velocidad y aceleración. Estos valores deseados varían según lo que se requiera controlar

ya sea regulación o seguimiento. De igual manera para ciertos casos se necesita el error de posición, el error de velocidad y el error acumulado. En el diagrama de flujo de la Figura 2.42, 2.43 y 2.44 se muestra la manera en que se programa la función.

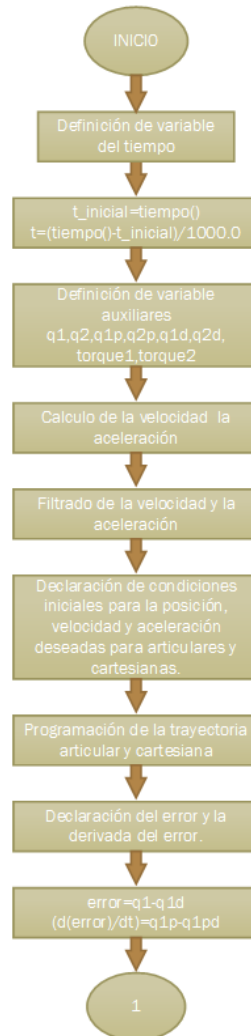


Figura 2.42: Diagrama de Flujo, Función de Control 1

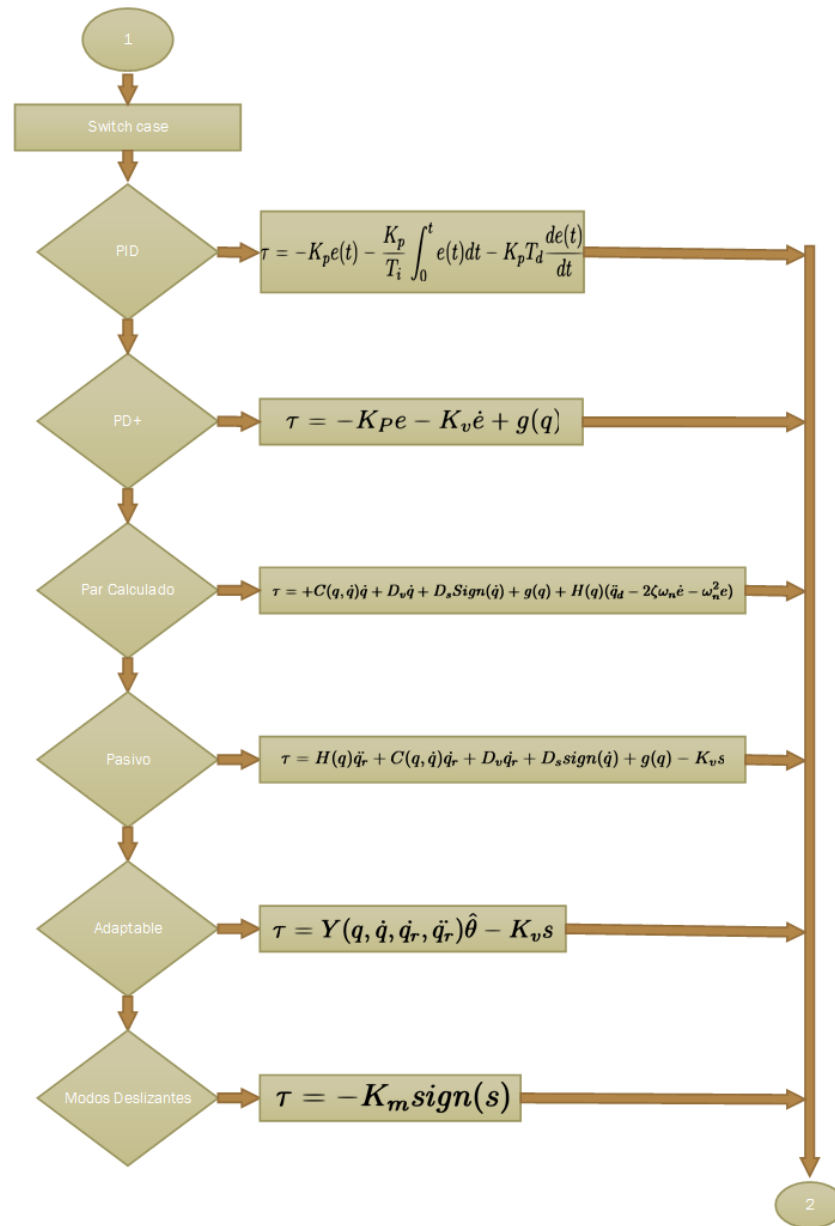


Figura 2.43: Diagrama de Flujo, Función de Control 2

El código completo correspondiente al programa en Visual Studio se encuentra en el Anexo B-Código de Visual Studio.



Figura 2.44: Diagrama de Flujo, Función de Control 3

2.6.3. Interfaz de Control

Con el programa completo se tiene una interfaz que permite conectarse con la plataforma del manipulador, la conexión, el test de la plataforma y selección de un controlador se realiza de la siguiente manera.

Conexión

Para conectar la interfaz con la plataforma solo basta con presionar el botón de *Connect*, una vez realizado esto se presiona el botón de *Toggle LED* para verificar que existe comunicación con la plataforma. En la sección de *Test motor* se puede enviar un valor de par y recibir la posición articular, para verificar el funcionamiento de los motores y encoders. En la Figura 2.45 se muestra la ubicación de dichas partes de la interfaz.

Selección de Un Controlador

Para seleccionar el controlador basta con ir a la sección de Algoritmo de Control. En la primera lista *Algoritmo de control* se elige el controlador de 6 opciones PID, PD+, Par Calculado, Pasivo, Adaptable y Modos Deslizantes. En la segunda lista *Control Regulación/Seguimiento* podemos definir si el control se va aplicar para regulación o seguimiento. En la tercera lista *Seguimiento Articulares/Cartesianas* tenemos dos opciones si el control por seguimiento se va a dar en una trayectoria Cartesiana o en una trayectoria articular. En la figura 2.46 se muestra esta sección.

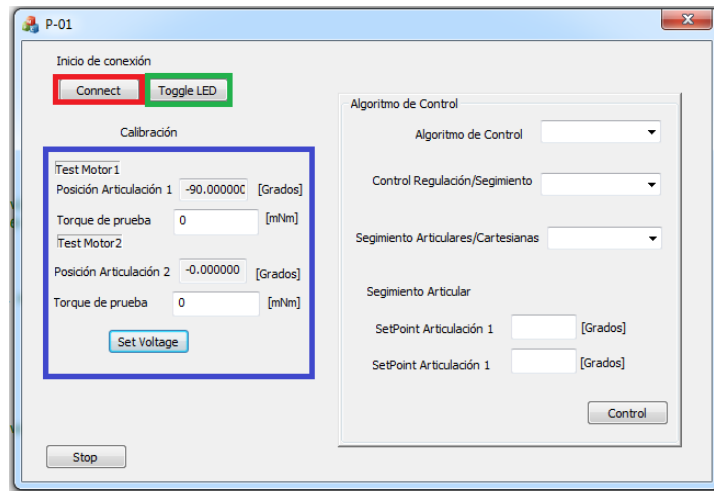


Figura 2.45: Sección de Comunicación y Prueba de la Interfaz Gráfica de Usuario

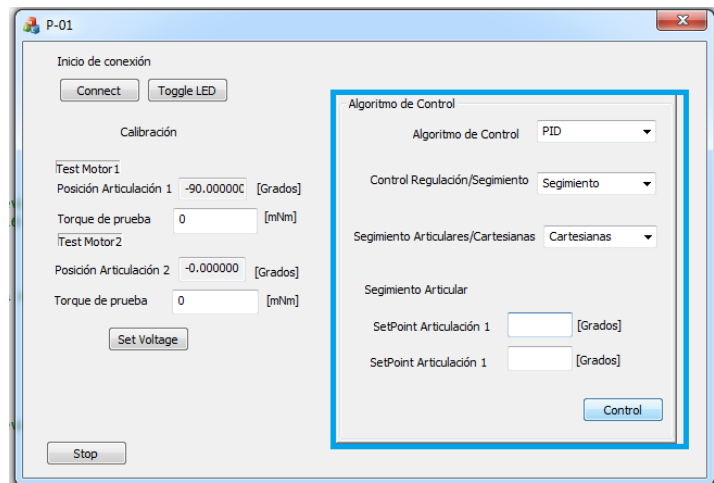


Figura 2.46: Sección de Elección de Algoritmo de Control

2.6.4. Calibración y Test de la Plataforma

Para verificar que los motores y los encoders estén funcionando de manera correcta, es necesario utilizar la sección de test motor 1 y test motor 2, de la interfaz del programa. En esta sección podemos indicar una valor de par que deseamos que entreguen los motores y también podemos leer la posición angular de las articulaciones, para confirmar su buen o mal funcionamiento y con ello tomar medidas para su corrección y calibración. En la Figura 2.47 se muestra la sección correspondiente en la interfaz que se utiliza para esto.

Calibración

Test Motor 1

Posición Articulación 1 [Grados]

Torque de prueba [mNm]

Test Motor 2

Posición Articulación 2 [Grados]

Torque de prueba [mNm]

Figura 2.47: Sección de Prueba de Encoders y de Motores

Como se logra observar en la interfaz test de motores, es posible leer la posición del encoder en grados, de la misma manera se puede enviar un valor de par y verificar que el motor inicie el movimiento si el par es suficiente para mover el robot..

Para realizar la calibración es necesario cerrar la interfaz de usuario, desconectar la plataforma de la conexión USB y apagar la fuente de alimentación, esperar unos segundos a que se consuma la energía almacenada en capacitores y con esto se procede a posicionar al manipulador en la posición de inicio, la que se puede modificar según las necesidades del usuario, pero para mayor comodidad se sugiere utilizar como posición de inicio lo descrito en la Figura 2.48 b).

Una vez realizado este proceso se conecta todo de nuevo y se vuelve abrir la interfaz de control, en sección test motores se verifica que los datos del encoder corresponda con la posición inicial, presionando el botón *Set Voltage*.

A continuación se muestran algunas fotos del modelo real del manipulador, como evidencia de la construcción y funcionamiento del mismo.

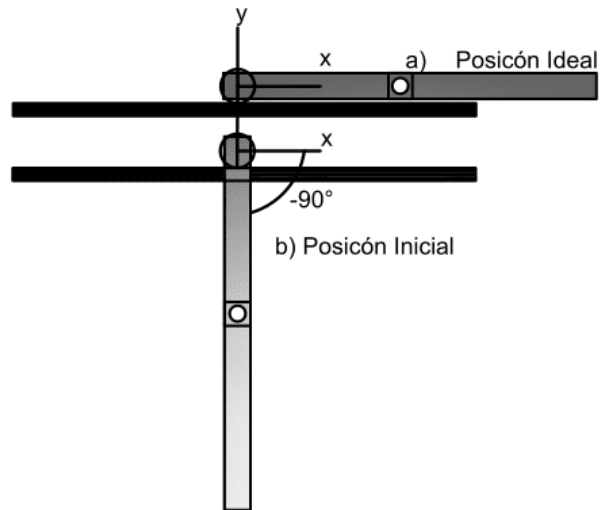
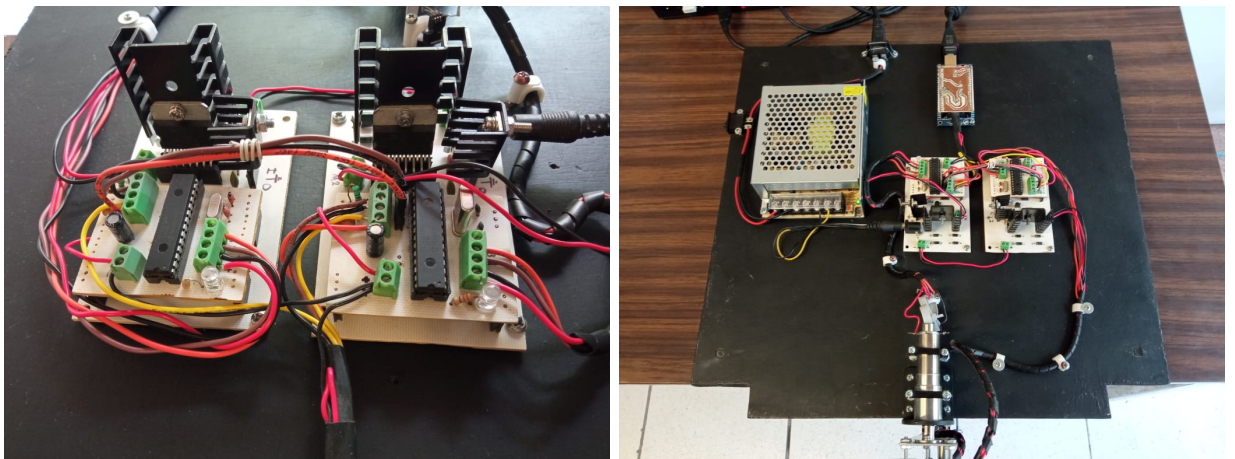


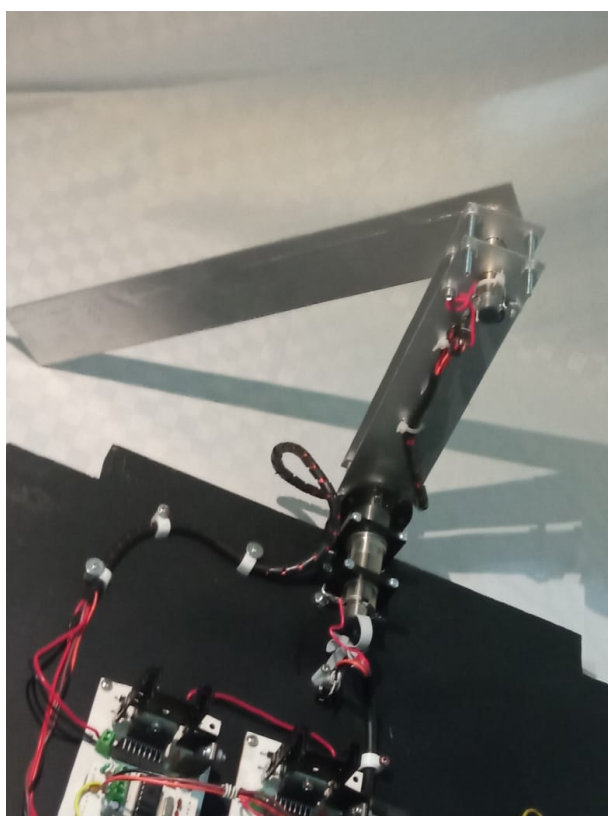
Figura 2.48: Posición de Referencia Para Calibración



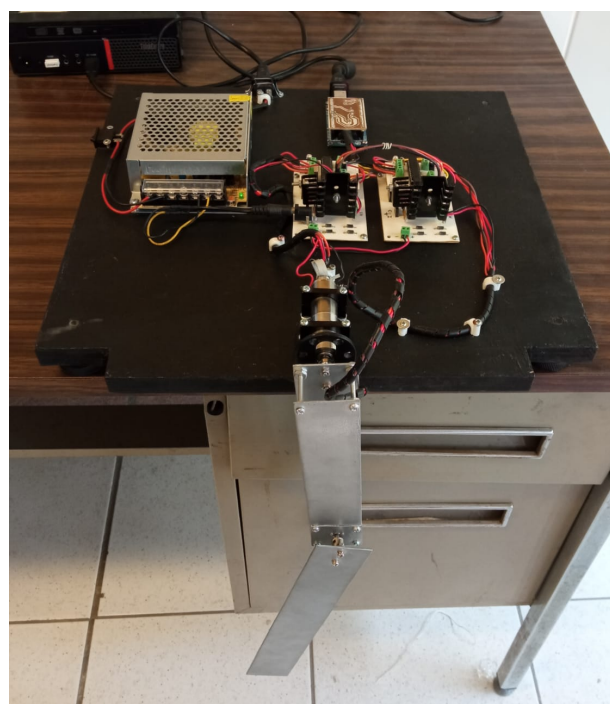
(a) Módulos

(b) Plataforma

Figura 2.49: Sistema de Adquisición de Señales, Control y Comunicación



(a) Manipulador vista trasera



(b) Manipulador vista frontal

Figura 2.50: Manipulador robótico de dos grados de libertad

2.7. Espacio de Trabajo del Manipulador

En la Figura 2.51 se muestra el espacio de trabajo del manipulador robótico de dos grados de libertad planar construido.

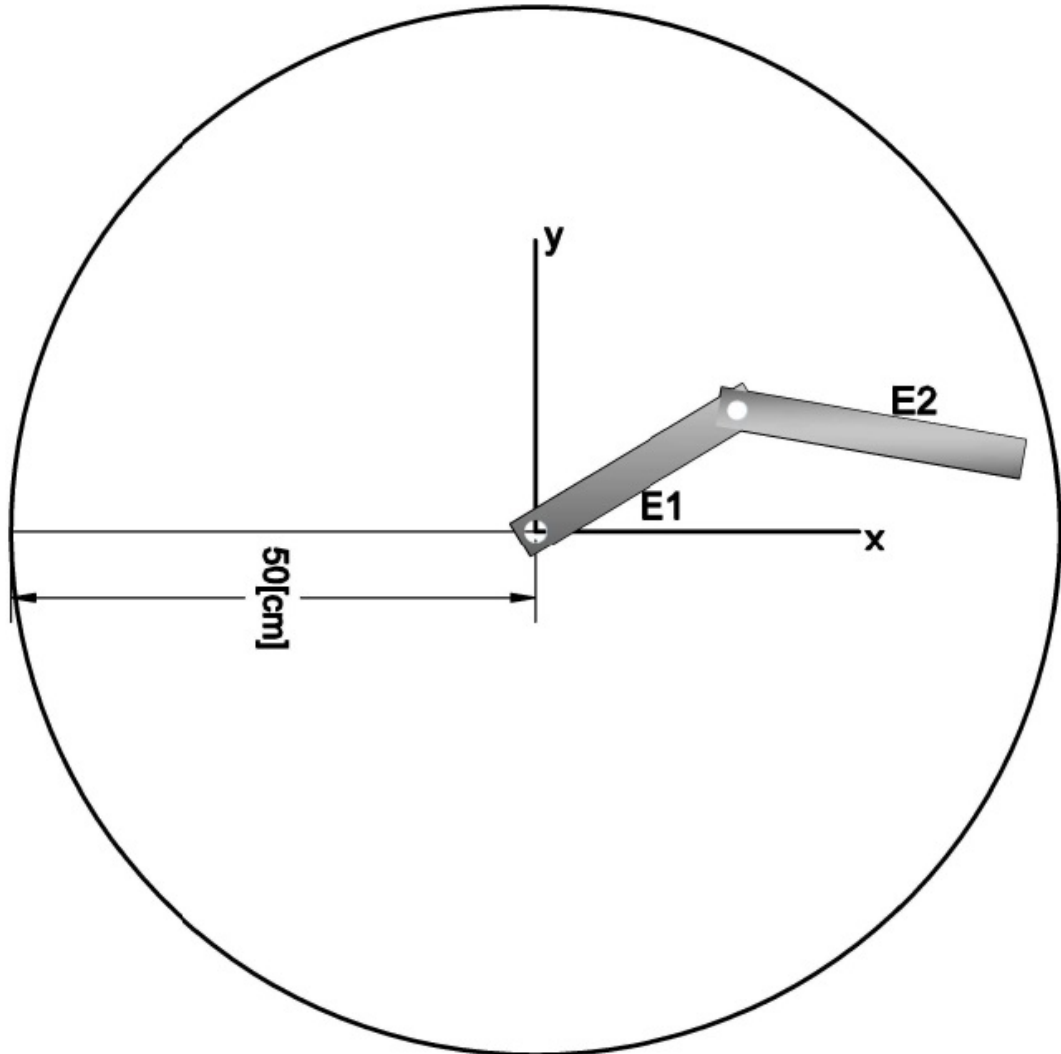


Figura 2.51: Espacio de Trabajo del Manipulador Robótico

Como se muestra en la Figura 2.51 el manipulador solo se mueve en el área de una circunferencia en el plano xy , limitado por un alcance máximo de 50 [cm] con respecto a la primera articulación y el efector final, limitado a dos revoluciones en sentido horario y antihorario.

2.8. Costos Generales

Recopilando la información de cada uno de las secciones anteriores tenemos la lista de costos mostrada en la Tabla 2.23.

Tabla 2.23: Balance de costos generales del prototipo

Sección	Costo General
Estructura del manipulador	\$251.5 MXN
Eléctrica y Electrónica del Manipulador sin motores	\$630.77 MXN
Eléctrica y Electrónica del Manipulador con motores	\$19,481.87 MXN
Módulo de Control y Adquisición	\$1,777.32 MXN

El costo final de prototipo sin incluir los motores fue de \$2,659.59 MXN mientras que el costo final incluyendo motores es de \$21,510.69 MXN

Capítulo 3

Modelado Matemático y Planeación de Trayectorias

3.1. Modelo Matemático de Manipuladores Robóticos

El análisis matemático se realizó por diferentes etapas, comenzando por la cinemática del sistema. Posterior a ésta se obtuvo la dinámica del manipulador de dos grados de libertad planar y al final se diseñó una trayectoria que nos ayudó como referencia para así visualizar el comportamiento de la aplicación los diferentes algoritmos de control. El desarrollo siguiente es específico para el manipulador planar de 2 grados de libertad, pero es aplicable a todos los tipos de manipuladores, como por ejemplo, los vistos en la Sección 1.4.1. En la Figura 3.1 se muestra el bosquejo de un manipulador de dos grados de libertad, que sirve como referencia para el análisis.

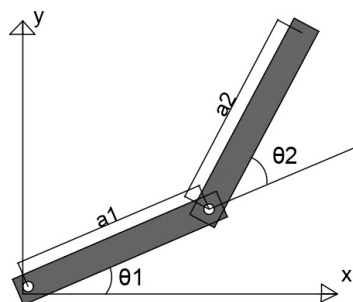


Figura 3.1: Robot Planar de 2 Grados de Libertad

3.2. Cinemática Directa

La cinemática directa representa la posición del efector final del robot en función de las variables articulares, que generalizaremos como:

$$q_i = \begin{cases} d_i & \text{si la articulación es Prismática} \\ \theta_i & \text{si la articulación es de Revolución} \end{cases} \quad (3.1)$$

Una vez definido esto, desglosaremos una serie de conceptos importantes que nos ayudaron a entender y a determinar la cinemática directa del manipulador.

El primer concepto con el que nos encontramos es la cadena cinemática, que se define como una sucesión de eslabones conectados mediante articulaciones, de la cual se derivan dos tipos: abiertas y cerradas, como se muestra en la Figura 3.1 que es una cadena cinemática abierta. Para nuestro caso nos enfocaremos en las cadenas abiertas, que son las que constituyen a los manipuladores serie como ya se planteó en la Sección 1.3.3.

Una cadena cinemática abierta es aquella en la que una articulación conecta con dos eslabones, por ello cuenta con n articulaciones y siguiendo la definición tendría $n+1$ eslabones, tomando como referencia la base que es el eslabón cero y último eslabón que es el efector final. En cambio una cadena cinemática cerrada es aquella en la que una articulación puede unir a dos o mas eslabones.

Esta representación cinemática está en función de la composición de transformaciones homogéneas, que es la imagen de un punto sobre el robot con respecto al sistema coordenado base, donde cada uno de los eslabones tiene asignado un sistema de referencia. De esta manera se puede tener la transformación homogénea del sistema base al efector final. Dicha transformación cuenta con propiedades particulares, como es el hecho de que las transformaciones dependen de las variables articulares. De manera general son requeridos al menos 6 parámetros: 3 de posición y 3 de rotación, para representar un movimiento rígido entre dos eslabones cualesquiera. Si se cumplen ciertas restricciones (de Denavit-Hartenberg), se pueden tener un mínimo de 4 parámetros para representar un movimiento rígido, dando como resultado la matriz

$${}^{i-1}\mathbf{H}_i = \begin{bmatrix} C_{\theta_i} & -S_{\theta_i}C_{\alpha_i} & S_{\theta_i}S_{\alpha_i} & a_iC_{\theta_i} \\ S_{\theta_i} & C_{\theta_i}C_{\alpha_i} & -C_{\theta_i}S_{\alpha_i} & a_iS_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

donde:

- a_i es la longitud del eslabón y es constante.
- α_i es la torcedura del eslabón, además es constante
- θ_i es el ángulo de la articulación, puede ser constante o variable según sea el tipo de articulación.
- d_i desplazamiento de la articulación, también puede ser constante o variable según el tipo de articulación

Con esto podemos determinar la orientación y posición del efector final en función de transformaciones homogéneas de eslabones adyacentes, de la siguiente manera.

$${}^0\mathbf{H}_n = {}^0\mathbf{H}_1 {}^1\mathbf{H}_2 \dots {}^{n-1}\mathbf{H}_n \quad (3.3)$$

Para poder utilizar la representación anterior, recurrimos al trabajo realizado por Jacques Denavit y Richard Hartenberg, que se define y se aplica como sigue.

3.2.1. Denavit-Hartenberg

Jacques Denavit y Richard Hartenberg proponen que solo basta con utilizar 4 parámetros para determinar las transformaciones homogéneas, como se observa en la ecuación (3.2), siempre que se cumplan las restricciones siguientes.

- DH1- El eje \mathbf{x}_i siempre interseca al eje \mathbf{z}_{i-1}
- DH2- El eje \mathbf{x}_i siempre es perpendicular al eje \mathbf{z}_{i-1}

Considerando estas restricciones procedemos a describir el algoritmo de Denavit-Hartenberg. Se aplicará al robot planar de dos grados de libertad, como se muestra en la Figura 3.2, pero es aplicable a cualquier tipo de robot.

Paso1. El primer paso consiste en colocar los ejes \mathbf{z} , desde $\mathbf{z}_0 \dots \mathbf{z}_{n-1}$. Este eje se ubica dependiendo del tipo de articulación. Para el caso presente en el eje de giro dado que es una articulación de revolución, si fuera prismática sería en la dirección de desplazamiento, esto se ve en la Figura 3.2.

Paso2. Para este paso se coloca el sistema coordenado de la base, posicionado el origen de

la base \mathbf{o}_0 sobre el eje \mathbf{z} de la base \mathbf{z}_0 , los ejes \mathbf{x}_0 y \mathbf{y}_0 se colocan de tal manera que se forme un sistema dextrógiro.

Paso3. En este paso se coloca el origen, $\mathbf{o}_1 \dots \mathbf{o}_{n-1}$, de las siguientes articulaciones, en función de 3 casos posibles.

- Caso1. Si el eje \mathbf{z}_i actual interseca al eje \mathbf{z}_{i-1} anterior, el origen actual se coloca sobre la intersección.
- Caso2. Si el eje \mathbf{z}_i y \mathbf{z}_{i-1} son paralelos, se coloca el origen actual en cualquier punto de eje \mathbf{z}_i actual de la manera en que mas convenga.
- Caso3. Si el eje \mathbf{z}_i actual y \mathbf{z}_{i-1} anterior no cumple con alguno de los dos casos anteriores, el origen se coloca en la intersección de una normal común entre ambos ejes \mathbf{z} .

Paso4. Una vez que ya terminamos de colocar los orígenes de cada articulación procedemos a colocar los ejes \mathbf{x} , $\mathbf{x}_1 \dots \mathbf{x}_{n-1}$, del cual también contamos con tres casos.

- Caso1. Si \mathbf{z}_i actual y \mathbf{z}_{i-1} anterior se intersecan, colocamos el eje \mathbf{x}_i en la perpendicular al plano que forman \mathbf{z}_i y \mathbf{z}_{i-1} , y que pase por el origen \mathbf{o}_i actual.
- Caso2. Si los ejes \mathbf{z}_i actual y \mathbf{z}_{i-1} anterior son paralelos, colocamos el eje \mathbf{x}_i actual en una normal al ambos ejes y que pase por el origen \mathbf{o}_i actual.
- Caso3. Si los ejes \mathbf{z}_i actual y \mathbf{z}_{i-1} anterior, no cumplen ninguno de los dos casos anteriores, se coloca el eje \mathbf{x}_i actual, sobre la normal común y que pase por el origen \mathbf{o}_i actual.

Paso5. Ya que tenemos los ejes \mathbf{z} y \mathbf{x} , ahora procedemos a colocar los ejes \mathbf{y} , $\mathbf{y}_1 \dots \mathbf{y}_{n-1}$, de tal manera que se formen sistemas dextrógiros.

Paso6. Este paso consiste en colocar el sistema coordinado del efector final, $\mathbf{O}_n, \mathbf{x}_n, \mathbf{y}_n, \mathbf{z}_n$, tomando en cuenta las siguientes consideraciones, que el origen \mathbf{o}_n se coloque en el centro del efector final, que \mathbf{z}_n del efector sea paralela a \mathbf{z}_{n-1} anterior y que pase por el origen del efector \mathbf{o}_n , finalmente colocamos \mathbf{x}_n siguiendo las consideraciones de los pasos anteriores y \mathbf{y}_n de tal manera que forme un sistema dextrógiro.

Con esto tenemos la cinemática directa del manipulador de dos grados de libertad, como se muestra a continuación, que se muestra en la Figura 3.2. **Paso7.** Ya que se obtuvo lo

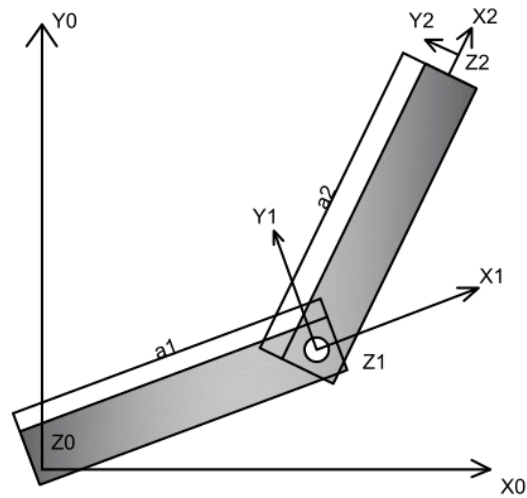


Figura 3.2: Cinemática Directa

anterior podemos crear una tabla de parámetros, que serán de gran utilidad para el último paso. Estos parámetros son 4, como se planteó al inicio de la Sección 3.2.1, y se describen a continuación.

- a_i es la dirección desde la intersección de \mathbf{x}_i y \mathbf{z}_{i-1} hasta el origen actual \mathbf{o}_i , considerando que el signo de a_i se determina con la dirección del eje \mathbf{x}_i .
- d_i es la distancia desde el origen anterior \mathbf{o}_{i-1} hasta la intersección del eje \mathbf{x}_i con el eje \mathbf{z}_{i-1} , considerando que si la articulación es prismática entonces d_i es variable y se va a diferenciar por d_i^*
- α_i es el ángulo entre \mathbf{z}_{i-1} y \mathbf{z}_i , utilizando a \mathbf{x}_i como eje de giro, el signo de α_i se asigna mediante la regla de la mano derecha.
- θ_i es el ángulo entre \mathbf{x}_{i-1} y \mathbf{x}_i , utilizando a \mathbf{z}_{i-1} como eje de giro. Si en este caso la articulación es de revolución entonces θ_i es variable y se va a diferenciar por θ_i^*

Paso 8. El último paso consiste en formar las matrices de transformación homogénea, a partir de los valores obtenidos en la tabla del paso 7, quedando de la siguiente manera.

Tabla 3.1: Tabla de parámetros DH

Articulación	a_i	d_i	α_i	θ_i	CI
1	a_1	0	0	θ_1	0
2	a_2	0	0	θ_2	0

$${}^0\mathbf{H}_1 = \begin{bmatrix} C_{\theta_1} & -S_{\theta_1} & 0 & a_1 C_{\theta_1} \\ S_{\theta_1} & C_{\theta_1} & 0 & a_1 S_{\theta_1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$${}^1\mathbf{H}_2 = \begin{bmatrix} C_{\theta_2} & -S_{\theta_2} & 0 & a_2 C_{\theta_2} \\ S_{\theta_2} & C_{\theta_2} & 0 & a_2 S_{\theta_2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

$${}^0\mathbf{H}_2 = {}^0\mathbf{H}_1 {}^1\mathbf{H}_2 = \begin{bmatrix} C_{\theta_1\theta_2} & -S_{\theta_1\theta_2} & 0 & a_2 C_{\theta_1\theta_2} + a_1 C_{\theta_1} \\ S_{\theta_1\theta_2} & C_{\theta_1\theta_2} & 0 & a_2 S_{\theta_1\theta_2} + a_1 S_{\theta_1} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

3.3. Cinemática Inversa

La cinemática inversa consiste en obtener los valores articulares $\mathbf{q} \in \mathbb{R}^n$, dada una pose deseada del efector final, es decir, una orientación deseada ${}^0\mathbf{R}_d$ y posición deseada ${}^0\mathbf{O}_d$, basado en la siguiente igualdad.

$${}^0\mathbf{H}_n(\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3 \dots \mathbf{q}_n) = {}^0\mathbf{H}_d \quad (3.7)$$

donde:

- ${}^0\mathbf{H}_d$ es la matriz de transformación homogénea
- \mathbf{q}_i es la variable articular

Resolver el problema de la cinemática inversa de manera analítica implica de resolver 12 ecuaciones no lineales, que para un robot con un gran número de articulaciones puede

ser un proceso demasiado complicado e incluso puede ser imposible obtener una solución, por lo que se recurre a utilizar un método geométrico, que se describe a continuación.

3.3.1. Método Geométrico

Este método se basa en analizar la geometría del robot y con ayuda de herramientas trigonométricas determinar el valor de las variables articulares. Para este método no hay pasos específicos a seguir, dependerá de la visualización y la destreza de cada persona, por lo que no existe una única forma de solucionarlo. Sin embargo, se pueden seguir algunas recomendaciones que se pueden utilizar para facilitar el cálculo, por ejemplo:

- Realizar primero la cinemática directa utilizando el algoritmo de Denavit-Hartenberg
- Si el robot cuenta con muñeca esférica, realizar el desacople cinemático.
- Mover el robot de tal manera que las variables articulares no tomen valores como $\theta_i = 0, \pm\frac{\pi}{2}, \pm\pi$ y $d_i = 0$
- Si la articulación actual es de revolución proyectar en el plano xy anterior y encontrar θ_i , en cambio si es prismática buscar donde se proyecte el eje z_i y así encontrar d_i

Para el robot planar de dos grados de libertad se resolvió como sigue. Considérese la asignación mostrada en la Figura 3.3.

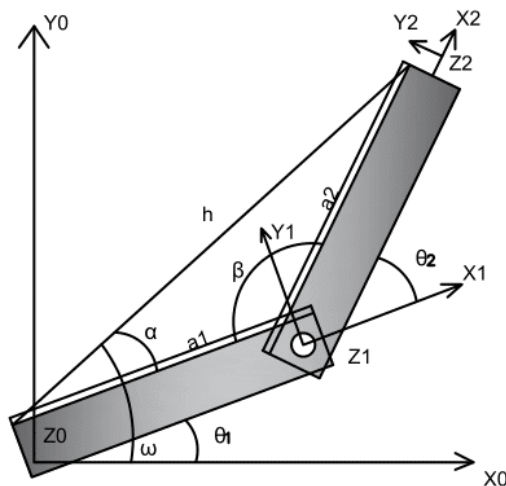


Figura 3.3: Referencia para el Análisis de la Cinemática Inversa

El origen del efector final se ubica en.

$$\begin{aligned} o_{cx} &= a_1 C_1 + a_2 C_{12} \\ o_{cy} &= a_1 S_1 + a_2 S_{12} \end{aligned} \quad (3.8)$$

Si unimos con una línea recta el origen de la base y el origen del efector final, podemos calcular la distancia entre ambos puntos de la siguiente manera

$$h = \sqrt{O_{cx}^2 + O_{cy}^2} \quad (3.9)$$

Utilizando leyes de senos y cosenos es posible determinar el valor del ángulo β , como se indica en la Figura 3.3. Aplicando la ley de cosenos

$$h^2 = a_1^2 + a_2^2 - 2a_1 a_2 \cos(\beta) \quad (3.10)$$

Despejando el ángulo $\cos(\beta)$, se tiene:

$$\cos(\beta) = \frac{-h^2 + a_1^2 + a_2^2}{2a_1 a_2} \triangleq D \quad (3.11)$$

De la Figura 3.3, se observa que

$$\beta = \pi - \theta_2 \quad (3.12)$$

Sustituyendo (3.12) en (3.11) :

$$\cos(\beta) = \cos(\pi - \theta_2) = \cos(-\theta_2) = \cos(\theta_2) \quad (3.13)$$

Por lo tanto:

$$\cos(\beta) = \cos(\theta_2) \triangleq D \quad (3.14)$$

De esta manera θ_2 se puede expresar en términos de $\text{atan2}(y, x)$.

$$\theta_2 = \text{atan2}(\pm \sqrt{1 - D^2}, D) \quad (3.15)$$

El ángulo ω es:

$$\omega = \text{atan2}(O_{cy}, O_{cx}) \quad (3.16)$$

Finalmente para determinar θ_1 se utiliza la Figura 3.4.

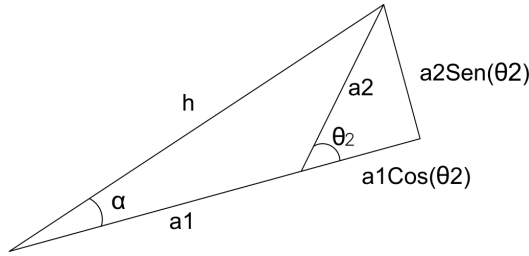


Figura 3.4: Cinemática Inversa

$$\alpha = \text{atan2}(a_2 \text{sen}(\theta_2), a_1 + a_2 \text{cos}(\theta_2)) \quad (3.17)$$

$$\theta_1 = \omega - \alpha = -\text{atan2}(a_2 \text{sen}(\theta_2), a_1 + a_2 \text{cos}(\theta_2)) + \text{atan2}(o_{cy}, o_{cx}) \quad (3.18)$$

Con esto finalmente tenemos la cinemática inversa del manipulador.

3.4. Jacobiano del Manipulador

En la cinemática directa e inversa se describió el espacio Cartesiano y el espacio articular y como pasar de uno a otro. Ahora se hablará de las razones de cambio instantáneo o cinemática diferencial, es decir, la relación entre las velocidades del espacio articular y las velocidades del espacio Cartesiano. Para realizar esto introduciremos un nuevo término, que es una de las herramientas con mayor peso en la actualidad para la robótica que es el Jacobiano. Para determinar el Jacobiano también contamos con dos perspectivas diferentes, una desde la perspectiva analítica y la otra desde la perspectiva geométrica.

El Jacobiano analítico parte de función explícita de la cinemática directa con la forma siguiente:

$$\begin{bmatrix} {}^0\mathbf{o}_n \\ \nu \end{bmatrix} = f(\mathbf{q}) \quad (3.19)$$

donde:

- ν es una parametrización de la rotación, que es el ángulo entre el último eslabón y la horizontal.

Si derivamos con respecto al tiempo la ecuación (3.19), tenemos entonces que:

$$\begin{bmatrix} \frac{d}{dt} {}^0\mathbf{o}_n \\ \frac{d}{dt} \nu \end{bmatrix} = \frac{\partial}{\partial \mathbf{q}} f(\mathbf{q}) \frac{d\mathbf{q}}{dt} = J_a(\mathbf{q}) \dot{\mathbf{q}} \quad (3.20)$$

donde:

- $J_a(q)$ es el Jacobiano analítico.

Pero este método tiene sus desventajas, por ejemplo el hecho de que es difícil lograr obtener una expresión de la orientación, que sea continua y diferenciable, además de que las funciones diferenciables para orientación y posición son muy complejas. Finalmente tenemos que la derivada de la parametrización de la orientación no tiene un significado físico, por lo tanto no es de utilidad. En cambio el Jacobiano geométrico muestra una relación del espacio articular con la velocidad lineal ${}^0\mathbf{v}_n$ y la velocidad angular ${}^0\boldsymbol{\omega}_n$, ecuación (3.7). De esta manera pasamos a la descripción y solución de Jacobiano del manipulador.

$$\begin{bmatrix} \frac{d}{dt} {}^0\mathbf{v}_n \\ \frac{d}{dt} {}^0\boldsymbol{\omega}_n \end{bmatrix} = \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}_{6 \times n} = \begin{bmatrix} \mathbf{J}_v(\mathbf{q})_{3 \times n} \\ \mathbf{J}_\omega(\mathbf{q})_{3 \times n} \end{bmatrix} \dot{\mathbf{q}} \quad (3.21)$$

donde:

- $\mathbf{J}_v(\mathbf{q})_{3 \times n}$ es el Jacobiano de velocidad lineal.
- $\mathbf{J}_\omega(\mathbf{q})_{3 \times n}$ es el Jacobiano de velocidad angular.

3.4.1. Jacobiano Geométrico

De esta manera procedemos a realizar el cálculo del Jacobiano para el manipulador de 2 grados de libertad, tomando en consideración la siguiente regla general para los robots rígidos serie. Esta regla se basa en el tipo de articulación que se esté analizando, se obtiene de manera individual para cada articulación y se representa en una matriz como en (3.7). Para el cálculo del Jacobiano de la articulación i -ésima se procede a lo siguiente:

$$\begin{aligned} \text{▪ } \mathbf{J}_{v_i} &= \begin{cases} {}^0\mathbf{z}_{i-1} \times ({}^0\mathbf{o}_n - {}^0\mathbf{o}_{i-1}) & \text{si la articulación es de revolución} \\ {}^0\mathbf{z}_{i-1} & \text{si la articulación es prismática} \end{cases} \\ \text{▪ } \mathbf{J}_{\omega_i} &= \begin{cases} {}^0\mathbf{z}_{i-1} & \text{si la articulación es de revolución} \\ 0_{3 \times 1} & \text{si la articulación es prismática} \end{cases} \end{aligned}$$

Considerando las matrices de transformación homogéneas, podemos determinar el Jacobiano del manipulador, usando el método geométrico.

$${}^0\mathbf{H}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

$${}^0\mathbf{H}_1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.23)$$

$${}^0\mathbf{H}_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_2 C_{12} + a_1 C_1 \\ S_{12} & C_{12} & 0 & a_2 S_{12} + a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.24)$$

Se comienza con el cálculo de las velocidades tangenciales

$$\begin{aligned} \mathbf{J}_{v1} &= {}^0 \mathbf{z}_0 \times ({}^0 \mathbf{o}_2 - {}^0 \mathbf{o}_0) \\ \mathbf{J}_{v1} &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left[\begin{bmatrix} a_2 C_{12} + a_1 C_1 \\ a_2 S_{12} + a_1 S_1 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right] \\ \mathbf{J}_{v1} &= \begin{bmatrix} -a_2 C_{12} - a_1 C_1 \\ a_2 S_{12} + a_1 S_1 \\ 0 \end{bmatrix} \end{aligned} \quad (3.25)$$

$$\begin{aligned} \mathbf{J}_{v2} &= {}^0 \mathbf{z}_1 \times ({}^0 \mathbf{o}_2 - {}^0 \mathbf{o}_1) \\ \mathbf{J}_{v2} &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \times \left[\begin{bmatrix} a_2 C_{12} + a_1 C_1 \\ a_2 S_{12} + a_1 S_1 \\ 0 \end{bmatrix} - \begin{bmatrix} a_1 C_1 \\ a_1 S_1 \\ 0 \end{bmatrix} \right] \\ \mathbf{J}_{v2} &= \begin{bmatrix} -a_2 C_{12} \\ a_2 S_{12} \\ 0 \end{bmatrix} \end{aligned} \quad (3.26)$$

Ahora se procede al cálculo de la velocidades angulares

$$\mathbf{J}_{\omega 1} = \mathbf{J}_{\omega 2} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.27)$$

Finalmente el Jacobiano geométrico del manipulador de 2 grados de libertad es.

$$\mathbf{J}_a(\mathbf{q}) = \begin{bmatrix} -a_2 C_{12} - a_1 C_1 & -a_2 C_{12} \\ a_2 S_{12} + a_1 S_1 & a_2 S_{12} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \quad (3.28)$$

Ya que se utilizará una trayectoria Cartesiana se calcula también la derivada del Jacobiano, tomando las siguientes consideraciones.

$$\begin{aligned} \dot{\mathbf{x}}_d &= \mathbf{J}_a(\mathbf{q})\dot{\mathbf{q}}_d \\ \dot{\mathbf{q}} &= \mathbf{J}_a(\mathbf{q})^{-1}\dot{\mathbf{x}}_d \\ \ddot{\mathbf{x}}_d &= \dot{\mathbf{J}}_a(\mathbf{q})\dot{\mathbf{q}}_d + \mathbf{J}_a(\mathbf{q})\ddot{\mathbf{q}}_d \\ \ddot{\mathbf{q}}_d &= \mathbf{J}_a(\mathbf{q})^{-1}(\ddot{\mathbf{x}}_d - \dot{\mathbf{J}}_a(\mathbf{q})\dot{\mathbf{q}}_d) \end{aligned} \quad (3.29)$$

Para lo cual $\dot{\mathbf{J}}_a(\mathbf{q})$ es:

$$\dot{\mathbf{J}}_a(\mathbf{q}) = \begin{bmatrix} -a_2 C_{12}\dot{q}_1 - a_1 C_1\dot{q}_1 - a_2 C_{12}\dot{q}_2 & -a_2 C_{12}(\dot{q}_1 + \dot{q}_2) \\ -a_2 S_{12}(\dot{q}_1 + \dot{q}_2) - a_1 S_1\dot{q}_1 & a_2 S_{12}(\dot{q}_1 + \dot{q}_2) \end{bmatrix} \quad (3.30)$$

3.5. Dinámica del Manipulador

Parte del análisis matemático ya se describió en las secciones anteriores, desde la perspectiva del movimiento, es decir, la cinemática, que conlleva hablar de posición, velocidad y la relación entre el espacio articular y el Cartesiano que se presentan en el robot. Ahora en esta sección nos centraremos en desarrollar el modelo desde la perspectiva de las fuerzas que actúan para generar movimiento, es decir, la dinámica. Aquí es donde se consideran fuerzas como la gravedad, la fricción, entre otras. Dicho modelo tiene la capacidad de representar la física del sistema y con ella mostrar el comportamiento del manipulador bajo

diferentes circunstancias.

Existen diferentes métodos para obtener el modelo matemático de la dinámica de un sistema físico, pero para el caso de los manipuladores se plantea bajo el método de Euler–Lagrange. Este método se basa en el Lagrangiano, que representa la resta matemática entre la energía cinética y la potencial. Además el Lagrangiano debe expresarse en términos de las coordenadas generalizadas y para los robots manipuladores \mathbf{q} es un conjunto válido de estas.

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}(q, \dot{q}) - \mathcal{P}(q) \quad (3.31)$$

donde:

- \mathcal{L} Lagrangiano
- \mathcal{K} Energía Cinética
- \mathcal{P} Energía Potencial

Si la ecuación de Euler–Lagrange se cumple, entonces se cumple el principio de mínima acción

$$\frac{d}{dt} \left\{ \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \right\} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = \tau - \tau_f \quad (3.32)$$

Pero antes de aplicar esto, primero debemos de conocer los términos de la energía cinética y potencial de un manipulador de manera general, para después obtener el modelo dinámico del robot manipulador de dos grados de libertad. Para el caso de la energía cinética tenemos dos términos, uno que surge del movimiento lineal de los eslabones y otro que surge de la rotación de los mismos, por lo tanto se involucra la velocidad lineal y angular

$$\mathcal{K} = \frac{1}{2} \sum_{i=1}^n \{ m_i {}^0\mathbf{v}_i^T {}^0\mathbf{v}_i + {}^0\boldsymbol{\omega}_i^T {}^0\mathbf{I}_i {}^0\boldsymbol{\omega}_i \} \quad (3.33)$$

Como podemos observar, los términos que involucran tanto a la velocidad lineal como a la angular, son términos con los que ya hemos trabajado en la cinemática diferencial del manipulador, que surgen a partir del Jacobiano, la única diferencia con el Jacobiano geométrico del manipulador y las velocidades para determinar la energía cinética es que la segunda se determina de manera individual para cada uno de los eslabones, considerando como punto de referencia el centro de masa del eslabón, a diferencia del Jacobiano geométrico

que se analiza desde el centro del efector final. Para continuar con nuestro análisis podemos recurrir a lo siguiente:

$${}^0\mathbf{v}_i = \mathbf{J}_{v_{C_i}} \dot{\mathbf{q}} \quad {}^0\boldsymbol{\omega}_i = \mathbf{J}_{\omega_{C_i}} \dot{\mathbf{q}} \quad (3.34)$$

donde:

- $\mathbf{J}_{v_{C_i}}$ Jacobiano de velocidad lineal del centro de masas
- $\mathbf{J}_{\omega_{C_i}}$ Jacobiano de velocidad angular del centro de masa

El método para obtener el Jacobiano de cada uno de los eslabones es similar al que se planteó en el Jacobiano geométrico, pero con las siguientes condiciones:

$$\begin{aligned} \text{▪ } \mathbf{J}_{v_{C_{ji}}} &= \begin{cases} \begin{cases} {}^0\mathbf{z}_{i-1} \times ({}^0\mathbf{o}_{C_j} - {}^0\mathbf{o}_{i-1}) & \text{si es de revolución} \\ {}^0\mathbf{z}_{i-1} & \text{si es prismática} \end{cases} & \text{si } i \leq j \\ 0 & \text{si } i > j \end{cases} \\ \text{▪ } \mathbf{J}_{\omega_{C_{ji}}} &= \begin{cases} \begin{cases} {}^0\mathbf{z}_{i-1} & \text{si es de revolución} \\ 0 & \text{si es prismática} \end{cases} & \text{si } i \leq j \\ 0 & \text{si } i > j \end{cases} \end{aligned}$$

Ya resuelta una gran parte de las incógnitas necesarias para determinar la energía cinética, ahora nos centraremos en el problema del tensor de inercia, dado que así como está planteado depende del sistema de la base, de tal manera depende de las variables articulares, por lo que podemos recurrir a lo siguiente que no solo depende de su orientación y que utiliza transformaciones de similitud para referirlo al sistema del mismo eslabón

$${}^0\mathbf{I}_i = {}^0\mathbf{R}_i {}^i\mathbf{I}_i {}^0\mathbf{R}_i^T \quad (3.35)$$

donde:

- ${}^0\mathbf{R}_i$ Matriz de rotación de la cinemática directa.
- ${}^i\mathbf{I}_i$ Tensor de inercia constante.

Esto nos da la ventaja de que el tensor de inercia ${}^i\mathbf{I}_i$ es una matriz constante de la forma

$${}^i\mathbf{I}_i = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \quad (3.36)$$

Además las matrices de rotación se obtienen de la cinemática directa. Con esto finalmente tenemos completos los términos para determinar la energía cinética

$$\mathcal{K} = \frac{1}{2} \sum_{i=1}^n \{m_i \dot{\mathbf{q}}^T \mathbf{J}_{v_{C_i}}^T \mathbf{J}_{v_{C_i}} \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{J}_{\omega_{C_i}}^T {}^0\mathbf{R}_i {}^i\mathbf{I}_i {}^0\mathbf{R}_i^T \mathbf{J}_{\omega_{C_i}} \dot{\mathbf{q}}\} \quad (3.37)$$

Una manera de expresar lo anterior es de la forma siguiente: si se factoriza el vector de coordenadas generalizadas.

$$\mathcal{K}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{H} \dot{\mathbf{q}}, \quad (3.38)$$

donde:

- \mathbf{H} es la matriz de inercia del sistema y depende de \mathbf{q}
- \mathbf{q} Vector de coordenadas generalizadas.

Otra de las ventajas de este método es la facilidad del cálculo de la energía potencial, ya que se puede hacer de manera individual para cada uno de los eslabones y luego sumar los efectos ocasionados por cada uno de ellos, para lo que solo basta con conocer la masa, la constante de aceleración gravitatoria y la altura del centro de masa de cada uno de los eslabones con respecto al sistema coordenado de la base, es decir

$$\mathcal{P}(\mathbf{q}) = \sum_{i=1}^n \{m_i \mathbf{g}^T {}^0\mathbf{o}_{C_i}\} \quad (3.39)$$

Donde:

- \mathbf{g}^T vector de aceleración gravitacional, este se define con dirección hacia donde incrementa la energía potencial y magnitud igual a la constante de aceleración gravitatoria.
- ${}^0\mathbf{o}_{C_i}$ Vector de posición del centro de masa del eslabón a partir del origen actual visto desde la base.

Es importante aclarar que todo esto depende de la asignación de la cinemática directa, por lo que la conformación de los vectores puede variar para cada eslabón. Con el Lagrangiano completo, procedemos a aplicar la formulación de Euler-Lagrange. Para ello debemos de expresara el Lagrangiano de la siguiente forma.

$$\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n h_{ij} \dot{q}_i \dot{q}_j - \mathcal{P} \quad (3.40)$$

Aplicando las ecuaciones de Euler–Lagrange llegamos a un modelo matemático de ecuaciones no lineales que se suele expresar de la siguiente manera.

$$\sum_{i=1}^n h_{ki} \ddot{q}_i + \sum_{i=1}^n \sum_{j=1}^n \frac{1}{2} \left\{ \frac{\partial h_{ki}}{\partial q_j} + \frac{\partial h_{kj}}{\partial q_i} - \frac{\partial h_{ij}}{\partial q_k} \right\} + \frac{\partial \mathcal{P}}{\partial q_k} = \tau_k - \tau_{fk} \quad (3.41)$$

Además si este modelo se representa para las n ecuaciones no lineales, podemos expresarlo de forma matricial, que nos ayudará a identificar el modelo de manera sencilla [Arteaga et al., 2021]. A este modelo se le añadió la matriz \mathbf{D}_v que representa a las fuerzas de fricción viscosa y \mathbf{D}_c que representa a las fuerzas de fricción de Coulomb [Bona and Indri, 2005].

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}_v\dot{\mathbf{q}} + \mathbf{D}_c\text{sign}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (3.42)$$

donde:

- $\mathbf{H}(\mathbf{q})$ es la matriz de inercia.
- $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ es la matriz de fuerzas centrífugas y de Coriolis.
- \mathbf{D}_v es la matriz de coeficientes de fricción viscosa.
- \mathbf{D}_c es la matriz de coeficientes de fricción de Coulomb.
- $\mathbf{g}(\mathbf{q})$ es el vector de pares debido a la gravedad
- $\boldsymbol{\tau}$ son los pares de entrada generalizados.

Finalmente procedemos a determinar el modelo matemático del robot planar de dos grados de libertad, basándonos en la Figura 3.5.

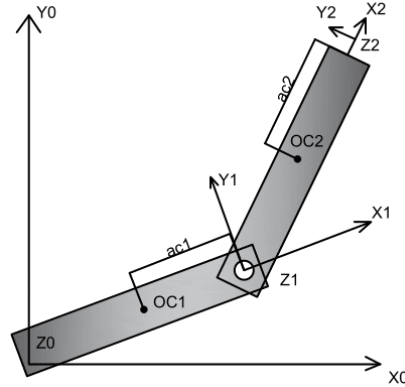


Figura 3.5: Centro de Masa Eslabones

El cálculo del centro de masa de los eslabones se realiza mediante

$${}^0\mathbf{o}_{c_1} = {}^0\mathbf{o}_1 - a_{c_1}^0 \mathbf{x}_1 = \begin{bmatrix} (a_1 - a_{c_1})C_1 \\ (a_1 - a_{c_1})S_1 \\ 0 \end{bmatrix} = \begin{bmatrix} l_{c_1} C_1 \\ l_{c_1} S_1 \\ 0 \end{bmatrix} \quad (3.43)$$

$${}^0\mathbf{o}_{c_2} = {}^0\mathbf{o}_2 - a_{c_2}^0 \mathbf{x}_2 = \begin{bmatrix} a_1 C_1 + (a_2 - a_{c_2})C_{12} \\ a_1 S_1 + (a_2 - a_{c_2})S_{12} \\ 0 \end{bmatrix} = \begin{bmatrix} a_1 C_1 + l_{c_2} C_{12} \\ a_1 S_1 + l_{c_2} S_{12} \\ 0 \end{bmatrix} \quad (3.44)$$

Además del cálculo del centro de masa del robot en función de la cinemática directa, se calcula el tensor de inercia considerando a los eslabones como barras planas, como se muestra en la Figura 3.6.



Figura 3.6: Barra Plana

El cálculo del tensor de inercia para cada una de las articulaciones se realiza utili-

zando las integrales correspondientes de la siguiente manera:

$$I_{xx} = \rho \int_{z=-c/2}^{z=c/2} \int_{y=-b/2}^{y=b/2} \int_{x=-a/2}^{x=a/2} (y^2 + z^2) \cdot dx dy dz = \frac{m}{12} ((b^*)^2 + (c^*)^2) \quad (3.45)$$

$$I_{yy} = \rho \int_{z=-c/2}^{z=c/2} \int_{y=-b/2}^{y=b/2} \int_{x=-a/2}^{x=a/2} (x^2 + z^2) \cdot dx dy dz = \frac{m}{12} ((a^*)^2 + (c^*)^2) \quad (3.46)$$

$$I_{zz} = \rho \int_{z=-c/2}^{z=c/2} \int_{y=-b/2}^{y=b/2} \int_{x=-a/2}^{x=a/2} (y^2 + x^2) \cdot dx dy dz = \frac{m}{12} ((a^*)^2 + (b^*)^2) \quad (3.47)$$

Dada la forma que considera para los eslabones del robot, se sabe que los términos de inercia cruzada son igual a cero.

$$\begin{aligned} I_{xy} &= I_{yx} = 0 \\ I_{xz} &= I_{zx} = 0 \\ I_{yz} &= I_{zy} = 0 \end{aligned} \quad (3.48)$$

Por lo tanto el tensor de inercia para cada una de las articulaciones queda como:

$${}^1\mathbf{I}_1 = \begin{bmatrix} \frac{m_1}{12} ((b_1^*)^2 + (c_1^*)^2) & 0 & 0 \\ 0 & \frac{m_1}{12} ((a_1^*)^2 + (c_1^*)^2) & 0 \\ 0 & 0 & \frac{m_1}{12} ((a_1^*)^2 + (b_1^*)^2) \end{bmatrix} \quad (3.49)$$

$${}^2\mathbf{I}_2 = \begin{bmatrix} \frac{m_2}{12} ((b_2^*)^2 + (c_2^*)^2) & 0 & 0 \\ 0 & \frac{m_2}{12} ((a_2^*)^2 + (c_2^*)^2) & 0 \\ 0 & 0 & \frac{m_2}{12} ((a_2^*)^2 + (b_2^*)^2) \end{bmatrix} \quad (3.50)$$

Con el tensor de inercia calculado para cada una de las articulaciones, se procede a calcular el Jacobiano para la velocidad angular y velocidad tangencial, para cada una de las articulaciones. Utilizando el método geométrico y las matrices homogéneas de la cinemática directa, se tiene que

$$\begin{aligned}
\mathbf{J}_{v_{11}} &= {}^0 \mathbf{z}_0 \times ({}^0 \mathbf{o}_{c_1} - {}^0 \mathbf{o}_0) = \begin{bmatrix} -l_{c_1} S_1 \\ l_{c_1} C_1 \\ 0 \end{bmatrix} \\
\mathbf{J}_{v_{12}} &= 0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\
\mathbf{J}_{v_{21}} &= {}^0 \mathbf{z}_0 \times ({}^0 \mathbf{o}_{c_2} - {}^0 \mathbf{o}_0) = \begin{bmatrix} -a_1 S_1 - l_{c_2} S_{12} \\ a_1 C_1 + l_{c_2} C_{12} \\ 0 \end{bmatrix} \\
\mathbf{J}_{v_{22}} &= {}^0 \mathbf{z}_1 \times ({}^0 \mathbf{o}_{c_2} - {}^0 \mathbf{o}_1) = \begin{bmatrix} -l_{c_2} S_{12} \\ l_{c_2} C_{12} \\ 0 \end{bmatrix}
\end{aligned} \tag{3.51}$$

Por lo tanto el Jacobiano de velocidad tangencial \mathbf{J}_{v_1} y \mathbf{J}_{v_2} es:

$$\mathbf{J}_{v_1} = \begin{bmatrix} -l_{c_1} S_1 & 0 \\ l_{c_1} C_1 & 0 \\ 0 & 0 \end{bmatrix} \tag{3.52}$$

$$\mathbf{J}_{v_2} = \begin{bmatrix} -a_1 S_1 - l_{c_2} S_{12} & -l_{c_2} S_{12} \\ a_1 C_1 + l_{c_2} C_{12} & l_{c_2} C_{12} \\ 0 & 0 \end{bmatrix} \tag{3.53}$$

$$\begin{aligned}
\mathbf{J}_{\omega_{11}} = {}^0 \mathbf{z}_0 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
\mathbf{J}_{\omega_{12}} &= 0 \\
\mathbf{J}_{\omega_{21}} = {}^0 \mathbf{z}_0 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\
\mathbf{J}_{\omega_{22}} = {}^0 \mathbf{z}_1 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\end{aligned} \tag{3.54}$$

Por lo tanto el Jacobiano de velocidad angular \mathbf{J}_{ω_1} y \mathbf{J}_{ω_2} es:

$$\mathbf{J}_{\omega_1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \tag{3.55}$$

$$\mathbf{J}_{\omega_2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 1 \end{bmatrix} \tag{3.56}$$

Con esto podemos realizar el cálculo del modelo dinámico, comenzando por la matriz de inercia $\mathbf{H}(\mathbf{q})$, quedando de la siguiente manera.

$$\mathbf{H}(\mathbf{q}) = \sum_{i=1}^n \{m_i \mathbf{J}_{v_{C_i}}^T \mathbf{J}_{v_{C_i}} + \mathbf{J}_{\omega_{C_i}}^T {}^0 \mathbf{R}_i {}^i \mathbf{I}_i {}^0 \mathbf{R}_i^T \mathbf{J}_{\omega_{C_i}}\} \tag{3.57}$$

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} m_1 l_{c_1}^2 + m_2(a_1^2 + 2a_1 l_{c_2} C_2 + l_{c_2}^2) + \beta_1 + \beta_2 & m_2(a_1 l_{c_2} C_2 + l_{c_2}^2) + \beta_2 \\ m_2(a_1 l_{c_2} C_2 + l_{c_2}^2) + \beta_2 & m_2 l_{c_2}^2 + \beta_2 \end{bmatrix} \tag{3.58}$$

donde:

- $\beta_1 = \frac{m_1}{12}(a_1^{*2} + b_1^2)$
- $\beta_2 = \frac{m_2}{12}(a_2^{*2} + b_2^2)$

Siguiendo la forma del modelo dinámico de la ecuación (3.42) tenemos que $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ se puede calcular a partir de la matriz de inercia $\mathbf{H}(\mathbf{q})$, mediante la obtención de los símbolos de Christoffel, que para el caso del manipulador de dos grados de libertad son 8 símbolos, que se obtienen mediante

$$c_{ijk} = \frac{1}{2} \left\{ \frac{\partial h_{ki}}{\partial q_j} + \frac{\partial h_{kj}}{\partial q_i} - \frac{\partial h_{ij}}{\partial q_k} \right\} \quad (3.59)$$

Los símbolos de Christoffel son

$$\begin{aligned} c_{111} &= 0 & c_{112} &= a_1 m_2 l_{c_2} S_2 & c_{121} &= -a_1 m_2 l_{c_2} S_2 & c_{122} &= 0 \\ c_{211} &= -a_1 m_2 l_{c_2} S_2 & c_{212} &= 0 & c_{221} &= -a_1 m_2 l_{c_2} S_2 & c_{222} &= 0 \end{aligned}$$

Para la construcción de la matriz $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})$ se utiliza

$$C_{kj} = \sum_{i=1}^n c_{ijk} \dot{q}_i \quad (3.60)$$

Dichos elementos quedan de la siguiente manera

$$\begin{aligned} C_{11} &= c_{111} \dot{q}_1 + c_{211} \dot{q}_2 \\ C_{12} &= c_{121} \dot{q}_1 + c_{221} \dot{q}_2 \\ C_{21} &= c_{112} \dot{q}_1 + c_{212} \dot{q}_2 \\ C_{22} &= c_{122} \dot{q}_1 + c_{222} \dot{q}_2 \end{aligned}$$

De esta manera obtenemos la matriz de fuerzas centrífugas y Coriolis.

$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -m_2 a_1 l_{c_2} S_{\theta_2} \dot{q}_2 & -m_2 a_1 l_{c_2} S_{\theta_2} (\dot{q}_1 + \dot{q}_2) \\ m_2 a_1 l_{c_2} S_{\theta_2} \dot{q}_1 & 0 \end{bmatrix} \quad (3.61)$$

Para determinar la matriz de los coeficientes debidos a la fricción, solo basta con acomodar cada coeficiente de tal manera que se presente en cada una de las ecuaciones del modelo dinámico. Para este caso se utilizaron dos matrices debido a la existencia de fricción de Coulomb en los motores, quedando de la siguiente manera

Fricción viscosa

$$D_{fv} = \begin{bmatrix} C_{fv1} & 0 \\ 0 & C_{fv2} \end{bmatrix} \quad (3.62)$$

Fricción de Coulomb

$$D_{fs} = \begin{bmatrix} C_{fs1} & 0 \\ 0 & C_{fs2} \end{bmatrix} \quad (3.63)$$

Finalmente se obtiene el vector de pares debidos a la gravedad. Para ello primero es indispensable obtener el vector de gravedad, que se elige con la dirección del crecimiento de la energía potencial. Dicho vector es de la forma

$$\mathbf{g} = [0 \quad g \quad 0]$$

Mediante la derivada de la ecuación (3.39) con respecto a la coordenada generalizada q obtenemos que el vector de pares debidos a la gravedad:

$$\mathbf{g}(\mathbf{q}) = \left(\frac{\partial \mathcal{P}(\mathbf{q})}{\partial \mathbf{q}} \right) = \begin{bmatrix} m_1 g l_{c_1} C_1 + m_2 g a_1 C_1 + m_2 g l_{c_2} C_{12} \\ m_2 g l_{c_2} C_{12} \end{bmatrix} \quad (3.64)$$

3.6. Planeación y Generación de Trayectoria tanto en Coordenadas Articulares como en Cartesianas

Para validar el funcionamiento de la plataforma del manipulador es necesario realizar una gran variedad de experimentos aplicando las leyes de control, pero no basta con mostrar los resultados para regulación, sino también se puede realizar control de seguimiento de trayectoria ya sea en variables articulares o en variables Cartesianas, El diseño de las trayectorias para ambos casos se detalla a continuación.

3.6.1. Trayectoria Articular

Una trayectoria articular es una función del tiempo $q(t)$ de tal manera que se tiene un punto de inicio y un punto de fin $q(t_0) = q_{inicial}$ y $q(t_f) = q_{final}$, por lo que $t_f - t_0$ representa el tiempo en que se lleva a cabo ese trayecto. Para diseñar una trayectoria basta con proponer una función basada en un polinomio. Entre mayor sea el grado del polinomio mayor será el control que se tendrá del diseño de la posición, velocidad y aceleración, es decir, con un polinomio de primer grado se tendría la posición variable, la velocidad constante y por lo tanto la aceleración sería nula, en cambio con un polinomio de quinto grado se tiene la posición variable, la velocidad variable y la aceleración igual variable. Esto tiene muchas ventajas claras a la hora de aplicarlo a un robot manipulador. Para la plataforma se propuso el diseño de una trayectoria basada en un polinomio de quinto grado de la forma

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (3.65)$$

Si se proponen condiciones de inicio (q_0, v_0, ac_0) y condiciones de fin (q_f, v_f, ac_f) , además la velocidad y la aceleración son la primera y segunda derivada temporal de la posición entonces se tiene que:

$$\begin{aligned}
 q_0(t_0) &= \delta_0 + \delta_1 t_0 + \delta_2 t_0^2 + \delta_3 t_0^3 + \delta_4 t_0^4 + \delta_5 t_0^5 \\
 v_0(t_0) &= \delta_1 + 2\delta_2 t_0 + 3\delta_3 t_0^2 + 4\delta_4 t_0^3 + 5\delta_5 t_0^4 \\
 ac_0(t_0) &= 2\delta_2 + 6\delta_3 t_0 + 12\delta_4 t_0^2 + 20\delta_5 t_0^3 \\
 q_f(t_f) &= \delta_0 + \delta_1 t_f + \delta_2 t_f^2 + \delta_3 t_f^3 + \delta_4 t_f^4 + \delta_5 t_f^5 \\
 v_f(t_f) &= \delta_1 + 2\delta_2 t_f + 3\delta_3 t_f^2 + 4\delta_4 t_f^3 + 5\delta_5 t_f^4 \\
 ac_f(t_f) &= 2\delta_2 + 6\delta_3 t_f + 12\delta_4 t_f^2 + 20\delta_5 t_f^3
 \end{aligned}$$

Estas ecuaciones se pueden escribir en su forma matricial:

$$\begin{bmatrix}
 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\
 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\
 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\
 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\
 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\
 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3
 \end{bmatrix}
 \begin{bmatrix}
 \delta_0 \\
 \delta_1 \\
 \delta_2 \\
 \delta_3 \\
 \delta_4 \\
 \delta_5
 \end{bmatrix}
 =
 \begin{bmatrix}
 q_0 \\
 v_0 \\
 ac_0 \\
 q_f \\
 v_f \\
 ac_f
 \end{bmatrix}$$

De esta manera tenemos la siguiente ecuación $M\delta = b$, de la cual podemos resolver para el vector de coeficientes δ y así dar solución para cada uno de los coeficientes

$$\begin{aligned}
 a_0 &= q_{d_0} & a_1 &= 0 & a_2 &= 0 \\
 a_3 &= \frac{10(q_{d_f} - q_{d_0})}{t_f^3} & a_4 &= -\frac{15(q_{d_f} - q_{d_0})}{t_f^4} & a_5 &= \frac{6(q_{d_f} - q_{d_0})}{t_f^5}
 \end{aligned}$$

Cabe aclarar que es necesario programar una trayectoria para cada una de las articulaciones, pero ya que se rigen bajo el mismo principio de diseño solo basta con dar una solución general. De esta manera la trayectoria funciona para las condiciones que el usuario de la plataforma desee utilizar, basta con proponer $(q_{d_0}, v_0, ac_0, q_{d_f}, v_f, ac_f, t_0, t_f)$.

3.6.2. Trayectoria Cartesiana

Una trayectoria Cartesiana al igual que la articular es una función del tiempo $x(t)$ y cumple las mismas condiciones. La diferencia radica en el espacio en que trabajan. Mientras

que la articular lo hace en el espacio de las articulaciones para trasladarse de un punto a otro, la Cartesiana lo hace en el espacio de Cartesiano. Gracias a la cinemática directa e inversa se puede hacer la conversión de un espacio a otro, de tal manera que podemos indicar los movimientos articulares para llegar a un punto en el espacio Cartesiano, que es mas fácil de visualizar. De la misma manera se propone un polinomio de quinto orden con la diferencia de que la posición va a quedar en función de coordenadas Cartesianas

$$x(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (3.66)$$

Como el diseño es el mismo que para trayectoria articular, la solución a cada uno de los coeficientes es la misma, pero en función de coordenadas Cartesianas.

$$a_0 = x_{d_0} \quad a_1 = 0 \quad a_2 = 0$$

$$a_3 = \frac{10(x_{d_f} - x_{d_0})}{t_f^3} \quad a_4 = -\frac{15(x_{d_f} - x_{d_0})}{t_f^4} \quad a_5 = \frac{6(x_{d_f} - x_{d_0})}{t_f^5}$$

Solo basta con elegir las condiciones de diseño $(x_{d_0}, v_{x_0}, ac_{x_0}, x_{d_f}, v_{x_f}, ac_{x_f}, t_0, t_f)$.

Capítulo 4

Algoritmos de Control Lineal, No Lineal y Determinación de Parámetros.

4.1. Control Lineal

4.1.1. Control PID

La siglas del control PID hacen referencia a tres elementos que constituye la acción de control que son el proporcional, derivativo e integral. Por separado cada una de ellas tiene sus ventajas y desventajas, pero en combinación se pueden aprovechar esas ventajas para tener mejores resultados [R. Kelly and Loría, 2005].

La ecuación general de un controlador PID, se expresa de la siguiente manera.

$$\tau = -K_p e(t) - \frac{K_p}{T_i} \int_0^t e(t) dt - K_p T_d \frac{de(t)}{dt} \quad (4.1)$$

Donde:

- $e(t)$ es el error y se expresa en términos de las coordenadas generalizadas del manipulador como $e(t) = \mathbf{q} - \mathbf{q}_d$.
- K_p es la ganancia de control proporcional.
- T_i es el tiempo integral.

- T_d es el tiempo derivativo.

Estas ganancias que constituyen al controlador se eligen de tal manera que entregue un resultado acorde a lo que se requiere. Se pueden determinar de manera directa para los más experimentados o utilizando alguno de los métodos de sintonización ya establecidos, por ejemplo el método de Ziegler–Nichols.

4.2. Algoritmo de Determinación de Parámetros, Regresor Lineal

En un modelo dinámico, en el caso de modelos de manipuladores, hay términos que caracterizan a dicho sistema y hace la diferencia entre uno y otro. Esto es debido a las características físicas que presentan como son las dimensiones, el peso, la gravedad y los términos de fricción. Los términos en un modelo dinámico representan parámetros que definen al manipulador, se denotan mediante un vector de parámetros, que generalmente se representa con la letra griega ' π ' o ' θ ' y se determinan midiendo las posiciones, velocidades y aceleraciones articulares, para valores de torques conocidos. Estos valores se obtienen en cada instante de tiempo para una trayectoria definida y mediante el uso de la expresión

$$\boldsymbol{\tau} = \begin{bmatrix} \tau(t_1) \\ \cdot \\ \cdot \\ \tau(t_N) \end{bmatrix} = \begin{bmatrix} Y(t_1) \\ \cdot \\ \cdot \\ Y(t_N) \end{bmatrix} \boldsymbol{\pi} = \mathbf{Y} \boldsymbol{\pi} \quad (4.2)$$

donde:

- $\boldsymbol{\tau}$ par en las articulaciones
- $\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ regresor del modelo dinámico
- $\boldsymbol{\pi}$ vector de parámetros

Si de la expresión(4.2) se resuelve para el vector de parámetros $\boldsymbol{\pi}$, tenemos que:

$$\boldsymbol{\pi} = (\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T \boldsymbol{\tau} \quad (4.3)$$

Como no existe matriz inversa directa de \mathbf{Y} se utiliza la pseudo-inversa $(\mathbf{Y}^T \mathbf{Y})^{-1} \mathbf{Y}^T$. Con esto podemos determinar los parámetros que caracterizan al manipulador. Cabe señalar

que la cantidad de parámetros depende del modelo y de la agrupación que se realice. De manera experimental se sugiere que el número de muestras obtenidas de \mathbf{Y} deben de ser lo suficientemente grande para tener una mejor estimación. Cuando se realiza la identificación de parámetros se debe tener en cuenta que algunos no se pueden determinar y todo depende de la distribución y acomodo de los eslabones y articulaciones. Otra recomendación importante es que a la hora de elegir los parámetros del vector $\boldsymbol{\pi}$ se deben agrupar los términos de tal manera que se reduzca su numero. Existe literatura con información detallada sobre este tema e incluso existen otros métodos de determinación de parámetros que son igual de importantes que el descrito aquí, se puede consultar la bibliografía [Spong et al., 2006] para profundizar en el tema.

Basándose en el modelo dinámico de la Sección 3.5 y aplicando lo antes descrito, se puede determinar el vector de parámetros para el robot planar de 2 grados de libertad, en este vector se incluyeron parámetros como la fricción seca y la fricción viscosa. El regresor y el vector de parámetros queda de la siguiente manera.

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \begin{bmatrix} \ddot{q}_1 & y_{12} & y_{13} & \dot{q}_1 & 0 & \text{sign}(\dot{q}_1) & 0 & C_{\theta_1} & C_{\theta_1\theta_2} \\ 0 & y_{22} & y_{23} & 0 & \dot{q}_2 & 0 & \text{sing}(\dot{q}_2) & 0 & C_{\theta_1\theta_2} \end{bmatrix} \quad (4.4)$$

Donde:

- $y_{12} = 2C_2\ddot{q}_1 + C_2\ddot{q}_2 - S_2\dot{q}_1\dot{q}_2 - S_2(\dot{q}_1 + \dot{q}_2)\dot{q}_2$
- $y_{13} = y_{23} = \ddot{q}_1 + \ddot{q}_2$
- $y_{22} = C_2\ddot{q}_1 + S_2\dot{q}_1^2$

El vector de parámetros queda como:

$$\begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \\ \pi_6 \\ \pi_7 \\ \pi_8 \\ \pi_9 \end{bmatrix} = \begin{bmatrix} m_1 l_{c_1}^2 + m_2 a_1^2 + \frac{m_2}{12} (b_1^2) \\ m_2 a_1 l_{c_2} \\ m_2 l_{c_2}^2 + \frac{m_2}{12} (b_2^2) \\ c_{fv1} \\ c_{fv2} \\ c_{fs1} \\ c_{fs2} \\ (m_1 l_{c_1} + m_2 a_1)g \\ m_2 l_{c_1} g \end{bmatrix} \quad (4.5)$$

4.3. Control No Lineal

4.3.1. Control PD con Compensación de Gravedad

Este controlador consta de un control basado en un PD más un término perteneciente al vector de pares gravitatorios. Guarda ciertas similitudes con un controlador lineal, pero con la existencia del vector de pares gravitacionales, que lo convierte en un control no lineal. La sintonización del mismo es muy sencilla ya que solo basta con sintonizar las ganancias de la parte proporcional y parte derivativa. Con ello se satisface el objetivo de control de posición en forma global. La única desventaja clara es que se requiere del conocimiento previo de una parte del modelo del robot y que el método de sintonización muchas veces se lleva a cabo de manera experimental.

Por las características del control y la poca dependencia que se tiene del modelo se puede demostrar que es más robusto ante incertidumbres y perturbaciones, pero ya que parte del control es lineal la sintonización para un punto puede ser inadecuada para otro punto, por la falta de conocimiento de las no linealidades del sistema [R. Kelly and Loria, 2005].

Dicha ley de control se expresa en la ecuación siguiente

$$\boldsymbol{\tau} = -\mathbf{K}_p \mathbf{e} - \mathbf{K}_v \dot{\mathbf{e}} + \mathbf{g}(\mathbf{q}) \quad (4.6)$$

4.3.2. Control Par Calculado

Para introducirse en el control Par Calculado [R. Kelly and Loria, 2005] previamente debemos de tener conocimiento del modelo dinámico del sistema y las coordenadas articulares deseadas si queremos que el robot siga una trayectoria o regule en un punto, estas son la posición, la velocidad y la aceleración deseadas. El objetivo es hacer que las coordenadas articulares tiendan a las coordenadas articulares deseadas, donde las coordenadas articulares se denotan por \mathbf{q} .

Si definimos el error como:

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_d \quad (4.7)$$

$$\dot{\mathbf{e}} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_d \quad (4.8)$$

$$\ddot{\mathbf{e}} = \ddot{\mathbf{q}} - \ddot{\mathbf{q}}_d \quad (4.9)$$

Y el objetivo del control es llegar a que:

$$(\mathbf{e}, \dot{\mathbf{e}}) = (0, 0) \quad \text{cuando } t \rightarrow \infty \quad (4.10)$$

Por lo tanto se debe definir una ley de control que depende de las variables articulares reales y de las deseadas. Para ello se comienza analizando el modelo Lagrangiano, que tiene la forma.

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}_v\dot{\mathbf{q}} + \mathbf{D}_s\text{sign}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} \quad (4.11)$$

Para comenzar podemos eliminar parte de la dinámica del sistema si la entrada de control es igual a esta mas un termino que representa la nueva entrada que podemos diseñar y se expresa de la siguiente manera.

$$\boldsymbol{\tau} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}_v\dot{\mathbf{q}} + \mathbf{D}_s\text{sign}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{u} \quad (4.12)$$

De las propiedades del modelo dinámico sabemos que la matriz de inercias es siempre invertible por lo tanto el nuevo sistema con la ley de control queda como.

$$\ddot{\mathbf{q}} = \mathbf{H}(\mathbf{q})^{-1}\mathbf{u} \quad (4.13)$$

Si la nueva entrada de control \mathbf{u} se define en términos de la misma matriz de inercias de la siguiente manera tenemos que:

$$\mathbf{u} = \mathbf{H}(\mathbf{q})\mathbf{v} \quad (4.14)$$

Por lo tanto tenemos un sistema lineal de segundo orden, que es mas sencillo de analizar a lo hora de crear una ley de control:

$$\ddot{\mathbf{q}} = \mathbf{v} \quad (4.15)$$

Un sistema de segundo orden en función del error se expresa de la siguiente manera

$$\ddot{\mathbf{e}} + 2\zeta\omega_n\dot{\mathbf{e}} + \omega_n^2\mathbf{e} = 0 \quad (4.16)$$

De esta manera la entrada \mathbf{v} se define como

$$\mathbf{v} = \ddot{\mathbf{q}}_d - 2\zeta\omega_n\dot{\mathbf{e}} - \omega_n^2\mathbf{e} \quad (4.17)$$

Con este análisis la ley de control queda definida de la siguiente manera.

$$\boldsymbol{\tau} = \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}_v\dot{\mathbf{q}} + \mathbf{D}_s\text{sign}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) + \mathbf{H}(\mathbf{q})(\ddot{\mathbf{q}}_d - 2\zeta\omega_n\dot{\mathbf{e}} - \omega_n^2\mathbf{e}) \quad (4.18)$$

4.3.3. Control Basado en Pasividad

Para hablar de control basado en pasividad nos remontamos a las propiedades del modelo dinámico, en las cuales encontramos la propiedad de pasividad. Esta propiedad considera una entrada, una salida y una función de almacenamiento. En el caso de los manipuladores se define como el Hamiltoniano y se expresa de la siguiente forma

$$\mathbf{H} = \mathbf{K} + \mathbf{P} \quad (4.19)$$

Con esto se llega a la desigualdad de pasividad, que se define como:

$$\dot{\mathbf{H}} = \dot{\mathbf{q}}^T \boldsymbol{\tau} \quad (4.20)$$

Si se toman algunas consideraciones se puede utilizar el Hamiltoniano como función candidata de Lyapunov y además utilizando la desigualdad de pasividad se puede probar estabilidad en el punto de equilibrio para una entrada $\boldsymbol{\tau} = 0$ o $\boldsymbol{\tau} = -\mathbf{K}_v \dot{\mathbf{q}}$.

Analizando el modelo dinámico se observa que la propiedad de pasividad depende en particular de la matriz antisimétrica $\dot{\mathbf{H}} - 2\mathbf{C}$, se puede utilizar el algoritmo de Slotine-Li para modificar el modelo dinámico de tal manera que podamos tener al sistema en términos de una sola variable que permita estabilizar el error de posición y de velocidad. Con esto el modelo no pierda la propiedad de pasividad haciendo al sistema fácil de estabilizar con las mismas condiciones que el modelo original. El algoritmo de Slotine-Li propone los siguientes cambios de variable:

$$\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \boldsymbol{\Lambda} \mathbf{e} \quad (4.21)$$

$$\mathbf{s} = \dot{\mathbf{q}} - \dot{\mathbf{q}}_r = \dot{\mathbf{e}} + \boldsymbol{\Lambda} \mathbf{e} \quad (4.22)$$

Donde:

- \mathbf{s} es la nueva salida pasiva
- $\boldsymbol{\Lambda}$ es una matriz de ganancias simétrica y positiva definida.
- $\dot{\mathbf{q}}_r$ es una variable auxiliar del algoritmo.

Sustituyendo dichas variables en el modelo Lagrangiano, se obtiene:

$$\mathbf{H}(\mathbf{q})\dot{\mathbf{s}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\mathbf{s} + \mathbf{D}_v \mathbf{s} = \boldsymbol{\tau} - \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r - \mathbf{D}_v \dot{\mathbf{q}}_r - \mathbf{D}_s \text{sign}(\dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q}) \quad (4.23)$$

Este nuevo modelo es pasivo y por lo tanto es fácil de estabilizar. Una de las ventajas del algoritmo es que se puede estabilizar el error de posición y velocidad con una sola variable s , [Spong et al., 2006, p. 276]. Por lo tanto la ley de control pasivo queda como:

$$\boldsymbol{\tau} = \mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{D}_v\dot{\mathbf{q}}_r + \mathbf{D}_s\text{sign}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) - \mathbf{K}_v\mathbf{s} \quad (4.24)$$

Donde:

- \mathbf{K}_v es una matriz de ganancias simétrica y positiva definida.

El termino $\mathbf{K}_v\mathbf{s}$ es equivalente a un control PD, por la definición de la variable \mathbf{s} . Además la diferencia de este con el control Para Calculado radica en que el modelo con $\dot{\mathbf{q}}_r$ es una prealimentación de la dinámica al depender de las velocidades y aceleraciones deseadas, siendo este un controlador más robusto que el Par Calculado.

4.3.4. Control Adaptable

Como se mencionó en la sección anterior, el control pasivo tiene ventajas sobre otros controladores, pero aun así se requiere del conocimiento exacto del modelo dinámico. Una manera de omitir el conocimiento exacto del modelo dinámico es utilizando la propiedad de parametrización del modelo Lagrangiano.

$$\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{D}_v\dot{\mathbf{q}} + \mathbf{D}_s\text{sign}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\boldsymbol{\pi}$$

Si se parte del modelo pasividad de la ecuación (4.23), se observa que el modelo $(\mathbf{H}(\mathbf{q})\ddot{\mathbf{q}}_r + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}_r + \mathbf{D}\dot{\mathbf{q}}_r + \mathbf{D}_s\text{sign}(\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}))$ también se puede parametrizar, con algo de diferencia siendo éste

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\boldsymbol{\pi}$$

Recordando que $\dot{\mathbf{q}}_r = \dot{\mathbf{q}}_d - \Delta(\mathbf{q} - \mathbf{q}_d)$, el nuevo modelo parametrizado queda como

$$\mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r) = \begin{bmatrix} \ddot{q}_{r1} & z_{12} & z_{13} & \dot{q}_{r1} & 0 & \text{sign}(\dot{q}_1) & 0 & C_{\theta_1} & C_{\theta_1\theta_2} \\ 0 & z_{22} & z_{23} & 0 & \dot{q}_{r2} & 0 & \text{sign}(\dot{q}_2) & 0 & C_{\theta_1\theta_2} \end{bmatrix}$$

Donde:

- $z_{12} = 2C_2\ddot{q}_{r1} + C_2\ddot{q}_{r2} - S_2\dot{q}_2\dot{q}_{r1} - S_2(\dot{q}_1 + \dot{q}_2)\dot{q}_{r2}$
- $z_{13} = z_{23} = \ddot{q}_{r1} + \ddot{q}_{r2}$

$$\blacksquare z_{22} = C_2\ddot{q}_{r1} + S_2\dot{q}_1\dot{q}_{r1}$$

Con esto tenemos un regresor modificado en el cual no se requiere del conocimiento de la aceleración, debido a que depende unicamente de los valores deseados. Por lo tanto el controlador Adaptable queda dado por

$$\boldsymbol{\tau} = \mathbf{Y}(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\hat{\boldsymbol{\pi}} - \mathbf{K}_v \mathbf{s} \quad (4.25)$$

Donde: \mathbf{K}_v es una matriz de ganancias positiva definida.

El vector $\hat{\boldsymbol{\pi}}$ es un aproximado del vector de parámetros $\boldsymbol{\pi}$. Recordemos que el control Adaptable calcula este vector de parámetros conforme se va desarrollando el movimiento, por lo que es un vector variable. Siendo variable se puede definir como cambia la derivada con respecto al tiempo. El cambio se conoce como ley de adaptación, y con esto solo se tienen dos estados $\hat{\boldsymbol{\pi}}$ y \mathbf{s} . En la literatura se demuestra que \mathbf{s} tiende a cero de manera asintótica si se considera lo siguiente:

$$\dot{\hat{\boldsymbol{\pi}}} = -\boldsymbol{\Gamma}\mathbf{Y}^T(\mathbf{q}, \dot{\mathbf{q}}, \dot{\mathbf{q}}_r, \ddot{\mathbf{q}}_r)\mathbf{s}$$

Donde: $\boldsymbol{\Gamma}$ es una matriz diagonal de ganancias positivas.

Aunque no todo el estado $(\hat{\boldsymbol{\pi}}, \mathbf{s})$ tiende a cero de manera asintótica, solo para \mathbf{s} , se puede demostrar que $\hat{\boldsymbol{\pi}}$ tiende a cero si existe excitación persistente, es decir, que se excite al robot en todas las frecuencias posibles. Como consecuencia no se espera que los valores aproximados tiendan a los valores reales, pero sí se espera que el error de posición y velocidad tiendan a cero [Spong et al., 2006, p. 277].

4.3.5. Control por Modos Deslizantes

Como se mencionó anteriormente, los controladores no lineales anteriores dependen del modelo dinámico del sistema. Una manera de omitir el modelo y aplicar control, es posible si se conoce una cota máxima de la norma siguiente

$$\|\mathbf{H}\ddot{\mathbf{q}}_r + \mathbf{C}\dot{\mathbf{q}}_r + \mathbf{D}_v\dot{\mathbf{q}}_r + \mathbf{D}_s\text{sign}(\dot{\mathbf{q}}) + \mathbf{g}\| \leq \rho$$

De igual manera se parte desde el modelo pasivizado (4.23). A partir del análisis de estabilidad en el sentido de Lyapunov, considerando el peor de los casos se obtiene la siguiente ley de control:

$$\boldsymbol{\tau} = -\mathbf{K}_m \text{sign}(\mathbf{s}) \quad (4.26)$$

Donde: \mathbf{K}_m es una matriz de ganancias diagonal positiva definida y la función **sign** se define como:

$$\text{sign}(\mathbf{s}) = \begin{cases} 1 & \text{si } s > 0 \\ 0 & \text{si } s = 0 \\ -1 & \text{si } s < 0 \end{cases}$$

Para hacer más sencillo el análisis se considera una sola cota, pero si se desea un mejor rendimiento se puede considerar una cota para cada una de las articulaciones del robot y de esta manera se pueden sintonizar las ganancias \mathbf{K}_m a partir de $k_{mi} > \rho$

Si se desea profundizar en la teoría de control de robots manipuladores, se puede consultar la bibliografía [Sciavicco and Siciliano, 1996, p. 333], en ella se encuentra a detalle las demostraciones y demás aplicaciones de los diferentes tipos de controladores.

Capítulo 5

Resultados Experimentales

5.1. Algoritmo de Determinación de Parámetros

Antes de mostrar los resultados obtenidos de los diferentes tipos de controladores se va a describir la manera en que se desarrollaron los experimentos para la determinación de los parámetros del manipulador mediante el método propuesto en el Capítulo 4. Además se muestra una comparativa entre los datos obtenidos de manera experimental contra los datos obtenidos con los valores nominales.

5.1.1. Experimento para Determinar los Parámetros del Modelo Completo

Para llevar a cabo este experimento se tuvo que someter al manipulador robótico a una señal de control en lazo abierto, del cual se obtuvieron datos suficientes que permiten determinar los parámetros del manipulador. Dicha señal de control se compone de una suma de senos y cosenos con amplitudes diferentes y frecuencias distintas más un término constante que mantiene al manipulador en una posición de excitación diferente al punto de equilibrio y más cercano al punto de máximo par ejercido por la gravedad, esta señal de control es

$$\begin{aligned}\tau_1 &= 0.22 + 0.08 \sin(2\pi t) + 0.1 \sin\left(\frac{2\pi t}{0.9}\right) + 0.12 \sin\left(\frac{2\pi t}{0.7}\right) \\ \tau_2 &= 0.025 + 0.016 \cos(2\pi t) + 0.032 \cos\left(\frac{2\pi t}{0.9}\right) + 0.048 \cos\left(\frac{2\pi t}{0.7}\right)\end{aligned}\tag{5.1}$$

Con los datos obtenidos a partir de esta entrada de control en lazo abierto se logró

estimar los parámetros del manipulador, mediante la solución a la ecuación (4.3). Estos parámetros se expresa a continuación.

$$\begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \\ \pi_6 \\ \pi_7 \\ \pi_8 \\ \pi_9 \end{bmatrix} = \begin{bmatrix} 0.00559 \\ 0.00022 \\ 0.00148 \\ 0.01467 \\ 0.00303 \\ 0.05617 \\ 0.00374 \\ 0.20793 \\ 0.02807 \end{bmatrix} \quad (5.2)$$

5.1.2. Cálculo de Parámetros con Valores Conocidos

Para determinar los parámetros del modelo calculado es necesario conocer los valores nominales del manipulador los cuales se obtuvieron midiendo las propiedades de cada uno de los eslabones como lo es la masa, la longitud, el centro de masa y la aceleración de la gravedad. Estos valores nominales se presentan en la tabla siguiente.

Tabla 5.1: Valores nominales

Valor nominal	Cantidad	Unidad
Masa eslabón 1	0.148	Kg
Masa eslabón 2	0.069	Kg
Longitud eslabón 1	0.165	m
Longitud eslabón 2	0.285	m
Longitud centro de masa 1	0.094	m
Longitud centro de masa 2	0.110	m
Aceleración de la gravedad	9.8	m/s ²

$$\begin{bmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \\ \pi_5 \\ \pi_6 \\ \pi_7 \\ \pi_8 \\ \pi_9 \end{bmatrix} = \begin{bmatrix} 0.003522 \\ 0.001252 \\ 0.001302 \\ - - - - \\ - - - - \\ - - - - \\ - - - - \\ 0.2479 \\ 0.0744 \end{bmatrix} \quad (5.3)$$

5.1.3. Comparación entre Parámetros

Como se puede observar en la tabla siguiente existe una diferencia muy clara entre los parámetros calculados mediante el experimento en lazo abierto y los obtenidos mediante los valores nominales, esto puede deberse a muchos factores entre ellos la compensación de parámetros entre gravedad y fricción a la hora de estimarlos por medio del regresor, otra razón puede ser que los valores nominales de torque no sean exactos como se indica en la hoja de datos del fabricante o que la señal PWM no sea muy precisa, pero aun así las técnicas de control que dependen del modelo, mostraron resultados satisfactorios. En la Tabla 5.2 se muestra una comparativa entre parámetros. En los experimentos, específicamente en el de Par Calculado, se realizó seguimiento tanto para valores nominales como para valores estimados y la comparación del rendimiento que tuvo cada uno de ellos.

Con esto finalmente pasamos a los experimentos de control aplicados a la plataforma del manipulador. Se implementaron 6 controladores propuestos en el capítulo anterior, utilizando los parámetros de la Tabla 5.2 para los controladores que dependen del modelo dinámico.

Tabla 5.2: Tabala de parámetros identificados y calculados

Parámetro	Unidades	Estimados	Calculados	Error de estimación
π_1	Kgm ²	0.00559	0.003522	0.587
π_2	Kgm ²	0.00022	0.001252	0.824
π_3	Kgm ²	0.00148	0.001302	0.136
π_4	Kgm ² /rads	0.01467	— — —	— — —
π_5	Kgm ² /rads	0.00303	— — —	— — —
π_6	Nm	0.05217	— — —	— — —
π_7	Nm	0.00374	— — —	— — —
π_8	Nm	0.20793	0.2479	0.161
π_9	Nm	0.02807	0.0744	0.622

5.2. Control Regulación

El control por regulación permite mantener al manipulador en un punto fijo. Sin importar en donde inicie el manipulador, éste va a tender a una posición de referencia. En la Figura 5.1 se muestra en graficas el comportamiento de la posición de las dos articulaciones del manipulador para un control por regulación. Con diferentes colores se representan a los 6 controladores utilizados y con ello se puede comparar el desempeño que tuvo cada uno. Como se observa en las gráficas algunos controladores presentaron sobrepaso principalmente en la articulación 2. Este tipo de control es muy brusco y puede provocar daños físicos al manipulador, ya que el control reacciona de manera “instantánea” tratando de llevar al manipulador a la referencia sin importar en que posición esté, o se encuentre. Como consecuencia entre más alejado se encuentre el robot de la referencia más grande será la señal de control que se requiera para corregir el error y el manipulador tendrá que recibir un impacto mayor en las articulaciones. A continuación se muestran graficas correspondientes a la posición angular de las dos articulaciones, así como el error de posición y la señal de control.

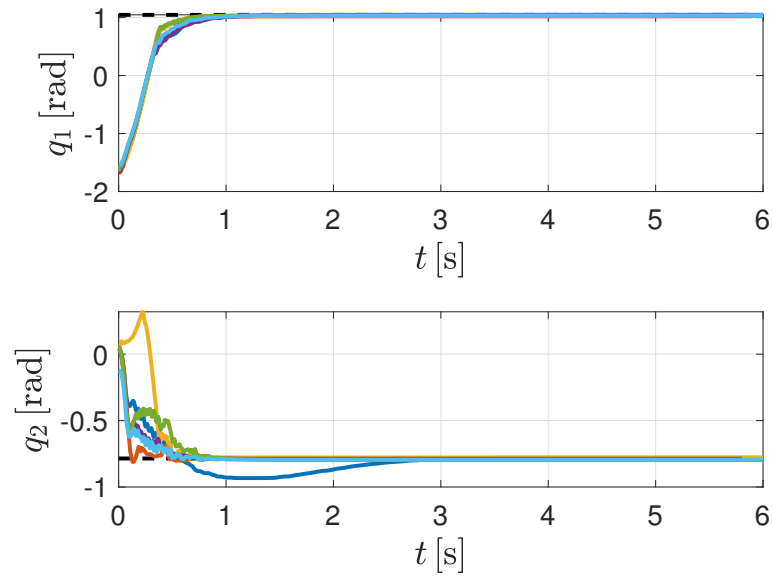


Figura 5.1: Regulación q_1 y q_2 : Deseada (---), PID (—), PD+G (—), Par Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).

Para observar el buen desempeño de un controlador la mejor manera es analizando el error de posición que presenta. Con ello podemos determinar qué tan bueno es un controlador cuando se desea que llegue a la referencia deseada. También nos dice en que puntos la señal de control presentó el máximo valor cuando se trata de regulación. El error tiende a cero conforme la posición tiende al valor de referencia. Por esta razón, el comportamiento de la señal de posición y el error son similares. Todo esto se visualiza en las gráficas de la Figura 5.2.

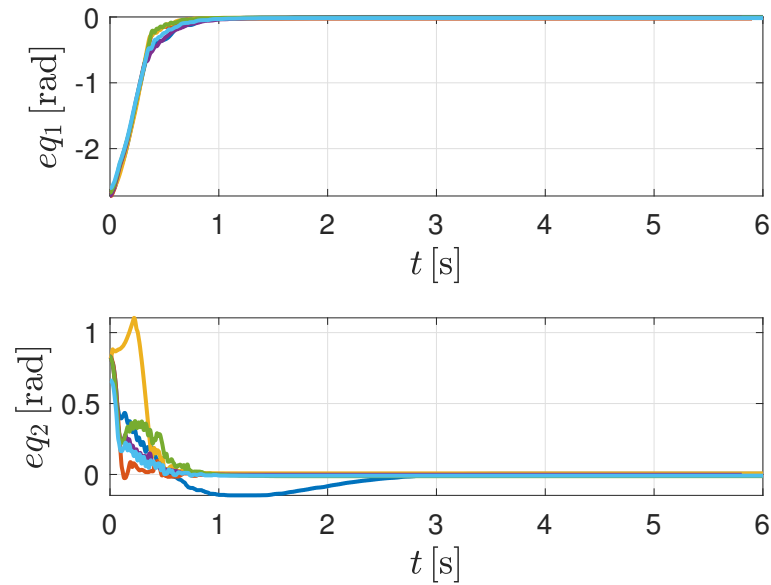


Figura 5.2: Error de Posición eq_1 y eq_2 : PID (—), PD+G (—), Par Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).

El par que entrega cada uno de los motores con los 6 controladores se representa en las gráficas de la Figura 5.3. En estas gráficas se puede observar que conforme avanza el tiempo el control tiende a cero, aunque nunca llega a ser cero, esto traducido a posición significa que el manipulador llegó a la referencia, por lo tanto no requiere de un par muy grande para mantenerlo en esa posición, a diferencia de cuando inició, que demandó un par bastante grande para hacerlo llegar al valor de referencia. Como se puede observar el control comenzó con valores bastante grandes pero eso no significa que los motores entregaran un par tan grande, dada la saturación incluida en la programación.

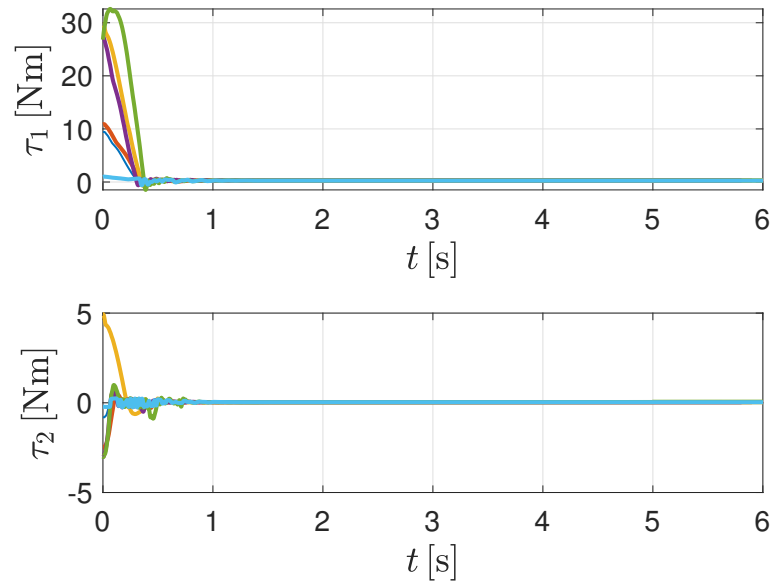


Figura 5.3: Par τ_1 y τ_2 : PID (—), PD+G (—), Par Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).

La gráfica de la Figura 5.4 muestra la evolución de los parámetros calculados por el control adaptable en regulación. Con esto se muestran las características del modelo dinámico, o mejor dicho, lo que interpreta el control adaptable como modelo dinámico. En esta gráfica es donde se ve presente la ley de adaptación. Conforme el control avanza los parámetros cambian y tienden a valores constantes, que son suficientes para llevar y mantener estable al manipulador en ese punto que es la referencia.

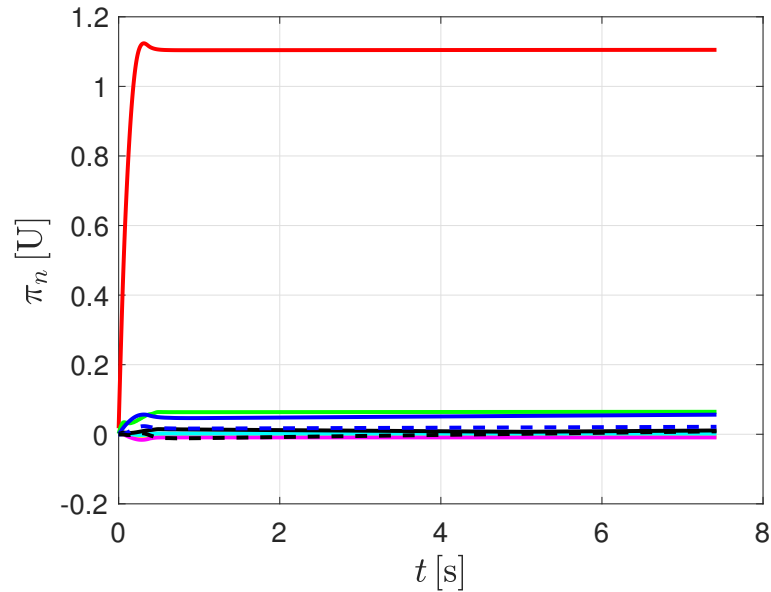


Figura 5.4: Parámetros: π_1 (—), π_2 (—), π_3 (—), π_4 (—), π_5 (—), π_6 (—), π_7 (—), π_8 (--), π_9 (--).

5.3. Control de Seguimiento de Trayectoria Articular

En las gráficas de la Figura 5.5 se presenta la posición en control de seguimiento de trayectoria de las articulaciones del manipulador, para el que se aplicaron de nuevo los 6 algoritmos de control descritos anteriormente. En este caso el control, a diferencia del control de regulación, es más suave ya que en lugar de tratar de corregir la posición de un punto a otro de manera brusca, lo hace siguiendo una serie de puntos deseados hasta llegar al valor final. Por lo tanto solo tiene que corregir el error en un intervalo muy pequeño haciendo que la señal de control sea en magnitud más pequeña, como se observa en las gráficas. Cada control se representa con un color y cada uno de ellos presenta resultados diferentes. Por ejemplo, el control PID y el control por Modos Deslizantes se alejan de la trayectoria deseada principalmente en la articulación 2, lo que puede ser causado por la ganancias de sintonización del controlador o por la influencia que tiene la primera articulación sobre la segunda articulación en el momento en el que se efectúa el control. En este experimento se aplicó lo visto en la Tabla 5.2, donde se comparan los parámetros calculados y los parámetros identificados. Lo anterior fue aplicado principalmente al control Par Calculado, que es el control que más depende del modelo dinámico. Por lo tanto es el que más se ve afectado por el valor de los parámetros y por esta misma razón es que se aplicó este mismo control para

las dos condiciones. Ambos mostraron resultados satisfactorios para las dos articulaciones, pero se observa que el Par Calculado para la articulación 2 con valores nominales fue mejor que el control con parámetros identificados.

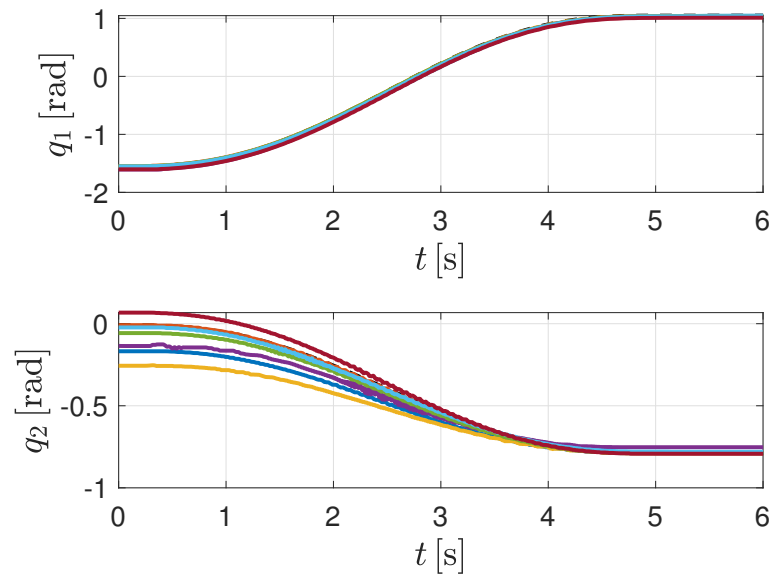


Figura 5.5: Trayectoria de q_1 y q_2 : Deseada (---), PID (—), PD+G (—), Par Calculado Identificado (—), Par Calculado Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).

En el error se ve de manera clara el comportamiento del control Par Calculado, para valores identificados y valores nominales, como se muestra en la Figura 5.6. Comparando el error se puede ver qué tan aproximada fue la trayectoria que efectuó el manipulador comparada con la trayectoria deseada. Se observa en este caso que el error tiende a cero conforme el manipulador va siguiendo la trayectoria. En algunos puntos se presentó una variación un poco más grande, es decir, el manipulador no fue capaz de alcanzar dichos valores. Aún así la variación fue mínima en comparación con el valor deseado, logrando alcanzar el objetivo de control.

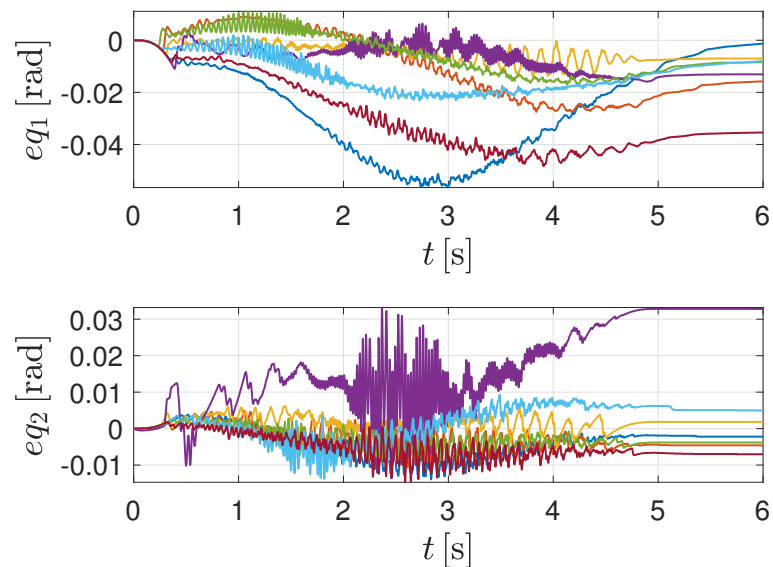


Figura 5.6: Error de Posición eq_1 y eq_2 : PID (—), PD+G (—), Par Calculado Identificado (—), Par Calculado Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).

En la Figura 5.7 se muestran las gráficas de la señal de control para seguimiento de la trayectoria articular. Como se observa, estas señales de control presentan valores mucho más pequeños que el control por regulación, pero más grandes que el control por seguimiento Cartesiano, ya que la trayectoria que siguieron estos controladores requirió de mover al manipulador desde el punto de inicio hasta el punto final, necesitando un mayor par ya que el punto de inicio es el punto de mínima energía potencial y finalizó en un punto de mayor energía potencial, además en un tiempo preestablecido que se asigna en el diseño de la trayectoria. No obstante, como se vio en la graficas de posición, se obtuvieron resultados satisfactorios.

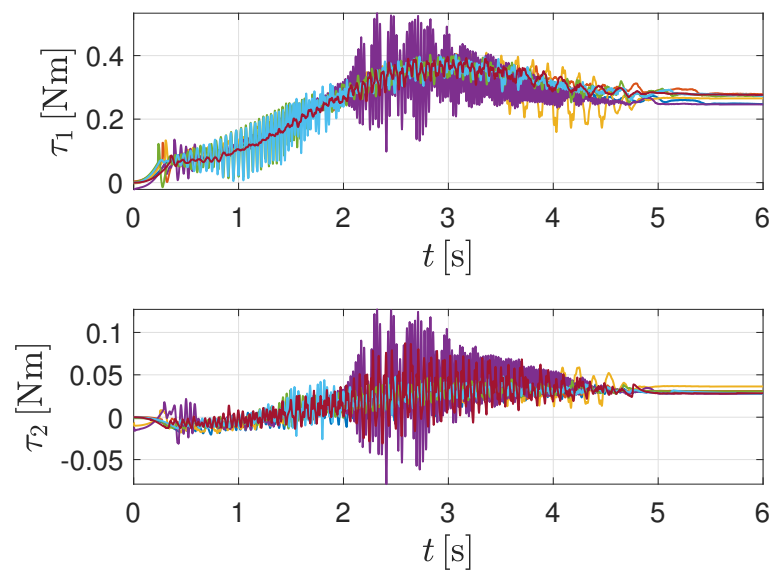


Figura 5.7: Par τ_1 y τ_2 : PID (—), PD+G (—), Par Calculado Identificado(—), Par Calculado Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).

Por último, tenemos la gráfica de la evolución del valor de los parámetros calculados por el control adaptable en seguimiento de trayectoria articular, mostrados en la Figura 5.8. Estos parámetros, al igual que en el control anterior, van cambiando conforme el control se va desarrollando, hasta llegar a un punto donde permanecen en un valor constante, que se interpreta como los parámetros del modelo dinámico. Se puede observar que son muy diferentes a los obtenidos por control de regulación.

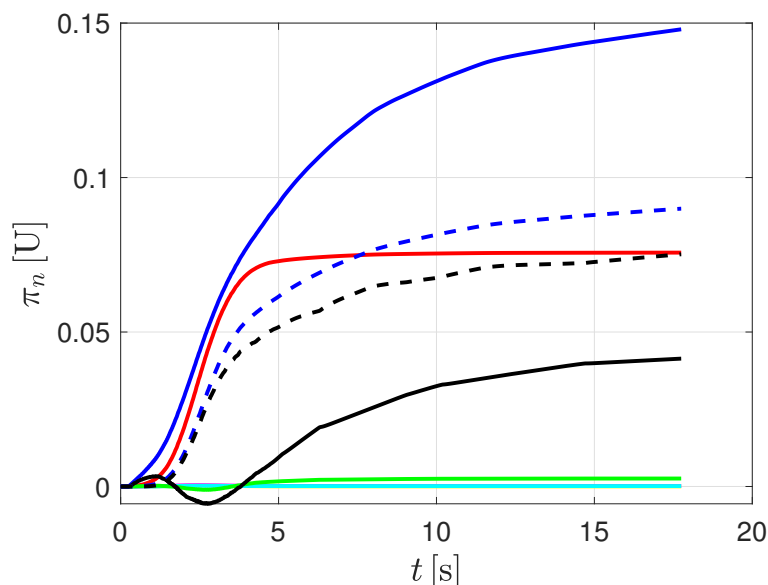


Figura 5.8: Parámetros: π_1 (—), π_2 (—), π_3 (—), π_4 (—), π_5 (—), π_6 (—), π_7 (—), π_8 (--), π_9 (--).

5.4. Control de Seguimiento de Trayectoria Cartesiana

El control de seguimiento de trayectoria cartesiana, es un experimento que muestra de manera clara el desempeño que tuvo cada uno de los seis controladores, ya que el objetivo de control es el seguimiento de una línea recta para el efector final. En primera instancia podemos observar en las gráficas de la Figura 5.9, que el control PID y el control por Modos Deslizantes, si bien funcionaron, no fueron capaces de seguir la trayectoria propuesta, ya que presentan mucha variación. El resto de controladores aunque también presentaron variaciones lograron seguir el objetivo de control propuesto y se acercaron más al valor deseado. Para este control se requirió de aplicar previamente control por seguimiento de trayectoria articular, para así llevar al manipulador al punto de inicio del control de seguimiento Car-

tesiano, y de esta manera evitar pasar al manipulador por alguna singularidad. Por lo tanto en las gráficas solo se muestra esta parte del control, lo demás ya se mostró en la Sección 5.3.

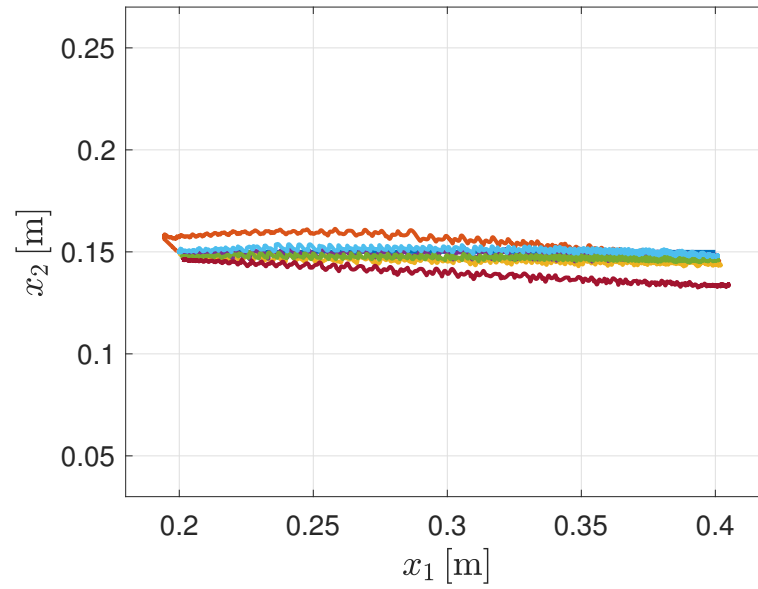


Figura 5.9: Trayectoria: Deseada (—), PID (—), PD+G (—), Par Calculado (—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).

El error en el seguimiento Cartesiano muestra qué tan diferentes fueron las posiciones Cartesianas con respecto a la trayectoria deseada. En ella se reafirmó que los peores controladores fueron el PID y el control por Modos Deslizantes y que los mejores controladores fueron el Adaptable y el Pasivo. Al igual que en el control por regulación y seguimiento articular se espera que el error tienda a cero, lo que nos muestran las gráficas de la Figura 5.10.

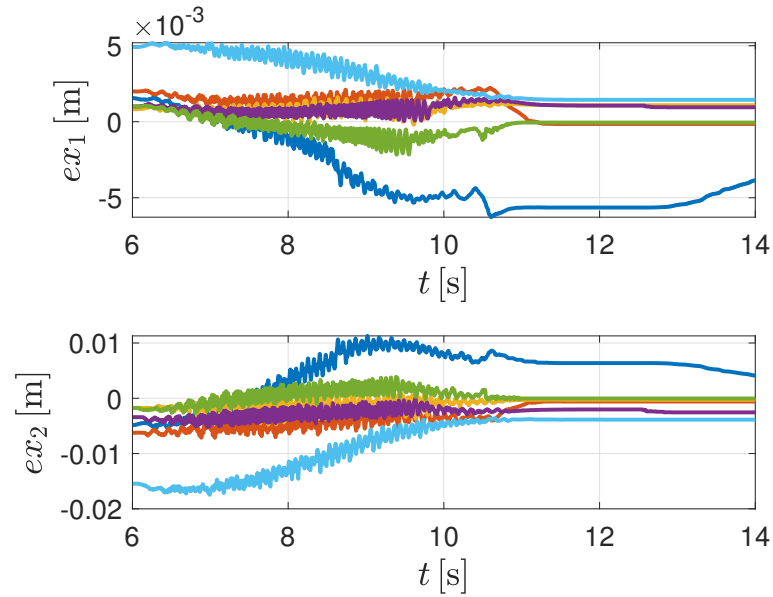


Figura 5.10: Error de Posición ex_1 y ex_2 : PID (—), PD+G (—), Par Calculado(—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).

En las señales de control de la Figura 5.11 se muestra que la magnitud del control es pequeña, comparado con el experimento de regulación. Esto se debe a que el control no tuvo que realizar alguna corrección muy grande de la posición, ya que al igual que el seguimiento articular va pasando una serie de puntos en los cuales no hay variaciones muy grandes entre ellos por lo que el control responde de manera más suave. Finalmente con estas gráficas se logra ver que el par necesario para realizar seguimiento Cartesiano es relativamente menor que para los otros experimentos.

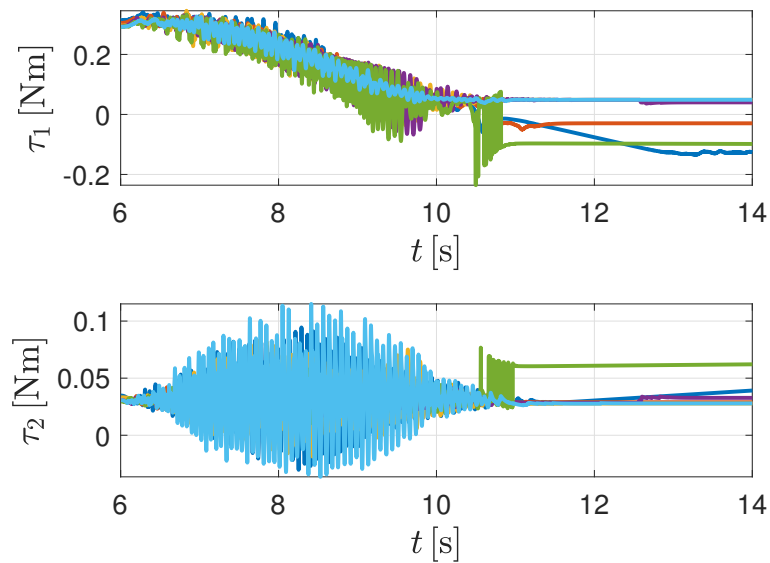


Figura 5.11: Par τ_1 y τ_2 : PID (—), PD+G (—), Par Calculado(—), Pasivo (—), Adaptable (—), Modos Deslizantes (—).

Para el caso de los parámetros obtenidos con el control adaptable, se muestra solo el comportamiento del control para el seguimiento Cartesiano ya que el seguimiento articular que permitió llevar al manipulador a la posición de inicio ya se describió en la Sección 5.3. En la Figura 5.12 solo se muestra cómo evolucionaron los parámetros después de que iniciara el control de seguimiento Cartesiano. Como se puede observar el cambio fue mínimo ya que el control no tuvo que iniciar todo el proceso de cálculo de los parámetros, si no que utilizó la referencia de seguimiento articular.

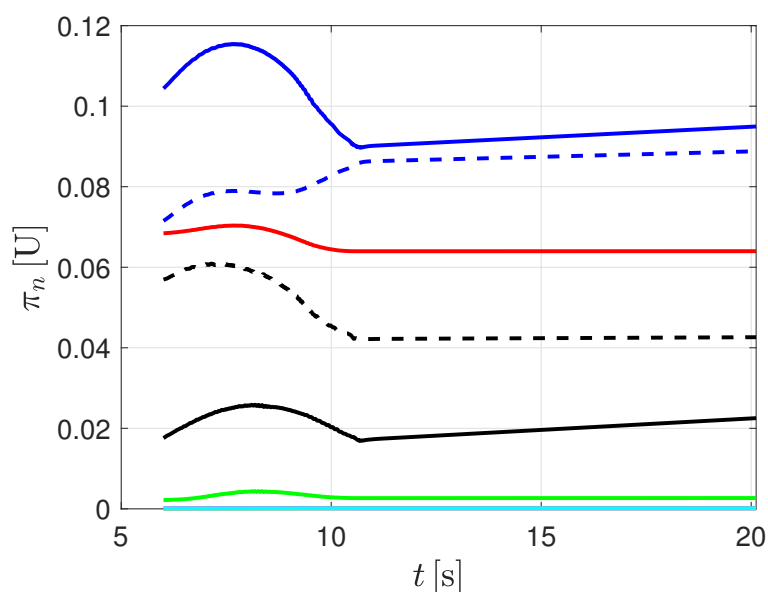


Figura 5.12: Parámetros: π_1 (—), π_2 (-·-), π_3 (—), π_4 (-·-), π_5 (-·-), π_6 (—), π_7 (-·-), π_8 (-·-), π_9 (—).

5.5. Caracterización de la Posición del Efector Final

Para determinar si el efector final del manipulador, llega al punto deseado en el espacio de trabajo, es necesario realizar un experimento donde se aplique una misma ley de control en repetidas ocasiones y medir la posición del efector final. Para verificar si existe una desviación con respecto al punto deseado y determinar mediante mediciones si lo realiza después de varios experimentos bajo las mismas condiciones.

El experimento propuesto consiste en aplicar 5 veces una ley de control Lineal y otra de control no lineal, midiendo la posición del efector final, para fines demostrativos se utiliza el algoritmo de control lineal PID y el de control no lineal Adaptable, para experimentos de

regulación y seguimiento articular. Dicho experimento se aplica en estado estacionario, que es cuando la ley de control ya surtió efecto y se alcanzaron los valores deseados.

Control Regulación

Tabla 5.3: Posición del efector final control por regulación

Punto PID	x	y	Punto Adaptable	x	y
P_{ref}	37.46	13.02		37.46	13.02
P_1	39.4	10.5		37	13.9
P_2	38.1	11.4		36.9	13.6
P_3	38.3	12		37.5	13.2
P_4	37.8	11.9		37.5	12.9
P_5	38.2	12		38.1	12.8

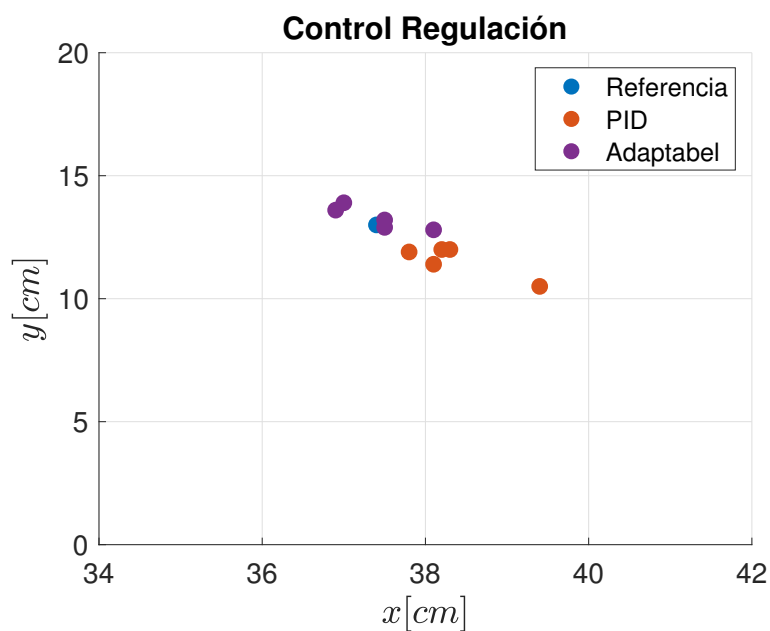


Figura 5.13: Posición del Efector Final Control de Regulación

Como se observa en la Figura 5.13 el efector final cae más cerca del valor de referencia para el control Adaptable en experimentos de regulación.

Control Seguimiento

Tabla 5.4: Posición del efector final control por seguimiento

Punto PID	x	y	Punto Adaptable	x	y
P_{ref}	37.46	13.02		37.46	13.02
P_1	37.3	14		37.6	14.1
P_2	37.6	13		37.1	14.6
P_3	38	12.5		37.5	13.9
P_4	37.8	12.3		37.7	13.5
P_5	37.6	12.5		37.2	12.8

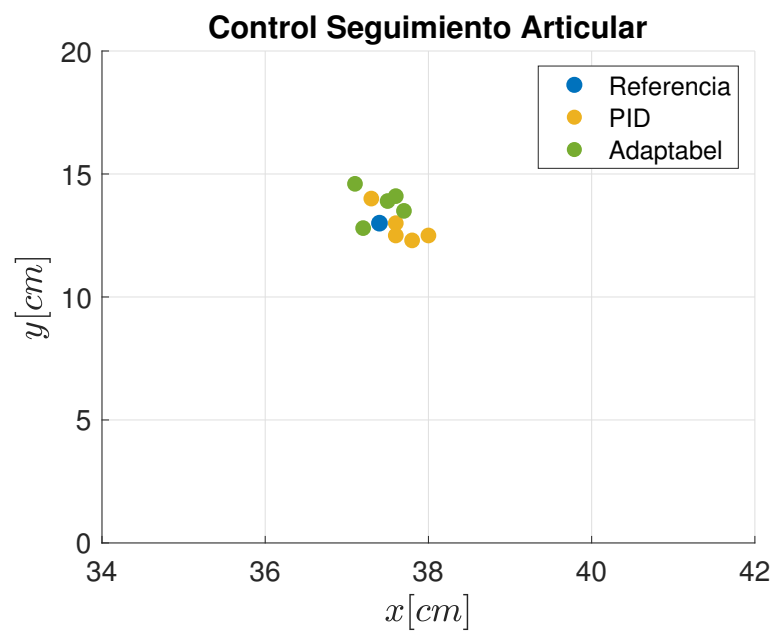


Figura 5.14: Posición del Efector Final Control de Seguimiento

En el caso de control de seguimiento articular el efector final se acerca más al punto de referencia tanto para experimentos de control PID como Adaptable.

Capítulo 6

Conclusión

6.1. Construcción del Manipulador

Se lograron cumplir los objetivos establecidos al diseñar, construir y aplicar los algoritmos de control a la plataforma, con resultados satisfactorios, desarrollados en los Capítulos 2 y 5. Por lo tanto, la plataforma cumple con las características necesarias para ser utilizada con fines de enseñanza e investigación. El diseño de las tarjetas es compacto, modular por lo que el manipulador es portátil y como consecuencia se puede escalar a un número mayor de grados de libertad y de bajo costo. El software donde se diseñó la interfaz de usuario es flexible ya que el lenguaje y la plataforma que se utilizó para su desarrollo es accesible y robusto. Además, el diseño de la interfaz es sencillo e intuitivo por lo tanto no se requiere de capacitación exhaustiva para su utilización. Si se desea programar un controlador o un observador diferente a los presentados, se puede realizar con relativa facilidad, con el único requisito de conocimiento básico del lenguaje C/C++. Por último el diseño mecánico influye bastante en la posición del manipulador.

6.2. Comparación de Algoritmos

Los controladores aplicados a la plataforma muestran resultados satisfactorios, con lo que se demuestra la funcionalidad de la plataforma para la realización de experimentos de control aplicado a la robótica de manipuladores, para experimentos de regulación y seguimiento en coordenadas articulares o Cartesianas. Comparando las señales de control se puede dar por hecho que el control lineal PID no muestra buenos resultados comparado con

los controladores no lineales ya que es superado por mucho y esto se nota con gran claridad en el error de posición de todos los experimentos. Más aun, en el seguimiento Cartesiano sigue la línea recta propuesta como trayectoria. Aunque no todos los controladores no lineales mostraron resultados satisfactorios, como en el caso del control por Modos Deslizantes que si bien funcionó y tuvo un rendimiento similar al PID, no es lo que se esperaba para un control no lineal. Esto puede deberse a diversas causas como el tiempo de muestreo del sistema de adquisición y control, que es lo que usualmente le afecta más a un control de este tipo. El controlador que presentó los mejores resultados aplicados a esta plataforma fue el control adaptable, con errores de posición muy pequeños, para experimentos de regulación y seguimiento de trayectoria articular y Cartesiana que dieron resultados muy cercanos a lo deseado. Finalmente, podemos concluir que la plataforma es capaz de efectuar las técnicas de control más conocidas en la literatura y por lo tanto ser utilizada en la investigación o incluso por alumnos que deseen implementar técnicas de control reportadas en actividades relacionadas con el tema.

6.3. Áreas de Oportunidad

Como desarrollo a futuro se propone la posibilidad de utilizar el manipulador con más de dos grados de libertad. Esta posibilidad es inmediata dado que debido a las condiciones de diseño, la plataforma es capaz de controlar hasta 28 motores.

Otra opción es modificar el manipulador de tal manera que se pueda utilizar para experimentos de robots del tipo *acrobot* y *pendubot*.

A futuro se plantea diseñar y aplicar más experimentos de caracterización del manipulador tanto para la posición del efector final como para el rendimiento real de las piezas mecánicas, eléctricas y electrónicas que constituyen la plataforma, como consecuencia de esto se propone diseñar un manual de usuario para su utilización y en complemento un instructivo para replicar y construir dicha plataforma.

Una posible mejora sería reducir el tamaño de la plataforma para que se pueda integrar en cualquier lugar con un espacio reducido y convertir cada módulo de adquisición y control en un diseño inspirado en la tarjeta de adquisición NI cRIO, donde todo se trabaja con slots acumulables que se pueden introducir o extraer de manera sencilla. Finalmente se espera que dicha plataforma se utilice con fines de enseñanza e investigación en alguna universidad o centro de investigación.

Apéndice A

Código de Microcontroladores PIC

Código PIC18F2331

```
1  /*
2  * File:   PIC18F2331
3  * Author: Julio César Flores Díaz, Iván Alejandro Gutiérrez Giles
4  */
5
6  #include <xc.h>
7  #include "config.h"
8  #include <math.h>
9
10 #define _XTAL_FREQ 20000000
11
12 unsigned char voltageL = 0, voltageH = 0, encoderL = 0, encoderH = 0,
13             encoderM=0;
14 short voltage = 0;
15 unsigned long enc=0, enc1=0;
16 unsigned int i=0;
17 unsigned short maxcounts=0 ;
18 unsigned long data=0;
19 void __interrupt(high_priority) SPI_Slave_Read(void) {
20     if(SSPIF == 1)
21     {
22
23         voltageL = SSPBUF;    // Recibe datos del buffer
24     }
```

```

25     encoderL = (unsigned char)(enc); //Envia datos del encoder
26     SSPBUF = encoderL;      /
27     SSPIF = 0;             //Limpia bandera de transmisión completa
28     while(!SSPIF);
29
30     voltageH = SSPBUF;     // Recibe datos del buffer
31
32     encoderM = (unsigned char)(enc>>8); //voltageH;
33     SSPBUF=encoderM;
34     SSPIF = 0;
35     while(!SSPIF);
36     char dummy = SSPBUF;
37
38     encoderH= (unsigned char)(enc>>16); //voltageH;
39     SSPBUF=encoderH;
40     SSPIF = 0;
41     while(!SSPIF);
42
43     dummy = SSPBUF;      // Lee los residuos del buffer
44
45
46     voltage = (((short)voltageH)<<8)|voltageL; //Voltaje de 16 bits
47
48
49     if(voltage==0)      // Testigo de comunicación SPI
50     LATAbits.LA0 ^= 1;
51
52     SSPIF = 0;
53 }else if(IC2QEIF==1)
54 {
55     i ^=1;
56
57     if(i==0)
58     {
59         enc1=0;
60     }else
61     {
62         enc1=maxcounts;
63     }
64

```

```

65     IC2QEIF=0;
66     }
67     return ;
68 }
69
70 void initialize(void)
71 {
72     TRISA = 0b11111110; // Test LED
73     LATA = 0x00;
74     ANSEL0 = 0x00;
75
76     TRISCBits.TRISC5 = 1; //SCK como entrada
77     TRISCBits.TRISC6 = 1; // SS esclavo
78     TRISCBits.TRISC7 = 0; // SDO
79     TRISCBits.TRISC4 = 1; // SDI
80
81     SSPSTAT = 0b00000000; // SPI_IDLE;
82     SSPCON1 = 0b00100100;
83
84     QEICON = 0b10001000; // Configuración de la interfaz de encoder de
           cuadratura
85     maxcounts =512*103;// PPR x 4 for the quadrature x gear reduction
86     MAXCNL = (unsigned char)maxcounts;
87     MAXCNH = (unsigned char)(maxcounts>>8);
88
89     POSCNTL = 0;
90     POSCNTH = 0;
91
92     GIE = 1; //Habilitar interrupcion por SSPIF
93     PEIE = 1;
94     SSPIF = 0;
95     SSPIE = 1;
96     IC2QEIE=1;//Habilitar interrupcion por IC2QEIF
97     IC2QEIF=0;
98
99
100    TRISBbits.TRISB1 = 0; // PWM salida
101    TRISBbits.TRISB4 = 0; // Motor dirección
102    PTCON0 = 0b00000000;
103    PWMCON0 = 0b00010011; // PWM0 and PWM1 independientes

```

```

104     PTPERH = 0x01;
105     PTPERL = 0xFF; // PWM Freq = (Fosc/4)/(PTPER+1) = 5E6/1024 = 4.882 Khz
106     PTEN = 1; // Habilitación del reloj
107
108     return;
109 }
110
111 void main(void) {
112
113     initialize();
114
115     while(1)
116     {
117
118         //Determinación de la posición a mayor número de cuentas
119         data=0xFFFF&(POSCNTH<<8)|POSCNTL;
120         enc=enc1+data;
121
122         //Dirección de movimiento
123         if(voltage >=0)
124             LATBbits.LATB4 = 1;
125         else
126             LATBbits.LATB4 = 0;
127         //Generación de PWM
128         unsigned short duty = abs(voltage)/8; 12 bits de resolución
129         PDC0L = duty;
130         PDC0H = duty>>8;
131         __delay_us(10);
132
133     }
134 }

```

Código PIC18F4550

```

1     switch( ReceivedDataBuffer [0]) //revisa los datos en elbuffer USB
2     {
3         case COMMAND_TOGGLE_LED: //LED en modo Toggle
4             LED_Toggle(LED_USB_DEVICE_STATE);
5             break;
6
7         case COMMAND_SET_VOLTAGE:

```

```

8      {
9      unsigned char voltageL1 = 0, voltageH1 = 0, encoderL1 = 0, encoderH1 =
        0, encoderM1=0;
10     unsigned char voltageL2 = 0, voltageH2 = 0, encoderL2 = 0, encoderH2 =
        0, encoderM2=0;
11     voltageL1 = ReceivedDataBuffer [1];
12     voltageH1 = ReceivedDataBuffer [2];
13     voltageL2 = ReceivedDataBuffer [3];
14     voltageH2 = ReceivedDataBuffer [4];
15
16     unsigned char dummy;
17
18     LATDbits.LD0 = 0; // Selecciona esclavo 1
19     __delay_us(10);
20
21     SSPBUF=voltageL1; // Copia datos en SSBUF a transmitir */
22     while(!SSPIF); //Espera a que se complete la transmisión */
23     SSPIF=0; // limpia la bandera */
24     dummy = SSPBUF;
25     __delay_us(10);
26     SSPBUF=voltageH1; // Copia datos en SSBUF a transmitir */
27     while(!SSPIF); //Espera a que se complete la transmisión */
28     SSPIF=0; // limpia la bandera */
29     encoderL1=SSPBUF;
30     __delay_us(10);
31     SSPBUF = dummy;
32     while(!SSPIF); //Espera a que se complete la transmisión */
33     SSPIF=0; // limpia la bandera */
34     encoderM1=SSPBUF;
35     __delay_us(10);
36     SSPBUF = dummy;
37     while(!SSPIF); //Espera a que se complete la transmisión */
38     SSPIF=0; // limpia la bandera */
39     encoderH1=SSPBUF;
40     LATDbits.LD0 = 1;
41
42     LATDbits.LD1 = 0; // Selecciona esclavo 2
43     __delay_us(10);
44
45     SSPBUF=voltageL2; // Copia datos en SSBUF a transmitir */

```

```

46     while(!SSPIF); /*Espera a que se complete la transmisión */
47     SSPIF=0;      /* limpia la bandera */
48     dummy = SSPBUF;
49     __delay_us(10);
50     SSPBUF=voltageH2; /* Copia datos en SSBUF a transmitir */
51     while(!SSPIF); /*Espera a que se complete la transmisión */
52     SSPIF=0;      /* limpia la bandera */
53     encoderL2=SSPBUF;
54     __delay_us(10);
55     SSPBUF = dummy;
56     while(!SSPIF); /*Espera a que se complete la transmisión */
57     SSPIF=0;      /* limpia la bandera */
58     encoderM2=SSPBUF;
59     __delay_us(10);
60     SSPBUF = dummy;
61     while(!SSPIF); /*Espera a que se complete la transmisión */
62     SSPIF=0;      /* limpia la bandera */
63     encoderH2=SSPBUF;
64     LATDbits.LD1 = 1;
65
66     if(!HIDTxHandleBusy(USBInHandle))
67     {
68         ToSendDataBuffer[0] = 0x82;
69         ToSendDataBuffer[1] = encoderL1;
70         ToSendDataBuffer[2] = encoderM1;
71         ToSendDataBuffer[3] = encoderH1;
72         ToSendDataBuffer[4] = encoderL2;
73         ToSendDataBuffer[4] = encoderM2;
74         ToSendDataBuffer[4] = encoderH2;
75         //Prepara al modulo USB para el envio de datos
76         USBInHandle = HIDTxPacket(CUSTOM_DEVICE_HID_EP, (uint8_t*)&
77             ToSendDataBuffer[0],64);
78     }
79     break;
80
81 }

```


Apéndice B

Código de Visual Studio

Código función transmisión de datos

```
1 void usbHidDaqSendAndRead(long voltage1 , long voltage2 , unsigned int &
   encoder1 , unsigned int &encoder2 , HANDLE& hndl)
2 {
3
4     DWORD BytesWritten = 0;
5     DWORD BytesRead = 0;
6     unsigned char OutputPacketBuffer[65]; //Allocate a memory buffer equal to
   our endpoint size + 1
7     unsigned char InputPacketBuffer[65]; //Allocate a memory buffer equal to
   our endpoint size + 1
8
9     OutputPacketBuffer[0] = 0; //The first byte is the "Report ID". This
   number is used by the USB driver, but does not
10 //get transmitted accross the USB bus. The custom HID class firmware is
   only configured for
11 //one type of report, therefore, we must always initialize this byte to
   "0" before sending
12 //a data packet to the device.
13
14     OutputPacketBuffer[1] = 0x82; //0x82 is the "Set Voltage" command in the
   firmware
15 //For simplicity, we will leave the rest of the buffer uninitialized, but
   you could put real
16 //data in it if you like.
17
```

```

18   OutputPacketBuffer [2] = (unsigned char)voltage1;
19   OutputPacketBuffer [3] = (unsigned char)(voltage1 >> 8);
20   OutputPacketBuffer [4] = (unsigned char)voltage2;
21   OutputPacketBuffer [5] = (unsigned char)(voltage2 >> 8);
22   WriteFile(hndl, &OutputPacketBuffer, 65, &BytesWritten, 0); //Blocking
      function, unless an "overlapped" structure is used
23   ReadFile(hndl, &InputPacketBuffer, 65, &BytesRead, 0); //Blocking
      function, unless an "overlapped" structure is used
24
25   //InputPacketBuffer[0] is the report ID, which we don't care about.
26   //InputPacketBuffer[1] is an echo back of the command.
27   //InputPacketBuffer[2-7] contains the encoder data.
28   encoder1 = InputPacketBuffer [2] + (InputPacketBuffer [3] << 8)+ (
      InputPacketBuffer [4] << 16);
29   encoder2 = InputPacketBuffer [5] + (InputPacketBuffer [6] << 8)+ (
      InputPacketBuffer [7] << 16);
30   return ;
31
32 }

```

Código función Acondicionamiento

```

1   void CUSBServoDlg::SendAndReceivedTorqueAndPosition(double torque1, double
      torque2, double &q1, double &q2)
2   {
3       const double torque1max = 0.7, torque2max = 0.125;
4       /*if (torque1 > 0) torque1 += 5;
5       if (torque1 < 0) torque1 -= 5;
6       if (torque2 > 0) torque2 += 0;
7       if (torque2 < 0) torque2 -= 0;*/
8
9       if (torque1 > torque1max) torque1 = torque1max;
10      if (torque1 < -torque1max) torque1 = -torque1max;
11      if (torque2 > torque2max) torque2 = torque2max;
12      if (torque2 < -torque2max) torque2 = -torque2max;
13
14
15      float T1 = 0, T2 = 0;
16      T1 = -torque1;//El cambio de signo es para cambiar el sentido de giro del
      eje z
17      T1 = (T1 / 1.133) * 17000.0;//1.133 torque maximo del motor por reduccion

```

```

    de engranajes
18  T2 = -torque2; //El cambio de signo es para cambiar el sentido de giro del
    eje z
19  T2 = (T2 / 0.1947) * 17000.0; //0.1947 torque maximo del motor por
    reduccion de engranajes
20
21  unsigned int encoder1 = 0, encoder2 = 0;
22  usbHidDaqSendAndRead(T1, T2, encoder1, encoder2, devHandle);
23  q1 = encoder1;
24  q1 = (-q1 / 105472) * 360.0 - 90.0; //El cambio de signo es para cambiar el
    sentido de giro del eje z
25  q2 = encoder2;
26  q2 = (-q2 / 105472) * 360.0; //El cambio de signo es para cambiar el
    sentido de giro del eje z
27
28  if (q1 > 180.0) q1 -= 360.0;
29  if (q1 <= -180.0) q1 += 360.0;
30  if (q2 > 180.0) q2 -= 360.0;
31  if (q2 <= -180.0) q2 += 360.0;
32
33  q1 *= pi / 180.0;
34  q2 *= pi / 180.0;
35
36  return ;
37
38 }

```

Código función de tecnicas de control

```

1  void CALLBACK CUSBServoDlg::ControlTimerProc(UINT uID, UINT uMsg, DWORD
    dwUser, DWORD dw1, DWORD dw2)
2  {
3      static long t_initial = timeGetTime();
4      double t = (timeGetTime() - t_initial) / 1000.0;
5      static double torque1 = 0.0, torque2 = 0.0;
6      double q1, q2;
7      SendAndReceivedTorqueAndPosition(torque1, torque2, q1, q2);
8
9
10     double q1p = 0.0, q2p = 0.0;
11     static double q1a = q1, q2a = q2;

```

```

12
13 q1p = (q1 - q1a) / T;
14 q2p = (q2 - q2a) / T;
15 const double lamfil = 10.0;
16 static double q1f = q1, q2f = q2;
17 double q1fp = lamfil * (q1 - q1f);
18 q1f += q1fp * T;
19 double q2fp = lamfil * (q2 - q2f);
20 q2f += q2fp * T;
21
22 static double q1pf = 0.0, q2pf = 0.0;
23 double q1fpp = lamfil * (q1fp - q1pf);
24 q1pf += q1fpp * T;
25 double q2fpp = lamfil * (q2fp - q2pf);
26 q2pf += q2fpp * T;
27
28 double xd[2] = { 0.0 }, xdp[2] = { 0.0 }, xdpp[2] = { 0.0 };
29 const double aDH1 = 0.165, aDH2 = 0.285;
30 double x[2] = {aDH1*cos(q1)+aDH2*cos(q1+q2),aDH1 * sin(q1) + aDH2 * sin(q1
    + q2) };
31 double Ja[2][2] = { {-aDH2 * sin(q1 + q2) - aDH1 * sin(q1),-aDH2 * sin(q1
    + q2)},{aDH2 * cos(q1 + q2) + aDH1 * cos(q1),aDH2 * cos(q1 + q2)} };
32 double qpf[2] = { q1fp, q2fp };
33 double xpf[2];
34 MatrixMul(Ja, qpf, 2, 2, 1, xpf);
35 static double th[9] = { 0.0 };
36
37 //seguimiento
38 double qd[2]={q1,q2}, qdp[2] = {0.0}, qdpp[2] = {0.0};
39
40 if (CartesianasType == 0)
41 {
42     const double qfin[2] = { 60.0 * pi / 180.0, -45.0 * pi / 180.0 }, tf =
        5.0;
43     static double q1ini = q1, q2ini = q2;
44     const double a0[2] = { q1ini, q2ini }, a1[2] = { 0.0,0.0 }, a2[2] = {
        0.0,0.0 }, a3[2] = { 10.0 * (qfin[0] - q1ini) / (tf * tf * tf) ,10.0
        * (qfin[1] - q2ini) / (tf * tf * tf) }, a4[2] = { -15.0 * (qfin[0]
        - q1ini) / (tf * tf * tf * tf),-15.0 * (qfin[1] - q2ini) / (tf * tf
        * tf * tf) }, a5[2] = { 6.0 * (qfin[0] - q1ini) / (tf * tf * tf * tf

```

```

    * tf),6.0 * (qfin[1] - q2ini) / (tf * tf * tf * tf * tf) };
45 for (int i = 0; i < 2; i++)
46 {
47     if (t <= tf && TrayectType == 1)
48     {
49         qd[i] = a0[i] + a1[i] * t + a2[i] * t * t + a3[i] * t * t * t + a4[i]
           ] * t * t * t * t + a5[i] * t * t * t * t * t;
50         qdp[i] = a1[i] + 2.0 * a2[i] * t + 3.0 * a3[i] * t * t + 4.0 * a4[i]
           * t * t * t + 5.0 * a5[i] * t * t * t * t;
51         qdpp[i] = 2.0 * a2[i] + 6.0 * a3[i] * t + 12.0 * a4[i] * t * t +
           20.0 * a5[i] * t * t * t;
52     }
53     else
54     {
55
56         qd[i] = qfin[i];
57         qdp[i] = 0.0;
58         qdpp[i] = 0.0;
59     }
60 }
61 }
62 else
63 {
64     TrayectType = 1; // Forzar seguimiento
65
66
67     const double xdini[2] = { 0.40, 0.15 }, xdfin[2] = { 0.20, 0.15 };
68
69     const double tf1 = 5.0, tf2 =tf1+1.0, tf3 = tf2+5.0;
70
71     double qfin[2];
72
73     if (t <= tf2)
74     {
75         double xd[2] = {xdini[0], xdini[1]}, xdp[2] = {0.0}, xdpp[2] = {0.0};
76         double Dt = (-aDH1 * aDH1 - aDH2 * aDH2 + xd[0] * xd[0] + xd[1] * xd
           [1]) / (2.0*aDH1*aDH2);
77         qfin[1] = atan2(-sqrt(1.0 - Dt * Dt), Dt);
78         qfin[0] = atan2(xd[1], xd[0]) - atan2(aDH2 * sin(qfin[1]), aDH1 + aDH2
           * cos(qfin[1]));

```

```

79     const double tf = tf1;
80     static double q1ini = q1, q2ini = q2;
81     const double a0[2] = { q1ini, q2ini }, a1[2] = { 0.0, 0.0 }, a2[2] = {
        0.0, 0.0 }, a3[2] = { 10.0 * (qfin[0] - q1ini) / (tf * tf * tf)
        , 10.0 * (qfin[1] - q2ini) / (tf * tf * tf) }, a4[2] = { -15.0 * (
        qfin[0] - q1ini) / (tf * tf * tf * tf), -15.0 * (qfin[1] - q2ini) /
        (tf * tf * tf * tf) }, a5[2] = { 6.0 * (qfin[0] - q1ini) / (tf *
        tf * tf * tf * tf), 6.0 * (qfin[1] - q2ini) / (tf * tf * tf * tf *
        tf) };
82     for (int i = 0; i < 2; i++)
83     {
84         if (t <= tf1)
85         {
86             qd[i] = a0[i] + a1[i] * t + a2[i] * t * t + a3[i] * t * t * t + a4
                [i] * t * t * t * t + a5[i] * t * t * t * t * t;
87             qdp[i] = a1[i] + 2.0 * a2[i] * t + 3.0 * a3[i] * t * t + 4.0 * a4[
                i] * t * t * t + 5.0 * a5[i] * t * t * t * t;
88             qdpp[i] = 2.0 * a2[i] + 6.0 * a3[i] * t + 12.0 * a4[i] * t * t +
                20.0 * a5[i] * t * t * t;
89         }
90         else
91         {
92
93             qd[i] = qfin[i];
94             qdp[i] = 0.0;
95             qdpp[i] = 0.0;
96         }
97     }
98 }
99 else
100 {
101     const double tf = tf3 - tf2;
102     const double a0[2] = { xdini[0], xdini[1] }, a1[2] = { 0.0, 0.0 }, a2[2] =
        { 0.0, 0.0 }, a3[2] = { 10.0 * (xdfin[0] - xdini[0]) / (tf * tf * tf)
        , 10.0 * (xdfin[1] - xdini[1]) / (tf * tf * tf) }, a4[2] = { -15.0 *
        (xdfin[0] - xdini[0]) / (tf * tf * tf * tf), -15.0 * (xdfin[1] -
        xdini[1]) / (tf * tf * tf * tf) }, a5[2] = { 6.0 * (xdfin[0] - xdini
        [0]) / (tf * tf * tf * tf * tf), 6.0 * (xdfin[1] - xdini[1]) / (tf
        * tf * tf * tf * tf) };
103     double ta = t - tf2;

```

```

104
105     for (int i = 0; i < 2; i++)
106     {
107         if (t <= tf3)
108         {
109             xd[i] = a0[i] + a1[i] * ta + a2[i] * ta * ta + a3[i] * ta * ta *
                ta + a4[i] * ta * ta * ta * ta + a5[i] * ta * ta * ta * ta *
                ta;
110             xdp[i] = a1[i] + 2.0 * a2[i] * ta + 3.0 * a3[i] * ta * ta + 4.0 *
                a4[i] * ta * ta * ta + 5.0 * a5[i] * ta * ta * ta * ta;
111             xdpp[i] = 2.0 * a2[i] + 6.0 * a3[i] * ta + 12.0 * a4[i] * ta * ta
                + 20.0 * a5[i] * ta * ta * ta;
112         }
113         else
114         {
115
116             xd[i] = xdfin[i];
117             xdp[i] = 0.0;
118             xdpp[i] = 0.0;
119         }
120     }
121
122     double Dt = (-aDH1 * aDH1 - aDH2 * aDH2 + xd[0] * xd[0] + xd[1] * xd
                [1]) / (2.0 * aDH1 * aDH2);
123     qd[1] = atan2(-sqrt(1.0 - Dt * Dt), Dt);
124     qd[0] = atan2(xd[1], xd[0]) - atan2(aDH2 * sin(qd[1]), aDH1 + aDH2 *
                cos(qd[1]));
125
126
127     double Ja1[2][2];
128     double Jap[2][2] = { {-aDH1 * cos(q1) * (q1fp)-aDH2 * cos(q1 + q2) * (
                q1fp + q2fp),-aDH2 * cos(q1 + q2) * (q1fp + q2fp)},{-aDH1 * sin(q1
                ) * q1fp - aDH2 * sin(q1 + q2) * (q1fp + q2fp),-aDH2 * sin(q1 + q2
                ) * (q1fp + q2fp)} };
129     double Japqdp[2];
130     InvMatrix(Ja, 2, Ja1);
131     MatrixMul(Ja1, xdp, 2, 2, 1, qdp);
132     MatrixMul(Jap, qdp, 2, 2, 1, Japqdp);
133     double xdppJap[2] = {xdpp[0] - Japqdp[0], xdpp[1] - Japqdp[1] };
134     MatrixMul(Ja1, xdppJap, 2, 2, 1, qdpp);

```

```

135
136
137     }
138
139 }
140 //error
141 double err1 = q1 - qd[0], err2 = q2 - qd[1];
142
143 //derivada del error seguimiento
144 double err1p = q1fp - qdp[0], err2p = q2fp - qdp[1];
145
146
147 //seguimiento
148 static double err1i = 0.0, err2i = 0.0;
149
150
151 switch (controlType)
152 {
153     case 0: // PID
154     {
155         double kp1 = 0.0, kp2 = 0.0, kd1 = 0.0, kd2 = 0.0, ki1 = 0.0, ki2 =
156             0.0;
157         if (TrayectType == 0)
158         {
159             kp1 = 3.5; kp2 = 1.0; kd1 = 0.4; kd2 = 0.3; ki1 = 0.4; ki2 = 1.5; //
160                 regulacion
161         }
162         else
163         {
164             kp1 = 4.5; kp2 = 1.6; kd1 = 0.7; kd2 = 0.5; ki1 = 1.6; ki2 = 1.3; //
165                 Trayectoria
166         }
167     }
168
169     torque1 = -kp1 * err1 - kd1 * err1p - ki1 * err1i;
170     torque2 = -kp2 * err2 - kd2 * err2p - ki2 * err2i;
171     break;
172 }
173
174 case 1: // Lazo abierto
175 { //Gravedad identificacion

```



```

172 //torque1 = 0.32;
173 //torque2 = 0.0375;
174 //Identificacion Todo el modelo
175 const double am1[2] = { 0.08,0.016 }, am2[2] = { 0.1,0.032 }, am3[2] =
      { 0.12,0.048 };
176 const double Tam1[2] = { 1.0,1.0 }, Tam2[2] = { 0.9,0.9}, Tam3[2] = {
      0.7,0.7 };
177
178 torque1 = 0.22 + am1[0] * sin(2 * pi * t / Tam1[0]) + am2[0] * sin(2 *
      pi * t / Tam2[0]) + am3[0] * sin(2 * pi * t / Tam3[0]);
179 torque2 = 0.025 + am1[1] * cos(2 * pi * t / Tam1[1]) + am2[1] * cos(2
      * pi * t / Tam2[1]) + am3[1] * cos(2 * pi * t / Tam3[1]);
180 break;
181 }
182 case 2: // PD +
183 {
184 //const double kp1 = 300.0, kp2 = 150.0, kd1 = 1.2, kd2 = 0.3; //
      regulacion
185 double kp[2] = { 5.7,1.6 }, kd[2] = {0.96,0.12}; //segimiento
186 if (TrayectType == 0)
187 {
188 kp[0] = 4.0;
189 kp[1] = 3.0;
190 kd[0] = 0.25;
191 kd[1] = 0.08;
192 }
193
194 const double th[9] = { 0.00558989076470356, 0.000220531905494296,
      0.00148560866504172, 0.0146713718633967, 0.00302981808549104,
      0.0561741876431650, 0.00374528435555607, 0.207931028530169,
      0.0280702251555495 };
195 double H[2][2] = { {th[0] + 2.0 * cos(q2) * th[1] + th[2], cos(q2) * th
      [1] + th[2]} ,{cos(q2) * th[1] + th[2],th[2]} },
196 C[2][2] = { {-sin(q2) * q2fp * th[1],-sin(q2) * (q1fp + q2fp) * th
      [1]},{sin(q2) * q1fp * th[1],0.0} }, D[2] = { th[3],th[4] }, Ds[2]
      = { th[5],th[6] }, G[2] = { cos(q1) * th[7] + cos(q1 + q2) * th
      [8],cos(q1 + q2) * th[8] };
197 double Hqdp[2], Cqdp[2];
198 MatrixMul(H, qdpp, 2, 2, 1, Hqdp);
199 MatrixMul(C, qdp, 2, 2, 1, Cqdp);

```

```

200 torque1 = Hqdpp[0] + Cqdp[0] + D[0] * qdp[0] + Ds[0] * sign(q1fp) + G
      [0] - kp[0] * err1 - kd[0] * err1p;
201 torque2 = Hqdpp[1] + Cqdp[1] + D[1] * qdp[1] + Ds[1] * sign(q2fp) + G
      [1] - kp[1] * err2 - kd[1] * err2p;
202
203 break;
204 }
205
206 case 3: // Par calculado
207 {
208     double zeta[2] = { 1.0, 1.0 }, omegan[2] = { 40.0, 45.0 };
209     if (TrayectType == 0)
210     {
211         zeta[0] = 1.0;
212         zeta[1] = 1.0;
213         omegan[0] = 40.0;
214         omegan[1] = 45.0;
215     }
216
217
218     double Y[2][9] = { { q1fpp, 2.0 * cos(q2) * q1fpp + cos(q2) * q2fpp -
      sin(q2) * q1fp * q2fp - sin(q2) * (q1fp + q2fp) * q2fp, q1fpp +
      q2fpp, q1fp, 0.0, sign(q1fp), 0.0, cos(q1), cos(q1 + q2) },
219     { 0.0, cos(q2) * q1fpp + sin(q2) * q1fp * q1fp, q1fpp + q2fpp, 0.0, q2fp
      , 0.0, sign(q2fp), 0.0, cos(q1 + q2) } };
220     //parametros medidos m1 = 0.069[kg], m2 = 0.148[kg], l1 = 0.165[m], l2
      = 0.285[m], lc1 = 0.094[m], lc2 = 0.110[m]
221     //const double th[9] = { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.2825,
      0.0375 };
222     //const double th[9] = { 0.003522, 0.001252, 0.001302,
      0.0146713718633967*0.0, 0.00302981808549104*0.0,
      0.0561741876431650*0.0, 0.00374528435555607*0.0, 0.2479, 0.0744
      }; // calculados solo gravedad
223     // modelo identificado
224     const double th[9] = { 0.00558989076470356, 0.000220531905494296,
      0.00148560866504172, 0.0146713718633967, 0.00302981808549104,
      0.0561741876431650, 0.00374528435555607, 0.207931028530169,
      0.0280702251555495 };
225     double Yth[2];
226     MatrixMul(Y, th, 2, 9, 1, Yth);

```

```

227     double H[2][2] = { {th[0] + 2.0 * cos(q2) * th[1] + th[2], cos(q2) * th
      [1] + th[2]} , {cos(q2) * th[1] + th[2], th[2]} },
228     C[2][2] = { {-sin(q2) * q2fp * th[1], -sin(q2) * (q1fp + q2fp) * th
      [1]}, {sin(q2) * q1fp * th[1], 0.0} }, D[2] = { th[3], th[4] }, Ds[2]
      = { th[5], th[6] }, G[2] = { cos(q1) * th[7] + cos(q1 + q2) * th
      [8], cos(q1 + q2) * th[8] };
229     double qde[2] = { qdpp[0] - 2.0 * zeta[0] * omegan[0] * err1p - omegan
      [0] * omegan[0] * err1, qdpp[1] - 2.0 * zeta[1] * omegan[1] * err2p
      - omegan[1] * omegan[1] * err2 };
230     double Cqp[2], Hqde[2], qfp[2] = {q1fp, q2fp};
231     MatrixMul(H, qde, 2, 2, 1, Hqde);
232     MatrixMul(C, qfp, 2, 2, 1, Cqp);
233     torque1 = Cqp[0]+D[0]*q1fp+Ds[0]*sign(q1fp)+G[0]+Hqde[0];
234     torque2 = Cqp[1] + D[1] * q2fp + Ds[1] * sign(q2fp) + G[1] + Hqde[1];
235     break;
236 }
237 case 4: // Pasivo
238 {
239     double Lambda[2] = { 9.0, 9.0 }, Kv[2] = {1.1, 0.2};
240     if (TrayectType == 0)
241     {
242         Lambda[0] = 9.0;
243         Lambda[1] = 9.0;
244         Kv[0] = 1.1;
245         Kv[1] = 0.4;
246     }
247     double qrp[2] = { qdp[0] - Lambda[0] * err1, qdp[1] - Lambda[1] * err2
      }, qrpp[2] = { qdpp[0] - Lambda[0] * err1p, qdpp[1] - Lambda[1] *
      err2p };
248     double s[2] = { q1fp - qrp[0], q2fp - qrp[1] };
249     const double th[9] = { 0.00558989076470356, 0.000220531905494296,
      0.00148560866504172, 0.0146713718633967, 0.00302981808549104,
      0.0561741876431650, 0.00374528435555607, 0.207931028530169,
      0.0280702251555495 };
250     double H[2][2] = { {th[0] + 2.0 * cos(q2) * th[1] + th[2], cos(q2) * th
      [1] + th[2]} , {cos(q2) * th[1] + th[2], th[2]} },
251     C[2][2] = { {-sin(q2) * q2fp * th[1], -sin(q2) * (q1fp + q2fp) * th
      [1]}, {sin(q2) * q1fp * th[1], 0.0} }, D[2] = { th[3], th[4] }, Ds[2]
      = { th[5], th[6] }, G[2] = { cos(q1) * th[7] + cos(q1 + q2) * th
      [8], cos(q1 + q2) * th[8] };

```

```

252     double Hqrpp[2], Cqrp[2];
253     MatrixMul(H, qrpp, 2, 2, 1, Hqrpp);
254     MatrixMul(C, qrp, 2, 2, 1, Cqrp);
255     torque1 = Hqrpp[0] + Cqrp[0] + D[0] * qrp[0] + Ds[0] * sign(q1fp) + G
           [0] - Kv[0] * s[0];
256     torque2 = Hqrpp[1] + Cqrp[1] + D[1] * qrp[1] + Ds[1] * sign(q2fp) + G
           [1] - Kv[1] * s[1];
257
258
259     break;
260 }
261 case 5: // Adaptable
262 {
263     double Lambda[2] = { 9.0, 10.0 }, Kv[2] = { 1.1, 0.2 };
264     double Gamma[9] = { 0.001, 0.001, 0.001, 0.15, 0.15, 0.15, 0.15, 0.15, 0.15 };
265     if (TrayectType == 0)
266     {
267         Lambda[0] = 9.0;
268         Lambda[1] = 9.0;
269         Kv[0] = 1.1;
270         Kv[1] = 0.4;
271         Gamma[0] = 0.0001; Gamma[1] = 0.0001; Gamma[2] = 0.0001; Gamma[3] =
           0.015; Gamma[4] = 0.015; Gamma[5] = 0.015; Gamma[6] = 0.015; Gamma
           [7] = 0.015; Gamma[8] = 0.015;
272     }
273     double qrp[2] = { qdp[0] - Lambda[0] * err1, qdp[1] - Lambda[1] * err2
           }, qrpp[2] = { qdpp[0] - Lambda[0] * err1p, qdpp[1] - Lambda[1] *
           err2p };
274     double s[2] = { q1fp - qrp[0], q2fp - qrp[1] };
275
276     double Yr[2][9] = { {qrpp[0], 2.0 * cos(q2) * qrpp[0] + cos(q2) * qrpp
           [1] - sin(q2) * qrp[0] * q2fp - sin(q2) * (q1fp + q2fp) * qrp[1],
           qrpp[0] + qrpp[1], qrp[0], 0.0, sign(q1fp), 0.0, cos(q1), cos(q1 + q2)},
277     {0.0, cos(q2) * qrpp[0] + sin(q2) * q1fp * qrp[0], qrpp[0] + qrpp
           [1], 0.0, qrp[1], 0.0, sign(q2fp), 0.0, cos(q1 + q2)} };
278
279     double thp[9], YrT[9][2];
280     //static double th[9] = { 0.313372901990620, 0.0206819573918073,
           0.101336229581604, 0.835980137097340, 0.200116370601950,
           3.19147767359876, 0.342503478586817, 11.4466433227176,

```

```

281         2.25676176044770 };
282     double Yrth[2], YrTs[9];
283     Transpose(Yr, 2, 9, YrT);
284     MatrixMul(YrT, s, 9, 2, 1, YrTs);
285     for (int i = 0; i < 9; i++)
286     {
287         thp[i] = -Gamma[i] * YrTs[i];
288         th[i] += thp[i] * T;
289     }
290     MatrixMul(Yr, th, 2, 9, 1, Yrth);
291     torque1 = Yrth[0] - Kv[0] * s[0];
292     torque2 = Yrth[1] - Kv[1] * s[1];
293
294     break;
295 }
296 case 6: // Modos deslizantes
297 {
298     double Lambda[2] = { 12.0, 9.0 }, Km[2] = { 2.2, 0.2 };
299
300     double qrp[2] = { qdp[0] - Lambda[0] * err1, qdp[1] - Lambda[1] * err2
301         }, qrpp[2] = { qdpp[0] - Lambda[0] * err1p, qdpp[1] - Lambda[1] *
302         err2p };
303     double s[2] = { q1fp - qrp[0], q2fp - qrp[1] };
304     double Yr[2][9] = { { qrpp[0], 2.0 * cos(q2) * qrpp[0] + cos(q2) * qrpp
305         [1] - sin(q2) * qrp[0] * q2fp - sin(q2) * (q1fp + q2fp) * qrp[1],
306         qrpp[0] + qrpp[1], qrp[0], 0.0, sign(q1fp), 0.0, cos(q1), cos(q1 + q2) },
307         { 0.0, cos(q2) * qrpp[0] + sin(q2) * q1fp * qrp[0], qrpp[0] + qrpp
308         [1], 0.0, qrp[1], 0.0, sign(q2fp), 0.0, cos(q1 + q2) } };
309     const double thn[9] = { 0.00558989076470356, 0.000220531905494296,
310         0.00148560866504172, 0.0146713718633967, 0.00302981808549104,
311         0.0561741876431650, 0.00374528435555607, 0.207931028530169,
312         0.0280702251555495 };
313     double Yrth[2];
314     MatrixMul(Yr, thn, 2, 9, 1, Yrth);
315
316     if (TrayectType == 0)
317     {
318         Lambda[0] = 12.0;

```

```

312         Lambda[1] = 9.0;
313         Km[0] = 2.2;
314         Km[1] = 0.2;
315     }
316     const double epsilon[2] = { 0.5,0.2 };
317     Km[0] = abs(Yrth[0])+epsilon[0];
318     Km[1] = abs(Yrth[1])+epsilon[1];
319     torque1 = - Km[0] * tanh(2.0 * s[0]);
320     torque2 = - Km[1] * tanh(2.0 * s[1]);
321     break;
322 }
323
324 default:
325     torque1 = 0.0;
326     torque2 = 0.0;
327     break;
328 }
329
330
331     err1i += err1 * T;
332     err2i += err2 * T;
333
334
335
336     q1a = q1;
337     q2a = q2;
338
339
340
341     graf[dataCount][0] = t;
342
343     graf[dataCount][1] = x[0]; //q1
344     graf[dataCount][2] = x[1]; //q2
345     graf[dataCount][3] = xdp[0]; //qdp[0]
346     graf[dataCount][4] = xdp[1]; //qdp[1]
347     graf[dataCount][5] = xpf[0]; //q1fp
348     graf[dataCount][6] = xpf[1]; //q2fp
349     graf[dataCount][7] = xd[0]; //qd[0]
350     graf[dataCount][8] = xd[1]; //qd[1]
351

```

```
352     /* graf[dataCount][1] = q1;
353     graf[dataCount][2] = q2;
354     graf[dataCount][3] = qdp[0];
355     graf[dataCount][4] = qdp[1];
356     graf[dataCount][5] = q1fp;
357     graf[dataCount][6] = q2fp;
358     graf[dataCount][7] = qd[0];
359     graf[dataCount][8] = qd[1]; */
360
361     graf[dataCount][9] = torque1;
362     graf[dataCount][10] = torque2;
363     graf[dataCount][11] = th[0];
364     graf[dataCount][12] = th[1];
365     graf[dataCount][13] = th[2];
366     graf[dataCount][14] = th[3];
367     graf[dataCount][15] = th[4];
368     graf[dataCount][16] = th[5];
369     graf[dataCount][17] = th[6];
370     graf[dataCount][18] = th[7];
371     graf[dataCount][19] = th[8];
372
373
374
375
376
377
378     dataCount++;
379     return ;
380 }
```

Bibliografía

- [Antonio B., 2007] Antonio B., Luis F. Peñin, C. B. R. A. (2007). *Fundamentos de Robótica*. Mc Graw Hill.
- [Arteaga et al., 2021] Arteaga, M. A., Gutiérrez-Giles, A., and Pliego-Jiménez, J. (2021). *Local Stability and Ultimate Boundedness in the Control of Robot Manipulators*. Springer.
- [Bona and Indri, 2005] Bona, B. and Indri, M. (2005). Friction compensation in robotics: an overview. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 4360–4367. IEEE.
- [Ernesto, 2009] Ernesto, O. (2009). *Modelado, Identificación y Control de un Robot Manipulador de Dos Grados de Libertad*. Instituto Politécnico Nacional.
- [F. Dreoni, 2020] F. Dreoni, M.G. Ortega, F. R. (2020). *IDENTIFICACIÓN Y CONTROL DE UN ROBOT MANIPULADOR*. Escuela Superior de Ingenieros. Universidad de Sevilla.
- [R. Kelly and Loria, 2005] R. Kelly, V. S. and Loria, A. (2005). *Control of Robot Manipulators in Joint Space*. Springer.
- [Sciavicco and Siciliano, 1996] Sciavicco, L. and Siciliano, B. (1996). *Modeling and Control of Robot Manipulators*. McGraw-Hill.
- [Spong et al., 2006] Spong, M. W., Hutchinson, S., and Vidyasagar, M. (2006). *Robot Modeling and Control*. John Wiley and Sons.
- [West Instruments, 2022] West Instruments (2022). *Manual de Aplicación de Encoders*.