

Rafael Guadalupe Chávez Moreno.

DESARROLLO DE CONTROL ADAPTABLE Y HMI
PARA ROBOT POLIARTICULADO SERIAL.

2011



Universidad Autónoma de Querétaro
Facultad de Ingeniería

DESARROLLO DE CONTROL ADAPTABLE Y HMI PARA ROBOT
POLIARTICULADO SERIAL.

Tesis

Que como parte de los requisitos para obtener el grado de
Maestro en Ciencias.

Instrumentación y Control Automático.

Presenta

Ing. Rafael Guadalupe Chávez Moreno.

Dirigido por:

Dr. Edgar Alejandro Rivas Araiza.

C.U. Querétaro, Qro. Noviembre 2011



Universidad Autónoma de Querétaro
 Facultad de Ingeniería.
 Maestría en Ciencias

DESARROLLO DE CONTROL ADAPTABLE Y HMI PARA ROBOT
 POLIARTICULADO SERIAL.

TESIS

Que como parte de los requisitos para obtener el diploma de
 Maestro en Ciencias.

Presenta:

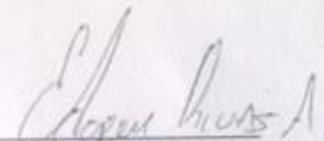
Ing. Rafael Guadalupe Chávez Moreno.

Dirigido por:

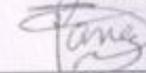
Dr. Edgar Alejandro Rivas Araiza.

SINODALES

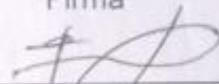
Dr. Edgar Alejandro Rivas Araiza
 Presidente


 Firma

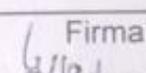
Dr. Yu Tang Xu
 Secretario


 Firma

Dr. Juvenal Rodríguez Reséndiz
 Vocal


 Firma

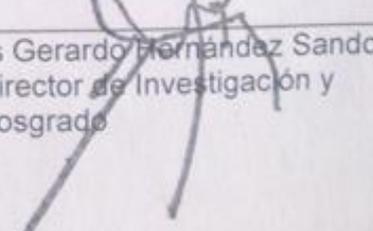
Dr. Manuel Toledano Ayala
 Suplente


 Firma

Dr. Víctor Manuel Hernández Guzmán
 Suplente


 Firma

Dr. Gilberto Herrera Ruíz
 Director de la Facultad


 Dr. Luis Gerardo Hernández Sandoval
 Director de Investigación y
 Posgrado

RESUMEN

Refiriéndose al problema del control de brazos manipuladores. Este trabajo, presenta el desarrollo de un sistema integral basado en variaciones del algoritmo de control PD (Proporcional Derivativo), para la solución del problema del control de trayectoria punto a punto. Se presenta el desarrollo de una interfaz máquina usuario (HMI) en tres dimensiones bajo el lenguaje Java y librerías OpenGL; permitiendo realizar simulaciones del proceso fuera de línea, minimizando costos y evitando accidentes humanos. Por otra parte, permite realizar un monitoreo del proceso en tiempo real. El desarrollo de la interfaz, involucra la implementación de algoritmos de cinemática directa e inversa. El sistema fue implementado utilizando la versatilidad proporcionada por los FPGAs (Field Programmable Gate Array), cuya funcionalidad, es procesar las señales de las articulaciones y generar las señales de control basadas en la modulación de ancho de pulso (PWM). Finalmente, se presentan los resultados obtenidos, al aplicar diferentes controladores ante una trayectoria circular en vacío y con carga.

(**Palabras clave:** brazo manipulador, control PD, cinemática directa e inversa, FPGA, Java, OpenGL)

SUMMARY

Referring to the problem of control of manipulator arms. This work presents the development of a comprehensive system based on variations of the PD (Proportional Derivative) control algorithm, to solve the point to point control problem. We report the development of user-machine interface (HMI) in three dimensions on the Java language and OpenGL libraries, enabling simulations of the process offline, minimizing costs and avoiding human accidents. On the other hand, allows monitoring real-time process. The development of the interface involves the implementation of algorithms of direct and inverse kinematics. The system was implemented using the flexibility provided by FPGAs (Field Programmable Gate Array), whose functionality is to process signals from the joints and generate control signals based on pulse width modulation (PWM). Finally, we present the results obtained by applying different controllers to a circular path at idle and under load.

(**Palabras clave:**manipulator arm, PD control, direct and inverse kinematics, FPGA, Java, OpenGL)

AGRADECIMIENTOS

Quiero expresar mis mas sincero agradecimiento a mi asesor de tesis. Dr. Edgar Alejandro Rivas Araiza, por haberme brindado la oportunidad de realizar una maestría bajo su tutela y por brindarme su amistad. De igual manera quiero dar un profundo agradecimiento a mi co-asesor de tesis. Dr. Yu Tang Xu, quien me ha brindado su apoyo incondicional tanto en lo académico como en lo personal, gracias por permitirme trabajar a su lado, pues su esfuerzo y dedicación, han sido un gran ejemplo y motivación que me ha permitido crecer como estudiante y como persona. Gracias por su amistad y por compartir sus conocimientos conmigo.

A mi amada esposa Ing. Bryanda Guadalupe Reyes Tesillo, por todo su amor y comprensión, quien me ha apoyado en todo momento y ha sacrificado gran parte del tiempo que deberíamos pasar juntos, con el fin de alcanzar esta meta que juntos nos propusimos. Gracias, por tus consejos y por lo mas valioso que me has brindado, la oportunidad de ser padre, TE AMO.

A mis padres Sr. Rafael Gonzalo Chávez Vizcaya y la Sra. Isabel Moreno Barrón, quienes me han brindado incondicionalmente su apoyo, y a lo largo de la vida me han enseñado a ser una persona de bien. A mis hermanos Efrain, Manuel y Francisco quienes han sido un ejemplo de perseverancia a lo largo de toda mi vida.

A mis sinodales. Dr. Juvenal Rodríguez Reséndiz, Dr. Manuel Toledano Ayala y en especial al Dr. Víctor Manuel Hernández Guzmán, por su paciencia y comprensión. Pero sobre todo, por compartir sus conocimientos y ayudarme a corregir mis errores. Sus enseñanzas son una de las partes mas importantes que me llevo del desarrollo de esta tesis.

A mis compañeros y amigos Ing. Alepth, Ing. Salvador, Ing. Claudia, Dr. Angel Luis y a todos los demás quienes por lo limitado del espacio no he mencionado, gracias por compartir experiencias y aprendizajes junto a mi.

Finalmente quiero agradecer al Consejo Nacional de Ciencia y Tecnología (CONACyT) y a los Programas de Apoyo a Proyectos de Investigación e Innovación Tecnológica (PAPIIT), por financiar el desarrollo de la investigación.

Índice general

1. INTRODUCCIÓN	12
1.1. Descripción del problema	13
1.2. Justificación	13
1.3. Hipótesis y Objetivos	14
1.3.1. Hipótesis	14
1.3.2. Objetivo general:	14
1.3.3. Objetivos particulares:	15
2. REVISIÓN DE LITERATURA	16
2.1. Sistemas robóticos	16
2.2. Sistemas de control	18
3. METODOLOGÍA	23
3.1. Historia de los sistemas robóticos	23
3.2. Movimientos rígidos y transformaciones homogéneas	26
3.2.1. Representación de rotaciones	27
3.2.2. Movimientos rígidos	30
3.2.3. Transformación homogénea	31
3.3. Cinemática	32
3.3.1. Cinemática directa	35
3.3.2. Asignación de marcos de coordenadas	35
3.3.3. Cinemática inversa	38
3.4. Velocidad cinemática	52
3.5. Implementación de controladores	54
3.6. PD+I (PID)	56
3.7. PD+W	57
3.8. PD+W (adaptable)	59
3.9. PD+G (Compensación de gravedad)	60
3.10. Modelado de motor CD	63
3.11. Interfaz gráfica	65
3.12. Herramientas de diseño (Sistemas VHDL)	69
3.13. Esquema general de interfaz (FPGA - PC)	69
3.13.1. Lectura de encoder (PPR - rad)	70
3.13.2. Bloque de selección de datos	73
3.13.3. Módulo de control PWM	75

3.13.4. PWM a 10 KHz	77
3.13.5. Driver de puente H	81
3.13.6. Comunicación serial	83
3.14. Desarrollo de hardware	91
3.14.1. Etapa de potencia	91
3.14.2. Tarjeta de conexión	95
4. RESULTADOS Y DISCUSIÓN	98
4.1. Resultados en vacío	99
4.1.1. PD+I	99
4.1.2. PD+W	102
4.1.3. PD+W (Adaptable)	106
4.1.4. PD+G	110
4.2. Resultados con carga	116
4.2.1. PD+I	116
4.2.2. PD+W	119
4.2.3. PD+W (Adaptable)	123
4.2.4. PD+G	127
4.3. Discusión	133
Bibliografía	133
5. ANEXOS	138
5.1. Control PD+I	138
5.2. Control PD+W	139
5.3. Control PD+W (adaptable)	140
5.4. Control PD+G	141
5.5. Puente H (L298n)	143
5.6. Compuerta Not (74ls14)	150

Índice de figuras

3.1. Una entidad geométrica p y dos marcos de referencia.	27
3.2. Marco coordenado $o_1x_1y_1$ orientado un ángulo θ sobre el marco $o_0x_0y_0$	28
3.3. Rotación sobre z_0 por un ángulo θ	29
3.4. Movimiento rígido del vector P_1	31
3.5. Diferencia entre cinemática directa e inversa.	33
3.6. Manipulador Rhino XR-3.	33
3.7. Marcos de coordenadas asignados al manipulador.	34
3.8. Marcos de coordenadas del robot Rhino XR-3.	37
3.9. Esquema general de robot Rhino XR-3.	39
3.10. Codo de manipulador Rhino XR-3.	41
3.11. Proyección de la muñeca sobre el plano $x_0 - y_0$	41
3.12. Arquitectura del manipulador Rhino XR-3.	42
3.13. Proyección de manipulador sobre el plano $z_0 - x_0$	44
3.14. Ley de coseno	45
3.15. Múltiples soluciones a cinemática inversa	46
3.16. Movimiento robot serial convencional	47
3.17. Movimiento robot Rhino XR-3	48
3.18. Configuración inicial (Home)	49
3.19. Relación de movimiento entre hombro y codo.	50
3.20. Relación de movimiento entre codo y muñeca.	51
3.21. Diagrama a bloques control PID.	57
3.22. Ubicación de centros de inercia.	61
3.23. Diagrama esquemático para motor de CD.	64
3.24. Representación virtual del manipulador.	66
3.25. Pantalla de controlador en modo manual.	67
3.26. Cuatro soluciones de posición para el manipulador serial.	67
3.27. Rotación sobre mundo virtual.	68
3.28. Diagrama a bloques de interfaz VHDL.	70
3.29. Diagrama a bloques del lector de encoder.	71
3.30. Ubicación de motores físicos.	72
3.31. Esquema de módulo Snd_Posicion.	73
3.32. Máquina de estados de bloque single pulse.	74
3.33. Máquina de estados de enviar posición.	75
3.34. Protocolo de recepción (Señal de control).	76
3.35. módulo Control_PWM	76

3.36. Máquina de estados de control PWM	77
3.37. Generador de señales PWM	78
3.38. Máquina de estados de generador de PWM	81
3.39. Diagrama de tiempos de PWM.	81
3.40. Álgebra de boole para Driver.	82
3.41. Diagrama de tiempos para comunicación serial.	83
3.42. Esquema general de módulo de comunicación Serial.	85
3.43. Máquina de estados de módulo de transmisión.	87
3.44. Diagrama de tiempos de módulo de transmisión.	87
3.45. Diagrama de tiempos de módulo de recepción.	88
3.46. Diagrama a bloques para módulo de recepción.	89
3.47. Diagrama de tiempo para registro de desplazamiento.	90
3.48. Máquina de estados de módulo receptor.	91
3.49. Esquema general de Driver.	92
3.50. Puesta en marcha de motor.	92
3.51. Circuito integrado L298n.	93
3.52. PCB para puente H.	94
3.53. Tarjeta de circuito de potencia.	95
3.54. Conjunto de tarjetas de potencia	95
3.55. Conector y cable tipo IDE de 40 pines	96
3.56. Tarjeta de circuito de interconexión.	96
3.57. Tarjeta de lectura de encoder	97
3.58. Sistema de control final	97
4.1. Trayectoria deseada	98
4.2. Posición articular obtenida (PD+I)	99
4.3. Señal de error articular (PD+I)	100
4.4. Señal de control (PD+I)	101
4.5. Posición final obtenida (PD+I)	102
4.6. Posición articular obtenida (PD+W)	103
4.7. Señal de error articular (PD+W)	104
4.8. Señal de control (PD+W)	105
4.9. Posición final obtenida (PD+W)	106
4.10. Posición articular obtenida (PD+W Adaptable)	107
4.11. Señal de error articular (PD+W Adaptable)	108
4.12. Señal de control (PD+W Adaptable)	109
4.13. Posición final obtenida (PD+W Adaptable)	110
4.14. Posición articular obtenida (PD+G)	111
4.15. Señal de error articular (PD+G)	112
4.16. Señal de control (PD+G)	113
4.17. Posición final obtenida (PD+G)	114
4.18. Comparativa de error para los controladores en vacío.	115
4.19. Comparativa de señal de control para los controladores en vacío.	115
4.20. Posición articular obtenida (PD+I) con peso	116
4.21. Señal de error articular (PD+I) con peso	117

4.22. Señal de control (PD+I) con peso	118
4.23. Posición final obtenida (PD+I) con peso	119
4.24. Posición articular obtenida (PD+W) con peso	120
4.25. Señal de error articular (PD+W) con peso	121
4.26. Señal de control (PD+W) con peso	122
4.27. Posición final obtenida (PD+W) con peso	123
4.28. Posición articular obtenida (PD+W Adaptable) con peso	124
4.29. Señal de error articular (PD+W Adaptable) con peso	125
4.30. Señal de control (PD+W Adaptable) con peso	126
4.31. Posición final obtenida (PD+W Adaptable) con peso	127
4.32. Posición articular obtenida (PD+G) con peso	128
4.33. Señal de error articular (PD+G) con peso	129
4.34. Señal de control (PD+G) con peso	130
4.35. Posición final obtenida (PD+G) con peso	131
4.36. Comparativa de error para los controladores en vacío.	132
4.37. Comparativa de señal de control para los controladores en vacío.	132

Índice de tablas

3.1. Propiedades del grupo de matrices $SO(n)$	29
3.2. Tabla de parámetros DH $SO(n)$	37
3.3. Valores nominales de articulación, para posición de home.	49
3.4. Resolución de encoder	72
3.5. Porcentaje de Duty cycle para PWM	79
3.6. Coeficientes de ancho de pulso correspondiente a porcentaje.	80
3.7. Tabla de verdad para Driver de puente H	82

Capítulo 1

INTRODUCCIÓN

La modelación matemática provee una vasta cantidad de información sobre el comportamiento dinámico de algún proceso, por esta razón, el modelado matemático es esencial en el control de Sistemas Robóticos Industriales (SRI). Debido a los avances que se ha tenido en algoritmos que conforman a los Sistemas de Control (SC), tales como Redes Neuronales Artificiales (RNA), Lógica Difusa (Fuzzy), controladores basados en visión, híbridos, etc. Es que se ha logrado satisfacer las demandas exigidas por las normativas de calidad, esto aunado con la capacidad de producción, han creado un amplio panorama de áreas de desarrollo para los SRI.

Algunos ejemplos de áreas de desarrollo que han tenido una gran atención son; el control multi-robot, control de seguridad, control de fuerza, visión 3D, supervisión a distancia y comunicación inalámbrica comenta Brogårdh (2007).

Los beneficios obtenidos por estos desarrollos se discuten; así como los problemas técnicos con los que se encuentran los fabricantes. Actualmente los SC se encuentran continuamente refinándose de acuerdo a las necesidades del proceso como puede ser: mayor rendimiento y la reducción de costos mediante el rediseño ingenieril lo que a su vez genera que sea más difícil su control al tener un modelo matemático mas complejo.

De aquí, es que se obtiene que el control basado en modelación, sea un factor clave para el control de robots y modelos industriales. Actualmente las áreas de mayor aplicación para los robots se encuentran en la industria automotriz, específicamente en las líneas de ensamble, en la industria del alimento, fundidoras, industria de la medicina, así como en la construcción de grandes estructuras. Así mismo, el uso de robots en los últimos años no se encuentra limitado exclusivamente a las grandes empresas, actualmente las Pequeñas y Medianas Empresas (PyMES) han comenzado a realizar inversiones en SRI, por lo que la investigación acerca de los SC se ha intensificado. En este documento, se discuten diversas técnicas de control; así como el diseño de trayectorias. Yahmadi (2007) comenta que el desarrollo del análisis dinámico de manipuladores flexibles es de gran interés debido al potencial en la aplicaciones industriales.

El futuro de la robótica se puede fortalecer cada vez más con el desarrollo de nuevas técnicas de control y modelado, así como de técnicas que permitan reducir costos de producción, tal es el caso de controladores sensor-less (sin sensor), los cuales no requieren de elementos primarios; sino que, mediante el análisis de la planta se puede conocer

las variables de interés. Con esto se puede concluir que el desarrollo de los robots industriales aún se encuentra lejos de alcanzar sus límites, actualmente diversos autores se encuentran trabajando en algoritmos de control adaptables, lo que permitirá que el sistema sea plug and play; una funcionalidad muy importante.

1.1. Descripción del problema

La necesidad de generar productos con una calidad superior a precios relativamente bajos, ha originado que las industrias mexicanas, específicamente la industria automotriz, hayan optado por realizar una inversión al adquirir algún sistema robótico. El cuál les permita alcanzar los niveles de calidad demandados y por otro lado liberan a los trabajadores de sufrir accidentes, al transferir los trabajos difíciles, sucios y peligrosos del hombre a la máquina.

Lo descrito anteriormente forma parte de algunas de las ventajas que se obtienen al adquirir algún tipo de sistema robótico. Desafortunadamente la adquisición de un brazo manipulador no es barata. Además, es requerido dar mantenimiento al sistema continuamente. Esto implica que se deba de solicitar los servicios de empresas extranjeras, para realizar dichas tareas.

Esto implica más gastos para la empresa. Es por ello, que actualmente sólo las empresas con capital para invertir son las que pueden contemplar la adquisición de un brazo manipulador. Dejando por fuera a las PyMES de lograr una automatización a sus procesos, siendo estas las empresas que conforman la mayor parte de la industria de México. De acuerdo a la (SEDECO, 2009), a pesar de los esfuerzos de los propietarios, lo cierto es que en México seis de cada diez empresas mexicanas mueren antes de los dos años de vida. Lo que ocurre es que ante la mayor competencia que existe en los diferentes sectores productivos del país, no basta la voluntad, el capital y tiempo. Hoy en día se necesita capacitación y tecnología para cualquier tipo de negocio.

Aunque el problema es grande y afecta a la mayoría de las empresas, aún no se cuenta con productos nacionales que permitan minimizar los costos y hacer factible que las PyMES puedan acceder a este tipo de tecnología, limitando su desarrollo y evitar que a pesar de los esfuerzos realizados no logren ser competitivos en el mercado.

1.2. Justificación

De acuerdo a lo publicado por IFR (International Federation of Robotics) en el mes de junio de 2010, se muestra que las ventas de los sistemas robóticos a nivel mundial, han tenido un decremento de 47 % comparado con el 2008, uno de los años mas exitosos. Especialmente los tres mayores mercados de la industria robótica decayeron dramáticamente (Japón, Alemania y Norte América).

Por lo tanto, la tendencia de la automatización fue interrumpida a nivel mundial. Esto se vio reflejado en las industrias mexicanas: automotriz, hule y plástico y la eléctrica/electrónica, las cuáles, debido a la situación económica en la que se encontraban, redujeron sus inversiones en la robótica.

1. INTRODUCCIÓN

Este fenómeno puede ser aprovechado por los investigadores mexicanos, quienes pueden desarrollar sistemas de control de alto rendimiento y bajo costo, que se adapten al proceso. Lo cual, originaría una oportunidad para hacer de las investigaciones nacionales una parte importante en la industria Mexicana.

Debido a que lamentablemente en el país no se cuenta con una industria de robots propia, no se ha realizado un avance significativo en la investigación y desarrollo de tecnología mexicana.

Es en este sentido en donde se pretende realizar una contribución a la investigación mexicana, al desarrollar un sistema de control *ROSCO* (Robotic System Control) capaz de generar el mismo rendimiento que los sistemas de control comerciales. Teniendo la ventaja de ser una tecnología nacional y de bajo costo. El sistema se encuentra basado en la tecnología de los circuitos integrados FPGA (Field Gate Programmable Array) y PC (Personal Computer); lo que permite ser implementado en diferentes sistemas de brazo manipulador, con la capacidad de adaptarse a las diferentes situaciones que presenta cada proceso, realizando la tarea definida con un margen de error similar a los sistemas comerciales.

En el sentido tecnológico, se cuenta con las herramientas necesarias para el desarrollo eficaz del controlador, logrando implementar los algoritmos necesarios para su funcionamiento a un costo razonable. De esta manera, las industrias verán una fuente viable para la inversión en sistema nacionales y se comience la independencia de la tecnología desarrollada en el extranjero.

Dentro de las herramientas utilizadas para el desarrollo del sistema se ha elegido el lenguaje de programación Java, dando la ventaja de ser un lenguaje de libre distribución. Como interfaz de comunicación se utilizó un FPGA, los cuales tienen la flexibilidad de ser reprogramados con el fin de realizar las modificaciones necesarias y/o requeridas por la industria.

1.3. Hipótesis y Objetivos

1.3.1. Hipótesis

- El desarrollo de un sistema de control y monitoreo integral, para un robot serial, utilizando diversas tecnologías como periféricos, permite generar un sistema viable, amigable y económico.

1.3.2. Objetivo general:

- Desarrollar un sistema embebido de control y monitoreo para un brazo robótico serial, con base en algoritmos de control PD (Proporcional Derivativo), trazo de trayectorias (mediante cinemática directa e inversa) y monitoreo mediante una interfaz gráfica utilizando técnicas visuales computacionales, todo ello formando parte de un sistema integral.

1.3.3. Objetivos particulares:

- Desarrollar un sistema de control integral, teniendo como sistema de control diversas variaciones del controlador PD.
- Desarrollar una interfaz gráfica, usuario-máquina, para el control y monitoreo de un brazo robótico.
- Vincular el hardware desarrollado con el software de control para pruebas funcionales.

Capítulo 2

REVISIÓN DE LITERATURA

En este capítulo se presenta de manera amena, integrada y resumida la información existente en la literatura nacional e internacional enfocada en el desarrollo de sistemas de control. Tal información es utilizada exclusivamente como referencia bibliográfica, con el fin de dar a conocer las técnicas de control actuales.

En esta tesis, las leyes de control utilizadas, están basadas en variaciones del controlador PD como son: controlador PID (Proporcional Integral Derivativo), controlador con compensación de gravedad y un controlador PD + compensación w (donde el parámetro w es definido para compensar las no-linealidades del sistema).

Los temas a tratar en este capítulo son los siguientes:

- Sistemas robóticos
- Sistemas de control

2.1. Sistemas robóticos

En esta sección se presentan los desarrollos que se han tenido en la robótica dentro de la industria, teniendo como fin incrementar la calidad de los productos y minimizar los costos de producción.

Las necesidades actuales de los seres humanos han originado que cada vez los procesos industriales requieran de sistemas más complejos que permitan alcanzar los niveles de calidad deseados, siendo esta la razón principal por la que los procesos con robots industriales han registrado un aumento.

Principalmente las industrias que han adoptado este sistema son las industrias donde el proceso implica estar en contacto directo con tareas peligrosas o de riesgo para el operador como el fresado, pulido, pintura, montaje, soldadura, etc. y en las tareas de precisión como en la asistencia médica.

Diversos autores han publicado artículos enfocados a los problemas anteriores.

(Bao *et al.*, 2009) presenta cómo una simple operación de pulir una pieza de trabajo es en realidad muy compleja, y reporta que el pulido mediante sistemas robóticos se está introduciendo como una nueva tecnología; ofreciendo una

2. REVISIÓN DE LITERATURA

mayor flexibilidad en comparación con el pulido manual; presentando una mejora económica en comparación con el uso de máquinas dedicadas. Reemplazando el trabajo manual y las máquinas pulidoras dedicadas con robots pulidores puede dar el indulto a los operadores de pulir en un ambiente difícil mejorando la calidad y eficiencia mientras se reducen costos.

(Daeinabi, 2007) menciona que “ Los robots industriales son importantes en los procesos de producción especialmente en la industria automotriz para la tasa de aumento de la producción, la promoción de productos de calidad, trabajo continuo y un bajo costo de fabricación. También los robots mejoran las condiciones de trabajos en las líneas de producción de automóviles para que puedan hacer procesos repetitivos y desplazamiento de piezas con la fuerza mínima posible”.

Igualmente (Fauadi and Jumali, 2008) comentan que por más de una década, los robots han sido ampliamente utilizados en asistencia al ser humano en la ejecución de tareas específicas y en diversos campos de aplicación para diferentes fines, por lo general se clasifican en dos categorías principales: robot de servicio y robots industriales. las organizaciones manufactureras hacen uso de robots industriales para realizar tareas de producción y su utilización ha mejorado la productividad de fabricación eficazmente.

En (Blomdell *et al.*, 2005) se presenta un documento donde se describe el diseño e implementación de una plataforma para la integración de sensores externos en un sistema de control para robot industrial. Como un ejemplo de aplicación presenta la implementación de un controlador de fuerza en procesos de rectificación.

(Andres *et al.*, 2009) detalla un análisis profuso de la cinemática de un sistema robótico específicamente un manipulador con 6 juntas rotativas montadas sobre un eje lineal y sincronizada con una mesa giratoria para un proceso de molienda. En este artículo se describen las ventajas y desventajas de su uso.

Otra aplicación en donde se ha hecho uso de robots es en los procesos de soldadura. Sin embargo, los robots soldadores son rara vez totalmente autónomos, sobre todo porque un brazo robótico, integrando una cadena de montaje, produce mejores resultados. Sin embargo, hay situaciones tales como la soldadura en tuberías o en los astilleros, donde los robots autónomos de soldadura son realmente útiles (Radaschin, 2010). Esta es la razón por la que el uso de la robótica en soldadura se ha incrementado.

En (Ting *et al.*, 2010) se presenta una investigación sobre el desarrollo de un sistema de soldadura robotizada, que contiene cuatro ejes de revoluta y será utilizado en la soldadura de las tuberías para la desalación de agua de mar. Este trabajo es apoyado por el uso de un sistema robótico en el proceso en el que la tarea es repetitiva o difícil, y la calidad de la soldadura puede garantizarse.

Otro trabajo enfocado sobre robots de soldadura es presentado por (Hazel and Cote, 2010) donde se presenta un procedimiento de soldadura sobre la base de un par de antorchas de oxiacetileno para realizar cortes, junto con un par de cámaras láser 2D para el seguimiento de la costura. Resultando en un sistema de alta precisión.

Pero los robots también están en procesos médicos, por lo que en los últimos años el campo de la medicina ha sido invadido por robots. Esto no significa que los robots reemplazarán al personal calificado, como médicos o enfermeras, pero pueden ayudar en el trabajo de rutina y las tareas de precisión. En esta zona existe una amplia variedad de aplicaciones en las que los robots pueden ser útiles.

En (Qinjun, 2008) se muestra el desarrollo de un robot cirujano de cinco grados de libertad para ayudar a los cirujanos para colocar la sonda con precisión en el blanco. Este robot ha sido diseñado de acuerdo a los requisitos y la seguridad de los pacientes y los cirujanos.

Otro tipo de robot médico es presentado por (Yangmin and Qingsong, 2007) quien propuso el concepto de un robot médico paralelo aplicable a la compresión del pecho en el proceso de reanimación cardiopulmonar (RCP).

Como hoy día los robots son de gran tamaño. Diversos investigadores se han enfocado en el desarrollo de microrobots capaces de entrar dentro del cuerpo humano y realizar un diagnóstico. Este es el caso de (Kosa *et al.*, 2005), quien menciona que los médicos usan radiología, endoscopía y píldoras inteligentes para inspeccionar el contenido interno del cuerpo humano.

Estas son sólo algunas de las obras que se ocupan de las áreas de aplicación de robots industriales. Otras investigaciones se pueden encontrar en (Kucuk and Bingul, 2004) y (Akan *et al.*, 2010).

2.2. Sistemas de control

Los sistemas de control tienen vastas áreas de aplicación, por ejemplo, en industrias del transporte, incluyendo la aeroespacial, procesos químicos y biológicos; sistemas mecánicos, eléctricos y electromecánicos; agroindustria, industrias de procesos y de manufactura; sistemas económicos, políticos y sociales; sólo por mencionar a algunos.

Los encontramos en nuestra vida cotidiana: desde la nevera hasta el sistema de control de combustión electrónica de los automóviles, así como en nuestro propio cuerpo: control de la temperatura corporal, presión arterial, equilibrio, etc. El simple acto de señalar con el dedo es un sistema de control.

La realización de un sistema de control conlleva un gran número de acciones a realizar tales como: la identificaciones del sistema. Es decir, la obtención de un modelo matemático que sea lo más próximo al comportamiento dinámico del proceso. Esta es una de las tareas de mayor complejidad debido a que en la naturaleza la mayor parte de los procesos no son lineales y en numerosas ocasiones no se puede contemplar todos los factores que interactúan en el sistema. Es por ello, que se adopta un modelo matemático aproximado del proceso.

En los últimos años se ha tenido un gran auge en el desarrollo e investigación de algoritmos genéticos tales como redes neuronales artificiales (RNA), controles difusos, neurodifusos (mezcla de RNA y Fuzzy), etc.

2. REVISIÓN DE LITERATURA

Es por ello que el presente documento está enfocado en el desarrollo de un control adaptativo para el control de un robot serial. (Hasan, 2010) describe una red neuronal como un procesador paralelo masivo distribuido que tiene una propensión natural para almacenar conocimiento experimental y hacerlo disponible para su uso. Se parece al cerebro humano en dos aspectos: la red a través de un proceso de aprendizaje adquiere los conocimientos, y los puntos de conexión interneuronal conocidos como pesos sinápticos se utilizan para almacenar el conocimiento.

Una red de retro-propagación se entrena mediante un procedimiento de dos pasos. La señal de entrada fluye hacia adelante a través de la red y la señal de error fluye hacia atrás para realizar el ajuste de pesos, donde los conocimientos adquiridos por la red son almacenados como un conjunto de pesos de conexión (Hasan, 2007).

Actualmente las RNA se han convertido en una importante implementación de computo, mediante las cuales se pueden obtener soluciones óptimas en tiempo real para ciertos problemas de optimización. De hecho, se han propuesto varios tipos de redes neuronales para la programación de sistemas lineales, no lineales, etc. tal como lo describen (Chen, 2002,G,H,K,T,X,Z).

Por su naturaleza de aprendizaje las RNA, han sido aceptadas en diferentes campos de la ciencia como controladores y/o predictores. Algunas de las áreas donde han tenido un fuerte impacto ha sido en la predicción de las variaciones climáticas tanto externas como internas en los conocidos edificios inteligentes tal como lo mencionan (Behrang, 2010,M).

Durante los últimos años se han publicado una gran cantidad de artículos, en el área de los sistemas robóticos mediante Redes neuronales. Algunos de los trabajos se discuten a continuación:

(Hasan, 2007) presenta un controlador basado en redes neuronales para la solución de la cinemática inversa de un brazo manipulador. Para ello se realizó una comparación entre los distintos métodos de obtención de la cinemática; llegando a la conclusión que actualmente las leyes que describen el modelo matemático de los manipuladores, asumen que el funcionamiento del brazo fue descrito a la perfección; lo cual no es correcto suponer, debido a que el modelo no incluye totalmente las variables que interactúan con el robot.

Para evitar el modelado, se propone el uso de Redes Neuronales Artificiales (RNA), con el fin de obtener un controlador adaptable. De tal manera que se evite el uso del modelo matemático del manipulador; simplemente usando las características naturales de aprendizaje de las RNA; con lo cual se realice una identificación del sistema lo más fielmente posible.

Autores alternos han realizado investigaciones para el control de manipuladores robóticos, basándose en diversas técnicas de control. Siendo una de ellas las redes neuronales obteniendo excelentes resultados. Algunas publicaciones fueron realizadas por (Chaoui, 2009,L,S) y (Zuo, 2010) , mientras que (Caccavale, 2009) presenta un artículo utilizando un observador como identificador de fallas

2. REVISIÓN DE LITERATURA

que presentan los manipuladores robóticos, utilizando como herramienta a las RNA. por otro lado algunos autores han optado por el uso de algoritmos de control híbridos, tal es el caso de (Karami, 2008) y (Lin, 2009).

Otros controladores alternativos han sido propuestos para el control de brazos manipuladores por diversos autores. Tal es el caso de (Owena *et al.*, 2008) quien presenta el desarrollo de un controlador con el fin de utilizar un brazo manipulador como sistema de escultura tradicional CNC; basándose en la optimización min-max y la búsqueda de un espacio nulo de trabajo, con el fin de producir una optimización global.

Mientras que en (Wallén, 2008) se propone un Control de Aprendizaje Iterativo (ILC) de primer orden, utilizado en un brazo manipulador comercial de seis grados de libertad. El funcionamiento del algoritmo ILC se basa en la medición de los ángulos del motor, pero adicionando a la evaluación convencional del ILC, el error por el lado del motor y el error de la trayectoria de la herramienta. Esta última fue evaluada mediante la utilización de un laser tracer (Dispositivo de medición laser).

Comenta que en muchas aplicaciones industriales la misma trayectoria es repetida una y otra vez. En tales casos el método ILC es una forma de compensar los errores repetitivos. Así mismo describe que normalmente los ángulos del motor son las únicas variables que pueden ser medidas en un robot comercial y a partir de ellas se tiene que desarrollar el controlador. Por lo que, asumiendo que los modelos de la cinemática y dinámica son correctos; la posición de la herramienta podría teóricamente ser derivada de los ángulos del motor. Sin embargo esto no es real, ya que requeriría la descripción exacta de los fenómenos físicos como la fricción, backlash, los rizados en el torque y un modelo completo de las flexibilidades mecánicas.

En (Solteiro *et al.*, 2007) el autor nos presenta el desarrollo de un algoritmo genético multi-objetivo. Utilizado para la generación de trayectorias de un manipulador con el fin de controlar múltiples objetivos y evitar colisiones con obstáculos. También nos da un amplio panorama de la utilización de los Algoritmos Genéticos (GAs) durante los últimos 20 años, en una enorme cantidad de campos, tales como: control, sistemas de identificación, robótica, planificación y programación, reconocimiento de patrones y reconocimiento de voz.

Durante el desarrollo del texto, comenta cuales son los posibles aprovechamientos que se tienen en la generación de trayectorias utilizando la cinemática inversa y la matriz jacobiana. Por lo que concluye que para poder usar esta técnica se debe de tener en cuenta las singularidades cinemáticas, las cuales son difíciles de abordar. Es por esa razón que el autor direcciona su investigación en el uso GAs de generación de trayectorias basadas en la cinemática directa.

En (Shi *et al.*, 2008) se presenta el desarrollo de un nuevo algoritmo de control basado en un controlador integral de modo deslizante. Basándose en investigaciones previas el autor asegura que para un robot de cuerpo rígido, el aprovechamiento del torque computarizado provee una estabilidad asintótica para

2. REVISIÓN DE LITERATURA

el rastreo de tareas de control. Sin embargo, es necesario conocer las matrices características del sistema, para completar el algoritmo computarizado de torque; las cuales normalmente son desconocidas y posiblemente demasiado complejas para realizar la implementación en tiempo real.

Debido a la complejidad del modelo de los manipuladores, los cuáles son del tipo MIMO (múltiples entradas-múltiples salidas) y no lineales. El autor selecciono el controlador deslizante con el fin de convertir el modelo en un modelo lineal similar. En base a lo anterior, el autor desarrolló un control integral en modo deslizante y un estimador de perturbación. Obteniendo como resultados, el hecho que para el control de manipuladores, es posible realizar el diseño por ubicación de polos sin el conocimiento exacto acerca del robot manipulador a controlar.

De igual manera en (Alavandar, 2009) el autor nos presenta una propuesta sobre un nuevo algoritmo híbrido neuro-difuso (ANFIS) para el control de manipuladores con incertidumbre. Inicialmente el autor describe que el control de un robot industrial incluye no linealidades, incertidumbre y perturbaciones externas que deberían ser consideradas en el diseño de las leyes de control. Es por ello que el control de un brazo manipulador ha sido siempre considerado como un problema desafiante.

Normalmente un robot manipulador rígido de N grados de libertad es caracterizado por N ecuaciones no lineales, y ecuaciones diferenciales. Los efectos de las no linealidades han llegado a ser muy prominentes en la presencia de fricción, backlash, histéresis y en la elasticidad de las juntas y uniones.

La dinámica del robot se vuelve más complicada cuando las condiciones de carga útil y las condiciones de la inercia de las uniones son desconocidas. Pero sobretodo el principal reto en este campo, es que siempre hay incertidumbre en los entornos no estructurados. Por lo tanto su algoritmo ANFIS consiste de un controlador adaptativo neuro-difuso y controladores convencionales, mediante el cual obtuvo como resultado, una mejora apreciablemente con respecto a los ANFIS convencionales.

Finalmente (Tang *et al.*, 2000) plantea el desarrollo de un control descentralizado, para sistemas mecánicos descritos por las ecuaciones de Euler-Lagrange incluyendo interconexiones de alto orden; Garantizando la estabilidad final uniforme cuando el objetivo de control es el seguimiento de una trayectoria deseada.

Debido a la complejidad de análisis y diseño del sistema de control a gran escala y la incapacidad física para el intercambio de información entre subsistemas. La implementación y diseño del sistema de control, conviene realizarse para cada subsistema un control descentralizado; el cual dependa sólo de las variables locales disponibles para medir.

El control descentralizado ofrece varias ventajas sobre el control centralizado, por ejemplo, robustez frente a fallos, eficiencia computacional y simplicidad de diseño. Esta clase de sistemas no lineales cubre una amplia gama de aplicaciones

2. REVISIÓN DE LITERATURA

de sistemas mecánicos, donde las interconexiones existen debido a la interacción física o los objetivos de control.

Capítulo 3

METODOLOGÍA

En este capítulo se presenta la fundamentación teórica mediante la cual fue basada la investigación, de igual manera se describe de forma detallada el proceso seguido para alcanzar el objetivo de esta tesis.

Los temas a tratar en este capítulo son los siguientes:

- Historia de los sistemas robóticos
- Movimientos rígidos y transformaciones homogéneas
- Cinemática
- Velocidad cinemática
- Dinámica
- Interfaz gráfica
- Sistemas VHDL
- Introducción a los sistemas de control

3.1. Historia de los sistemas robóticos

Actualmente los avances tecnológicos han tenido un gran impacto en la vida del hombre. Esto se ve reflejado en la forma en que interactuamos con el medio que nos rodea. Pero principalmente, se puede observar este fenómeno en la industria; donde las exigencias de las normas que regulan el control de calidad de los productos, han obligado a las empresas a buscar soluciones viables. Permitiendo obtener la calidad de productos deseada a precios aceptables, por lo que se ha optado por incorporar sistemas robóticos dentro de las líneas de producción.

Este documento se enfoca específicamente a este tipo de sistemas. Por lo tanto, como primer punto que se tomará en cuenta, es la definición de *Robótica*.

3. METODOLOGÍA

De acuerdo a la *Real Academia Española (RAE)*, se define como la *Técnica que aplica la informática al diseño y empleo de aparatos que, en sustitución de personas, realizan operaciones o trabajos, por lo general en instalaciones industriales*. Esta definición tan general, cubre muchos conceptos diferentes, pero todos giran en torno a la misma idea; hacer sistemas autónomos (robots) que realicen tareas complejas y/o peligrosas para los seres humanos. Lo cual nos lleva a buscar la definición de Robot.

Nuevamente la (RAE, 2010) define al término Robot como: *Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservadas solo a las personas*. Sin embargo, este termino fue utilizado inicialmente por el autor checo Karel Capek, en su obra de 1921 RUR(Rossum Universal Robota), en donde utilizó la palabra robot (robota en checo, que significa esclavo, o trabajador obligado).

A partir de entonces diversos autores han propiciado que en sentido literario o cinematográfico hayamos creado una imagen de Robot, a un humanoide capaz de realizar tareas que desplacen al hombre. Por lo tanto, y en sentido común, estos antecedentes impiden que veamos a las máquinas cuyas formas no son semejantes a la humana como robots.

Los robots pueden ser clasificados de acuerdo a su arquitectura como:

- Móviles
- Zoomórficos
- Médicos
- Industriales
- Teleoperadores
- Híbridos

Móviles: Robots provistos de patas o ruedas que los capacitan para desplazarse de acuerdo a su programación. Elaboran la información que reciben a través de sus propios sistemas de sensores y se emplean en determinado tipo de instalaciones industriales, sobre todo para el transporte de mercancías en cadenas de producción y almacenes. También se utilizan en la investigación.

Zoomórficos: Robots caracterizados principalmente por su sistema de locomoción que imita a diversos seres vivos. Los androides pueden ser considerados robots zoomórficos.

Médicos: Los robots médicos son, fundamentalmente, prótesis para discapacidades físicas que se adaptan al cuerpo y están dotados de potentes sistemas de mando. Con ellos se logra suplir las extremidades o incluso órganos de los seres humanos.

Industriales: Los robots industriales son artilugios mecánicos y electrónicos destinados a realizar de forma automática determinados procesos de fabricación o manipulación.

3. METODOLOGÍA

Teleoperadores: Pueden clasificarse o no como robots. Se controlan remotamente por un operador humano. Son muy sofisticados y se utilizan en desactivación de bombas.

Híbridos: Corresponden a aquellos de difícil clasificación cuya estructura es una combinación de las anteriores.

Es precisamente sobre la división de los robots industriales, específicamente en los brazos robóticos donde se enfocará este documento. El desarrollo de los brazos robóticos manipuladores tienen sus orígenes en el año 1947 cuando *Goetz* construye el primer manipulador eléctrico servocontrolado. Es decir, capaz de conocer su posición, y modificarla para conseguir la posición deseada por el programador.

Pero no fue sino hasta el año de 1954 cuando *Devol* desarrolla el que puede considerarse el primer robot: un manipulador con memoria R/W leíble y escribible (es decir, programable) que almacena trayectorias punto a punto. Devol fundó entonces Unimation, la que sería la mayor compañía productora de robots durante los años siguientes.

Al mismo tiempo que los robots industriales avanzaban en complejidad durante la década de los años 60's y 70's, la robótica comenzó a introducirse como disciplina académica y comenzó a formar parte de la investigación de algunos centros universitarios. Por citar a algunas de las instituciones con mayor aporte al área de la robótica se encuentra la universidad de Stanford (California), cuyos investigadores en el año de 1969 construyeron el prototipo Shakey, un robot con sistema de visión artificial muy rudimentario, dotado con sensores ultrasónicos que le permitían interactuar con el medio, y una estructura cilíndrica.

En 1970 se introdujo el primer brazo de Stanford, cuyo uso universitario se ha extendido enormemente, el robot contiene seis grados de libertad, y su sistema de control es electrónico y servocontrolado.

En 1978 la compañía Unimation, lanzó al mercado al robot PUMA (Programmable Universal Machine for Assembly, o Programmable Universal Manipulation Arm). El cual, es probablemente el robot más difundido comercialmente hasta hoy día, por su arquitectura, geometría y robustez.

Este robot cuenta con seis grados de libertad, y al igual que el robot Stanford, está dotado con un sistema de control electrónico y servocontrolado. Actualmente la industria ha buscado realizar modificaciones al control de los sistemas con el fin de obtener un mejor rendimiento que se vea reflejado directamente en la calidad del producto final. Es por ello, que se han adaptado sistemas de visión artificial, interactuando con diversos tipos de sensores de acuerdo a la aplicación que se este realizando, llevando a los robots a un nuevo nivel de funcionalidad.

Con lo anterior mencionado se puede tener una idea general de la historia de los robots. Una vez comprendido el significado de los términos (robot) y (robótica), comenzaremos describiendo una aplicación típica de los sistemas manipuladores

industriales. Imaginemos un robot soldador que debe seguir una trayectoria planeada con una precisión de $\pm 0.5mm$.

Para lograr su objetivo se deben de involucrar diversas areas de investigación tales como movimientos rígidos y transformaciones homogéneas, cinemática directa e inversa, velocidad cinemática, planeación de trayectorias, dinámica y distintas leyes de control.

3.2. Movimientos rígidos y transformaciones homogéneas

Una gran parte de la cinemática de los manipuladores es concebida con el establecimiento de varios sistemas de coordenadas para representar la posición y orientación de los objetos rígidos. Por lo tanto la geometría del espacio tridimensional y de movimientos rígidos juega un papel central en todos los aspectos de la manipulación robótica.

Las transformaciones homogéneas combinan las operaciones de rotación y traslación dentro de una única multiplicación matricial. Para entender un poco más del uso de las matrices de transformación comenzaremos examinando la representación de puntos y vectores en el espacio Euclideo conformado de múltiples marcos de coordenadas. Tales transformaciones nos permiten representar un vector o punto desde la perspectiva de diferentes marcos de referencia, una cualidad que será explotada al máximo en la sección 3.3.

Antes de desarrollar esquemas para la representación de puntos y vectores, es de utilidad distinguir entre los dos enfoques fundamentales para el razonamiento geométrico: el enfoque *sintético* y el enfoque *analítico*. En el primero, se actúa directamente sobre las entidades (ejemplo, líneas o puntos), mientras en el último se representan estas entidades usando coordenadas o ecuaciones, y el razonamiento es realizado mediante ecuaciones algebraicas.

En robótica, típicamente utilizamos el razonamiento analítico, debido a que las tareas a realizar por el robot son a menudo usando coordenadas Cartesianas. Por supuesto que con el fin de asignar coordenadas es necesario especificar un marco de coordenadas.

La figura 3.1, muestra dos marcos de referencia que difieren en orientación por un ángulo de 45° , por lo tanto podemos especificar las coordenadas del punto p con respecto a cada marco de referencia $o_0x_0y_0z_0$ o el marco $o_1x_1y_1z_1$. En el primer caso, podemos asignar a p el vector $(6, 6)^T$, y con respecto al segundo marco de referencia el punto se puede definir como $(-2.8, 4.2)^T$. Como se puede apreciar ambos vectores representan exactamente el mismo punto pero expresado en diferentes marcos de referencia. Por lo tanto, siempre se debe de tener en cuenta bajo qué marco se está trabajando.

3. METODOLOGÍA

Para facilitar esta cuestión se adoptará la siguiente notación, en la cual, un número colocado como superíndice nos indicara el marco sobre el cual se entra, es decir:

$$\begin{aligned} p^0 &= \begin{bmatrix} 6 \\ 6 \end{bmatrix} \\ p^1 &= \begin{bmatrix} -2.8 \\ 4.2 \end{bmatrix} \end{aligned} \quad (3.1)$$

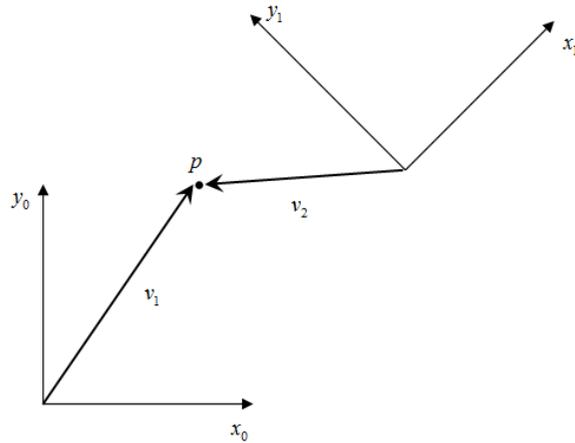


Figura 3.1: Una entidad geométrica p y dos marcos de referencia.

En este punto cabe resaltar, que geoméricamente, el punto p corresponde a una ubicación específica en el espacio, mientras que p^0 y p^1 son vectores de coordenadas que representan la ubicación de ese punto en el espacio con respecto a los marcos de coordenadas $o_0x_0y_0z_0$ y $o_1x_1y_1z_1$, respectivamente.

Debido a que el origen de un sistema coordenado es sólo un punto en el espacio, podemos asignar coordenadas que representen la posición del origen de una sistema de coordenadas con respecto a otro. Por ejemplo tomando como referencia nuevamente a la figura 3.1 es posible expresar a cada origen de los marcos de coordenadas expresados con respecto al otro como:

$$\begin{aligned} o_1^0 &= \begin{bmatrix} 10 \\ 5 \end{bmatrix} \\ o_0^1 &= \begin{bmatrix} -10.6 \\ 3.5 \end{bmatrix} \end{aligned} \quad (3.2)$$

3.2.1. Representación de rotaciones

Con el fin de representar la posición y orientación relativa de un cuerpo rígido con respecto a otro, fijaremos rígidamente un marco de coordenadas a cada cuerpo, y posteriormente se especificarán las relaciones existentes entre cada marco coordenado.

3. METODOLOGÍA

Rotación en un plano

La figura 3.2, muestra dos marcos coordenados, con el marco $o_1x_1y_1$ siendo obtenido al rotar el marco $o_0x_0y_0$ por un ángulo θ .

Para poder expresar la orientación del marco 1, es necesario especificar vectores coordenados para los ejes del marco $o_1x_1y_1$ con respecto al marco coordenado $o_0x_0y_0$:

$$R_1^0 = [x_1^0 | y_1^0] \quad (3.3)$$

donde x_1^0 y y_1^0 son las coordenadas en el marco $o_0x_0y_0$ de los vectores unitarios x_1 y y_1 , respectivamente. Una matriz que tiene esta forma es llamada *Matriz de rotación*.

Para aclarar esta situación tomemos en cuenta la figura 3.2 anterior, de la cual es posible obtener la proyección del marco coordenado $o_1x_1y_1$ con respecto al marco $o_0x_0y_0$.

$$\begin{aligned} x_1^0 &= \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} \\ y_1^0 &= \begin{bmatrix} -\sin\theta \\ \cos\theta \end{bmatrix} \end{aligned} \quad (3.4)$$

Lo cual nos resulta en

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad (3.5)$$

Por lo tanto las columnas de R_1^0 especifican los cosenos directores de los ejes coordenados de $o_1x_1y_1$ relativos a los ejes coordenados $o_0x_0y_0$.

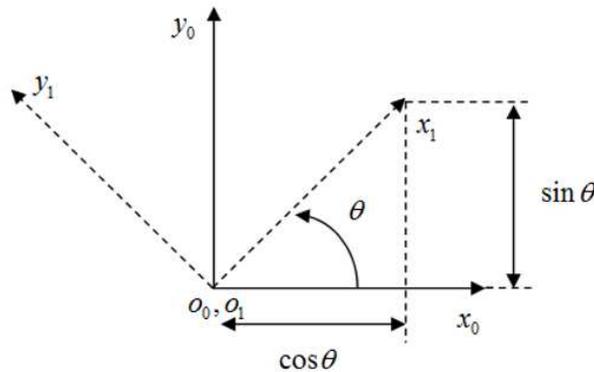


Figura 3.2: Marco coordenado $o_1x_1y_1$ orientado un ángulo θ sobre el marco $o_0x_0y_0$.

3. METODOLOGÍA

Lo descrito anteriormente es fácilmente escalable a un caso tridimensional. Cada eje del marco $o_1x_1y_1z_1$ es proyectado dentro del marco $o_0x_0y_0z_0$. La matriz de rotación resultante es definida como

$$R_1^0 = \begin{bmatrix} x_1 \cdot x_0 & y_1 \cdot x_0 & z_1 \cdot x_0 \\ x_1 \cdot y_0 & y_1 \cdot y_0 & z_1 \cdot y_0 \\ x_1 \cdot z_0 & y_1 \cdot z_0 & z_1 \cdot z_0 \end{bmatrix} \quad (3.6)$$

Dichas matrices de rotación en cualquier caso de n dimensión son $SO(n)$ es decir, pertenecen **A un grupo especial ortogonal de orden n** , es decir

$$R_1^0 = (R_0^1)^{-1} = (R_0^1)^T \quad (3.7)$$

cuyas propiedades pueden ser listadas en la siguiente tabla

Tabla 3.1: Propiedades del grupo de matrices $SO(n)$

-
- Las columnas (y por lo tanto los renglones) de R son mutuamente ortogonales.
 - Cada columna (y cada renglón) de R es un vector unitario.
 - $\det R = 1$
-

Dentro del estudio de la robótica se utilizan principalmente las rotaciones sobre un eje fijo, ya sea el eje x, y o z , es por esta razón que a continuación se mostrará la metodología seguida para la obtención de la matriz de rotación sobre un eje.

La figura 3.3, la cual muestra una rotación sobre el eje z .

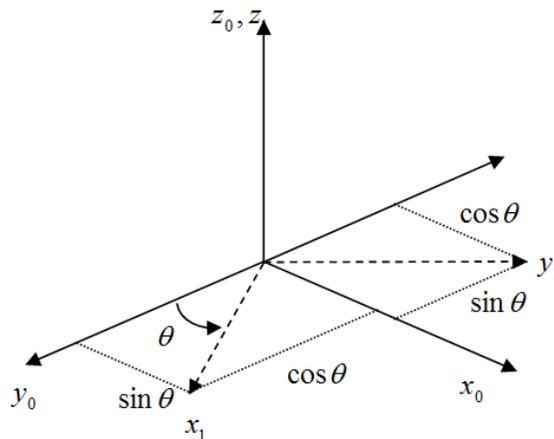


Figura 3.3: Rotación sobre z_0 por un ángulo θ .

Suponga que el marco $o_1x_1y_1z_1$ es rotado un ángulo θ sobre el eje z_0 y que se desea encontrar la matriz de transformación resultante R_1^0 .

Por lo tanto, a partir de la figura 3.3 se puede apreciar que las proyecciones de los ejes del marco $o_1x_1y_1z_1$ con respecto a los ejes del marco $o_0x_0y_0z_0$, están dadas por

$$\begin{aligned}x_1 \cdot x_0 &= \cos\theta, & y_1 \cdot x_0 &= -\sin\theta \\x_1 \cdot y_0 &= \sin\theta, & y_1 \cdot y_0 &= \cos\theta\end{aligned}\tag{3.8}$$

y

$$z_0 \cdot z_1 = 1\tag{3.9}$$

mientras que todos los demás productos son cero. Por lo tanto la matriz de rotación R_1^0 tiene una forma particular simple para este caso, siendo

$$R_{1,z}^0 = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\tag{3.10}$$

la cual recibe el nombre de **matriz de rotación básica** (sobre el eje z). Realizando un procedimiento análogo sobre los ejes x e y , se obtienen fácilmente las matrices de rotación para cada eje.

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}\tag{3.11}$$

$$R_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}\tag{3.12}$$

Una vez que se ha comprendido el significado de las matrices de rotación y cual es su utilidad en el campo de la robótica se procederá a describir lo que se conoce como movimientos rígidos.

3.2.2. Movimientos rígidos

Una vez que se ha visto como representar tanto la posición como la orientación, se procede a realizar la combinación de ambos conceptos con el fin de definir un **movimiento rígido**.

Definición. Un movimiento rígido es un par ordenado (d, R) donde $d \in R^3$ y $R \in SO(3)$. El grupo de todos los movimientos rígidos es conocido como el **Grupo especial Euclidiano** y es denotado por $SE(3)$. se observa que $SE(3) = R^3 \times SO(3)$

3. METODOLOGÍA

Un movimiento rígido es una traslación pura junto con una rotación pura, refiriéndose a la figura 3.4, se observa que el marco $o_1x_1y_1z_1$ es obtenido del marco $o_0x_0y_0z_0$ aplicando primeramente una rotación de ángulo θ seguido por una traslación (con respecto al marco $o_0x_0y_0z_0$) de longitud d_1^0 , entonces las coordenadas p^0 están dadas por

$$p^0 = R_1^0 p^1 + d_1^0 \quad (3.13)$$

Una vez comprendido lo que conlleva realizar un movimiento rígido, se puede continuar con la descripción de las transformaciones homogéneas, mismas que serán utilizadas para obtener la cinemática del manipulador.

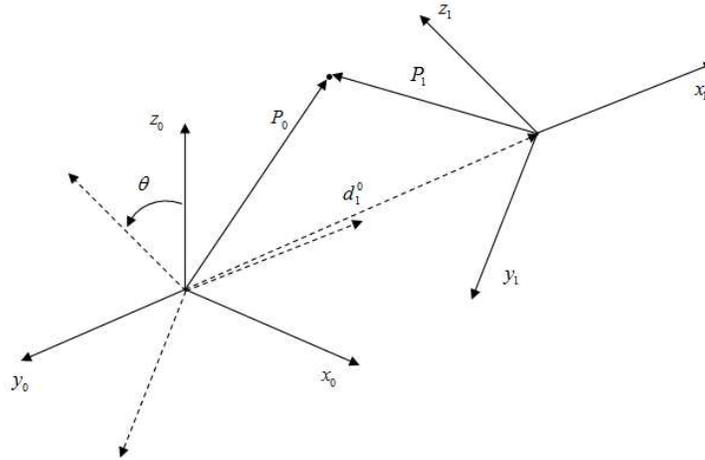


Figura 3.4: Movimiento rígido del vector P_1 .

3.2.3. Transformación homogénea

En esta sección se trata de describir la forma en que los movimientos rígidos pueden ser representados en forma matricial. De tal forma, que la composición de los movimientos rígidos pueda ser reducido a una multiplicación matricial, es decir las transformaciones homogéneas tienen la forma

$$H = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}; \quad R \in SO(3), \quad d \in R^3 \quad (3.14)$$

Por lo tanto un conjunto básico de **Transformaciones homogéneas** es dado por

$$Trans_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad Rot_{x,\alpha} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_\alpha & -s_\alpha & 0 \\ 0 & s_\alpha & c_\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (3.15)$$

$$Trans_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; Rot_{y,\beta} = \begin{bmatrix} c_\beta & 0 & s_\beta & 0 \\ 0 & 1 & 0 & 0 \\ -s_\beta & 0 & c_\beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (3.16)$$

$$Trans_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}; Rot_{z,\gamma} = \begin{bmatrix} c_\gamma & -s_\gamma & 0 & 0 \\ s_\gamma & c_\gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \quad (3.17)$$

Observe que para mayor comodidad se ha optado por representar al $\cos(\alpha)$ por c_α y al $\sin(\alpha)$ por s_α .

Por otro lado, la matriz homogénea descrita por la ecuación (3.14) es un caso especial de coordenadas homogéneas; la cual, ha sido ampliamente usada en el campo de los gráficos computacionales, donde uno esta interesado en escalar y/o transformar la perspectiva en conjunto con una traslación y una rotación. La transformación homogénea más general toma la forma

$$H = \left[\begin{array}{c|c} R_{3 \times 3} & d_{3 \times 1} \\ \hline f_{1 \times 3} & s_{3 \times 1} \end{array} \right] = \left[\begin{array}{c|c} Rotacion & Traslacion \\ \hline Perspectiva & Factordeescala \end{array} \right] \quad (3.18)$$

Para nuestro caso siempre tomaremos el ultimo vector renglón de H como $(0, 0, 0, 1)$, ahora que hemos comprendido como se realizan las transformaciones homogéneas, podemos continuar con el estudio de una de las partes mas importantes en el modelado del manipulador; la cinemática, debido a que a partir de ella podemos obtener la jacobiana que veremos en la sección 3.4 y la dinámica del manipulador; así como nos permitirá desarrollar la interfaz gráfica mediante la cual se realizará las simulaciones y el monitoreo descrito en la sección 3.12.

3.3. Cinemática

El problema de la cinemática consiste en describir el movimiento del manipulador sin la consideración de las fuerzas y torques que causan el movimiento. Por lo tanto, la descripción de la cinemática es puramente geométrica. En esta sección primeramente se considerará la *cinemática directa*, la cual es utilizada para determinar la posición y orientación del efector final, dados los valores de las variables de articulación del robot. Finalmente se presenta la solución de la *cinemática inversa*, cuyo problema es determinar los valores de las variables de articulación dada la posición y orientación del efector final. La figura 3.5 describe la diferencia radical entre la cinemática directa y la inversa.

3. METODOLOGÍA

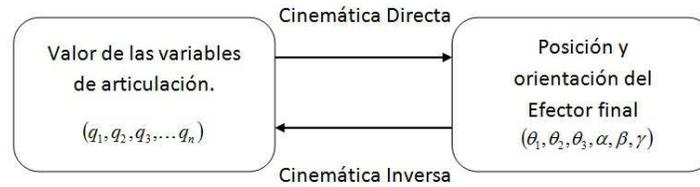


Figura 3.5: Diferencia entre cinemática directa e inversa.

Como se describió anteriormente, un robot manipulador está compuesto de un conjunto de eslabones conectados por medio de articulaciones. Las articulaciones pueden ser muy simples, tales como una articulación de revolución, una articulación prismática, o pueden ser más complejas, tales como una articulación de rótula. Recordando que una revoluta es como una bisagra y permite una rotación relativa sobre un único eje, y una articulación prismática permite un movimiento lineal a lo largo de un eje, es decir una extensión o una contracción.

La diferencia entre las dos situaciones es que, en la primera, la articulación tiene un movimiento de un solo grado de libertad: el ángulo de rotación en el caso de una revoluta, y la cantidad de desplazamiento lineal en el caso de la articulación prismática. En contraste, una articulación de tipo rótula tiene dos grados de libertad.

Para el desarrollo de esta tesis, el manipulador utilizado es un robot serial tipo *Rhino XR-3*, como el que se muestra en la figura 3.6.

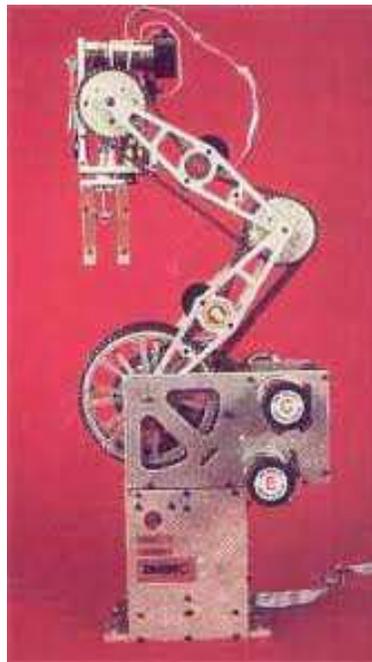


Figura 3.6: Manipulador Rhino XR-3.

3. METODOLOGÍA

De acuerdo a las características descritas anteriormente, es posible identificar que el manipulador consiste de un robot manipulador serial de cinco grados de libertad, para las cuales, cada una de sus cinco articulaciones son de tipo revoluta.

Un robot manipulador con n articulaciones tendrá $n + 1$ eslabones, debido a que cada articulación conecta dos eslabones. Por lo tanto, enumeramos los eslabones de 0 a n , comenzando a partir de la base. Para esta convención la articulación i conecta al eslabón $i - 1$, seguidamente a la i^{th} articulación y asociamos la variable de articulación denotada por q_i .

$$q_i = \begin{cases} \theta_i & \text{si la articulación } i \text{ es revoluta} \\ d_i & \text{si la articulación } i \text{ es prismática} \end{cases}$$

Para realizar el análisis cinemático, fijamos un marco de coordenadas rígidamente a cada eslabón. En particular, fijamos $o_i x_i y_i z_i$ al eslabón i . Esto significa que cualquier movimiento que el robot ejecute, las coordenadas de cada punto sobre la articulación i son constantes cuando son expresadas en el i^{th} marco de coordenada. Por otro lado, cuando la articulación i actúa, el eslabón i y su marco fijo experimentan un movimiento resultante.

El marco $o_0 x_0 y_0 z_0$, el cual es fijado a la base del robot, hace referencia al marco inercial, la figura 3.7.

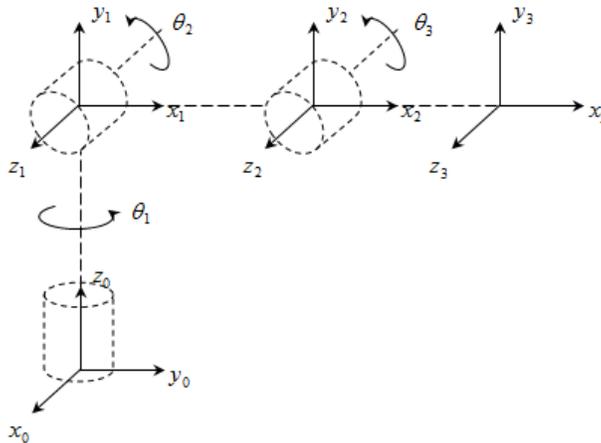


Figura 3.7: Marcos de coordenadas asignados al manipulador.

Una vez comprendido lo anterior se comenzará a describir la metodología utilizada en el cálculo de la cinemática directa.

3.3.1. Cinemática directa

El objetivo del análisis de la cinemática directa es determinar el efecto acumulativo del conjunto completo de las variables de articulación; esto es, determinar la posición y orientación del efector final dando los valores de sus variables de articulación.

Para el cálculo de la cinemática directa, es comúnmente usada la convención de Denavit-Hartenberg (DH). En esta convención, cada transformación homogénea A_i es representada como un producto de cuatro transformaciones básicas.

$$\begin{aligned}
 A_i &= Rot_{z,\theta_i} Trans_{z,d_i} Trans_{x,a_i} Rot_{x,\alpha_i} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i} & 0 & 0 \\ s_{\theta_i} & c_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &\times \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_{\alpha_i} & -s_{\alpha_i} & 0 \\ 0 & s_{\alpha_i} & c_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}s_{\alpha_i} & a_i c_{\alpha_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & -c_{\theta_i}s_{\alpha_i} & a_i s_{\alpha_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.19}
 \end{aligned}$$

Donde las cuatro cantidades θ_i , a_i , d_i , α_i son parámetros asociados con el eslabón i y la articulación i . Cuyos nombres generalmente son **ángulo de articulación**, **longitud de eslabón**, **offset de eslabón** y **giro del eslabón** respectivamente.

3.3.2. Asignación de marcos de coordenadas

Para un robot manipulador dado, siempre es posible elegir los marcos $0, \dots, n$ de tal manera que las condiciones anteriores sean satisfechas.

Para comenzar, se eligió la dirección de los ejes z_i ; por lo tanto, asignamos los ejes z_0, \dots, z_{n-1} en una forma estandarizada. Específicamente, se asigna z_i como el eje de actuación para la articulación $i - 1$. Por lo tanto, z_0 es el eje de actuación para la articulación 1, z_1 es el eje de actuación para la articulación 2, etc. Sin embargo, existen dos casos a considerar:

- Si la articulación $i + 1$ es una junta rotatoria, z_i es el eje de revolución de la articulación $i + 1$.

3. METODOLOGÍA

- Si la articulación $i + 1$ es una junta prismática, z_i es el eje de traslación de la articulación $i + 1$.

Una vez establecido el eje z para cada una de las articulaciones, establecimos el marco base. La elección de un marco base es ciertamente arbitraria. Por lo tanto, la elección del origen o_0 del marco base puede ser asignado en cualquier punto a lo largo del eje z_0 , por consiguiente elegimos a los ejes x_0, y_0 de manera conveniente, siempre y cuando se cumpla la regla de la mano derecha.

Como paso siguiente, localizamos el origen o_i donde la normal común a z_i y z_{i-1} interseca a z_i . Si z_i interseca a z_{i-1} se ubica a o_i en esa intersección. Si z_i y z_{i-1} son paralelos, se ubica a o_i en cualquier posición conveniente a lo largo de z_1 .

Una vez establecidos los orígenes de los marcos de referencia, se procedió a ubicar los ejes x_i a lo largo de la normal común entre z_{i-1} y z_i a través de o_i , o en la dirección normal al plano $z_{i-1} - z_i$ si z_{i-1} y z_i intersecan.

Con los ejes z_i y x_i establecidos, se procedió con la ubicación de los ejes y_i para completar el marco de referencia siguiendo la regla de la mano derecha.

El siguiente paso consistió en establecer el marco del efector final $o_n x_n y_n z_n$. Asumiendo que la n^{th} articulación es una revoluta, se estableció a $z_n = a$ a lo largo de la dirección z_{n-1} . Una vez elegido, fue establecido el origen o_n convenientemente a lo largo de la dirección z_n ; preferentemente en el centro del gripper o en la punta de cualquier herramienta que maneje el manipulador. El eje $y_n = s$ es elegido en la dirección de cierre del gripper y a $x_n = s \times a$, si el efector final no fuese un gripper se establecen los ejes x_n y y_n convenientemente para formar un marco establecido por la regla de la mano derecha.

Una vez establecidos cada uno de los marcos de referencia como se muestra en la figura 3.8, se procedió a la creación de la tabla de parámetros de los eslabones $a_i, d_i, \alpha_i, \theta_i$ (Tabla *DH*).

3. METODOLOGÍA

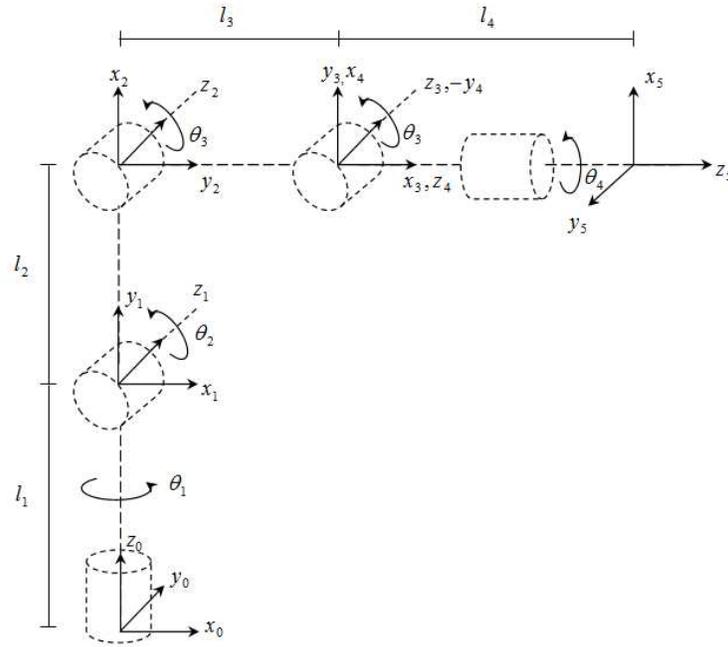


Figura 3.8: Marcos de coordenadas del robot Rhino XR-3.

Con el fin de conocer la definición de cada uno de los parámetros de la tabla DH, se han consolidado las definiciones de los parámetros en la siguiente lista.

a_i = Distancia a lo largo del eje x_i de o_i a la intersección de los ejes z_i y z_{i-1} .

d_i = Distancia a lo largo del eje z_{i-1} de o_{i-1} a la intersección de los ejes x_i y z_{i-1} .
 d_i es variable si la articulación i es prismática.

α_i = El ángulo entre z_{i-1} y z_i medido sobre x_i .

θ_i = El ángulo entre x_{i-1} y x_i medido sobre z_{i-1} . θ_i es variable si la articulación i es una revoluta.

En la tabla se muestra el consolidado de los parámetros DH, del robot Rhino XR-3.

Tabla 3.2: Tabla de parámetros DH $SO(n)$

Eslabón	a_i	α_i	d_i	θ_i
1	0	$-\frac{\pi}{2}$	l_1	θ_1^*
2	l_2	0	0	θ_2^*
3	l_3	0	0	θ_3^*
4	0	$-\frac{\pi}{2}$	0	θ_4^*
5	0	0	l_4	θ_4^*

Utilizando los parámetros obtenidos mediante la tabla DH, se procedió a la obtención de las matrices de transformación homogénea A_i . Sustituyendo los

3. METODOLOGÍA

parámetros obtenidos dentro de la ecuación (3.19) se obtuvieron las siguientes matrices.

$$\begin{aligned}
 A_1 &= \begin{bmatrix} c\theta_1 & 0 & -s\theta_1 & 0 \\ s\theta_1 & 0 & c\theta_1 & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_2 &= \begin{bmatrix} c\theta_2 & -s\theta_1 & 0 & l_2C\theta_2 \\ s\theta_2 & c\theta_1 & 0 & l_2s\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_3 &= \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & l_3C\theta_3 \\ s\theta_3 & c\theta_3 & 0 & l_3s\theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & A_4 &= \begin{bmatrix} c\theta_4 & 0 & -s\theta_4 & 0 \\ s\theta_4 & 0 & c\theta_4 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.20)
 \end{aligned}$$

A través de las transformaciones homogéneas, se formaron las matrices cambio de base $T_n^0 = A_1 \dots A_n$. Estas matrices dan la posición y orientación del marco de la herramienta expresada en coordenadas de la base.

$$\begin{aligned}
 T_1^0 &= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 T_2^0 &= \begin{bmatrix} c_1c_2 & -c_1s_2 & -s_1 & l_2c_1c_2 \\ s_1c_2 & -s_1s_2 & c_1 & l_2s_1c_2 \\ -s_2 & -c_2 & 0 & -l_2s_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 T_3^0 &= \begin{bmatrix} c_1c_{23} & -c_1s_{23} & -s_1 & l_3c_1c_{23} + l_2c_1c_2 \\ s_1c_{23} & -s_1s_{23} & c_1 & l_3s_1c_{23} + l_2s_1c_2 \\ -s_{23} & -c_{23} & 0 & -l_3s_{23} - l_2s_2 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 T_4^0 &= \begin{bmatrix} c_1c_{234} & s_1 & -c_1s_{234} & l_3c_1c_{23} + l_2c_1c_2 \\ s_1c_{234} & -c_1 & -s_1s_{234} & l_3s_1c_{23} + l_2s_1c_2 \\ -s_{234} & 0 & -c_{234} & -l_3s_{23} - l_2s_2 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.21)
 \end{aligned}$$

3.3.3. Cinemática inversa

A lo largo de esta sección, se desarrollará la metodología utilizada para la obtención de las ecuaciones de cinemática inversa del robot Rhino XR-3.

Para esta investigación solamente se utilizarán 4 gdl, correspondientes a las primeras cuatro articulaciones del robot, siendo las siguientes:

3. METODOLOGÍA

Waist Corresponde al movimiento de la cintura(base) del manipulador, su movimiento es rotacional y su variable de articulación será denotada como θ_0 .

Shoulder Corresponde al movimiento del hombro del manipulador, su movimiento es rotacional y su variable de articulación será denotada como θ_1 .

Elbow Corresponde al movimiento del codo del manipulador, su movimiento es rotacional y su variable de articulación será denotada como θ_2 .

Wrist Corresponde al movimiento de la muñeca del manipulador, su movimiento es rotacional y su variable de articulación será denotada como θ_3 .

Cada una de las partes del manipulador y su respectiva variable de articulación se pueden apreciar en la figura 3.9.

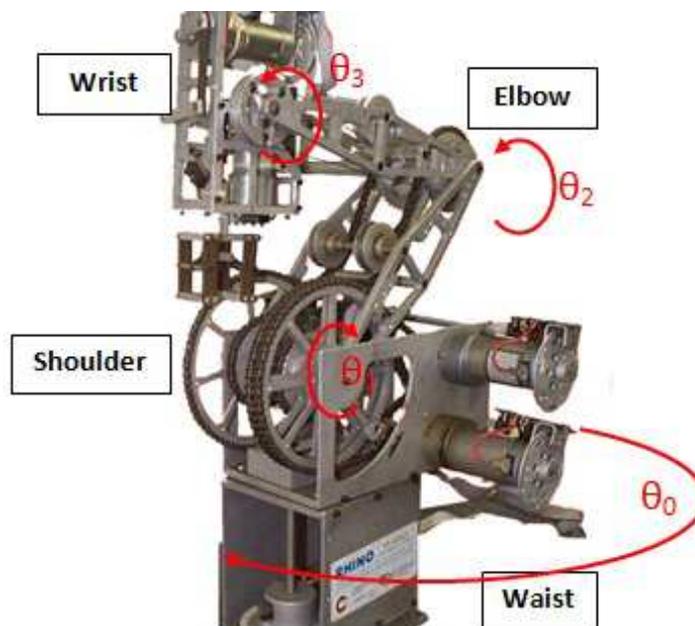


Figura 3.9: Esquema general de robot Rhino XR-3.

Antes de comenzar a desarrollar ecuaciones, es necesario definir el término de cinemática inversa. En la sección pasada, se mostró la metodología para la obtención de ecuaciones que nos permitieran determinar la posición y orientación del efector final en términos de las variables de articulación. Es decir, a través del conocimiento de los ángulos de rotación de las articulaciones se puede obtener la posición final del brazo manipulador.

En contra parte, esta sección tendrá como objetivo el análisis del problema cinemática inverso, el cual consiste en encontrar las variables de articulación en términos de la posición del efector final, la posición y la orientación. Esto en

3. METODOLOGÍA

general es a lo que se conoce como " Cinemática Inversa ". Siendo en general más difícil que el problema de cinemática directa.

En este capítulo, describiremos la solución al problema de la cinemática directa aprovechando la geometría del manipulador.

Posición inversa: Un aprovechamiento geométrico

Debido a la configuración del manipulador Rhino XR-3. Podemos hacer uso del aprovechamiento geométrico para encontrar las variables de articulación, $\theta_0, \theta_1, \theta_2, \theta_3$ correspondiente a o_c^0 , que es el centro de la muñeca del brazo manipulador.

donde:

$$o_c^0 = o - l_3 R \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.22)$$

$$R = R_4^0(q1, q2, q3, q4)$$

$$o = o_4^0(q1, q2, q3, q4)$$

Si los componentes de la posición del efector final o es denotado como o_x, o_y, o_z y los componentes del centro de la muñeca esférica como x_c, y_c, z_c , entonces la ecuación (3.22) da la relación.

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \begin{bmatrix} o_x - l_3 r_{11} \\ o_y - l_3 r_{21} \\ o_z - l_3 r_{31} \end{bmatrix} \quad (3.23)$$

Usando la ecuación (3.23) podemos encontrar los valores del origen del centro de la muñeca, la cual es de vital importancia para el uso del aprovechamiento geométrico.

Restringimos el uso del aprovechamiento geométrico por dos razones. Primero, la mayoría de los diseños de los manipuladores son cinemáticamente simples, usualmente consisten de uno de las cinco configuraciones básicas Revoluta - Revoluta - Revoluta (RRR), Manipulador esférico (RRP), Scara (Selective Compliant Assembly Robot Arm), Manipulador cilíndrico y manipulador cartesiano. De hecho, es parcialmente debido a la dificultad del problema de la cinemática inversa general, que los diseños de los manipuladores incluyen estas configuraciones. Segundo, hay pocas técnicas que pueden manejar el problema de la cinemática inversa general para configuraciones arbitrarias.

En general, la complejidad del problema de la cinemática inversa incrementa con el número de parámetros de eslabón diferentes de cero. Para la mayoría de los manipuladores, muchos de los parámetros a_i, d_i son cero, los α_i son 0 ó $\pm \frac{\pi}{2}$, etc. En esos casos especialmente, el aprovechamiento geométrico es más simple

3. METODOLOGÍA

y natural. La idea general del aprovechamiento geométrico, es resolver para la variable de articulación q_i proyectando el manipulador sobre el plano $x_{i-1} - y_{i-1}$ y resolviendo un problema trigonométrico simple.

Siguiendo con esta metodología, se presenta la forma de encontrar el valor de la variable de articulación θ_0 de la junta 0. Considere el codo del manipulador mostrado en la figura 3.10, con los componentes de o_c denotados por x_c, y_c, z_c .

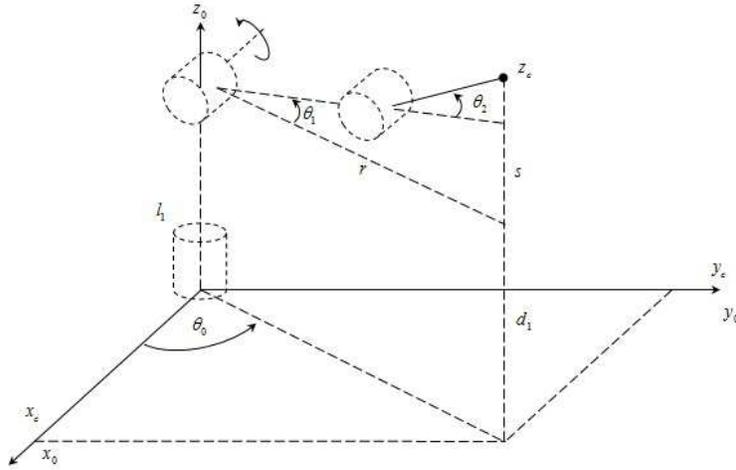


Figura 3.10: Codo de manipulador Rhino XR-3.

Primero proyectamos el origen de la muñeca del manipulador o_c sobre el plano $x_0 - y_0$ como se muestra en la figura 3.11.

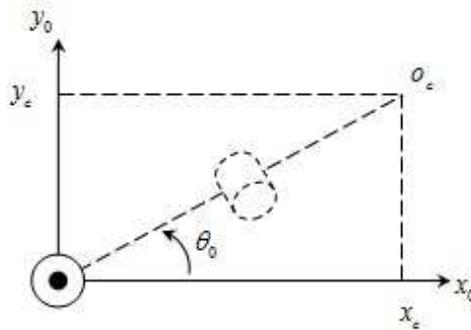


Figura 3.11: Proyección de la muñeca sobre el plano $x_0 - y_0$.

Como se puede observar, a partir de la proyección en la figura 3.11 es posible encontrar una relación trigonométrica para encontrar el valor de θ_0 :

$$\theta_0 = \text{atan2}(x_c, y_c) \quad (3.24)$$

En donde el término $\text{atan2}(x, y)$ denota, la función arcotangente de dos argumentos definida como:

3. METODOLOGÍA

$$\text{atan2}(x, y) = 2\text{atan}\left(\frac{y}{\sqrt{x^2 + y^2} + x}\right) \quad (3.25)$$

donde x_c e y_c representan las coordenadas del centro de la muñeca del manipulador.

Los valores de las coordenadas del centro de la muñeca, son hasta el momento desconocidos. Por lo tanto, aprovecharemos la arquitectura del manipulador, por lo tanto observando la figura 3.12, se aprecia que la ecuación (3.24) se puede reescribir haciendo uso de los términos de las coordenadas del efector final.

$$\theta_0 = \text{atan2}(x_c, y_c) = \text{atan2}(x_f, y_f) \quad (3.26)$$

$$\theta_0 = \text{atan2}(x_f, y_f) \quad (3.27)$$

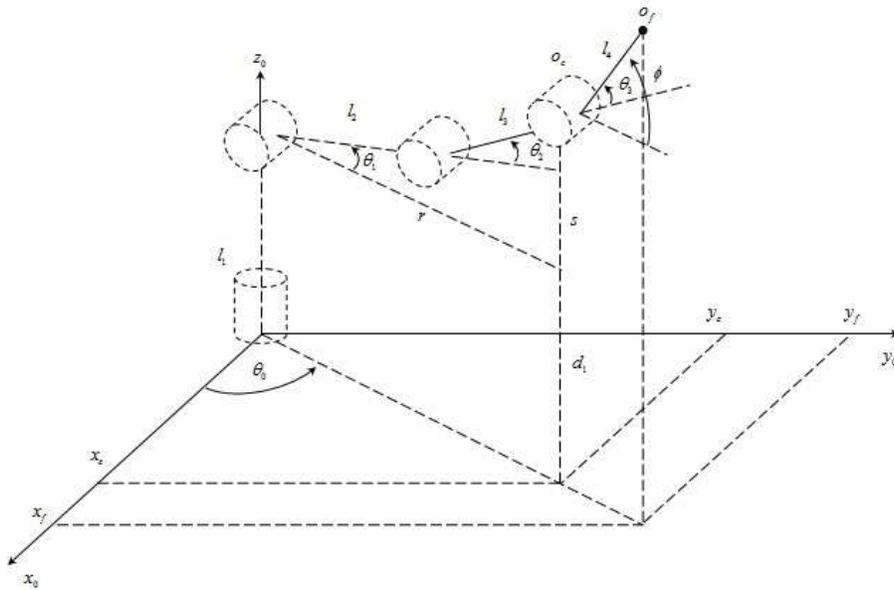


Figura 3.12: Arquitectura del manipulador Rhino XR-3.

El cálculo del ángulo θ_0 es en realidad el mas sencillo de realizar. A continuación, se detalla la metodología para encontrar las variables de articulación θ_1 , θ_2 y θ_3 .

Por lo tanto, el problema siguiente es encontrar las coordenadas x_c e y_c a partir de las coordenadas del efector final x_f e y_f y el ángulo de orientación ϕ (pitch).

En la figura 3.12, podemos apreciar la relación entre los ángulos de las variables de articulación y el ángulo de orientación deseado, por lo tanto, el ángulo ϕ puede definirse como:

3. METODOLOGÍA

$$\phi = \theta_1 + \theta_2 + \theta_3 \quad (3.28)$$

Una vez establecida la ecuación (3.28), se pueden encontrar las relaciones trigonométricas requeridas para el cálculo del origen de la muñeca.

Para el cálculo de la coordenada x_c establecemos la siguiente relación.

$$x_f = x_c + \Delta_x$$

\therefore

$$\Delta_x = l4\cos(\phi)\cos(\theta_0) \quad (3.29)$$

Despejando la coordenada x_c obtenemos la siguiente ecuación:

$$x_c = x_f - l4\cos(\phi)\cos(\theta_0) \quad (3.30)$$

De manera similar es posible encontrar el valor de la coordenada y_c . Siguiendo con la metodología anterior, definimos y_f como:

$$y_f = y_c + \Delta_y$$

\therefore

$$\Delta_y = l4\cos(\phi)\sin(\theta_0) \quad (3.31)$$

Despejando a y_c y sustituyendo a Δ_y obtenemos la siguiente relación.

$$y_c = y_f - l4\cos(\phi)\sin(\theta_0) \quad (3.32)$$

Con las ecuaciones anteriores se han encontrado las coordenadas x_c e y_c del manipulador. Solo resta encontrar la coordenada z_c , para ello observemos nuevamente la figura 3.12, mediante la cual podemos obtener la siguiente relación.

$$z_f = z_c + \Delta_z$$

\therefore

$$\Delta_z = l4\sin(\phi) \quad (3.33)$$

A partir de la ecuaciones anteriores es posible encontrar el valor de z_c .

$$z_c = z_f - l4\sin(\phi) \quad (3.34)$$

Una vez encontrado el centro de la muñeca del manipulador cuyas coordenadas están dadas por las ecuaciones (3.30),(3.32) y (3.34), podemos escribir las relaciones que nos permitan encontrar los ángulos θ_1 , θ_2 y θ_3 los cuales esta en función de las coordenadas del centro de la muñeca.

3. METODOLOGÍA

Para la formulación de los ángulos θ_1 , θ_2 y θ_3 es preciso realizar la proyección del manipulador sobre el plano $z_0 - x_0$ y realizar el desacoplo cinemático. Es decir, considerar al manipulador sin muñeca. El realizar el desacoplo facilita la resolución de las ecuaciones; por lo tanto, la posición final del manipulador será el punto de las coordenadas anteriormente calculado, es decir, el centro de la muñeca.

Al realizar la proyección obtendremos el diagrama de un manipulador planar tal como lo muestra la figura 3.13.

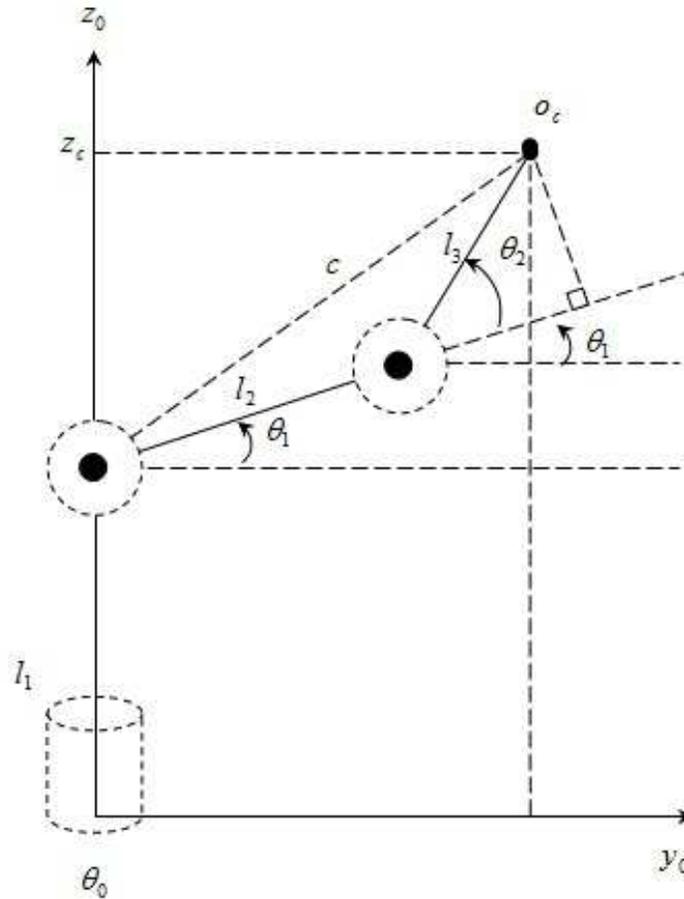


Figura 3.13: Proyección de manipulador sobre el plano $z_0 - x_0$

A partir de la figura 3.13, observamos que al trazar una línea a partir de la articulación 1 hacia la articulación 3, se dibuja un triángulo escaleno, mediante el cual es posible obtener el valor del ángulo θ_2 .

Utilizando la **Ley de Cosenos**, la cual se define como:

$$c^2 = a^2 + b^2 - 2ab\cos(\theta) \quad (3.35)$$

y su representación gráfica se muestra en la figura 3.14.

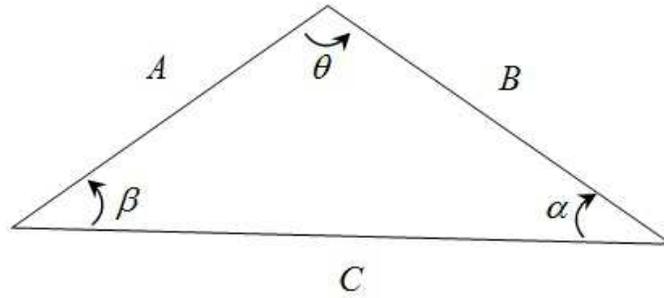


Figura 3.14: Ley de coseno

Aplicando la *ley de coseno* al triángulo formado observamos que el ángulo θ_2 está dado por:

$$\cos(\theta_2) = \frac{x_c^2 + x_c^2 - l_1^2 - l_2^2}{2l_1l_2} = D \quad (3.36)$$

Ahora podemos determinar a θ_2 como:

$$\theta_2 = \cos^{-1}(D) \quad (3.37)$$

Sin embargo, esta solución solo representa una solución para la configuración del manipulador, por lo tanto, si la ecuación 3.36 representa a $\cos(\theta_2)$ entonces el $\sin(\theta_2)$ puede ser expresado como:

$$\sin(\theta_2) = \pm\sqrt{1 - D^2} \quad (3.38)$$

Recordando la definición de tangente, podemos escribir el valor de θ_2 como:

$$\theta_2 = \tan^{-1}\left(\frac{\pm\sqrt{1 - D^2}}{D}\right) \quad (3.39)$$

La ventaja de realizar el cálculo del ángulo θ_2 de esta forma, es debido, a que mientras el punto del efector final deseado, se encuentre dentro del area de trabajo del manipulador, dicho punto puede ser alcanzado en dos configuraciones diferentes a las que llamaremos **Codo abajo** y **Codo arriba**, tal fenómeno se muestra en la figura 3.15.

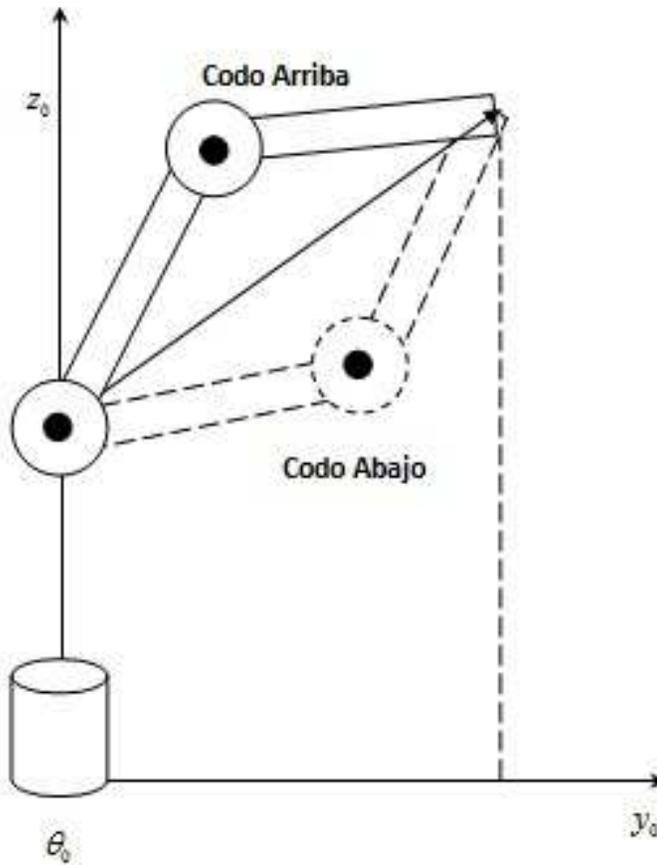


Figura 3.15: Múltiples soluciones a cinemática inversa

Observando nuevamente la figura 3.13, podemos definir las ecuaciones necesarias para encontrar el valor del ángulo θ_1 .

$$\theta_1 = \tan^{-1} \left(\frac{l_1 - z_c}{\sqrt{x_c^2 + y_c^2}} \right) - \tan^{-1} \left(\frac{l_3 \sin(\theta_2)}{l_2 + l_3 \cos(\theta_2)} \right) \quad (3.40)$$

Finalmente en la ecuación (3.28), definimos a ϕ como el ángulo de orientación del manipulador con respecto al eje x , por lo tanto, es posible obtener el valor de ángulo θ_3 :

$$\theta_3 = \phi - \theta_1 - \theta_2 \quad (3.41)$$

Las ecuaciones (3.27), (3.40), (3.39) y (3.41), resuelven el problema de la cinemática inversa del manipulador serial de 4 gdl convencional. Desafortunadamente el manipulador Rhino XR-3, presenta una variación en la configuración al momento de realizar sus movimientos.

3. METODOLOGÍA

En la figura 3.16, podemos apreciar el movimiento general de un manipulador convencional.

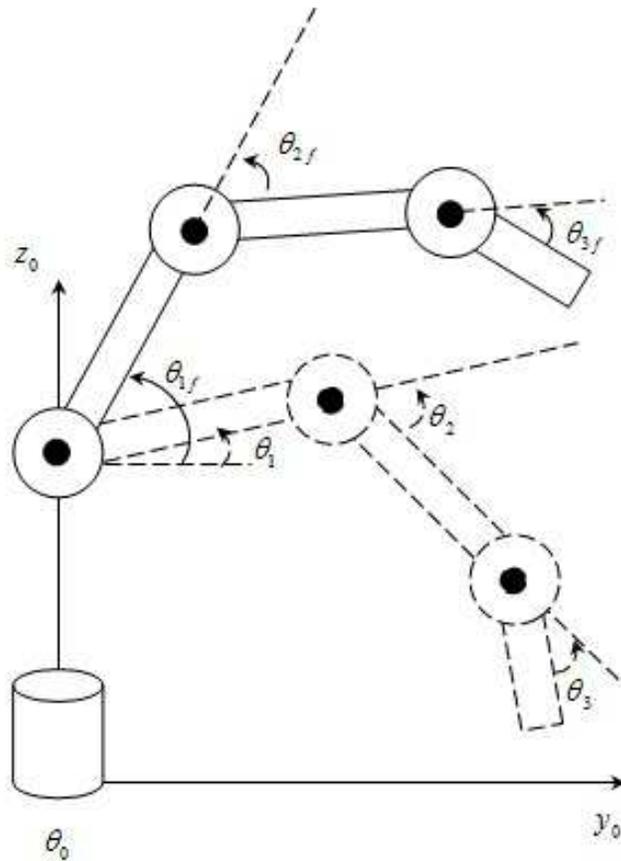


Figura 3.16: Movimiento robot serial convencional

Como puede apreciarse, el movimiento realizado por cada articulación es independiente a cada una de ellas. A diferencia del movimiento del manipulador Rhino XR-3, cuyos movimientos se encuentran relacionados internamente, tal como se muestra en la figura 3.17.

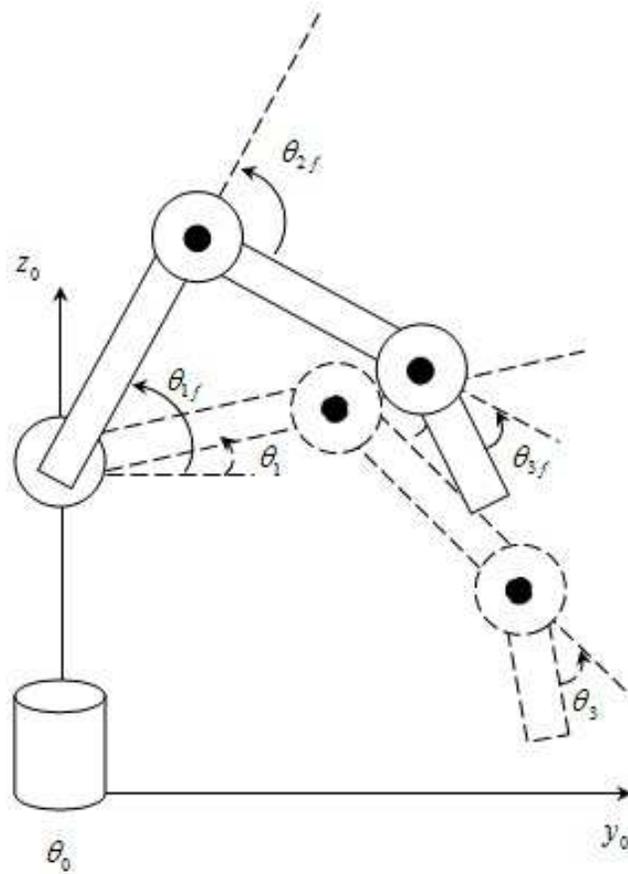


Figura 3.17: Movimiento robot Rhino XR-3

Debido a esto, es necesario encontrar la relación existente entre los movimientos de las articulaciones. Para ello consideremos la posición inicial del manipulador tal como se muestra en la figura 3.18.

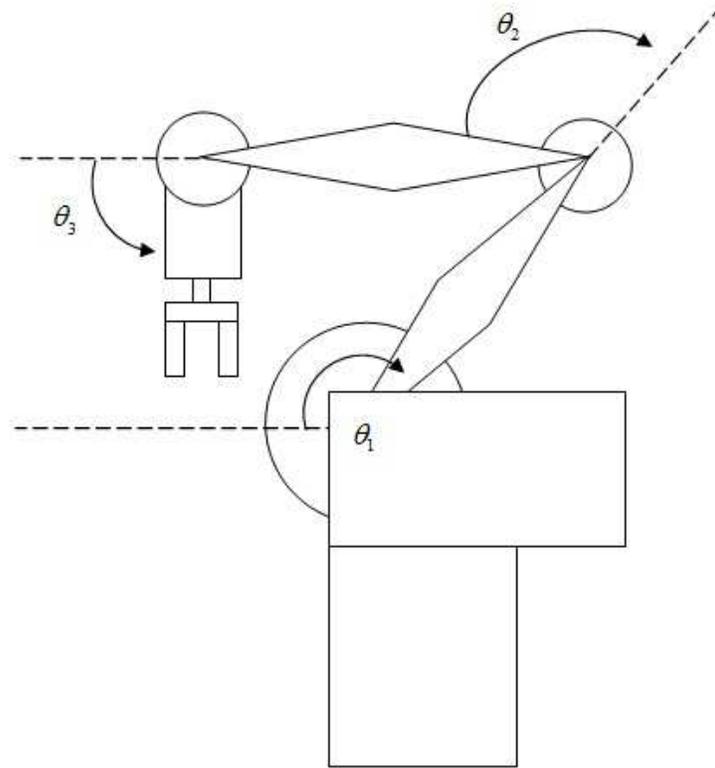


Figura 3.18: Configuración inicial (Home)

Como puede observarse, la posición inicial del robot, se encuentra dada por las variables de articulación cuyos valores se encuentran en la tabla 3.3.3.

Tabla 3.3: Valores nominales de articulación, para posición de home.

Articulación	Ángulo °
Base	0
Hombro	135
Codo	-135
Muñeca	-90

Tomando en consideración la posición inicial del manipulador, se procedió a obtener la relación que se mantiene entre las articulaciones. Un factor que facilita mucho el desarrollo de las ecuaciones, es el hecho que la relación de la transmisión de las articulaciones, es la misma. De esta manera se puede establecer que el valor de la variable de articulación θ_i realiza un cambio de la misma magnitud sobre la variable de articulación θ_{i+1} .

Una vez establecido lo anterior, se procedió a identificar el efecto que producía el realizar un movimiento sobre la articulación 2 (hombro) con respecto a la articulación 3 (Codo), este efecto es mostrado en la figura 3.19.

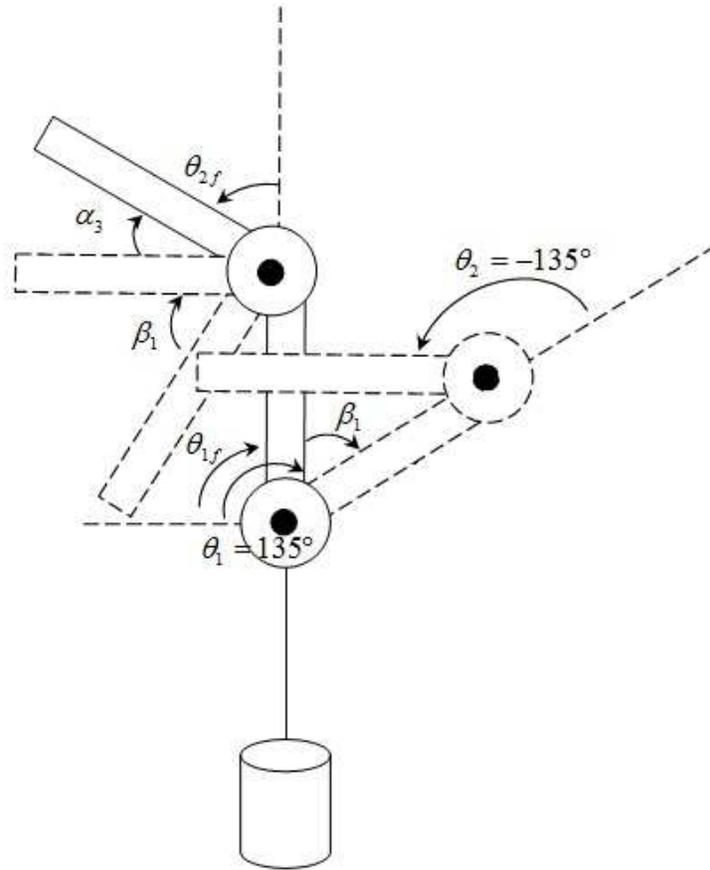


Figura 3.19: Relación de movimiento entre hombro y codo.

En la figura 3.19, muestra el efecto que se produce cuando la articulación 1 (Hombro) realiza una rotación, la posición obtenida por un manipulador convencional, la posición obtenida por el manipulador Rhino XR-3 y finalmente la posición obtenida en el momento en que la articulación 2 (codo), realiza una rotación adicional. Haciendo un análisis a la figura anterior, podemos comenzar definiendo algunos conceptos:

α_3 ; Ángulo de rotación debido al movimiento de la articulación y es la variable de control para la articulación 2.

$$\beta_1 := 135 - \theta_{1f}; \text{ Cambio en la variable de articulación 2} \quad (3.42)$$

Con las ecuaciones anteriores es posible derivar una ecuación para encontrar el valor θ_{2f} .

$$\theta_{2f} = -135 + \beta_1 + \alpha_3 \quad (3.43)$$

Sustituyendo la ecuación (3.42) en la ecuación (3.43).

3. METODOLOGÍA

$$\begin{aligned}\theta_{2f} &= -135 + 135 - \theta_1 + \alpha_3 \\ \theta_{2f} &= -\theta_1 + \alpha_3\end{aligned}\tag{3.44}$$

Por lo tanto, despejando a α_3 que es el parámetro de interés obtenemos la siguiente ecuación.

$$\alpha_3 = \theta_1 + \theta_{2f}\tag{3.45}$$

Hasta el momento se han obtenido las ecuaciones para la cinemática inversa de un robot de 3 grados de libertad. Lo único que falta es encontrar el valor para la articulación 3 (muñeca).

Para encontrar este parámetro, se realizará una metodología similar a la anterior. Para ello, tomaremos como referencia a la figura 3.20.

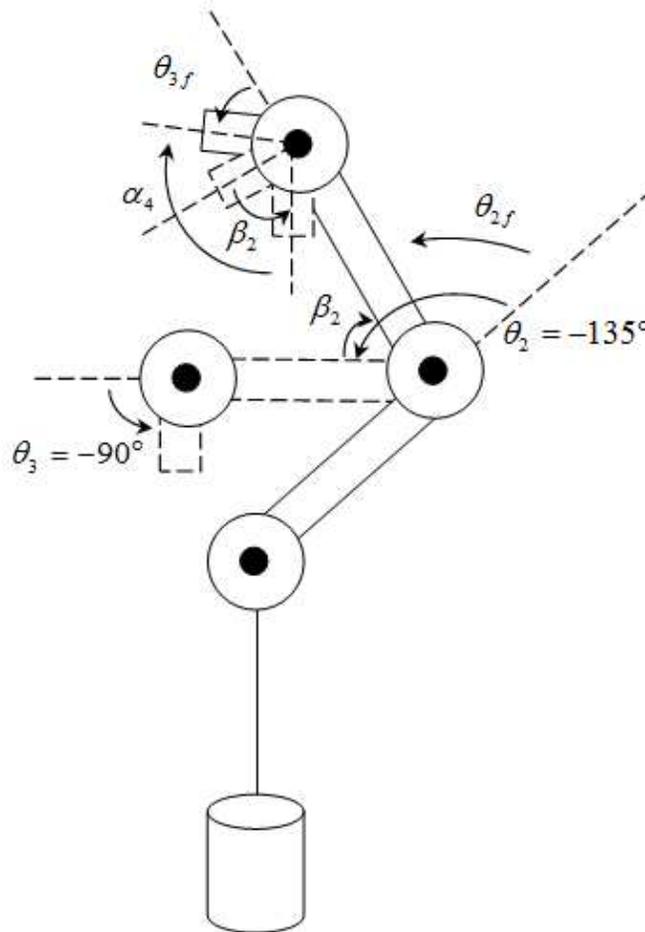


Figura 3.20: Relación de movimiento entre codo y muñeca.

En la figura 3.20, se muestran la posición obtenida por un manipulador convencional, la posición obtenida del manipulador Rhino XR-3, y la posición obtenida del manipulador considerando una rotación en la muñeca adicional.

Definiendo a α_4 como el ángulo de rotación ejercido por el motor de la articulación 3 y es la variable de control para la articulación 2.

β_2 es definido como el cambio en la variable de articulación 3.

$$\beta_2 := -135 - \theta_{2f} \quad (3.46)$$

$$\theta_{3f} := -90 + \beta_2 - \alpha_4 \quad (3.47)$$

Sustituyendo la ecuación (3.46) dentro de la ecuación (3.47), obtenemos:

$$\theta_{3f} := -90 - 135 - \theta_{2f} - \alpha_4$$

$$\theta_{3f} := -225 - \theta_{2f} - \alpha_4 \quad (3.48)$$

Despejando a α_3 , de la ecuación anterior se obtiene la variable de control deseada:

$$\alpha_4 := -225 - \theta_{2f} - \theta_3 \quad (3.49)$$

Con la derivación de la ecuación (3.49) se ha concluido el problema de la cinemática inversa del manipulador Rhino XR-3.

En la siguiente sección se derivaran las ecuaciones para obtener la velocidad cinemática del manipulador, mismas que son de utilidad en la formulación de las ecuaciones de la dinámica del manipulador.

3.4. Velocidad cinemática

En la sección anterior se derivaron las ecuaciones de posición a partir de la cinemática directa, relacionando las posiciones de las articulaciones a la posición y orientación del efector final. En esta sección se derivarán las relaciones de velocidad, relacionando las velocidades lineales y angulares de las velocidades de articulación del efector final.

Matemáticamente, las ecuaciones de cinemática directa definen una función entre el espacio de las posiciones, las orientaciones cartesianas y el espacio de la posición de las articulaciones. La relación de velocidad son entonces determinados por la **jacobi**ana de su función. La jacobiana es una de los parámetros mas importantes en el análisis y control del movimiento de robots. Esta se encuentra prácticamente en cada aspecto de los manipuladores robóticos: en la planeación y ejecución de

3. METODOLOGÍA

trayectorias, en la determinación de las configuraciones singulares, en la ejecución de movimiento de coordenadas antropomorfas, en la derivación de las ecuaciones dinámicas de movimiento y en la transformación de fuerzas y torques del efector final a las articulaciones del manipulador.

Por lo tanto una vez descrito la importancia de la *jacobiana*, se procederá a definir las características y los pasos a seguir para el cálculo de la jacobiana del manipulador.

Un marco de coordenadas en movimiento tiene ambas velocidades, velocidad lineal y velocidad angular. La velocidad lineal esta asociada a un punto en movimiento, mientras que la velocidad angular esta asociada a un marco rotativo. Por lo tanto, la velocidad lineal de un marco en movimiento es meramente la velocidad de su origen. La velocidad angular para un marco en movimiento esta relacionada con la derivada con respecto al tiempo de la matriz de rotación que describe la orientación instantánea del marco. En particular, si $R(t) \in SO(3)$, entonces:

$$R'(t) = S(w(t))R(t) \quad (3.50)$$

y el vector $w(t)$ es la velocidad angular instantánea del marco. El operador S nos indica una matriz antisimétrica definida como:

$$S(w) = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix} \quad (3.51)$$

El manipulador jacobiano relaciona el vector de las velocidades de articulación a la velocidad del cuerpo $\xi = (v, w)^T$ al efector final.

$$\xi = J\dot{q} \quad (3.52)$$

Esta relación puede ser escrita como dos ecuaciones, una para la velocidad lineal y una para la velocidad angular.

$$\begin{aligned} v &= J_v \dot{q} \\ w &= J_w \dot{q} \end{aligned} \quad (3.53)$$

La i^{th} columna de la matriz jacobiana corresponde a la i^{th} articulación del robot manipulador, y toma una de las dos formas, dependiendo si la i^{th} articulación es prismática o revoluta.

$$J_i = \begin{cases} \begin{bmatrix} z_{i-1} \times (o_n - o_{i-1}) \\ z_{i-1} \end{bmatrix} & \text{Si la articulación } i \text{ es revoluta.} \\ \begin{bmatrix} z_{i-1} \\ 0 \end{bmatrix} & \text{Si la articulación } i \text{ es prismática.} \end{cases}$$

3. METODOLOGÍA

Por lo tanto, aplicando la metodología anterior y teniendo el conocimiento de que las articulaciones del manipulador son solamente del tipo rotatorio, la jacobiana queda como sigue.

$$\begin{aligned}
 & J_1 \begin{bmatrix} z_0 \times (o_5 - o_0) \\ z_0 \end{bmatrix} J_2 \begin{bmatrix} z_1 \times (o_5 - o_1) \\ z_1 \end{bmatrix} J_3 \begin{bmatrix} z_2 \times (o_5 - o_2) \\ z_2 \end{bmatrix} \\
 & J_4 \begin{bmatrix} z_3 \times (o_5 - o_3) \\ z_3 \end{bmatrix} J_4 \begin{bmatrix} z_4 \times (o_5 - o_4) \\ z_4 \end{bmatrix} \tag{3.54}
 \end{aligned}$$

siendo los parámetros de la jacobiana los siguientes:

$$\begin{aligned}
 z_0 &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}; \quad z_1 = z_2 = z_3 = \begin{bmatrix} -s_1 \\ c_1 \\ 0 \end{bmatrix}; \quad z_4 = \begin{bmatrix} -c_1 s_{234} \\ -s_1 s_{234} \\ -c_{234} \end{bmatrix} \\
 o_0 &= \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; \quad o_1 = \begin{bmatrix} 0 \\ 0 \\ l_1 \end{bmatrix}; \quad o_2 = \begin{bmatrix} l_2 c_1 c_2 \\ l_2 s_1 c_2 \\ -l_2 s_2 + d_1 \end{bmatrix}; \quad o_3 = o_4 = \begin{bmatrix} l_3 c_1 c_{23} + l_2 c_1 c_2 \\ l_3 s_1 c_{23} + l_2 s_1 c_2 \\ -l_3 s_{23} - l_2 s_2 + d_1 \end{bmatrix}; \\
 o_5 &= \begin{bmatrix} -l_4 c_1 s_{234} + l_3 c_1 c_{23} + l_2 c_1 c_2 \\ -l_4 s_1 s_{234} + l_3 s_1 c_{23} + l_2 s_1 c_2 \\ -l_4 c_{234} - l_3 s_{23} - l_2 s_2 + d_1 \end{bmatrix} \tag{3.55}
 \end{aligned}$$

3.5. Implementación de controladores

Comenzaremos con la definición de los parámetros que intervienen en el desarrollo del controlador. Para ello, se comenzará con una breve introducción a los sistemas de control.

Tal como se describió en la sección *Revisión literaria*, los sistemas de control formar una parte de vital importancia en nuestras vidas hoy día. Puesto que todas las herramientas tecnológicas que usamos a diario son manejadas por algún tipo de controlador. Pero antes de analizar los S.C. deben definirse ciertos términos básicos como son *variable controlada* y *variable manipulada*; tal y como los describe (Ogata, 1998).

Variable controlada: es la cantidad o condición que se mide y controla, por lo común, la variable controlada es la salida (el resultado) del sistema.

3. METODOLOGÍA

Variable manipulada: es la cantidad o condición que el controlador modifica para afectar el valor de la variable controlada.

Controlar: medir el valor de la variable controlada del sistema y aplicar la variable manipulada al sistema para corregir o limitar una desviación del valor medido a partir de un valor deseado.

En el estudio de la ingeniería de control, necesitamos definir términos adicionales que resultan necesarios para describir los sistemas de control.

Planta: Puede ser una parte de un equipo, tal vez un conjunto de las partes de una máquina que funcionan juntas, el propósito de la cual es ejecutar una operación particular. Llamaremos planta a cualquier objeto físico que se va a controlar (tal como un dispositivo mecánico, un horno de calefacción, un reactor químico o una nave espacial).

Proceso: (Merriam-Webster, 2006) Lo define como una operación o un desarrollo natural progresivamente continuo, marcado por una serie de cambios graduales que se suceden uno al otro en una forma relativamente fija y que conducen a un resultado o propósito determinados; o una operación artificial o voluntaria progresiva que consiste en una serie de acciones o movimientos controlados, sistemáticamente dirigidos hacia un resultado o propósito determinados. En este escrito llamaremos proceso a cualquier operación que se va a controlar. Algunos ejemplos son los procesos químicos, económicos y biológicos.

Sistema: Es una combinación de componentes que actúan juntos y realizan un objetivo determinado. Un sistema no necesariamente es físico. El concepto de sistema se aplica a fenómenos abstractos y dinámicos, tales como los que se encuentran en la economía. Por tanto, la palabra sistema debe interpretarse como una implicación de sistemas físicos, biológicos, económicos y similares.

Perturbación: Una perturbación es una señal que tiende a afectar negativamente el valor de la salida de un sistema. Si la perturbación se genera dentro del sistema se denomina interna, en tanto que una perturbación externa se produce fuera del sistema y es una entrada.

Control retroalimentado: Se refiere a una operación que, en presencia de perturbaciones, tiende a reducir la diferencia entre la salida de un sistema y alguna entrada de referencia y lo continúa haciendo con base en esta diferencia.

Con las definiciones anteriores se puede tener un amplio panorama de lo que concierne al área de control industrial, cuyo principal objetivo es la de mantener en un rango de operación aceptable a una o varias variables en tiempo real, ya sea Temperatura(T), Voltaje(V), Corriente(I), Nivel(L), Flujo(F), etc.

Tal como se comentó en la sección 2.2, en los últimos años se han desarrollado una gran cantidad de algoritmos enfocados al control de posición de manipuladores industriales.

3. METODOLOGÍA

Por esta razón, esta investigación tiene como objetivo principal la implementación de controladores PD + compensación, dichos controladores PD, son robustos con respecto a incertidumbres en los parámetros de inercia. Se han elegido tres algoritmos de control, siendo los siguientes:

[**PD + I**] Algoritmo mejor conocido como PID, y es el algoritmo de control mayormente utilizado en la industria, debido a su gran robustez, confiabilidad y fácil implementación, tal como lo presentan (Tae-Yong and Woong, 2000), (Jose *et al.*, 2000) y (Rocco, 1996).

[**PD + W y PD + W (Adaptable)**] El cual fue presentado por (Tang *et al.*, 2000), cuyo parámetro w fue diseñado para realizar la compensación de los efectos producidos por la incertidumbre, interconexiones y perturbaciones del sistema.

[**PD + G**] Algoritmo de control adaptable con respecto a los parámetros de gravedad del robot manipulador presentado por (Tomei, 1991).

En las secciones siguientes, se presenta la descripción para los algoritmos de control.

3.6. PD+I (PID)

En esta sección se presenta la implementación del controlador PD + una compensación integral, mayormente conocido como controlador PID . Este tipo de controlador es adecuado para aplicaciones que no involucren movimientos rápidos, especialmente en robots con una relación grande entre el actuador y las articulaciones.

A lo largo de este controlador se considerará el sistema de la siguiente forma:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (3.56)$$

donde:

$q(t) \in \mathbb{R}$ son las coordenadas generalizadas de el sistema

$M(q)$ denota la matriz de inercia del sistema.

$C(q, \dot{q})\dot{q}$ son los términos de las fuerzas centrífugas y de coriolis.

$g(q)$ denota los términos de energía potencial.

τ las fuerzas generalizadas aplicadas al sistema.

Estas ecuaciones pueden ser obtenidas a través del modelado de sistemas mecánicos mediante ecuaciones de Euler-Lagrange. Para el desarrollo del controlador, las fuerzas generalizadas τ son consideradas de la forma

$$\tau = -K_P e(q) - K_D \dot{q} - K_I \int e(q) \quad (3.57)$$

Donde K_p representa a la constante proporcional del controlador; K_D especifica el valor derivativo del controlador; K_I representa al valor integral del controlador PID y $e(q)$ define al error de posición del sistema y es definido como

3. METODOLOGÍA

$$e(q) = q_d - q \quad (3.58)$$

Siguiendo con la metodología, en la figura 3.21 se observa el esquema general del controlador PID aplicado a un motor de CD.

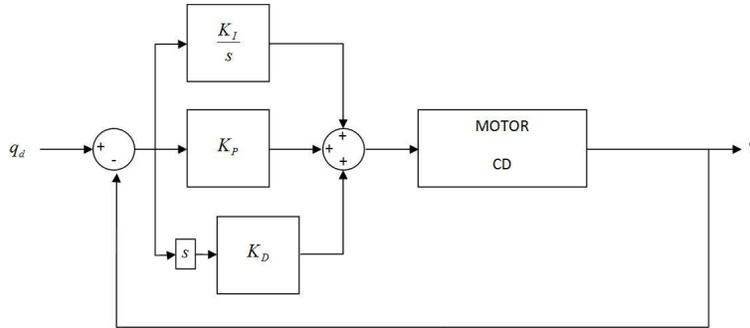


Figura 3.21: Diagrama a bloques control PID.

Por lo tanto, al utilizar un controlador tipo proporcional integral y derivativo, se asegura que ante una perturbación se anule el error en régimen permanente. Sin embargo, de usar sólo un regulador PD no se consigue este efecto. Por otra parte, es sabido que la acción integral K_i necesaria para eliminar el error en régimen permanente tiende a empeorar la respuesta dinámica de la articulación. También se ha demostrado que con la acción de prealimentación, es posible eliminar el error de seguimiento, siempre y cuando se tenga un buen modelo del sistema, es decir, el seguimiento absoluto de la trayectoria deseada q_d ante ausencia de perturbación, se conseguiría solo si los parámetros del sistema fuesen perfectamente conocidos. En la práctica, habrá que conformarse con tener una buena estimación de estos valores y admitir ciertas desviaciones entre q_d y q .

El código para la implementación del controlador bajo el lenguaje Java se puede apreciar en el anexo (5.1).

3.7. PD+W

Este controlador fue diseñado bajo la primicia de ser un control descentralizado, el cual garantice la estabilidad uniforme final, cuando el objetivo de control sea una trayectoria deseada suave.

Debido a la complejidad del análisis y diseño de los sistemas de control para sistemas a gran escala y la incapacidad de intercambiar información entre los subsistemas en la implementación del sistema de control, es más conveniente diseñar para cada subsistema un controlador descentralizado, el cual dependa solamente de las variables locales medibles. Un controlador descentralizado ofrece varias ventajas sobre los controles centralizados, por ejemplo, robustez frente a errores, eficiencia computacional y simplicidad de diseño (Siljak, 1991).

Para la realización de dicho controlador (Tang *et al.*, 2000), utilizó algunas propiedades físicas de los sistemas mecánicos y fue diseñado bajo la metodología

3. METODOLOGÍA

de Lyapunov para hacer frente a las no-linealidades existentes. De esta manera se evitó realizar la cancelación de las no-linealidades del sistema.

Considere el sistema mecánico descrito por:

$$\begin{aligned} M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) &= \tau_i \\ i &= 1, 2, \dots, N. \end{aligned} \quad (3.59)$$

donde:

$q_i(t) \in \mathbb{R}$ son las coordenadas generalizadas de el subsistema $i \in [1, N]$, $N \in \mathbb{Z}_+$

$M_i(q)$ denota la matriz de inercia con dimensiones $N \times N$.

n denota el número de grados de libertad del sistema. $C_i(q_i, \dot{q}_i)\dot{q}_i$ son los términos de las fuerzas centrífugas y de coriolis.

$g_i(q_i)$ denota los términos de energía potencial.

τ_i las fuerzas generalizadas aplicadas al subsistema.

Estas ecuaciones pueden ser obtenidas a través del modelado de sistemas mecánicos mediante ecuaciones de Euler-Lagrange. Para el desarrollo de este controlador, las fuerzas generalizadas τ_i son consideradas de la forma

$\tau_i = u_i - Z(q_i, \dot{q}_i) - D_i(q_i, \dot{q}_i)$, con u_i un vector n_i -dimensional de entradas de control, $q^T = [q^T q^T \dots q^N] \in \mathbb{R}^n$, $n = \sum_{i=1}^N n_i$. donde:

$Z(q_i, \dot{q}_i)$ implica el vector de interconexiones del sistema.

$D(q_i, \dot{q}_i)$ denota al vector de perturbaciones del sistema.

Por lo tanto, la ecuación 3.59, puede ser descrita como:

$$\begin{aligned} M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + g_i(q_i) + Z(q_i, \dot{q}_i) + D_i(q_i, \dot{q}_i) &= u_i \\ i &= 1, 2, \dots, N. \end{aligned} \quad (3.60)$$

Para el desarrollo del controlador, se utilizó la metodología basada en Lyapunov. Para ello (Tang *et al.*, 2000) propuso la siguiente función definida positiva.

$$V = \sum_{i=1}^N V_i \triangleq \sum_{i=1}^N \frac{1}{2} s_i^T M_i S_i \quad (3.61)$$

A través de la metodología anterior, el autor obtuvo la siguiente ley de control, siendo los términos definidos con el subíndice 0 el valor nominal de $(\cdot)_i$:

$$u_i = M_i^0(q_i)\ddot{q}_i^r + C_i^0(q_i, \dot{q}_i)\dot{q}_i^r + g_i^0(q_i) - K_i s_i + w_i \quad (3.62)$$

Definiendo a la señal de error del sistema como $\tilde{q}_i(t) \triangleq q_i(t) - q_i^d(t)$ con $q_i^d(t)$ definido como la trayectoria deseada. Haciendo uso de parametrización de la dinámica local de cada subsistema, definida como:

$$M_i(q_i)\ddot{q}_i + C(q_i, \dot{q}_i)\dot{q}_i + q_i(q_i) = Y_i(q_i, \dot{q}_i, \ddot{q}_i)\Theta_i \quad (3.63)$$

donde:

$Y_i(q_i, \dot{q}_i, \ddot{q}_i)$ indica al vector $1 \times n$ denominador regresor, correspondiente a los parámetros

3. METODOLOGÍA

conocidos del sistema.

Θ_i indica al vector $n \times 1$, correspondiente a las variables de articulación del sistema.

La ecuación (3.62), puede ser escrita como:

$$u_i = Y_i(q_i, \dot{q}_i, \ddot{q}_i^r, \ddot{q}_i^r) \Theta_i^0 - K_i s_i + w_i \quad (3.64)$$

donde:

\ddot{q}_i^r es la velocidad de referencia y s_i es una variable deslizante definida como:

$$\ddot{q}_i^r \triangleq \ddot{q}_i^d - \Lambda \tilde{q}_i \quad s_i \triangleq \dot{q}_i + \Lambda_i \tilde{q}_i = \dot{q}_i - \dot{q}_i^r$$

Con $K_i, \Lambda \in \mathbb{R}^{n_i \times n_i}$ definidas positivas, y $w(t) \in \mathbb{R}^{R_i}$ fue diseñada para compensar los parámetros de incertidumbre, perturbaciones e interconexiones.

$$w_i = -(\delta S_i)^2 \frac{s_i}{\delta \|s_i\| + \epsilon_i} \quad (3.65)$$

$$S_i \triangleq 1 + \|s_i\| + \dots + \|s_i\|^p \quad (3.66)$$

Con $\epsilon_i > 0$ un parámetro de diseño, $p \geq 1$, $\|x\|$ denota la norma euclideana, y $\delta > 0$ que depende de la cota de la fuerza de los parámetros de incertidumbre, perturbaciones e interconexiones (Tang *et al.*, 2000).

El código para la implementación del controlador bajo el lenguaje Java se puede apreciar en el anexo (5.2).

3.8. PD+W (adaptable)

En la práctica el parámetro δ de la ecuación (3.65) es complicado de conocer. Por lo tanto, es necesario realizar un algoritmo para realizar su estimación.

En el mismo artículo (Tang *et al.*, 2000) presenta, el desarrollo de una ecuación que permite realizar la estimación del parámetro δ .

Sea $\hat{\delta}(t)$ el cual denota el valor de δ en el instante t en el i th subsistema, el cual es actualizado de acuerdo a:

$$\begin{aligned} \dot{\hat{\delta}} &= -\sigma \hat{\delta} + \gamma_i S_i \|s_i\| \\ \hat{\delta}(t_0) &> 0 \end{aligned} \quad (3.67)$$

Donde $\sigma, \gamma > 0$ son parámetros de diseño. Por lo tanto, la señal de compensación w_i está ahora dada por:

$$w_i = -(\hat{\delta}_i S_i)^2 \frac{s_i}{\hat{\delta}_i S_i \|s_i\| + \epsilon_i} \quad (3.68)$$

El código para la implementación del controlador bajo el lenguaje Java se puede apreciar en el anexo (5.3).

3.9. PD+G (Compensación de gravedad)

El controlador utilizado en esta sección fue presentado por (Tomei, 1991), el cual basó su estudio en el problema de control de trayectorias punto a punto. El resultado de su investigación fue un algoritmo PD adaptable con respecto a los parámetros de gravedad del robot manipulador.

Como es conocido, las leyes de control basadas en retroalimentación de posición y velocidad de las articulaciones han demostrado ser asintóticamente estables, siempre que los términos de gravedad sean compensados. También se ha demostrado que los controladores PD pueden ser usados para el seguimiento de trayectoria, con una precisión relacionada a las ganancias de retroalimentación de velocidad. Además, tales algoritmos de control son robustos con respecto a incertidumbres en los parámetros de inercia; es decir, aún si los parámetros de inercia no son conocidos, la estabilidad asintótica global (GAS) es asegurada. A la inversa, las incertidumbres en los parámetros de gravedad (tales como la carga) pueden llevar a errores indeseables en estado estacionario.

Por lo tanto (Tomei, 1991) propone un controlador PD adaptable que lleve a la estabilidad asintótica del sistema, aún si los parámetros de inercia y gravedad son desconocidos.

Dentro de este artículo se considera la dinámica de un manipulador de n -articulaciones para un robot rígido descrito como:

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + e(q) + F\dot{q} = u \quad (3.69)$$

donde q es el vector $n \times 1$ de coordenadas de articulación; $B(q)$ es la matriz de inercia, la cual es simétrica y definida positiva; $C(q, \dot{q})$ toma en cuenta las fuerza centrífuga y de coriolis; F es una matriz diagonal semidefinida positiva de los coeficientes de fricción viscosa; u es el vector de torques aplicados y $e(q)$ es el vector de fuerzas de gravedad dado por

$$e(q) = \frac{\partial U(q)}{\partial q} \quad (3.70)$$

donde $U(q)$ es la energía potencial del robot.

Como es conocido, con referencia al control punto a punto de los manipuladores, un control PD con retroalimentación independiente para cada articulación, asegura la estabilidad global asintótica de todo el sistema, siempre que los términos de gravedad sean exactamente compensados, lo cual se puede escribir como

$$u = e(q_0) - k_P(q - q_0) - K_D\dot{q} \quad (3.71)$$

donde K_P y K_D son matrices simétricas definidas positivas y además se debe cumplir la hipótesis de que $\lambda_m(K_P) > M_i$, con M_i una constante positiva, tal que $\left\| \frac{\partial e(q)}{\partial q} \right\| \leq M_1$ y el punto de equilibrio $q = q_0$, $\dot{q} = 0$ es globalmente asintóticamente estable.

Para el desarrollo del controlador adaptable, (Tomei, 1991) utilizó la técnica de parametrización, debido a que el vector $e(q)$ es lineal en términos de los parámetros del robot, puede ser expresado como

3. METODOLOGÍA

$$e(q) = E(q)p \quad (3.72)$$

donde p es el vector $m \times 1$ de parámetros desconocidos, el cual es asumido constante y $E(q)$ es una matriz conocida. Por lo tanto, considerando la ley de control

$$u = -K_P \tilde{q} - K_D \dot{\tilde{q}} + E(q) \hat{p} \quad (3.73)$$

con el parámetro de adaptación dinámico

$$\dot{\hat{q}} = -\beta E^T(q) \left[\gamma \dot{\tilde{q}} + \frac{2\tilde{q}}{1 + 2\tilde{q}^T \tilde{q}} \right] \quad (3.74)$$

donde $\tilde{q} = q - q_0$ es el error de posición, K_P y K_D son matrices simétricas y definidas positivas; β es una constante positiva y γ es tal que

$$\gamma > \max \left\{ \frac{2\lambda_M(B)}{\sqrt{\lambda_m(B)\lambda_m(K_P)}}, \frac{1}{\lambda_m(K_D)} \cdot \left[\frac{(\lambda_M(K_D) + \lambda_M(F))^2}{2\lambda_m(K_P)} + 4\lambda_M(B) + \frac{K_C}{\sqrt{2}} \right] \right\} \quad (3.75)$$

Por lo tanto la ecuación (3.75), fue implementada bajo el lenguaje java, para lo cual fue requerido realizar la parametrización del manipulador Rhino XR-3, tal como se muestra a continuación.

Tomando como referencia la figura 3.22

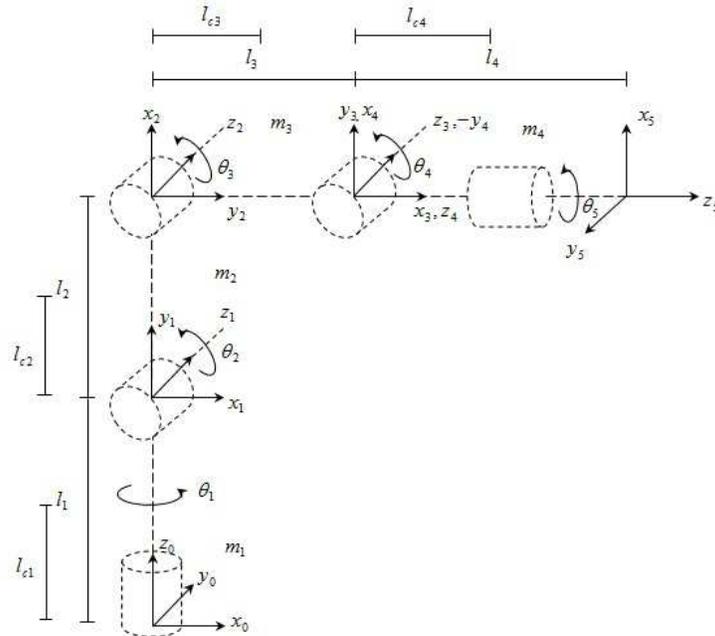


Figura 3.22: Ubicación de centros de inercia.

3. METODOLOGÍA

Se puede apreciar la ubicación de los centros de inercia denotados como L_{ci} . Dichos puntos son requeridos para la obtención de las ecuaciones de energía potencial, la cual puede ser escrita como:

$$U_i(q) = m_i g h \quad (3.76)$$

donde m define a la matriz de inercia definida positiva; g corresponde al coeficiente de gravedad y h corresponde a la distancia existente entre la posición de referencia y el centro de masa de la $i - th$ articulación.

Utilizando la ecuación (3.76), podemos obtener la energía potencial para cada eslabón del manipulador. Inicialmente se obtuvo la energía potencial del eslabón 1.

$$U_1 = m_1 g l_1 \quad (3.77)$$

Siguiendo con la metodología anterior, se encontró la energía potencial para las articulaciones restantes

$$U_2 = m_2 g (h + l_{c2} \sin(\theta_2)) \quad (3.78)$$

$$U_3 = m_3 g (h + l_2 \sin(\theta_2) + l_{c3} \sin(\theta_2 + \theta_3)) \quad (3.79)$$

$$U_4 = m_4 g (h + l_2 \sin(\theta_2) + l_3 \sin(\theta_2 + \theta_3) + l_{c4} \sin(\theta_2 + \theta_3 + \theta_4)) \quad (3.80)$$

finalmente la ecuación de energía potencial para todo el sistema es descrita como la suma de las energías potenciales de los subsistemas, es decir.

$$U = U_1 + U_2 + U_3 + U_4 \quad (3.81)$$

Una vez obtenida la energía potencial del sistema, es necesario obtener el vector de energía gravitacional, utilizando la ecuación (3.70) se obtuvieron las siguientes ecuaciones.

$$e(q_1) = \frac{\partial U(q)}{\partial q_1} = 0 \quad (3.82)$$

$$\begin{aligned} e(q_2) &= \frac{\partial U(q)}{\partial q_2} = m_2 g l_{c2} \cos(\theta_2) + m_3 g l_2 \cos(\theta_2) + m_3 g l_{c3} \cos(\theta_2 + \\ &\quad \theta_3) + m_4 g l_2 \cos(\theta_2) + m_4 g l_3 \cos(\theta_2 + \theta_3) + m_4 g l_{c4} \cos(\theta_2 + \theta_3 + \theta_4) \\ &= (m_2 l_{c2} + m_4 l_2 + m_3 l_2) g \cos(\theta_2) + (m_3 l_{c3} + m_4 l_3) g \cos(\theta_2 + \theta_3) + \\ &\quad m_4 l_{c4} g \cos(\theta_2 + \theta_3 + \theta_4) \end{aligned} \quad (3.83)$$

$$\begin{aligned} e(q_3) &= \frac{\partial U(q)}{\partial q_3} = m_3 g l_{c3} \cos(\theta_2 + \theta_3) + m_4 l_3 \cos(\theta_2 + \theta_3) \\ &\quad + m_4 g l_{c4} \cos(\theta_2 + \theta_3 + \theta_4) \\ &= (m_3 l_{c3} + m_4 l_3) g \cos(\theta_2 + \theta_3) + m_4 l_{c4} g \cos(\theta_2 + \theta_3 + \theta_4) \end{aligned} \quad (3.84)$$

$$e(q_4) = \frac{\partial U(q)}{\partial q_4} = m_4 l_{c4} g \cos(\theta_2 + \theta_3 + \theta_4) \quad (3.85)$$

3. METODOLOGÍA

Con las ecuaciones de gravedad obtenidas, se realizó la parametrización del sistema, cuya forma es definida por la ecuación (3.72), obteniendo lo siguiente

$$E(q)p := e(q)$$

$$E(q) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \cos(\theta_2) & 0 & 0 \\ 0 & \cos(\theta_2 + \theta_3) & \cos(\theta_2 + \theta_3) & 0 \\ 0 & \cos(\theta_2 + \theta_3 + \theta_4) & \cos(\theta_2 + \theta_3 + \theta_4) & \cos(\theta_2 + \theta_3 + \theta_4) \end{bmatrix}$$
$$p = \begin{bmatrix} 0 \\ g(m_2l_{c2} + m_3l_2 + m_4l_2) \\ g(m_3l_{c3} + m_4l_3) \\ gm_4l_{c4} \end{bmatrix}$$

donde $E(q)$ es una matriz conocida y p es el vector de parámetros definidos por el sistema y desconocidos, por lo tanto, las componentes del vector de gravedad pueden ser estimadas utilizando lo siguiente

$$\hat{e}(q) = E(q)\hat{p} \quad (3.86)$$

para lo cual \hat{p} es calculado mediante la ley de adaptación descrita en la ecuación (3.74).

El código para la implementación del controlador bajo el lenguaje Java se puede apreciar en el anexo (5.4).

3.10. Modelado de motor CD

Debido a que los controladores definidos en secciones anteriores, utilizan el vector τ como señal de control; se requiere encontrar la relación entre el voltaje aplicado a los motores y su torque generado.

Por lo tanto, se definirá la función de transferencia para una clase particular de motor, que es el servomotor controlado por armadura. En la figura 3.23 se muestra el diagrama esquemático del motor.

3. METODOLOGÍA

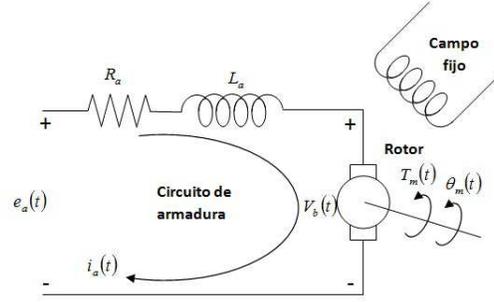


Figura 3.23: Diagrama esquemático para motor de CD.

Puesto que la armadura portadora de corriente se encuentra girando en un campo magnético, su voltaje es proporcional a la velocidad. Entonces,

$$v_b(t) = K_b \frac{d\theta_m(t)}{dt} \quad (3.87)$$

A $v_b(t)$ se le da el nombre de *fuerza contraelectromotriz (fem)*; K_b es una constante de proporcionalidad llamada constante de fuerza contraelectromotriz y $\frac{d\theta_m(t)}{dt} = \omega_m(t)$ es la velocidad angular del motor.

Realizando un análisis eléctrico de malla, se puede obtener el valor del voltaje de alimentación denotado como $e_a(t)$.

$$e_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + v_b(t) \quad (3.88)$$

El par generado por el motor es directamente proporcional a la corriente de armadura. Es decir,

$$\tau_m(t) = K_T i_a(t) \quad (3.89)$$

donde τ_m es el par creado por el motor y K_T , una constante de proporcionalidad, llamada constante de par del motor, que depende de las características del motor y del campo magnético.

Un reacomodo de la ecuación 3.89, permite obtener el valor de la corriente de armadura.

$$i_a(t) = \frac{\tau_m(t)}{K_T} \quad (3.90)$$

Sustituyendo las ecuaciones 3.90 y 3.87 en 3.88, podemos obtener la siguiente ecuación.

$$e_a(t) = R_a \frac{\tau_m(t)}{K_T} + L_a \frac{d\tau_m(t)}{K_T dt} + K_b \frac{d\theta_m(t)}{dt} \quad (3.91)$$

En la realidad, la inductancia del motor es mucho menor que la resistencia de armadura. Por lo tanto, se puede considerar a la inductancia como nula. Es decir, $L_a = 0$. Tomando en consideración esto, la ecuación 3.91 se puede escribir como:

$$e_a(t) = R_a \frac{\tau_m(t)}{K_T} + \frac{K_b}{dt} \quad (3.92)$$

A partir de la ecuación 3.92, se puede despejar el valor de τ .

$$\begin{aligned}\tau_m(t) &= e_a(t) \frac{K_T}{R_a} - \frac{K_b K_T}{R_a} \frac{d\theta_m(t)}{dt} \\ \tau_m(t) &= e_a(t) \frac{K_T}{R_a} - \frac{K_b K_T}{R_a} \omega_m(t)\end{aligned}\tag{3.93}$$

La ecuación 3.93, es la utilizada para implementar los controladores, con el fin de convertir la señal de par a voltaje.

3.11. Interfaz gráfica

Para el desarrollo de la interfaz gráfica se ha elegido el lenguaje de programación Java, debido a las amplias características que este lenguaje proporciona.

Java ha sido corregido, mejorado, ampliado y probado por una comunidad especializada de más de 6,5 millones de desarrolladores, la mayor y más activa del mundo. Gracias a su versatilidad, eficiencia y portabilidad, Java se ha convertido en un recurso inestimable ya que permite a los desarrolladores:

- Desarrollar software en una plataforma y ejecutarlo en prácticamente cualquier otra plataforma.
- Crear programas para que funcionen en un navegador web y en servicios web.
- Desarrollar aplicaciones para servidores como foros en línea, tiendas, encuestas, procesamiento de formularios HTML, etc.
- Combinar aplicaciones o servicios que usan el lenguaje Java para crear servicios o aplicaciones totalmente personalizados.
- Desarrollar potentes y eficientes aplicaciones para teléfonos móviles, procesadores remotos, productos de consumo de bajo costo y prácticamente cualquier tipo de dispositivo digital.

La interfaz fue desarrollada siguiendo la metodología de ser una interfaz amigable al usuario, mediante la cual, se puedan realizar simulaciones de la trayectoria a seguir por el manipulador fuera de línea. Es decir, el sistema muestra de forma gráfica un robot manipulador virtual, dando la flexibilidad de realizar pruebas de trayectorias y del sistema de control sin la necesidad de estar conectado al robot físicamente.

Debido a esto se ha incorporado la biblioteca OpenGL, permitiendo generar el ambiente tridimensional.

OpenGL (para la "Biblioteca de Open Graphics") es una interfaz de software para hardware de gráficos. La interfaz se compone de un conjunto de varios cientos de procedimientos y funciones que permiten a un programador especificar los objetos y las operaciones involucradas en la producción de imágenes gráficas de alta calidad, en concreto las imágenes en color de los objetos en tres dimensiones.

3. METODOLOGÍA

La mayoría de los comandos de OpenGL requiere que el hardware de gráficos contenga un framebuffer. Muchas llamadas de OpenGL se refieren a objetos de dibujo como puntos, líneas y polígonos, pero la forma en que algunos de estos dibujos se producen (por ejemplo, cuando se activa el antialiasing) se basa en la existencia de un framebuffer.

Tomando como base la biblioteca OpenGL, se procedió a realizar el simulador del manipulador, en la figura 3.24 se muestra el modelo virtual del manipulador.

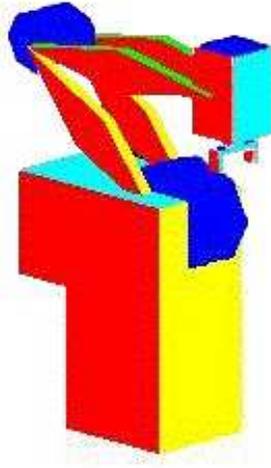


Figura 3.24: Representación virtual del manipulador.

El sistema consiste en la representación gráfica del movimiento virtual perteneciente al manipulador, para lo cual fueron requeridas las ecuaciones de cinemática, dinámica, y leyes de control. Dicha interfaz cuenta con diversos modos de operación, siendo los siguientes:

Modo de control manual Mediante la cual es posible manipular al robot de forma independiente a cada una de las articulaciones.

Modo de control automático Permite realizar el seguimiento de trayectorias previamente almacenadas, utilizando las leyes de control preestablecidas.

Modo de planeación de trayectorias Permite generar trayectorias punto a punto, y ser almacenadas en la PC mediante el formato de archivo de texto, que es utilizado por el modo de control automático.

La interfaz de movimiento manual, se puede apreciar en la figura 3.25, misma que contiene diversos *sliders* con el fin de controlar manualmente a cada articulación.

3. METODOLOGÍA

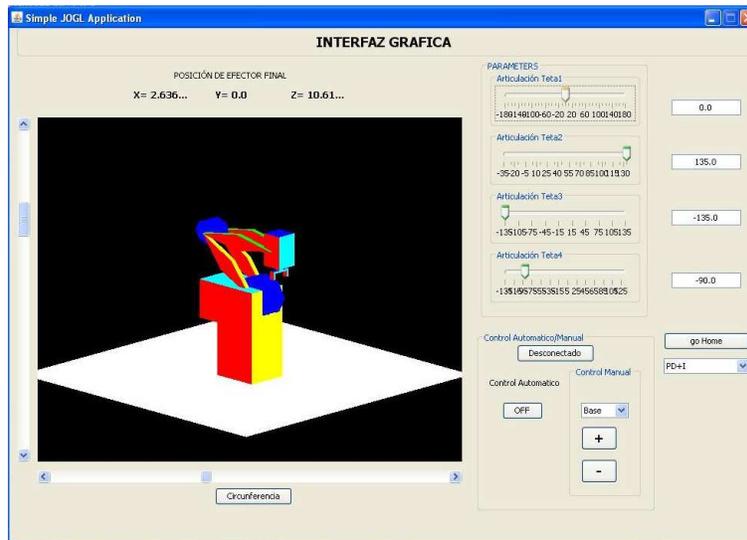


Figura 3.25: Pantalla de controlador en modo manual.

Una de las ventajas que ofrece este tipo de interfaz es que se pueden observar los distintos puntos de singularidad, es decir los puntos que el manipulador en cualquier configuración no puede alcanzar, así como visualizar los puntos que es posible alcanzar en diversas configuraciones, en la figura 3.26, se puede observar con detalle como es posible alcanzar un punto en el espacio en al menos cuatro configuraciones, conocidas como: codo arriba derecha, codo abajo derecha, codo arriba izquierda, codo abajo izquierda.

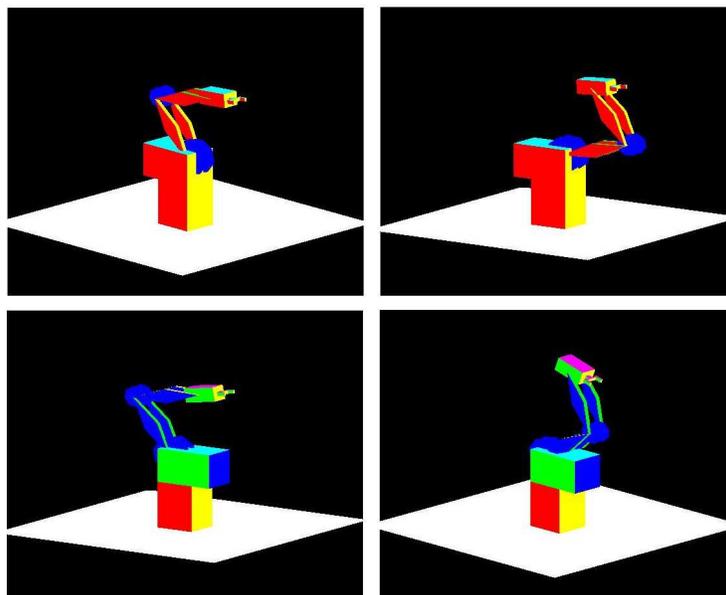


Figura 3.26: Cuatro soluciones de posición para el manipulador serial.

El sistema contiene adicionalmente un slider vertical y uno horizontal, mediante los cuales es posible realizar rotaciones sobre el mundo virtual, con el fin de observar de

3. METODOLOGÍA

diversas perspectivas los movimientos del manipulador, tal como se muestra en la figura 3.27.

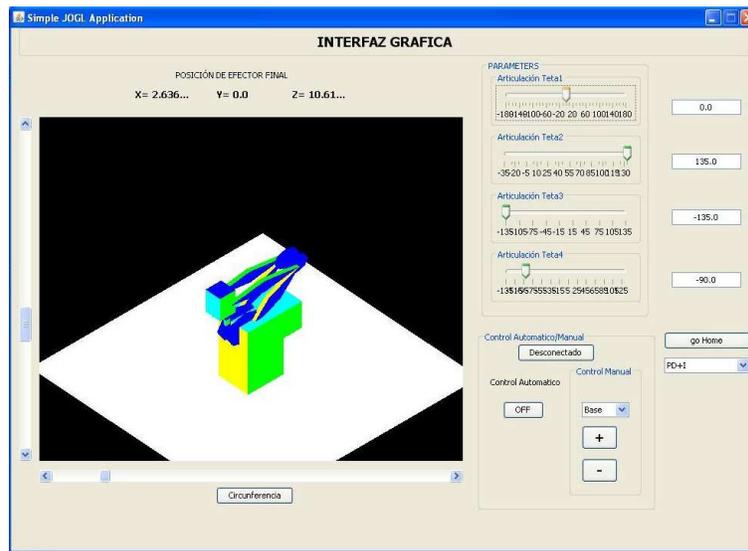


Figura 3.27: Rotación sobre mundo virtual.

Para iniciar la comunicación entre el manipulador y el sistema de control, se colocó un botón en la parte inferior de la interfaz, el cual se encuentra inicialmente etiquetado con la leyenda "Desconectado" y al ser presionado cambia su leyenda a "Conectado", lo cual da un gran nivel de abstracción. Es decir, mantiene oculto el proceso de comunicación de tal manera que el usuario solo requiera presionar el botón, y la comunicación será establecida.

Una vez iniciada la comunicación, se tiene dos modalidades de operación, las cuales son:

- Lazo abierto
- Lazo cerrado

El modo establecido por defecto, es la modalidad en lazo abierto, es decir el movimiento de cada una de las articulaciones es establecida manualmente por el usuario sin hacer uso de la retroalimentación de los codificadores ópticos.

Para realizar lo mencionado anteriormente, el sistema contiene un panel etiquetado como "Control Manual", en dicho panel se encuentran tres componentes, dos botones y un cuadro combinado.

A través del cuadro combinado podemos seleccionar la articulación que se desea mover, siendo las opciones de articulación las siguientes:

- Base
- Hombro
- Codo

- Muñeca

Una vez seleccionada la articulación, el movimiento de dicha articulación puede ser realizado en dos direcciones, para lo cual son utilizados los botones etiquetados como "+" y "-" respectivamente.

3.12. Herramientas de diseño (Sistemas VHDL)

Tal como lo menciona (Troncoso, 2007), una de las herramientas más poderosas con que se cuenta actualmente para el diseño de sistemas digitales, es el uso de lenguajes descriptivos de circuitos que permiten jerarquizar las diferentes etapas de diseño para lograr una repartición óptima del trabajo.

Con los lenguajes descriptivos, se puede diseñar y simular el funcionamiento de cada uno de los niveles de abstracción, tratando a los sistemas como cajas negras y por lo tanto pueden ser desarrollados en forma paralela, es decir, al mismo tiempo se pueden definir las características del microprocesador, su arquitectura, su estructura funcional y su realización a nivel de transistor de manera simultánea, disminuyendo los tiempos de diseño. Esto es posible gracias al uso de elementos repetitivos en cada una de las partes y sobre todo, a la existencia de lenguajes que permiten el modelado y simulación de sistemas digitales en cada uno de los diferentes niveles de abstracción. Solamente se necesita dar las especificaciones en cada nivel para que a la hora de conjuntar el diseño, se cumplan las expectativas originales. Aún sigue siendo un problema complejo, pero ya no tanto con la ayuda de estas herramientas.

A pesar de que existen diversos lenguajes descriptivos, el VHDL (Very high-speed integrated circuit Hardware Description Language) es el más popular y su uso es cada vez mayor, dejando relegados los otros lenguajes a un segundo plano. Su popularidad se debe principalmente a que es un estándar del IEEE, particularmente el estándar 1164.

En resumen, el estudio de la electrónica digital moderna incluye el conocimiento de las bases matemáticas de la lógica formal y álgebra de Boole, el manejo y conocimiento de los principales circuitos digitales que conforman el diseño y el dominio de un lenguaje descriptivo como herramienta de simulación y síntesis.

3.13. Esquema general de interfaz (FPGA - PC)

A continuación se presenta el desarrollo de la interfaz entre el FPGA y la PC, siendo los módulos descritos los siguientes:

- Comunicación serial
- Control PWM
- Módulo de PWM
- Lectura de encoder
- Driver de puente H

3. METODOLOGÍA

- Envío de posición

Al conjuntarse los módulos el producto final se puede describir a través del diagrama de la figura 3.28.

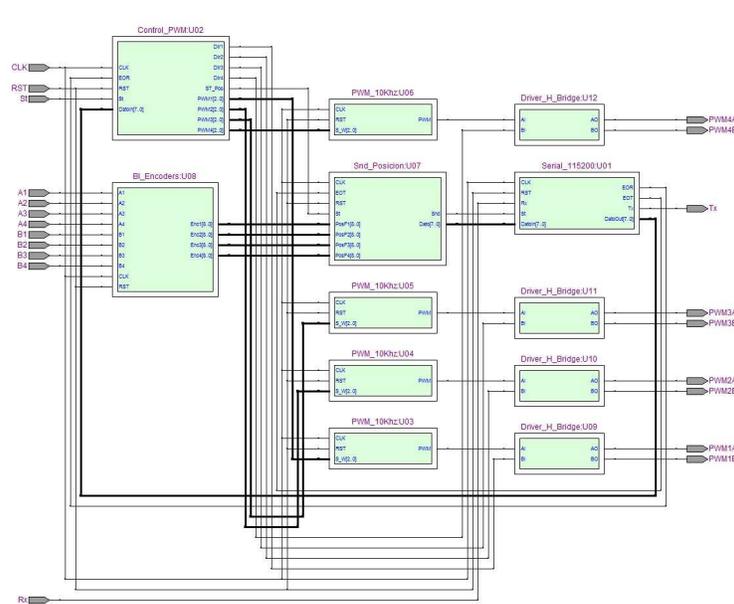


Figura 3.28: Diagrama a bloques de interfaz VHDL.

3.13.1. Lectura de encoder (PPR - rad)

Como puede observarse el bloque Bl.Encoder (Lectura de los encoder) permite realizar la lectura de cada uno de los encoders con una resolución de 1x, generando un vector de salida para cada canal de encoder en un formato 3.6 teniendo una longitud de 9 bits, que da la posición de la articulación en radianes.

El bloque completo de lectura de encoder puede apreciarse en la figura 3.29, el cual esta compuesto por los bloques Mux_Channel, L.Encoder y Multiplicador.

3. METODOLOGÍA

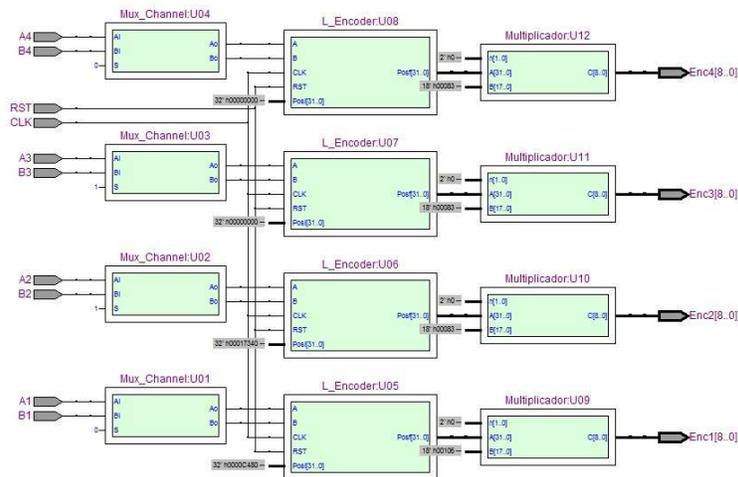


Figura 3.29: Diagrama a bloques del lector de encoder.

Siendo el bloque Mux_Channel el encargado de realizar la selección entre cambiar el canal A por B , o mantener la conexión directa. Este bloque es necesario debido a que el robot Rhino tiene ya conectado el cable de alimentación por lo que realizar el cambio físicamente se vuelve imposible.

Estas señales son enviadas al bloque L_Encoder, cuya función es indicar las cuentas del encoder y la dirección del motor, la dirección es indicada a través de una señal de 1 bit y el contador es de 32 bits, se ha elegido esta dimensión debido a que de acuerdo a investigaciones previas de los sistemas de control comerciales, manejan este tamaño de contador, con el fin de evitar que el sistema se sature y perdamos el control del robot.

Finalmente esta señal del contador es enviada al bloque Multiplicador el cual se encarga de convertir el valor del contador a radianes con un formato de 3.6, el factor de conversión fue elegido de acuerdo a la resolución de cada encoder, en la figura 3.30 se muestra la ubicación de los motores en el robot, y en la tabla 3.4 se muestra la resolución de cada encoder.

3. METODOLOGÍA

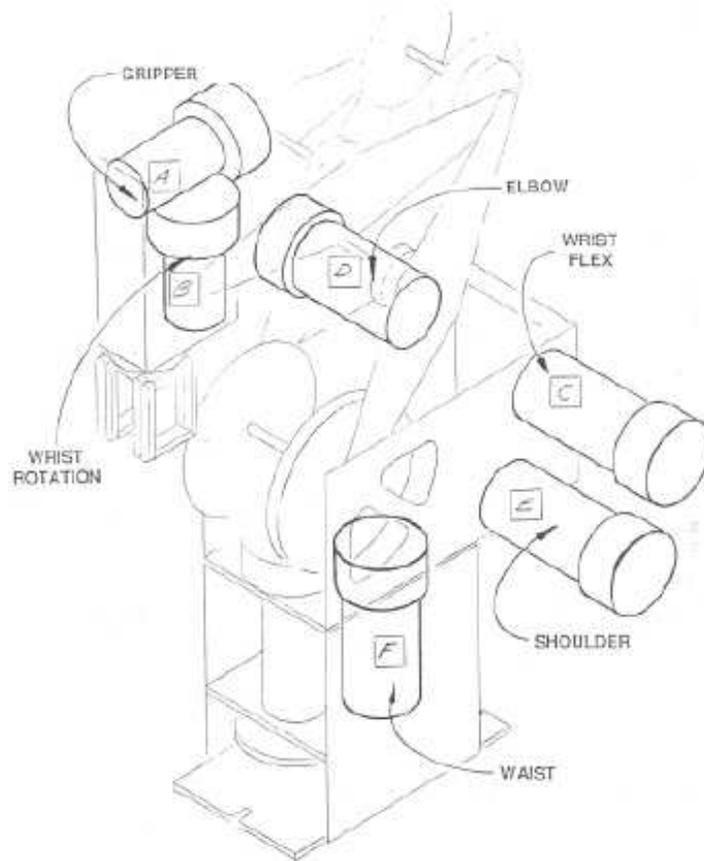


Figura 3.30: Ubicación de motores físicos.

Tabla 3.4: Resolución de encoder

Motor	Linkage	Total Reduction	Counts per Degree	Theoretical Counts per Revolution
B	Wrist Rotation	384:1	3.2	1152
C	Wrist Flex	524:1	8.7	3144
D	Elbow	524:1	8.7	3144
E	Shoulder	524:1	8.7	3144
F	Waist	524:1	4.4	1572

La intención de haber seleccionado un formato 3.6 fue debido, a que se requería un tiempo de transferencia corto, y al enviar solamente 9 bits por cada encoder, son requeridos 36 bits de datos, por lo tanto se sabe que a través del protocolo serial se envía una trama de 8 bits a la vez, siendo que:

$$\frac{36 \text{ bits}}{8 \text{ bits}} = 4.5 \text{ Tramas}$$

3. METODOLOGÍA

Por lo que al enviar solamente 5 tramas de información, es posible recuperar la información de los encoders para posteriormente ser procesada. La información acerca de la tasa de transmisión y la configuración (Paridad, n bits paro), será tratada en la sección siguiente donde se describe el módulo de transmisión UART.

Los códigos de cada uno de los bloques del módulo lectura de encoder, pueden ser consultados en el disco anexo de esta tesis en la subcarpeta *L_Encoder* de la carpeta *Control_Rhino*.

3.13.2. Bloque de selección de datos

En esta sección se describe el bloque *Snd_Posicion* el cual es encargado de tomar la información de los encoders, realizar una concatenación. Obteniendo un vector de 40 bits, para posteriormente seleccionar tramas de 8 bits a la vez, para ser enviados al bloque de comunicación serial. El diagrama a bloques del módulo *Snd_Posicion* se puede apreciar en la figura 3.31.

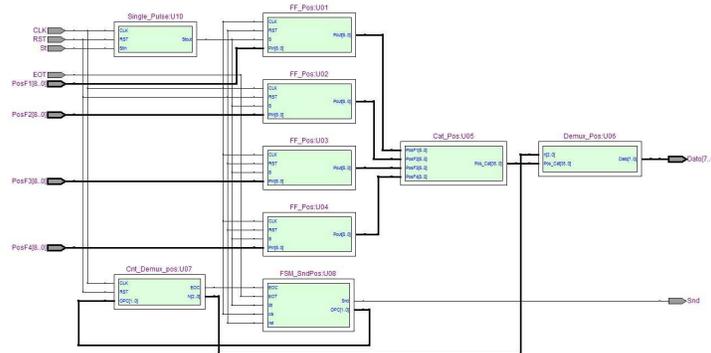


Figura 3.31: Esquema de módulo *Snd_Posicion*.

En esta imagen se pueden distinguir los siguientes módulos:

- *Single_Pulse*
- *FF_Pos*
- *Cnt_Demux_Pos*
- *FSM_SndPos*
- *Cat_Pos*
- *Demux_Pos*

El primer módulo consiste en hacer que una señal de activación de x tiempo de duración, sea convertida en un pulso de activación de un solo ciclo de reloj de duración, esto es con el fin de evitar que un proceso que sea sensible a la activación de esta señal

3. METODOLOGÍA

sea ejecutada en más de una ocasión, su máquina de estados se puede observar en la figura 3.32.

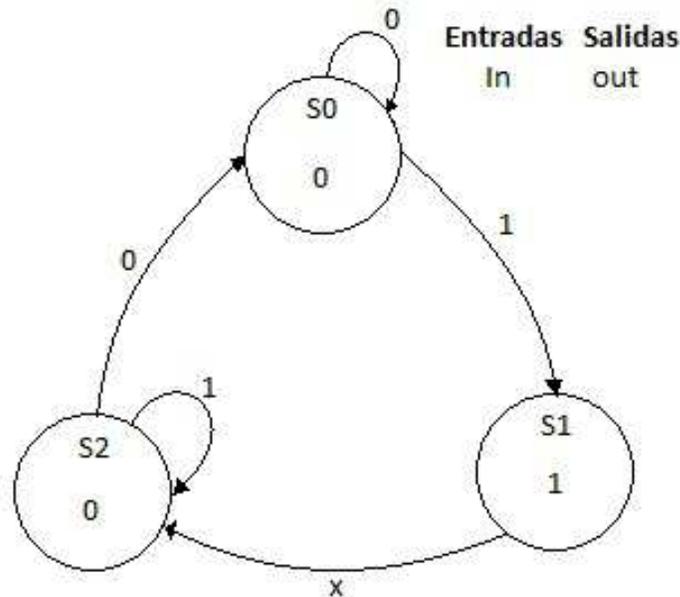


Figura 3.32: Máquina de estados de bloque single pulse.

Los módulos FF_Pos, consisten en bloques Flip-Flop tipo D, con un tamaño de bus de 9 bits, con el fin de mantener el dato proveniente del encoder hasta el momento en que se indique actualizar el dato.

Los datos provenientes de los bloques FF_Pos son enviados hacia el bloque Cat_Pos, el cual tal y como su nombre lo indica, es el encargado de realizar una concatenación de los contadores de los encoders generando un vector de 36 bits, al cual es concatenado 4 bits más de tipo *Don't Care*, a la parte baja (LSB) del vector, obteniendo finalmente un vector de 40 bits.

Este vector es transmitido hacia el módulo Demux_Pos el cual, selecciona los datos a enviar (tramas de 8 bits) a través del puerto serial. Los datos son seleccionados de acuerdo a un vector de 3 bits que funge como control del multiplexor, dicha señal proviene de un contador que a su vez es controlado por una máquina de estados, que es la encargada de sincronizar los módulos en conjunto.

Esta máquina de estados nombrada FSM_SndPos recibe como parámetros la señal de fin de transmisión generada por el módulo de comunicación serial, con la finalidad de indicar si la primera trama de información ha sido enviada, y de esta manera cargar la siguiente trama, para proceder a realizar la transmisión nuevamente, el grafo de la máquina de estados puede ser apreciado en la figura 3.33.

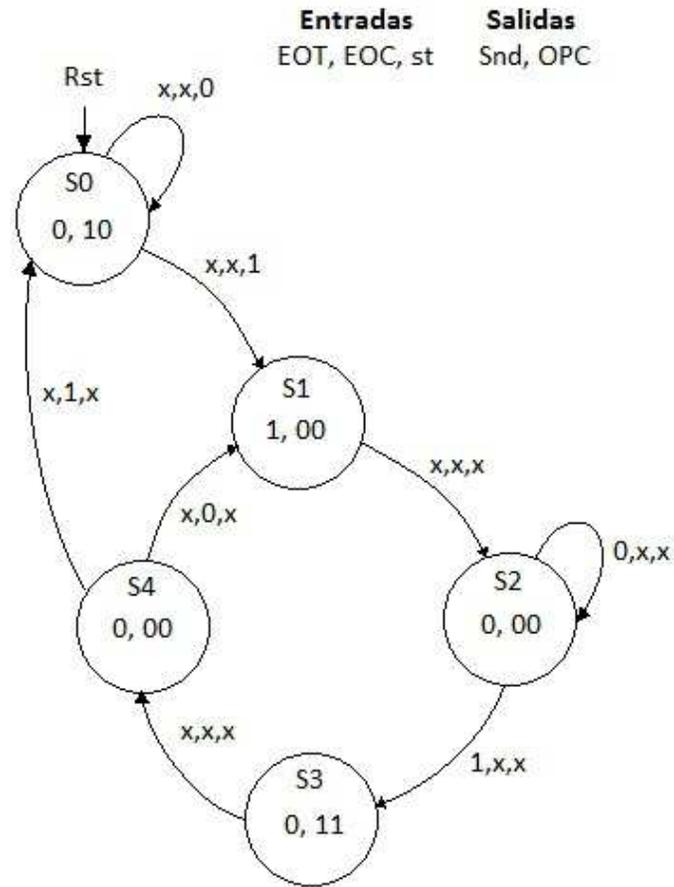


Figura 3.33: Máquina de estados de enviar posición.

De igual manera, contiene una señal de 2 bits, la cual es enviada hacia un contador, cuya función es indicar que parte del vector de 40 bits será seleccionado para ser enviado.

Los códigos de cada uno de los bloques del módulo selección de datos, pueden ser consultados en el disco anexo de esta tesis en la subcarpeta

3.13.3. Módulo de control PWM

La funcionalidad de este módulo radica en realizar la decodificación de información enviada por el controlador a través del puerto serial, para ser convertida en una información útil indicando el valor del *duty cycle* de cada PWM y el sentido de giro deseado para los motores.

La información completa de control de los 4 motores, es recibida con tan solo 2 tramas de 8 bits cada una. Esta información es recibida siguiendo el formato que se muestra en la figura 3.34.

3. METODOLOGÍA

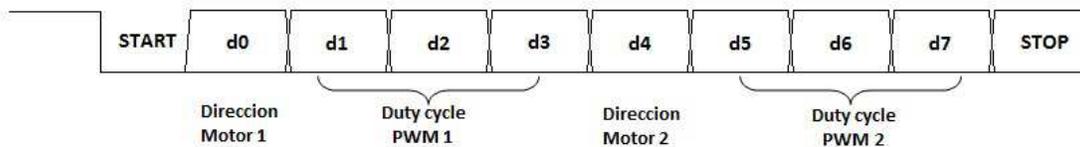


Figura 3.34: Protocolo de recepción (Señal de control).

Las señales de control para los motores 3 y 4 siguen exactamente el mismo protocolo. El esquema a bloques general del módulo de control PWM se muestra en la figura 3.35.

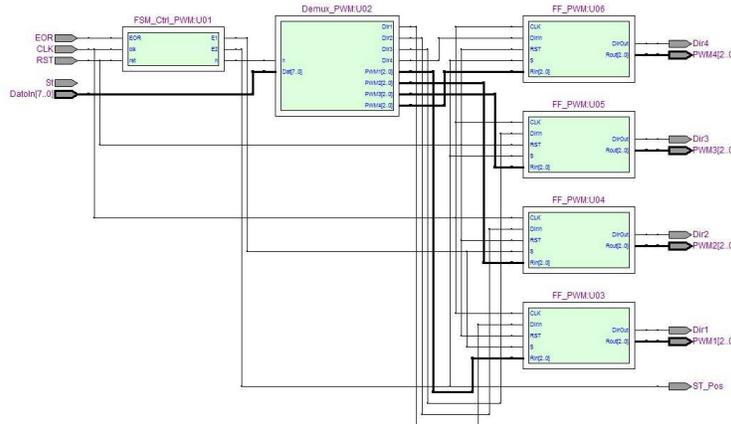


Figura 3.35: módulo Control_PWM .

Dicho módulo se encuentra por tres bloques principales los cuales se enlistan a continuación.

- FSM_Ctrl_PWM
- Demux_PWM
- FF_PWM

El bloque Demux_PWM se encarga de convertir los datos recolectados a través de la comunicación serial. Por lo tanto las tramas son separadas, obteniendo las señales de control y dirección para cada uno de los motores.

Estas señales de control son enviadas hacia un bloque de Flip-Flop tipo D cada una, con el fin de mantener la información mientras es actualizada.

La señal de habilitación de los Flip-Flop(FF), está dada por una máquina de estados, la cual es encargada de indicar a que numero de motor corresponde la información, de tal manera que los FF son activados en pares. Es decir, al momento de finalizar la recepción de la primera trama, la información es enviada hacia los FF y se realiza la habilitación de los FF correspondientes a los motores 1 y 2, para posteriormente recibir

3. METODOLOGÍA

la segunda trama de información y realizar la habilitación de los FF correspondientes a los motores 3 y 4.

Finalmente la máquina de estados, como ya se menciono, es la encargada de sincronizar cada uno de los bloques con el resto de los módulos.

El grafo de la máquina de estados puede apreciarse en la figura 3.36.

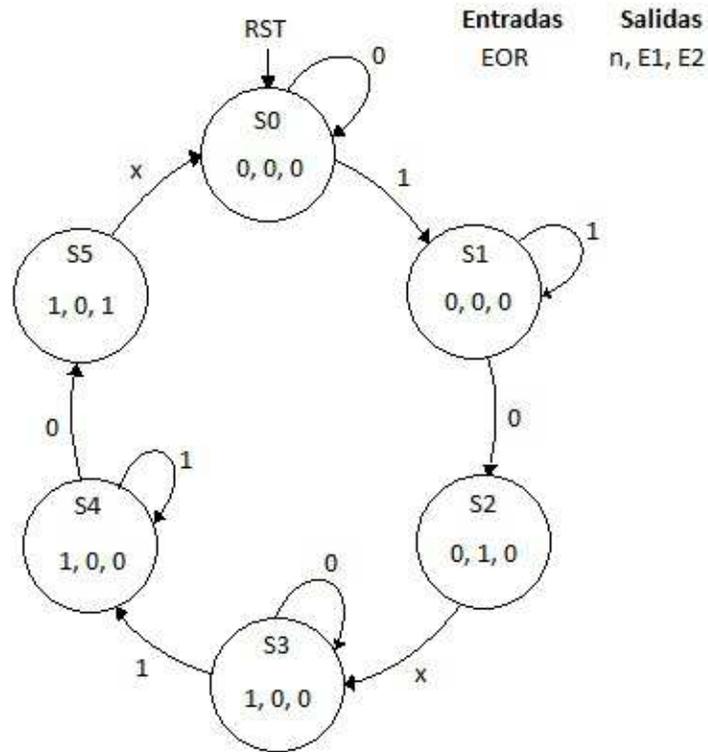


Figura 3.36: Máquina de estados de control PWM .

Los códigos de cada uno de los bloques del módulo control de PWM, pueden ser consultados en el disco anexo de esta tesis en la subcarpeta *Control_PWM* de la carpeta *Control_Rhino*.

3.13.4. PWM a 10 KHz

En esta sección se detallará el desarrollo del módulo generador de la señal PWM, este módulo permite variar el duty cycle de la señal en línea. Es decir, el usuario puede modificar su periodo en el momento que le sea requerido.

La frecuencia de 10 KHz ha sido seleccionada, con el fin de ser poco audible para el oído humano, y ser una frecuencia muy utilizado en el campo de los servomotores.

Para el desarrollo de este módulo son necesarios diversos componentes como lo fueron:

- Base de tiempo

3. METODOLOGÍA

- Mux de control
- Contador
- Máquina de estados

La interconexión de estos bloques se presenta en la figura 3.37.

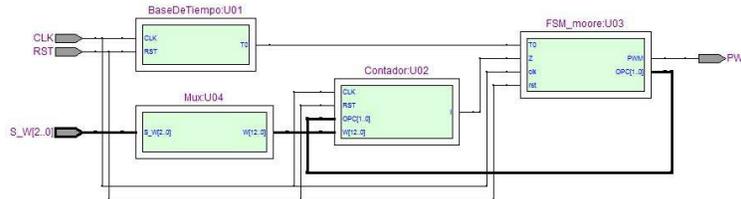


Figura 3.37: Generador de señales PWM .

Primeramente se creo una base de tiempo, que es la encargada de dar la señal de inicio y fin de periodo, para lo cual fue requerido realizar los cálculos necesarios para obtener una base de 10 ms.

Recordando que la tarjeta de desarrollo utilizada, fue una *Altera DE2*.

Esta tarjeta cuenta con un oscilador de 50 Mhz, es decir cada tick de reloj es producido cada 20 ns.

Por lo tanto el esquema seguido para el desarrollo de la base de tiempo consiste, en un contador libre, el cual tiene un valor inicial de 0 e incrementa una unidad con cada tick de reloj, hasta llegar a un valor establecido y reinicia nuevamente en 0.

Este valor establecido, define el periodo de la base de tiempo. La forma de obtención de este valor es la siguiente:

$$F_{osc} = 50Mhz$$

$$F_{des} = 10Khz$$

Recordando que $T = \frac{1}{f}$
donde:

- T = Periodo
- f = Frecuencia

por lo tanto:

$$T_{osc} = 20 \text{ ns}$$

$$T_{des} = 0.1 \text{ ms}$$

Finalmente el problema se reduce a, cuantos ticks de reloj son necesarios para obtener un periodo de 10 ms.

3. METODOLOGÍA

$$\frac{0.1 \times 10^{-3}}{20 \times 10^{-9}} = 5000 \text{ ticks}$$

A este número de ticks, es necesario restar una unidad debido a que el contador cuenta hasta cero, por lo tanto el número final que debemos establecer al contador es el.

$$4999_{10} \implies 1001110000111_2$$

Debido a que la tarjeta, realiza las operaciones en forma binaria, el coeficiente debe ser establecido en formato binario. Por lo tanto, todos los coeficientes y valores que se ingresen y generen estarán en formato binario (base 2).

Por otro lado es necesario, generar otro contador descendente con características similares a la base de tiempo, el cual generará una señal al terminar de contar, indicando el final del duty cycle de la señal PWM.

La diferencia entre el contador y la base de tiempo radica en que el contador es programable; es decir, se especifica el valor inicial del contador, mientras que la base de tiempo tiene un valor fijo.

Este valor inicial, es proporcionado por un bloque multiplexor.

Al multiplexor le llega la señal de selección proveniente de el bloque Control_PWM, la cual es una señal de 3 bits.

El tamaño del arreglo fue seleccionado puesto que con 3 bits, es posible obtener 8 salidas diferentes que se verán reflejadas en la duración del ancho de pulso.

Es decir se han elegido 8 diferentes valores de duty cycle fijos, siendo los que se muestran en la tabla 3.5.

Tabla 3.5: Porcentaje de Duty cycle para PWM

Selector	Duty cycle
000	0 %
001	15 %
010	30 %
011	45 %
100	60 %
101	75 %
110	90 %
111	100 %

Estos valores fueron elegidos de acuerdo a resultados experimentales, los cuales mostraron que no había diferencia en el tiempo de establecimiento, si se utilizaban mas valores de duty cycle. Por otro lado, permite tener un tiempo de recepción de las señales de control corto; es decir, por cada trama (8 bits) se pueden recibir las señales de control para dos diferentes motores.

1 bit de dirección (0) y 3 bits de control (1 ... 3) Motor 1
1 bit de dirección (4) y 3 bits de control (5 ... 7) Motor 2

3. METODOLOGÍA

Los valores establecidos para el contador, fueron obtenidos de la misma manera como se cálculo el valor de la base de tiempo.

ejemplo:

Si se requiere un duty cycle del 10 %, y sabiendo que la frecuencia de la señal es del 10 Khz con un periodo de 0.1 ms.

Es necesario encontrar el tiempo requerido para ese ancho de pulso, es decir.

$$0.1 \times 10^{-3} \times 0.1 = 10 \times 10^{-6}$$

Una vez obtenido este valor, se calcula el número de ticks de reloj necesario para alcanzar dicho valor con la frecuencia del oscilador, que en nuestro caso es de 50 Mhz.

$$\frac{10 \times 10^{-6}}{20 \times 10^{-9}} - 1 = 499$$

Una vez obtenido el valor, se debe de convertir a binario, debido a los requisitos del FPGA.

$$499_{10} \implies 111110100_2$$

Los coeficientes necesarios para cada porcentaje de ancho de pulso, han sido ordenados en la tabla.

Tabla 3.6: Coeficientes de ancho de pulso correspondiente a porcentaje.

Porcentaje	Valor Base 10	Valor Base 2
0 %	0.00	00000000000000
15 %	749.00	0001011101101
30 %	1499.00	0010111011011
45 %	2249.00	0100011001001
60 %	2999.00	0101110110111
75 %	3749.00	0111010100101
90 %	4499.00	1000110010011
100 %	4999.00	1001110000111

Finalmente y para completar el módulo, es requerido el diseño de una máquina de estados, la cual es la encargada de sincronizar cada uno de los bloques y generar la señal de salida PWM.

El grafo de la máquina de estados puede observarse en la figura 3.38 y el diagrama de tiempos del módulo PWM es observado en la figura 3.39.

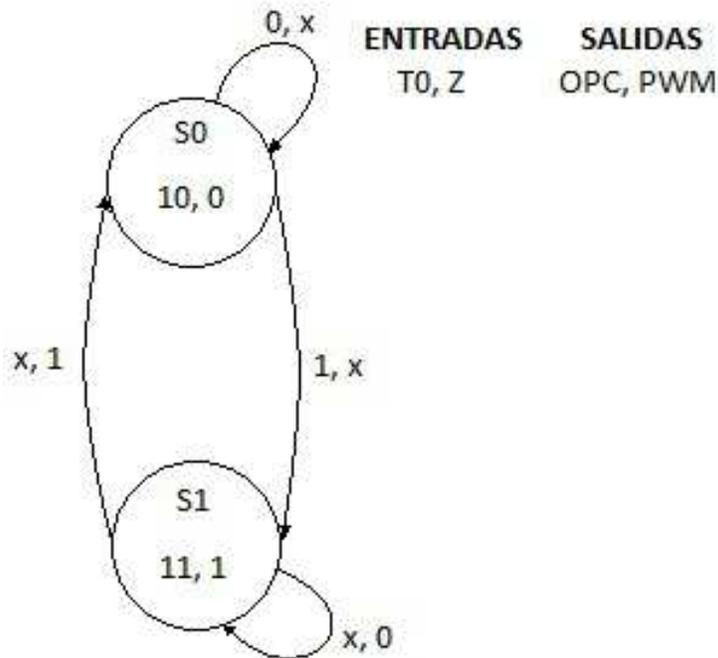


Figura 3.38: Máquina de estados de generador de PWM .

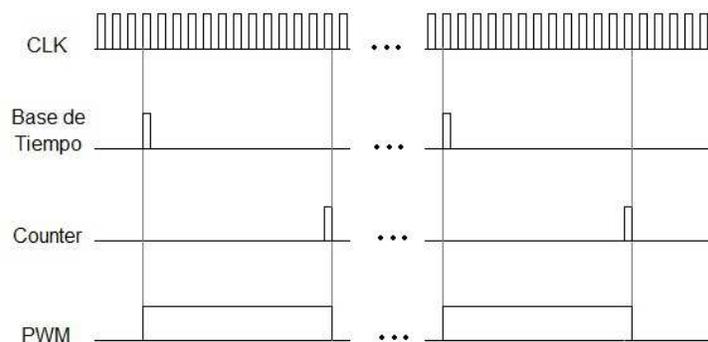


Figura 3.39: Diagrama de tiempos de PWM.

El código correspondiente al módulo PWM a 10 Khz, puede ser consultado en el disco anexo de esta tesis en la subcarpeta *PWM_10Khz* de la carpeta *Control_Rhino*.

3.13.5. Driver de puente H

El diseño de este módulo esta basado en la necesidad de controlar a la electrónica de la etapa de potencia (Puente H), y a su vez protegerla de corto circuitos, que pueden ocurrir al activar a la vez ambos lados del puente H.

Para comenzar con su desarrollo, se observa la tabla de verdad que se obtenía de la etapa de potencia, siendo la siguiente tabla 3.7.

3. METODOLOGÍA

Tabla 3.7: Tabla de verdad para Driver de puente H

PWM	Dir	A	B
0	0	1	1
0	1	1	1
1	0	0	1
1	1	1	0

Como puede apreciarse, la lógica requerida es inversa, debido al aislamiento que se colocó entre la electrónica y la tarjeta FPGA. Ergo cuando la señal PWM se encuentre en alto, la señal de activación sera su complementaria.

Para comprender mejor lo descrito anteriormente, imaginemos el siguiente escenario, se desea realizar un giro de motor en el sentido de las manecilla de reloj, con un duty cycle del 10 %.

Para lograr esto se activaría la señal *B* dejándola fija, y a la señal *A* sería aplicado el complemento de la señal PWM con un duty cycle del 10 %.

Por lo tanto, aplicando algebra booleana, las ecuaciones que describen a la tabla de verdad 3.7, es la siguiente:

$$\begin{aligned}A_o &= \overline{PWM} \cap Dir \\B_o &= \overline{PWM} \cap \overline{Dir}\end{aligned}$$

Aplicando el teorema de DeMorgan, el cual se describe como:

$$\text{Teorema de DeMorgan: } \begin{cases} \overline{A \cup B} = \overline{A} \cap \overline{B} \\ \overline{A \cap B} = \overline{A} \cup \overline{B} \end{cases}$$

se pueden convierten las ecuaciones, en las que se muestran en la figura 3.40.

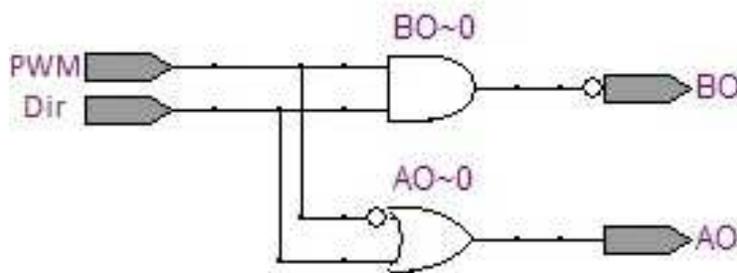


Figura 3.40: Álgebra de boole para Driver.

El código correspondiente al módulo del driver de puente H, puede ser consultado en el disco anexo de esta tesis en la subcarpeta *Control_PWM* de la carpeta *Control_Rhino*.

El diagrama y circuitos de la etapa de potencia puede observarse en el capítulo 3.14.

3.13.6. Comunicación serial

Continuando con el desarrollo del sistema de control y monitoreo, fue necesario implementar el protocolo de comunicación serial (RS-232) entre la tarjeta FPGA y la computadora.

Este protocolo fue seleccionado, debido a que actualmente en la industria, sigue siendo uno de los protocolos de comunicación más utilizados, gracias a su distancia máxima de transmisión, velocidad de transmisión, inmunidad al ruido, etc.

Aunque la velocidad de transmisión máxima es de 115.2 Kbits, es muy inferior a las velocidades alcanzadas en la comunicación USB, tiene la gran ventaja de contar con una distancia máxima de transmisión de 15 m, mientras que el USB tiene una distancia máxima de 5 m.

Cabe mencionar que lo anterior es considerando una conexión estándar tal como lo menciona cada una de sus normas, es decir, sin hacer uso de hubs o repetidores.

Por estas razones se decidió utilizar una comunicación serial full-duplex, a una tasa de transmisión de 115200 baudios, que como se menciono anteriormente es la velocidad máxima.

El diagrama de tiempos de la transmisión serial se puede observar en la figura 3.41, donde se muestra el conjunto de bits completo a enviar, donde cada bit puede ser descrito como:

- **Bit de inicio.** (Start)
- **Bits de datos.** (7/8)
- **Bit de paridad.** Este bit se utiliza para comprobar si los bits de datos han sido recibidos correctamente.
Existen tres posibilidades.
 - **Paridad Par.** Si la suma de los bits de datos es par, el bit de paridad es 1, si es impar, el bit de paridad es 0.
 - **Paridad Impar.** Si la suma de los bits de datos es impar, el bit de paridad es 1, si es par, el bit de paridad es 0.
 - **Sin paridad.** No se utiliza el bit de paridad.
- **Bit de paro.** (Stop) Pueden ser uno o dos bits.

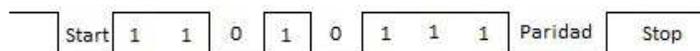


Figura 3.41: Diagrama de tiempos para comunicación serial.

Tal como se describió anteriormente, es necesario realizar el envío de al menos 10 bits por cada byte o trama (8 bits).

Por lo tanto, se decidió adoptar la siguiente configuración, con una tasa de transferencia de 115200 baudios:

3. METODOLOGÍA

- **Bit de inicio**
- **8 bits de datos**
- **Sin paridad**
- **Bit de paro**

Se ha elegido no realizar la verificación de paridad con el fin de disminuir el tiempo de transmisión.

Por lo tanto, el tiempo que tarda en realizar la transmisión de un byte de datos, puede ser calculada fácilmente multiplicando el periodo de la velocidad de transmisión por el número de bits enviados.

$$n_{bits} = 10.$$

$$T_{trans} = \frac{1}{115200baudios} = 8.68 \times 10^{-6}.$$

$$Tiempo = n_{bits} \times T_{trans} = 8.68 \text{ uS}.$$

Una vez obtenido el tiempo de transmisión de cada trama. Se realizó el cálculo del tiempo de transmisión total de la posición de los 4 encoders.

Recordando de la sección 3.13.1, sabemos que por cada articulación son enviados 9 bits de información, por lo tanto los bits requeridos para el vio de la información de los motores son 36, siendo requeridas 5 tramas para poder realizar el envío de la información.

Por lo tanto, anteriormente se cálculo que el tiempo de transmisión de cada trama es de 86.8 uS. Es decir, para conocer el tiempo de transmisión de la información de los 4 motores basta con realizar la multiplicación de el tiempo de transmisión por el número de tramas a enviar.

$$Tiempo_{Total} = Tiempo_{Trans.Trama} \times n_{tramas} = 0.43 \times 10^{-3} \text{ S}.$$

Como puede apreciarse, el tiempo que tarda el dispositivo en enviar la información de las posiciones de las articulaciones hacia la computadora, es inferior a medio milisegundo, siendo esto una gran ventaja a la hora de realizar el controlador, permitiendo contar con periodos de muestro demasiado cortos.

Con lo anterior hemos definido las características que nos interesa que tenga la comunicación serial, pero aun no hemos desarrollado las interfaces; por lo tanto, se comenzará dividiendo el problema en dos para mayor comodidad.

- **Transmisor serial.**
- **Receptor serial.**

3. METODOLOGÍA

Comenzaremos explorando el desarrollo del *Transmisor serial* para posteriormente introducirnos al desarrollo del receptor serial, el cual por ser asíncrono conlleva un poco mas de complejidad.

Dicha division puede ser observada a modo de diagrama de bloques a través de la figura 3.42, la cual muestra las señales de entrada y salida de dichos módulos.

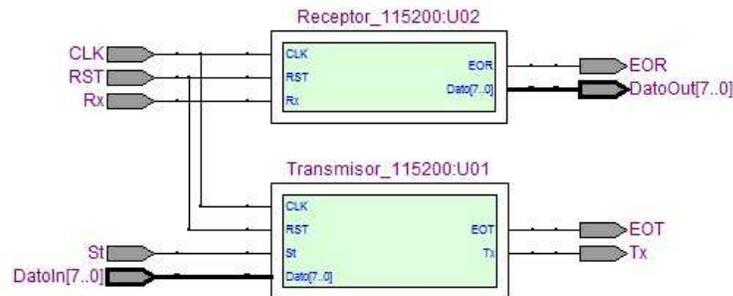


Figura 3.42: Esquema general de módulo de comunicación Serial.

Transmisor Serial

Como primer paso se diseñaron los bloques necesarios para que el módulo de transmisión realizara su cometido.

Los módulos necesarios son:

- Single Pulse.
- Base de tiempo.
- Contador.
- Demux.
- Máquina de estados.

El bloque llamado *Single Pulse*, se encarga de convertir una señal de activación de duración indefinida en una señal de activación de un solo tick de reloj. Dicho módulo fue descrito anteriormente en la subsección 3.13.2 y su máquina de estados puede observarse en la figura 3.32.

Uno de los módulos más importantes, debido a que, si la señal que genera la base de tiempo no es constante o se encuentra a una frecuencia diferente a la deseada, provocará que todo el módulo falle. Por lo tanto, se debe de tener un cuidado especial en el diseño de las bases de tiempo.

Recordando que la base de tiempo funciona como un contador de carrera libre, la cual genera una señal al llegar al límite máximo de su contador (ascendente) o al llegar al valor mínimo de su contador (descendente).

3. METODOLOGÍA

Debido a esto el principal problema se reduce a encontrar el valor al que deberá de llegar el contador, el cual nos dará la señal de base de tiempo con el periodo deseado.

Siendo la frecuencia deseada 115200 baudios, obtenemos su periodo T como:

$$T = \frac{1}{f}$$
$$T = \frac{1}{115200} = 8.68 \times 10^{-6}$$

Por lo tanto, recordando que la tarjeta FPGA, funciona con un oscilador de 50 Mhz.

Se puede observar que este oscilador genera una señal de reloj cada 20×10^{-9} S. El valor máximo del contador se puede calcular como el coeficiente entre el periodo deseado y el periodo del oscilador.

$$Valor_{maximo} = \frac{8.68 \times 10^{-6}}{20 \times 10^{-9} - 1} = 433.02$$

Aunque el valor obtenido es fraccionario, se debe de tomar el valor entero superior, por lo que, el valor del contador será 434 ticks.

Dicho valor decimal, convertido a valor binario es escrito como:

$$434_{10} \longrightarrow 110110010_2$$

El contador genera una señal de salida a la que hemos llamado To , dicha señal es producida en el momento en el que el contador alcanza su valor máximo, dicho señal tiene una duración de un ciclo de reloj, es decir 20×10^{-9} S.

Continuando con el desarrollo de los módulos, fue requerido el diseño de un bloque demultiplexor, cuya función es la de recibir el byte de información que se desea enviar y seleccionar, uno a uno los bits del vector a enviar.

La posición del bit a enviar es seleccionado, mediante una señal de control de 4 bits, la citada señal de control, es proveniente de un módulo contador, el cual tiene un valor límite de 10 unidades.

Finalmente el módulo del transmisor serial, no podría estar completo sin el desarrollo de un bloque de sincronización.

A este bloque se ha denominado como FSM_moore, la cual es una máquina de estados que se encarga de realizar la concordancia entre cada uno de los bloques del módulo Transmisor serial.

El grafo de la máquina de estados se puede apreciar en la figura 3.43, y en la figura 3.44 se muestra el diagrama de tiempo del módulo de transmisión.

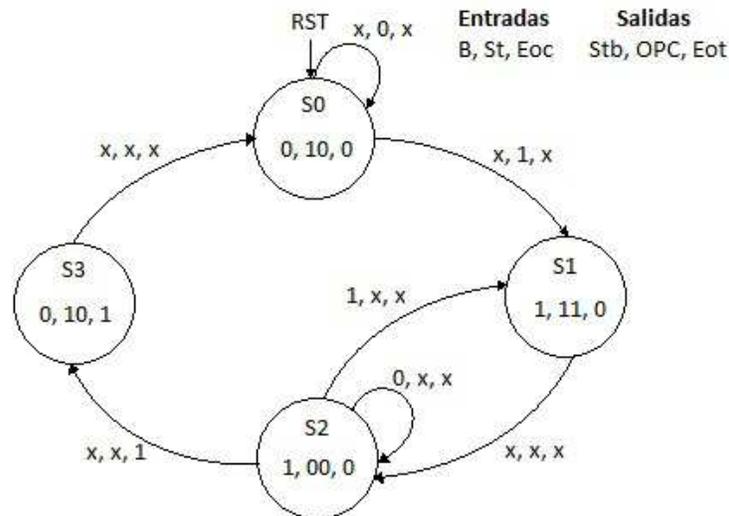


Figura 3.43: Máquina de estados de módulo de transmisión.

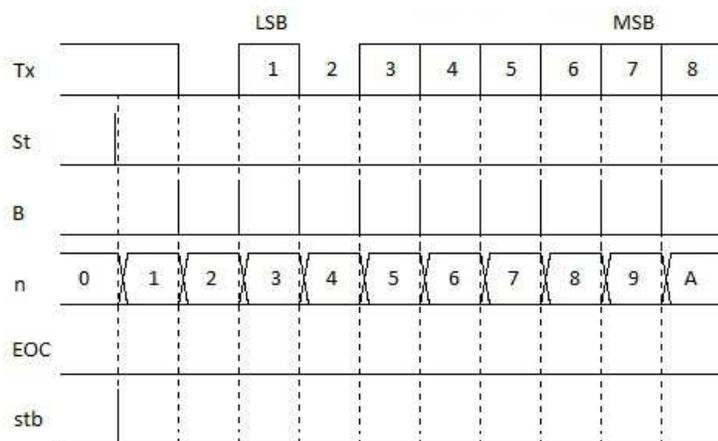


Figura 3.44: Diagrama de tiempos de módulo de transmisión.

Los códigos desarrollados para el módulo de transmisión serial, pueden ser consultados en el disco anexo de esta tesis en la subcarpeta *Transmisor_115200* de la carpeta *Control_Rhino*.

Receptor Serial

El desarrollo de este módulo, conlleva la contemplación de diversos factores que intervienen en el proceso como son:

- Sistema asíncrono.
- Velocidad de transmisión.

3. METODOLOGÍA

- **Configuración de trama.**

- n° bits de datos.
- Paridad.
- n° bits de paro.

cada uno de estos factores, deben de ser contemplados en el momento de realizar la máquina de estados, que se encargará de realizar la sincronización.

Una vez que se ha definido la configuración del puerto serial se realizó el diseño del diagrama de tiempos, mediante el cual se definieron los bloques necesarios para el funcionamiento del receptor serial, el diagrama de tiempos requerido se presenta en la figura 3.45.

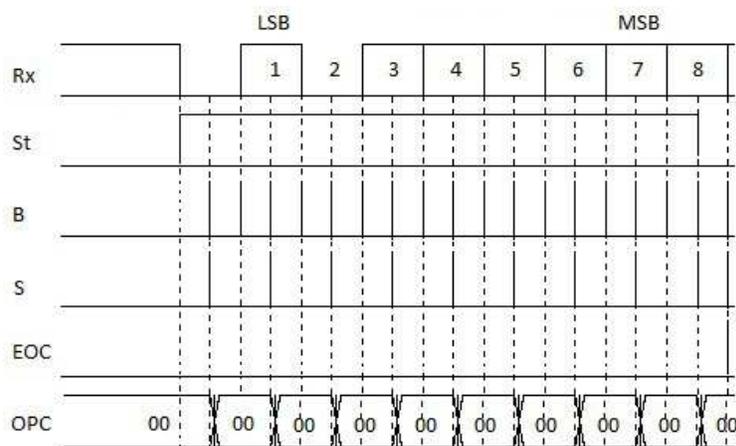


Figura 3.45: Diagrama de tiempos de módulo de recepción.

A partir de este diagrama, se definieron los bloques requeridos para generar el módulo de recepción del módulo de comunicación serial.

Como puede apreciarse en el diagrama de tiempos, son requeridos diversos bloques que ya han sido descritos con anterioridad, como son: bases de tiempo, contadores, Flip-Flop y máquinas de estados, pero este módulo requiere de un bloque nuevo que hasta el momento no se había necesitado, conocido como *Registro de Desplazamiento (Shift Register)*.

Los bloques necesarios para el funcionamiento del receptor se listan a continuación.

- Base de tiempo
- FSM (Máquina de estados)
- Contador
- Flip- Flop
- Registro de desplazamiento.

3. METODOLOGÍA

El diagrama a bloques del módulo de recepción se puede apreciar en la figura 3.46

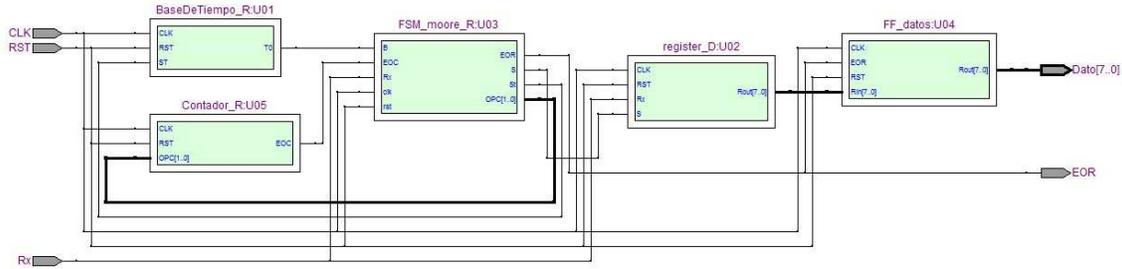


Figura 3.46: Diagrama a bloques para módulo de recepción.

Para este módulo fue necesario desarrollar una base de tiempo tal como se describió anteriormente. Con la diferencia, de ser del doble de la frecuencia de la velocidad de transmisión.

Se ha escogido que la base de tiempo sea del doble de la velocidad de transferencia, con el fin de obtener el valor de entrada a través de la señal Rx , exactamente a la mitad del periodo de transmisión. Esto puede ser apreciado más claramente en la figura 3.45, la cual muestra que el valor de la señal de entrada es obtenida en el momento que la señal S se activa.

Por lo tanto, al realizar los cálculos pertinentes se obtiene que el periodo de la base de tiempo requerida es de la mitad del periodo de la base de tiempo de 115200 baudios.

$$T_{115200} = 8.68 \times 10^{-6} S$$

$$T_{Base_{receptor}} = \frac{T_{115200}}{2} = 4.34 \times 10^{-6} S$$

Por lo tanto el valor máximo del contador de la base de tiempo, es calculado como:

$$Valor_{max} = \frac{T_{Base_{receptor}}}{T_{oscilador}} - 1 =$$

$$\frac{4.34 \times 10^{-6}}{20 \times 10^{-9}} - 1 = 216.014 \text{ unidades}$$

Debido a que el valor obtenido es fraccionario se requiere redondear el valor a la unidad entera superior, es decir $Valor_{max} = 217$. Recordando que la tarjeta funciona en formato binario, es necesario realizar la conversión del $Valor_{max}$ a binario.

$$217_{10} \longrightarrow 11011001_2$$

Para realizar el control de los datos, se emplea un contador cuyo valor máximo es 10. El valor máximo ha sido elegido debido al número de bits que son recibidos, por lo tanto el contador incrementa su valor en el momento que un bit de información es recibido y continua su incremento hasta llegar su cuenta a 10. Momento en que la transferencia de información es completada y se genera una señal de control indicando el fin de la recepción.

3. METODOLOGÍA

Por otra parte el módulo contiene un bloque de tipo *Registro de desplazamiento*, cuya tarea es la de recibir cada uno de los bits de información recibidos y generar un vector de longitud de un byte.

Para ello debemos de recordar que el protocolo del puerto serial describe que el primer bit recibido corresponde al bit menos significativo del vector de información. Es por esta razón que es imprescindible hacer uso de un registro de desplazamiento.

El funcionamiento del registro de desplazamiento, consiste principalmente en dos señales de un bit (Rx y S) y un vector de salida de un byte de longitud (Rout).

La señal Rx, se encuentra conectada al pin de recepción del puerto serial por lo tanto el registro tiene a la entrada en todo momento el dato de información, pero solamente realiza su adquisición en el momento en el que la señal shift (S) se activa. En ese momento el bit de información pasa al registro el cual realiza un recorrido de una unidad a la derecha. El diagrama de la figura 3.47 muestra en forma más clara el proceso del registro de desplazamiento.

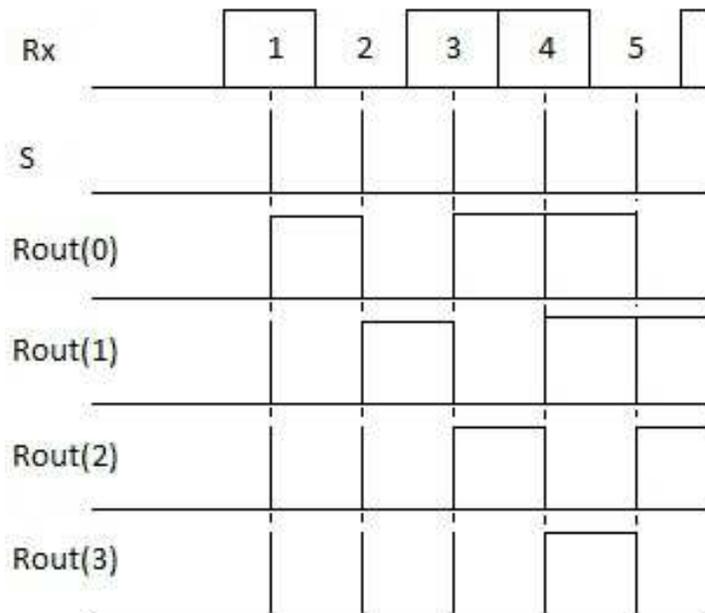


Figura 3.47: Diagrama de tiempo para registro de desplazamiento.

Al recibir el byte de información, la máquina de estados encargada de realizar la sincronización de los bloques genera una señal a la se ha denominado *EOR* por sus siglas en inglés *End Of Reception*; la cual habilita al Flip-Flop de un byte de longitud, el cual se encarga de mantener la información estable hasta el momento en que se realice nuevamente una recepción de información.

Finalmente el módulo no podría estar listo, sin desarrollar su máquina de estados.

El grafo de la máquina de estados encargada de realizar la sincronización de cada uno de los bloques, puede ser representado como el que muestra la figura 3.48.

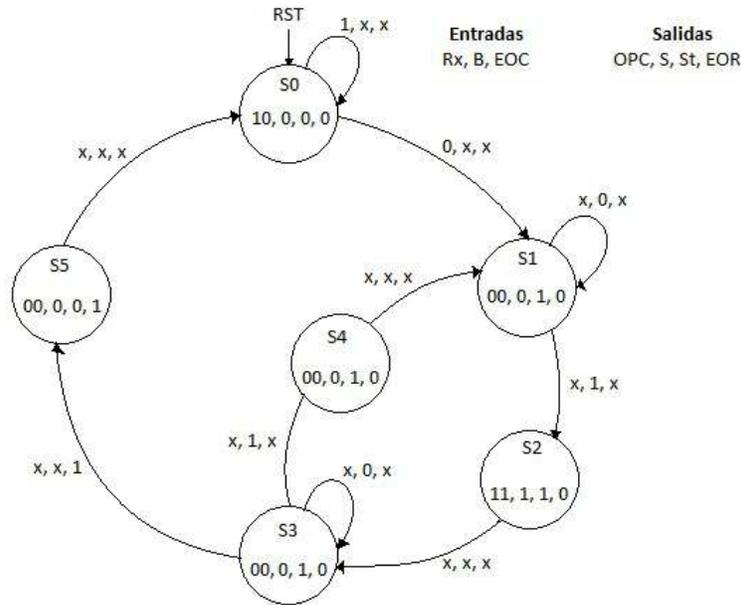


Figura 3.48: Máquina de estados de módulo receptor.

El código correspondiente al módulo de recepción serial, pueden ser consultados en el disco anexo de esta tesis en la subcarpeta *Receptor_115200* de la carpeta *Control_Rhino*.

3.14. Desarrollo de hardware

En secciones anteriores se presentó la descripción de circuitos integrados bajo el lenguaje VHDL y la implementación de controladores utilizando el lenguaje de programación Java, sin embargo, el sistema no puede estar terminado sin el desarrollo de una etapa de potencia y tarjeta de interconexión. La etapa de potencia es la encargada de proporcionar al sistema, el voltaje de alimentación equivalente a la señal de control requerida para realizar un movimiento sobre la articulación y la tarjeta de interconexión se encarga de realizar la interfaz y acondicionamiento de la señal de los codificadores ópticos y el FPGA.

En las siguientes secciones se presenta el desarrollo de la etapa de potencia y la tarjeta de interconexión.

3.14.1. Etapa de potencia

Para poder controlar la velocidad de los motores en ambos sentidos de giro y poder frenar la marcha, la topología a utilizar es el puente H completo (Full-Bridge).

Es posible utilizar distintas técnicas para controlar la potencia aplicada al motor, de manera de obtener el comportamiento deseado. Una técnica que presenta alto rendimiento es la técnica de modulación de ancho de pulso (PWM).

Esto es debido, a la arquitectura del driver la cual se muestra en la figura 3.49.

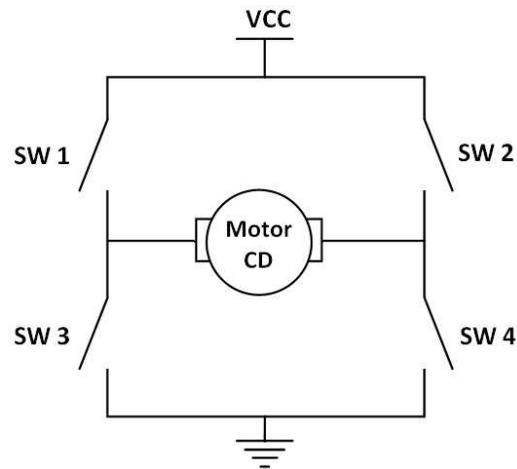


Figura 3.49: Esquema general de Driver.

Como puede apreciarse, el driver está constituido por cuatro interruptores denotados como $sw1, sw2, sw3$ y $sw4$, los cuales en realidad son sustituidos por transistores o mosfet. Independientemente de la familia que sea elegida los interruptores operan en corte y saturación, minimizando de esta manera las pérdidas de energía.

La configuración cuádruple presenta la flexibilidad de activar a los motores en marcha o reversa de acuerdo a la necesidad que se tenga, dicha acción es lograda al accionar a los interruptores en forma ordenada, es decir, tomemos como ejemplo el caso en el que los interruptores $sw1$ y $sw4$ se encuentran activados.

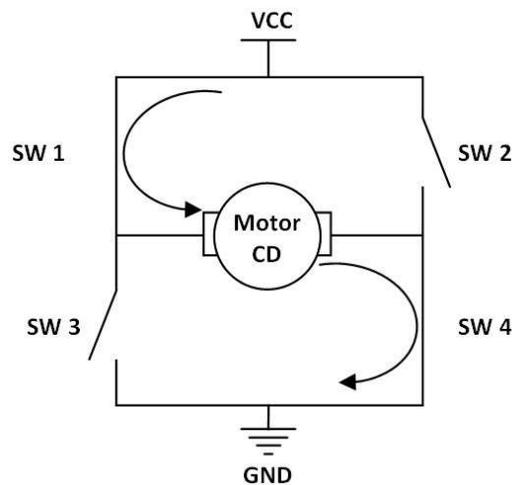


Figura 3.50: Puesta en marcha de motor.

Como se puede observar en la figura 3.50, en el momento en que el interruptor $sw1$ y $sw4$ son activados, forman un camino que permite que la corriente (i) fluya desde la fuente de alimentación pase a través del motor mediante el borne izquierdo y siga su camino hacia la masa (tierra), originando de esta manera la polarización del motor y provocando su giro.

3. METODOLOGÍA

Para ejecutar un giro en sentido contrario, como es de esperarse, deben de ser desactivados los interruptores *sw1* y *sw4* con el fin de evitar poner el driver en corto circuito y posteriormente activar los interruptores *sw2* y *sw3* que permitirán que la corriente fluya en sentido contrario, generando de esta manera una inversión de polaridad entre las terminales del motor.

Hasta este momento hemos definido el funcionamiento básico del driver de potencia, pero, ¿cómo podemos controlar su velocidad?, La respuesta es simple, haciendo que el par de interruptores necesarios para generar un movimiento en el sentido deseado, se abran y cierren a una determinada frecuencia. Debido a esto, en esta tesis se decidió hacer uso de la técnica basada en modulación de ancho de pulso.

El conjunto del circuito generador de PWM y el driver, son encargados de transformar pulsos lógicos (baja potencia) en pulsos de corriente (potencia) con el fin de excitar los bobinados del motor.

Para esta el desarrollo de esta tesis, primeramente se había decidido, desarrollar el propio driver de control basado en tecnología "Mosfet", desafortunadamente se presentaron diversos inconvenientes en el momento de realizar la integración del producto final, razón por lo que se opto por cambiar y hacer uso del circuito integrado (CI) *l298n* mostrado en la figura 3.51. El cual es construido como "DUAL FULL-BRIDGE DRIVER", lo que significa que en su interior contiene dos puente H completos.

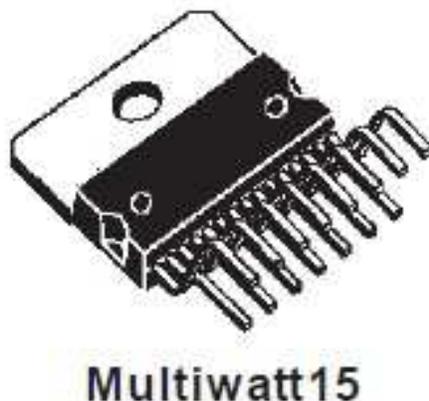


Figura 3.51: Circuito integrado L298n.

Tal circuito fue elegido debido a las características técnicas que presenta, como son amplio rango de voltaje de alimentación, corriente de salida, protección de corto circuito y la capacidad de realizar una conexión en paralelo doblando sus capacidades de operación. La hoja de datos correspondiente al circuito *L298n* se muestra en el anexo 5.5.

Una vez que se probó que el circuito funcionaba adecuadamente, se prosiguió con el desarrollo de la tarjeta impresa, para ello fue necesario realizar el diseño del circuito impreso (PCB).

Los circuitos impresos son una parte constitutiva y de fundamental importancia para

3. METODOLOGÍA

cualquier sistema electrónico. Su correcto diseño asegura un funcionamiento confiable en condiciones de operación no ideales; es decir, ante perturbaciones generadas por el medio en el que se desarrolla.

A continuación se presenta el diseño del circuito impreso para la etapa de potencia. Dicho diseño puede apreciarse en la figura 3.52.

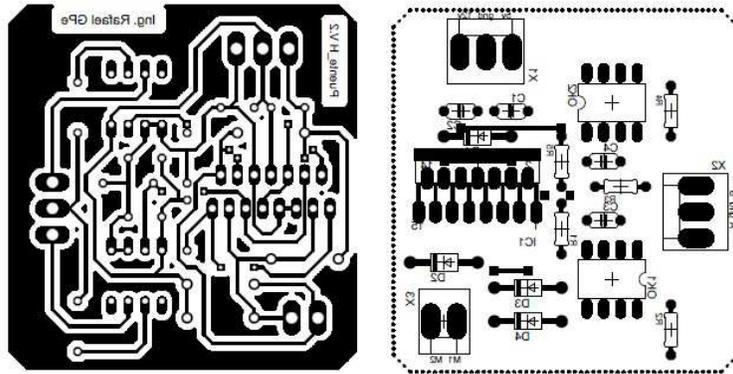


Figura 3.52: PCB para puente H.

En la figura anterior se puede apreciar los distintos componentes que integran a la etapa de potencia. Independientemente del circuito *L298n* se observan dos circuitos integrados *6N137* (Optoacoplador con salida a compuerta lógica NAND), encargados de proporcionar un aislamiento entre los circuitos de baja potencia (FPGA) y los componentes de potencia, con el fin de proporcionar protección al sistema de cortos circuitos.

El sistema contiene tres conectores (clemas) de conexión denotadas como "X1", "X2" y "X3". A través del conector "X1" se energiza el sistema, el cuál, para poder operar requiere de dos voltajes diferentes, $5v$ y $12v$.

Mediante el conector "X2" se aplican las señal enfocadas en el control de movimiento del motor, dichas señales provenientes de la FPGA son aisladas de los componentes de potencia mediante el uso de optoacopladores.

Finalmente el conector "X3", es utilizado para realizar la conexión de los motores. Es decir, a través del conector circulan las señales de control de potencia obtenidas por el controlador.

En la figura 3.53, se muestra la tarjeta de potencia terminada, como puede apreciarse el tamaño es relativamente pequeño, lo que da la flexibilidad de instalar el equipo en un área reducida.

3. METODOLOGÍA

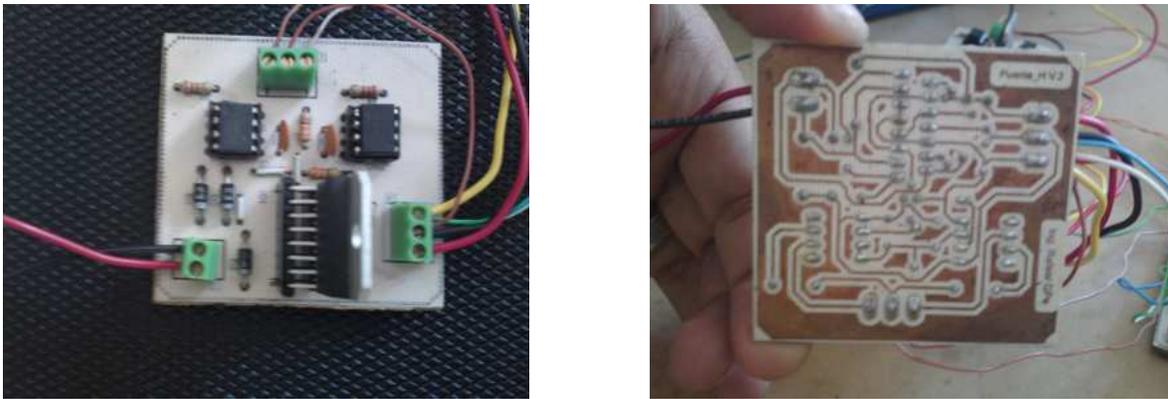


Figura 3.53: Tarjeta de circuito de potencia.

Por otra parte en la figura 3.54 se muestra el conjunto de tarjetas desarrolladas para el controlador de cuatro grados de libertad; en la figura, también se puede apreciar las conexiones de la tarjeta con el sistema.



Figura 3.54: Conjunto de tarjetas de potencia

3.14.2. Tarjeta de conexión

Con el fin de realizar la conexión entre los dispositivos, fue requerido el diseño de una tarjeta de interconexión capaz de conjuntar las diferentes tecnologías.

Teniendo en cuenta las conexiones originales del manipulador Rhino XR-3, y con el fin de realizar las mínimas modificaciones al robot, se diseñó la tarjeta bajo las especificaciones del manipulador.

Otra característica importante de la tarjeta es el hecho de contener un sistema de acondicionamiento de las señales provenientes de los canales del encoder, esto es indispensable para todo tipo de encoder, sin importar la marca y calidad, debido a

3. METODOLOGÍA

que en todo sistema existe el riesgo de tener ruido eléctrico, el cual provocaría que las mediciones que estemos realizando sean incorrectas.

Para obtener una inmunidad al ruido, se opto por utilizar un disparador Schmitt, específicamente el chip $74ls14$ (compuerta tipo not), la cualidad de esta compuerta es manejar voltajes pertenecientes a la familia TTL pero sobre todo tener entradas de tipo disparador Schmitt, cualidad que origina la inmunidad al ruido, para mayor información vea el anexo 5.6.

Para realizar la conexión entre la tarjeta y el dispositivo FPGA, se tomaron en cuenta las características de los headers de expansión de la FPGA. Dicha tarjeta utiliza conectores de expansión de tipo IDE de 40 pines. ver figura 3.55.



Figura 3.55: Conector y cable tipo IDE de 40 pines

Este tipo de conector es el clásico conector de 40 pines utilizado en la interconexión de los componentes en las computadoras de escritorio.

En la figura 3.56, se muestra el esquema PCB diseñado para la tarjeta de interconexión.

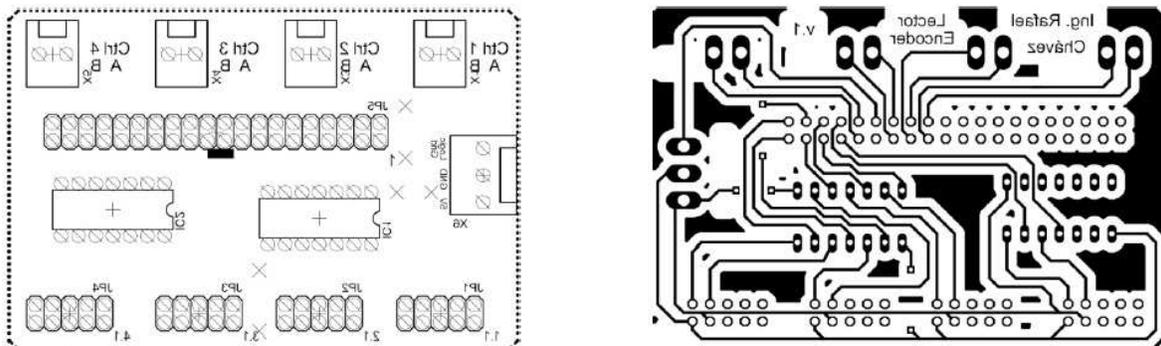


Figura 3.56: Tarjeta de circuito de interconexión.

En la figura pueden apreciarse los diferentes componentes de la tarjeta, como son cuatro conectores mediante los cuales se reciben las señales provenientes de los encoders, dos circuitos $74ls14$ encargados de proporcionar inmunidad al ruido eléctrico a cada uno de los canales de los encoder, un conector IDE para realizar la comunicación entre el FPGA y la tarjeta, cuatro conectores tipo clema de dos vias, mediante los cuales se emiten las señales de control a cada una de las tarjetas de potencia y un conector tipo clema de tres vias mediante el cual se realiza la alimentación de la tarjeta y de los encoder. En la figura 3.57 se muestra la tarjeta maquinada, desde una vista superior e inferior.

3. METODOLOGÍA

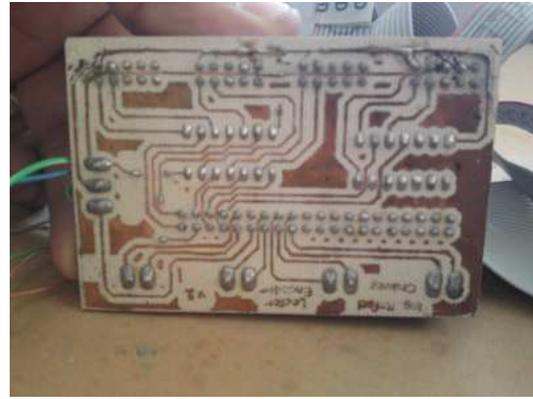
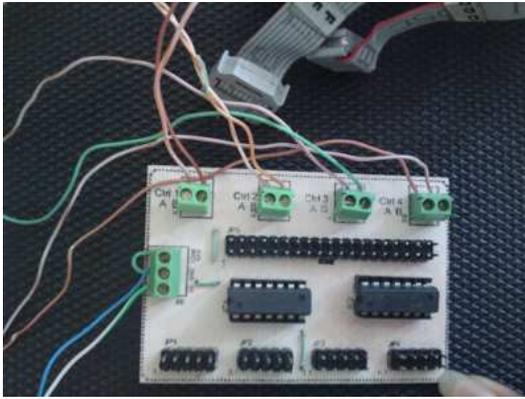


Figura 3.57: Tarjeta de lectura de encoder

Una vez finalizadas las tarjetas de adquisición y la tarjeta de etapa de potencia, se pudo realizar la interconexión del sistema completo y realizar pruebas de funcionalidad, las cuales arrojaron resultados satisfactorios, dichos resultados serán discutidos en la sección 4.

En la figura 3.58 se muestra el sistema de control finalizado integrado al manipulador Rhino XR-3 y la tarjeta FPGA.

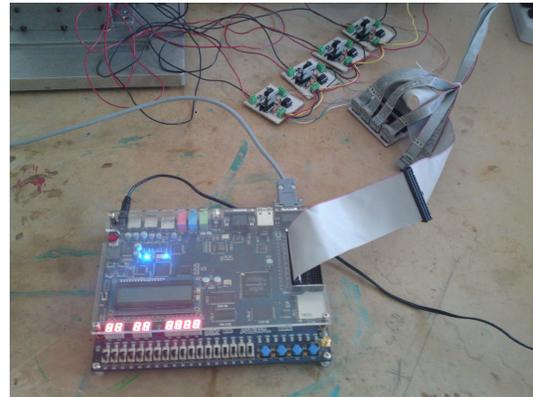
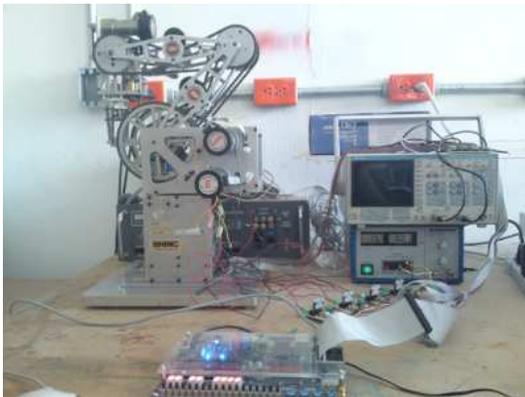


Figura 3.58: Sistema de control final

Capítulo 4

RESULTADOS Y DISCUSIÓN

En este capítulo se presentan de forma ordenada los resultados obtenidos del desarrollo de esta tesis. La forma de organización consiste en mostrar el desempeño de los controladores, que fueron caso de estudio. En cada prueba se considero el problema de seguimiento de trayectoria, siendo la trayectoria deseada un círculo de 10.16cm (4in) de radio sobre el plano $y_0 - z_0$, con un desplazamiento sobre el eje y de 38.1cm (15in) con respecto al origen o_0 . Primeramente se muestra el desempeño de cada controlador en vacío, es decir, en ausencia de carga. Posteriormente se presenta el desempeño de los controladores ante una carga de 2.3Kg . Después se muestra los resultados de los controladores aumentando la velocidad de la trayectoria deseada y finalmente se presenta una comparación entre los diferentes controladores.

La trayectoria deseada y que será la utilizada para la realización de las pruebas, se muestra en la figura 4.1.

La figura anterior muestra los puntos en el espacio que debe alcanzar el manipulador en cada eje.

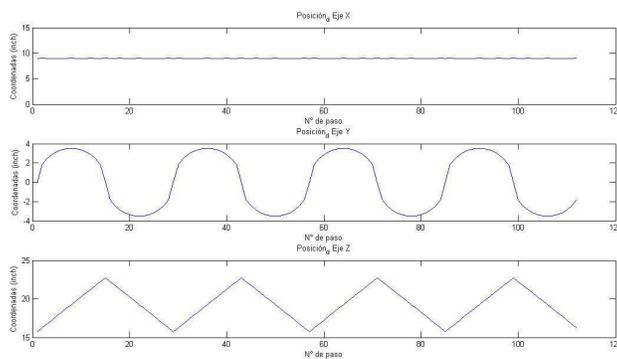


Figura 4.1: Trayectoria deseada

4.1. Resultados en vacío

4.1.1. PD+I

Como primer prueba se utilizó el controlador PD+I, mostrado en la sección 3.6, cuyos resultados fueron satisfactorios con respecto al seguimiento de trayectoria manteniendo una buena relación entre posición y la señal de control calculada.

Realizando una sintonización del controlador de forma empírica, se propusieron los siguientes valores para las ganancias del controlador.

$$P = \begin{bmatrix} 0.007 \\ 0.25 \\ 0.08 \\ 0.01 \end{bmatrix} \quad D = \begin{bmatrix} 0.00001 \\ 0.00001 \\ 0.000009 \\ 0.000001 \end{bmatrix} \quad I = \begin{bmatrix} 0.001 \\ 0.002 \\ 0.003 \\ 0.0001 \end{bmatrix}$$

La figura 4.2, muestra la comparación entre la posición de referencia y la posición obtenida por cada una de las articulaciones del manipulador.

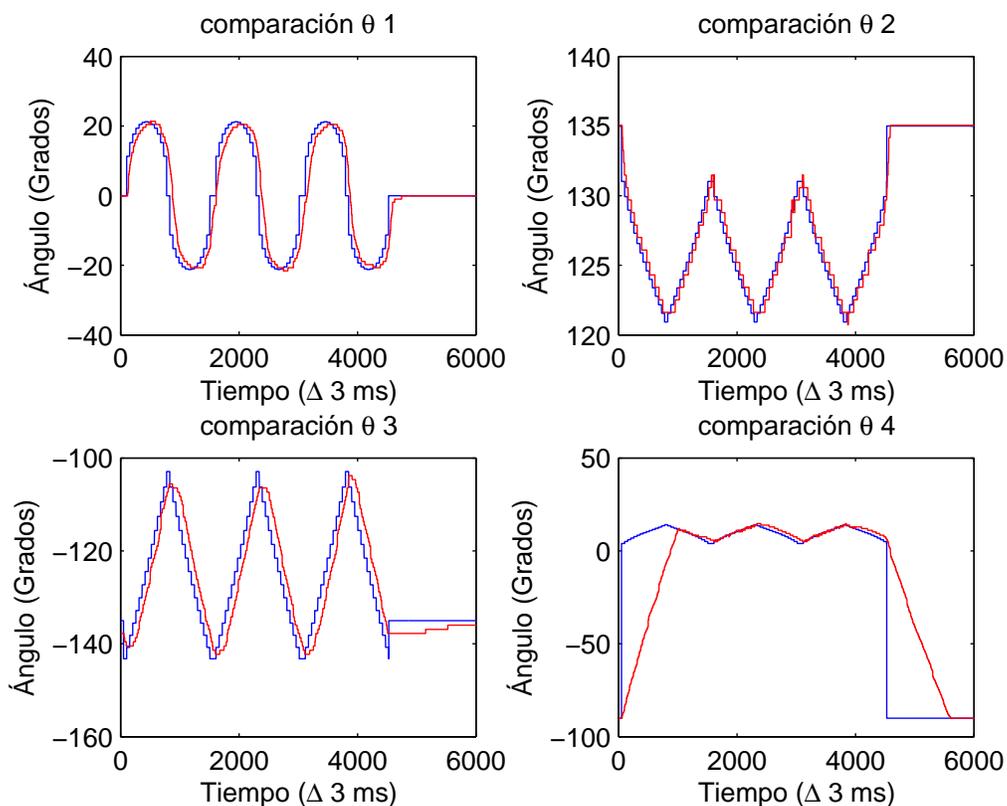


Figura 4.2: Posición articular obtenida (PD+I)

Como puede apreciarse el seguimiento es aceptable, ante lo esperado por hacer uso del controlador $PD + I$.

4. RESULTADOS Y DISCUSIÓN

En la figura 4.3, se muestra la señal de error articular, cuya amplitud puede apreciarse en unidades de grados. Como puede apreciarse la articulación con mayor desviación fue la base (articulación 4), debido a la inercia con la que cuenta.

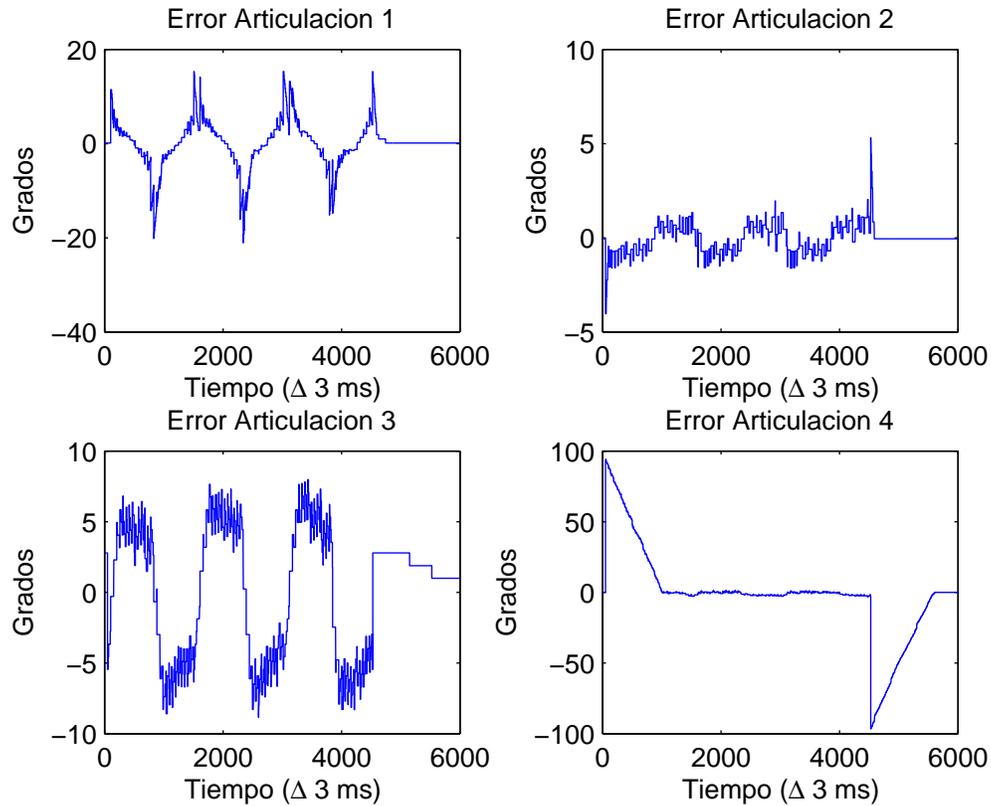


Figura 4.3: Señal de error articular (PD+I)

La figura 4.4, muestra la señal de control requerida para alcanzar la posición deseada, como puede apreciarse la señal de control obtenidas presentan un cierto grado de ruido.

4. RESULTADOS Y DISCUSIÓN

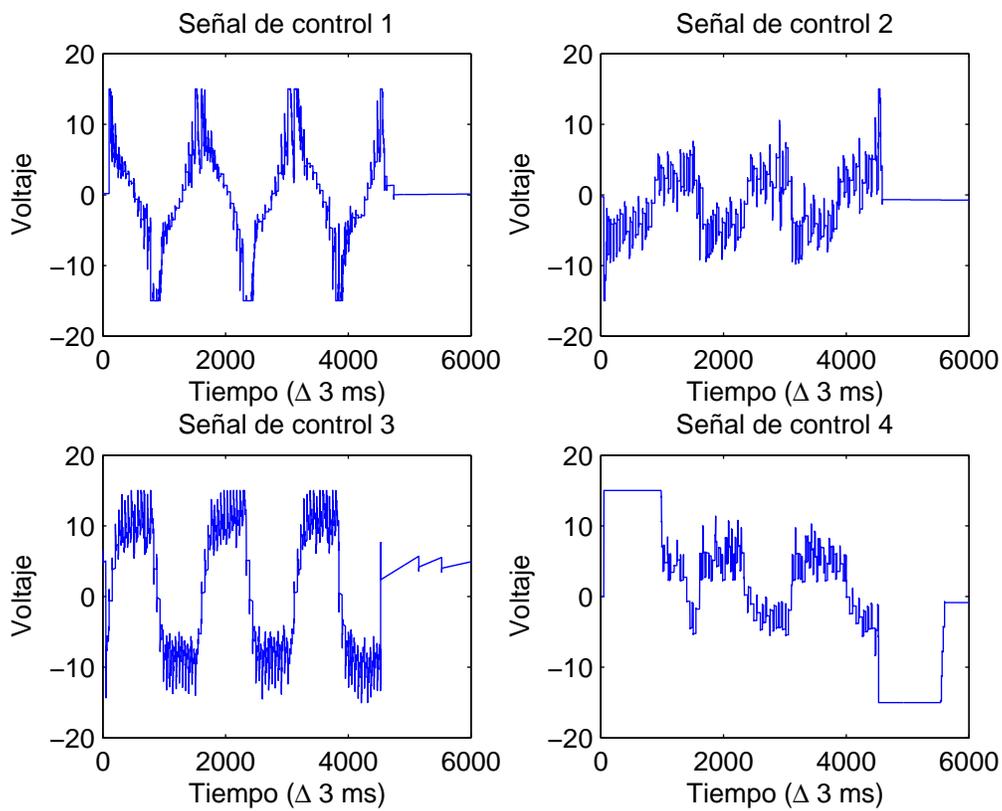


Figura 4.4: Señal de control (PD+I)

Finalmente en la figura 4.5, se muestra la posición del efector final obtenida al realizar la cinemática inversa con respecto a las posiciones articulares generadas.

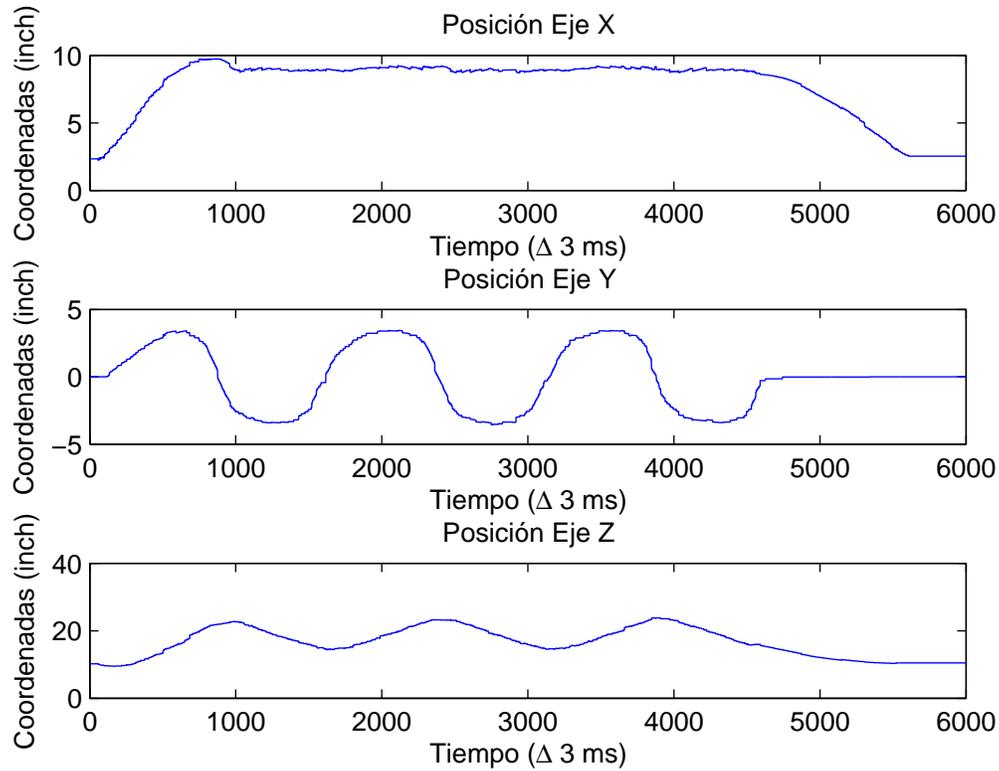


Figura 4.5: Posición final obtenida (PD+I)

4.1.2. PD+W

Continuando con la prueba de los controladores implementados, se describirá lo obtenido con el controlador $PD + W$ mostrado en la sección 3.7, cuyos resultados se muestran a continuación.

Con el fin de realizar una comparación justa entre los controladores, se utilizaron los mismos parámetros PD para este controlador. Por lo tanto, se realizó una sintonización del controlador de forma empírica, obteniendo los siguientes valores para las ganancias del controlador.

$$K = \begin{bmatrix} 0.00001 \\ 0.00001 \\ 0.000009 \\ 0.000001 \end{bmatrix} \quad \lambda = \begin{bmatrix} 700 \\ 2500 \\ 888.88 \\ 10000 \end{bmatrix} \quad \delta = \begin{bmatrix} 0.00001 \\ 0.000001 \\ 0.000003 \\ 0.000001 \end{bmatrix} \quad \epsilon = \begin{bmatrix} 0.00005 \\ 0.00002 \\ 0.0001 \\ 0.0001 \end{bmatrix}$$

Dichos valores, fueron establecidos para obtener las mismas ganancias PD del controlador PD + I, es por ello que se muestra valores muy elevados y muy pequeños.

La figura 4.6, muestra la comparación entre la posición de referencia y la posición obtenida por cada una de las articulaciones del manipulador.

4. RESULTADOS Y DISCUSIÓN

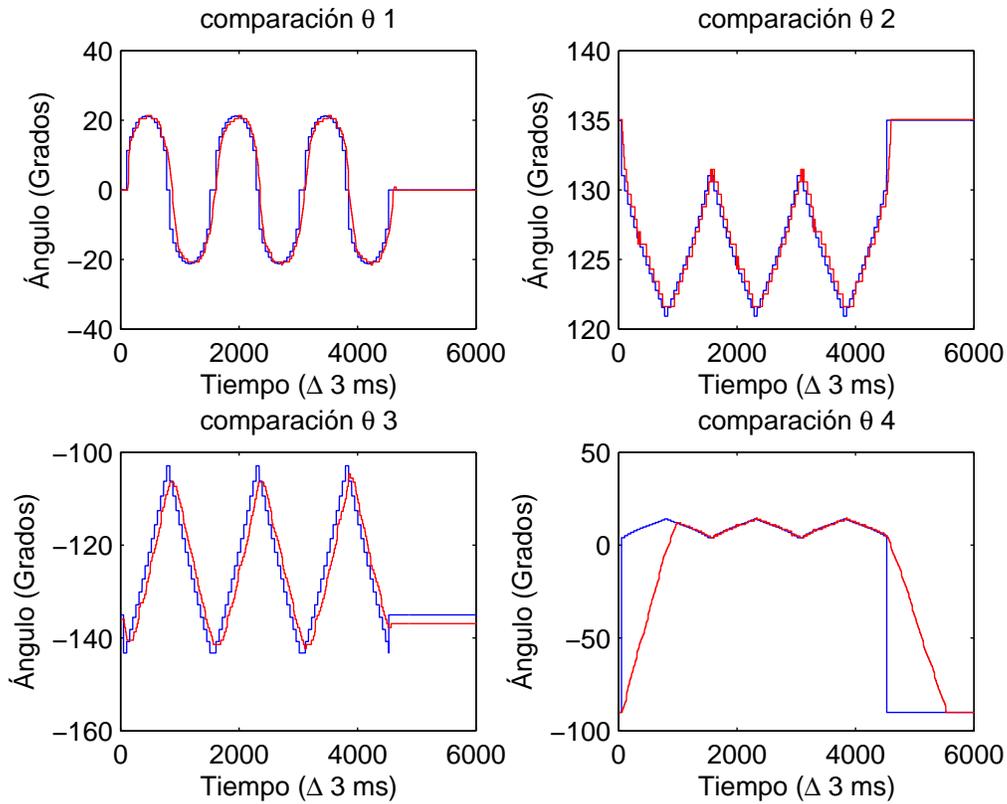


Figura 4.6: Posición articular obtenida (PD+W)

Como puede apreciarse el seguimiento es aceptable, y solamente la articulación 3 (codo), presenta un pequeño error en estado estacionario.

En la figura 4.7, se muestra la señal de error articular, cuya amplitud puede apreciarse en unidades de grados. Como puede apreciarse la articulación con mayor error fue la base al igual que en el controlador PID.

4. RESULTADOS Y DISCUSIÓN

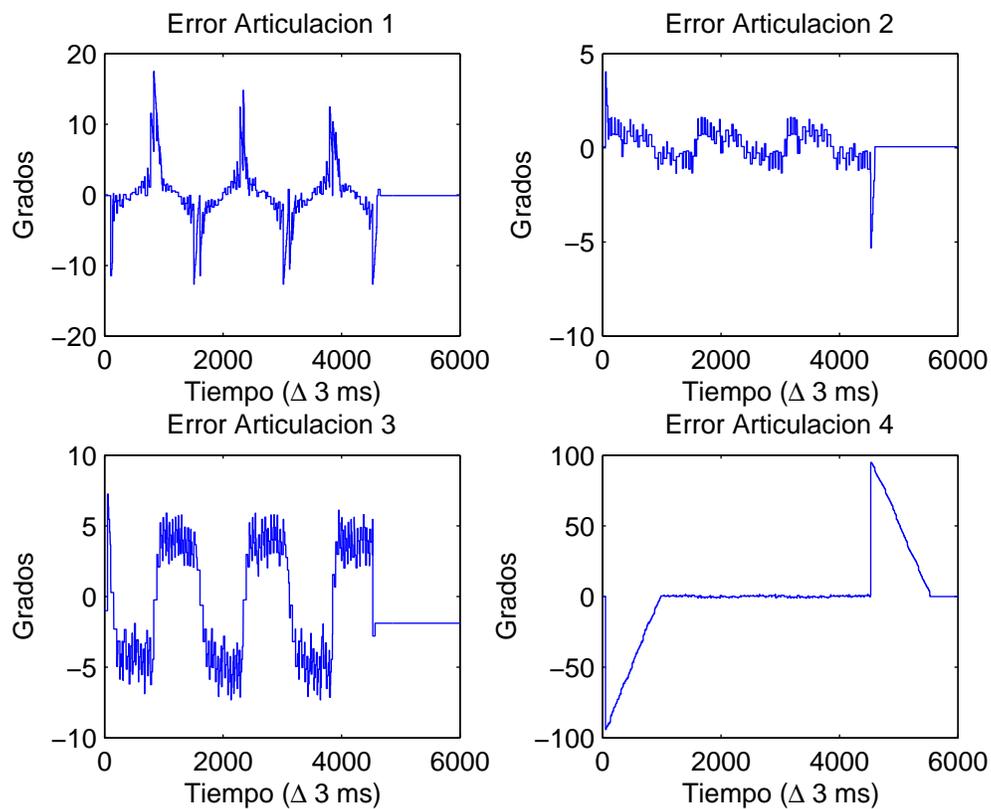


Figura 4.7: Señal de error articular (PD+W)

La figura 4.8, muestra la señal de control requerida para alcanzar la posición deseada, como puede apreciarse la señal de control obtenida es demasiado ruidosa, lo cual puede llevar a un desgaste mas rápido en los sistemas.

4. RESULTADOS Y DISCUSIÓN

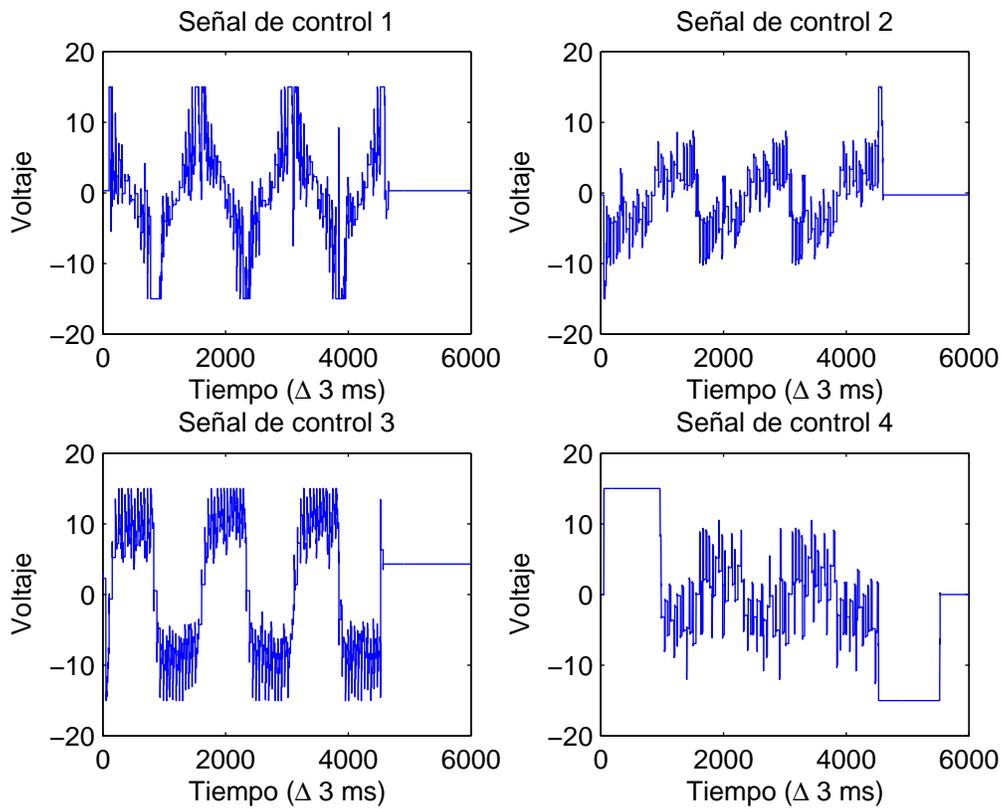


Figura 4.8: Señal de control (PD+W)

Finalmente en la figura 4.9, se muestra la posición del efector final obtenida al realizar la cinemática inversa con respecto a las posiciones articulares generadas.

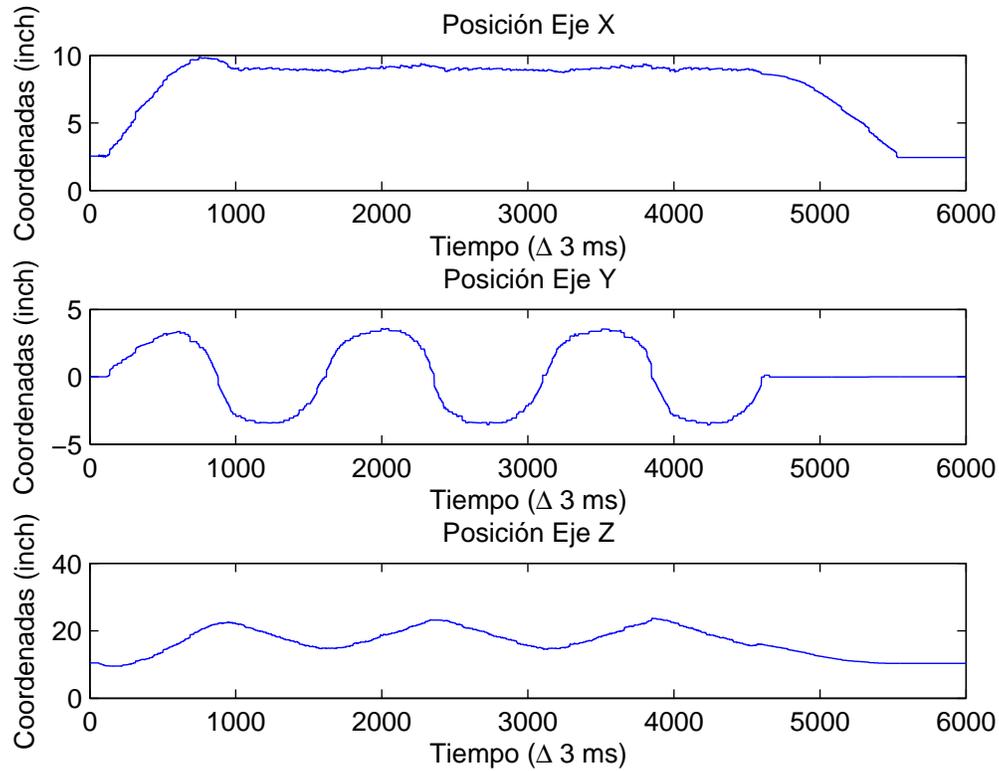


Figura 4.9: Posición final obtenida (PD+W)

4.1.3. PD+W (Adaptable)

Una vez obtenido los resultados para el controlador PD+W, se realizaron pruebas con el controlador PD+W en modo adaptable, obteniendo los siguiente resultados.

Con el fin de realizar una comparación justa entre los controladores, se utilizaron los mismos parámetros PD y los parámetros utilizados en el controlador PD+W para este controlador. Por lo tanto, se realizó una sintonización del controlador de forma empírica, obteniendo los siguientes valores para las ganancias del controlador.

Por lo tanto, el parámetro adaptable como lo indica la metodología vista en la sección 3.8, es el parámetro δ , cuyo valor inicial es de:

$$\delta = \begin{bmatrix} 0.00001 \\ 0.00001 \\ 0.000009 \\ 0.000001 \end{bmatrix}$$

La figura 4.10, muestra la comparación entre la posición de referencia y la posición obtenida por cada una de las articulaciones del manipulador.

4. RESULTADOS Y DISCUSIÓN

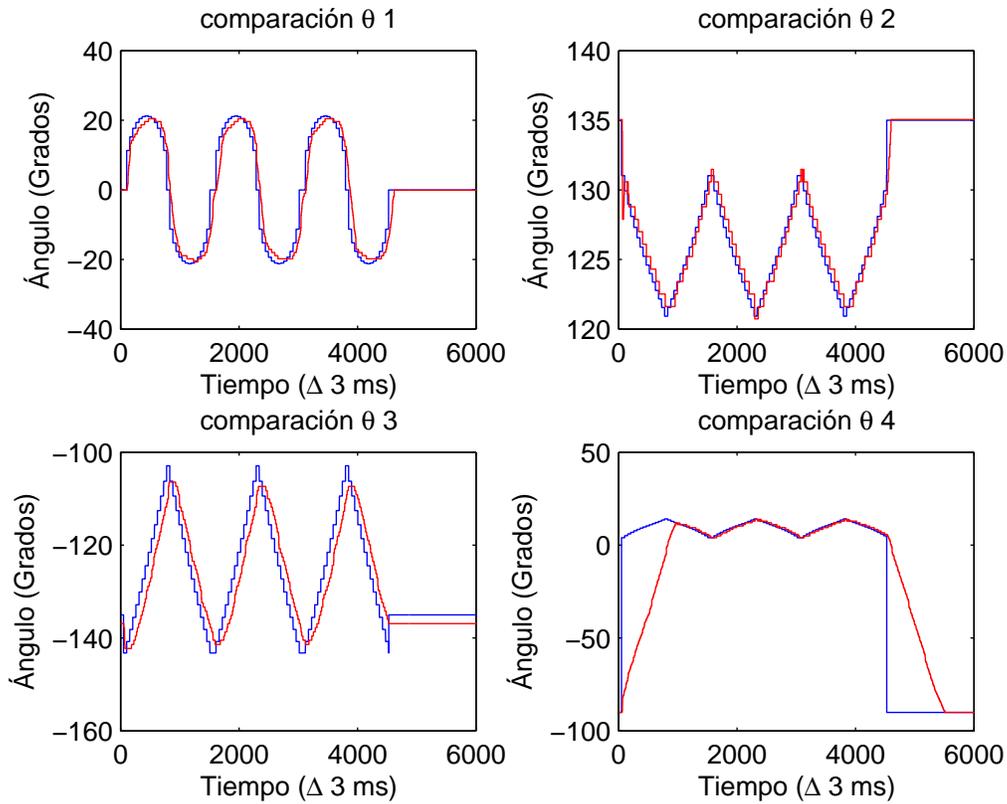


Figura 4.10: Posición articular obtenida (PD+W Adaptable)

Como puede apreciarse el seguimiento es suave, teniendo una pequeña desviación en estado estacionario para la articulación 3.

En la figura 4.11, se muestra la señal de error articular, cuya amplitud se aprecia en grados. Como puede apreciarse la articulación con mayor error continua siendo la base, debido a la inercia con la que cuenta.

4. RESULTADOS Y DISCUSIÓN

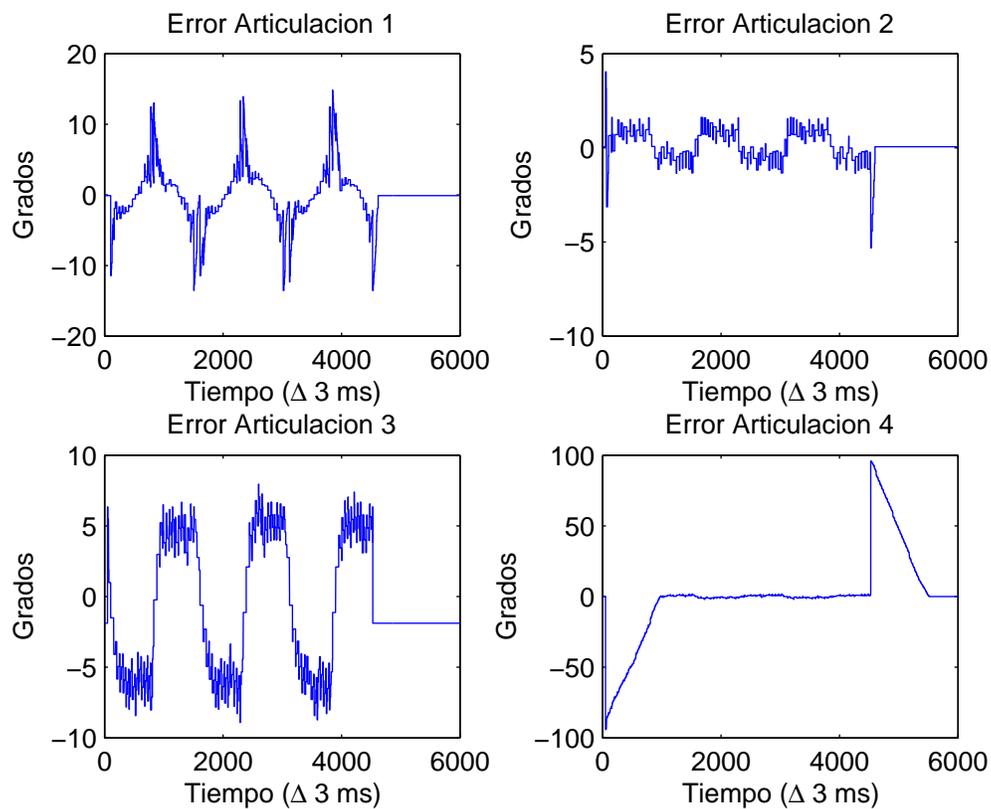


Figura 4.11: Señal de error articular (PD+W Adaptable)

La figura 4.12, muestra la señal de control requerida para alcanzar la posición deseada. La cuál, se encuentra definida y sin la presencia de ruido excesivo, lo cuál incrementaría el tiempo de vida de los componentes.

4. RESULTADOS Y DISCUSIÓN

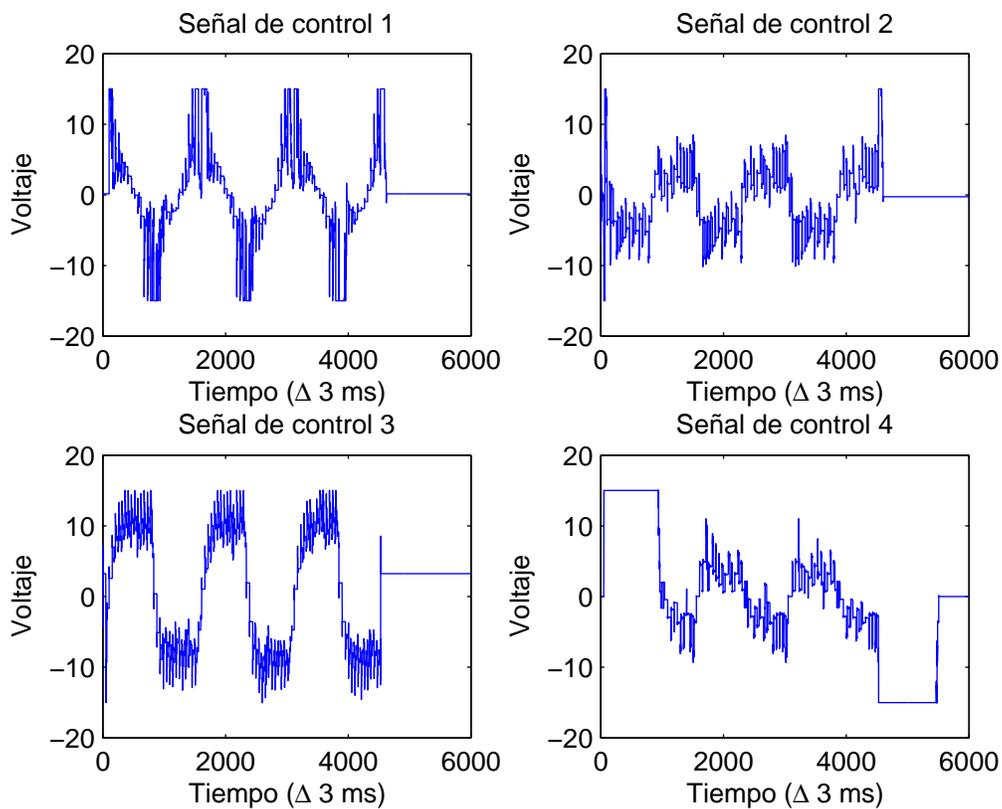


Figura 4.12: Señal de control (PD+W Adaptable)

Finalmente en la figura 4.13, se muestra la posición del efector final obtenida al realizar la cinemática inversa con respecto a las posiciones articulares generadas.

4. RESULTADOS Y DISCUSIÓN

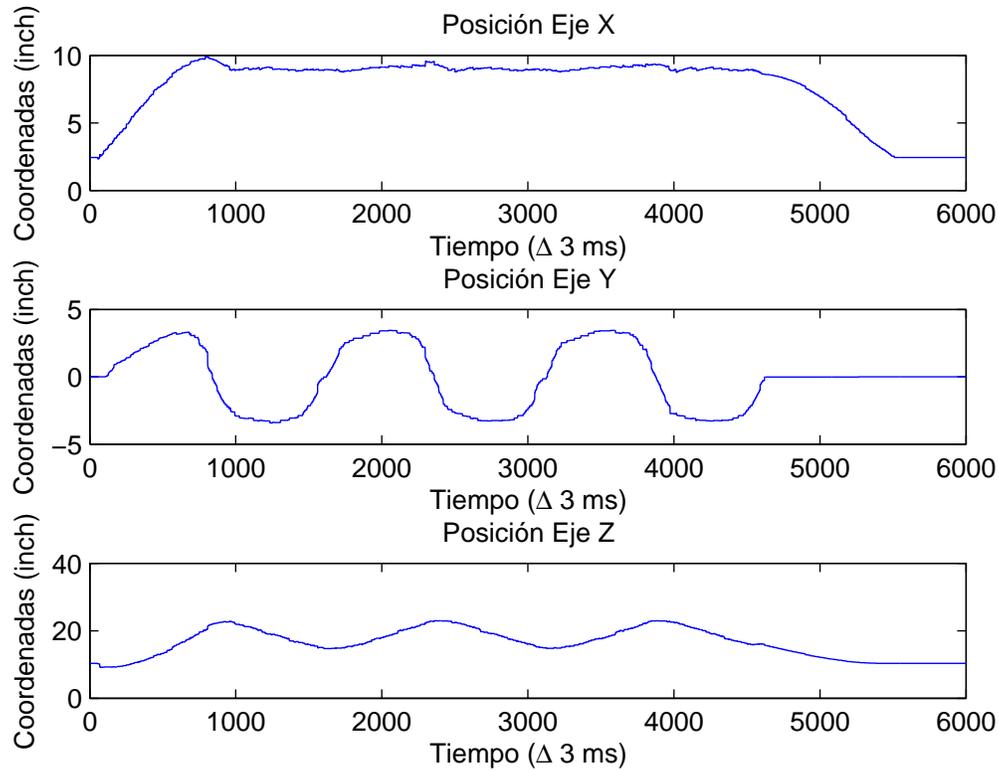


Figura 4.13: Posición final obtenida (PD+W Adaptable)

4.1.4. PD+G

Finalmente se presenta la implementación del controlador PD + G, mostrado en la sección 3.6, el cual está diseñado para adaptar los parámetros referidos a la carga.

Realizando una sintonización del controlador de forma empírica, se propusieron los siguientes valores para las ganancias del controlador, recordando que se utilizaron los mismos parámetros PD del controlador PD +I.

$$\beta = \begin{bmatrix} 0.00 \\ 0.025 \\ 0.008 \\ 0.01 \end{bmatrix} \quad \gamma = \begin{bmatrix} 0 \\ 0.00001 \\ 0.000009 \\ 0.000001 \end{bmatrix}$$

La figura 4.14, muestra la comparación entre la posición de referencia y la posición obtenida por cada una de las articulaciones del manipulador.

4. RESULTADOS Y DISCUSIÓN

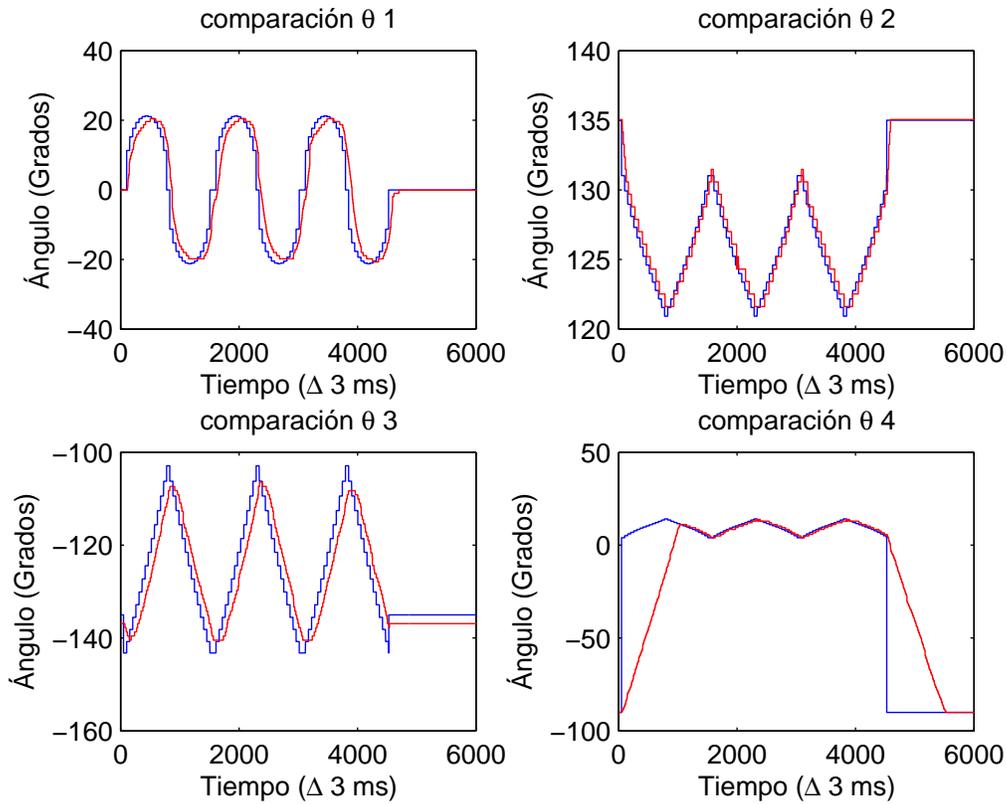


Figura 4.14: Posición articular obtenida (PD+G)

En la figura 4.15, se muestra la señal de error articular; cuya amplitud está en grados. Como puede apreciarse la articulación 1 (cintura), no se ve afectada por la compensación del controlador debido a que se consideró el origen en la base del manipulador; por lo tanto, no existe energía potencial en ese punto.

4. RESULTADOS Y DISCUSIÓN

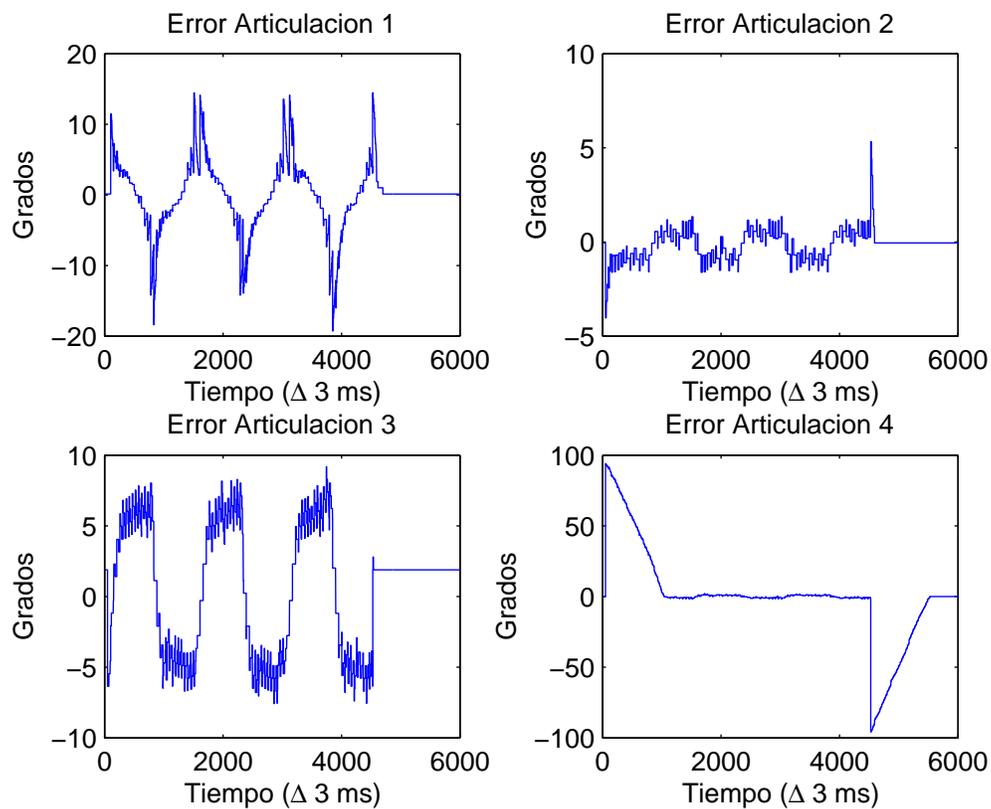


Figura 4.15: Señal de error articular (PD+G)

La figura 4.16, muestra la señal de control requerida para alcanzar la posición deseada. Como puede apreciarse estas señales presentan muy poco ruido incrementando la vida útil de los componentes del sistema.

4. RESULTADOS Y DISCUSIÓN

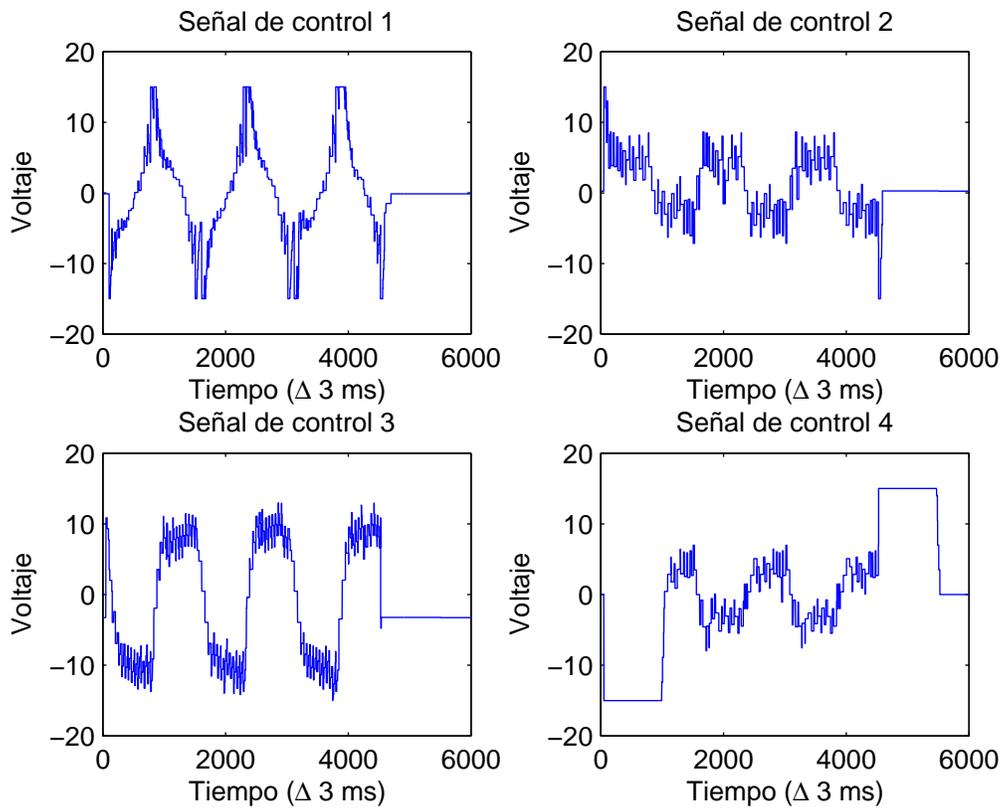


Figura 4.16: Señal de control (PD+G)

Finalmente en la figura 4.17, se muestra la posición del efector final obtenida al realizar la cinemática inversa con respecto a las posiciones articulares generadas.

4. RESULTADOS Y DISCUSIÓN

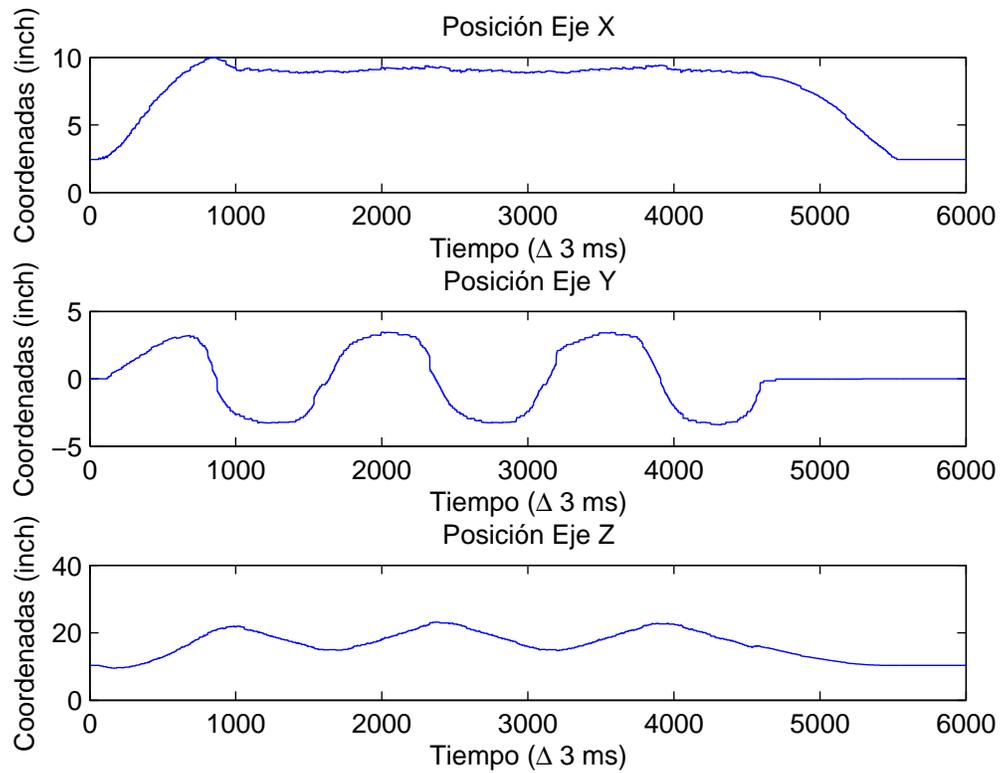


Figura 4.17: Posición final obtenida (PD+G)

Con el fin de realizar una comparación entre los controladores anteriormente mencionados; se presentan dos figuras, en las cuales se muestran las señales obtenidas por cada controlador. Dichas figuras se encuentran divididas en 4 cuadrantes, donde cada cuadrante muestra la señal generada por los controladores en el siguiente orden: PD+I, PD+W, PD+W (adaptable) y PD+G.

En la figura 4.36, se muestran las señales de error generadas por cada uno de los controladores.

4. RESULTADOS Y DISCUSIÓN

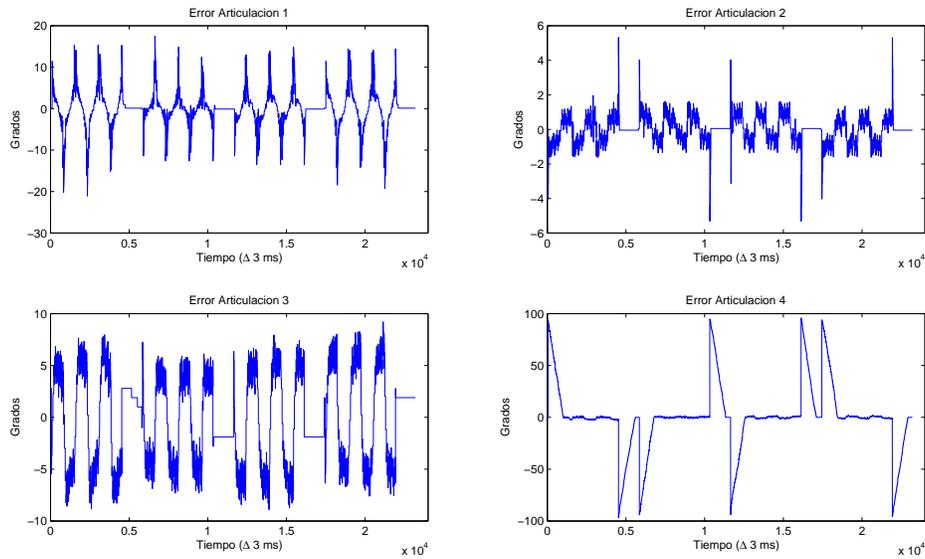


Figura 4.18: Comparativa de error para los controladores en vacío.

Como puede apreciarse, aunque el controlador PD+I, presenta un buen rendimiento, las variaciones del controlador PD, presentaron una desviación menor. En este punto se, observó que el controlador que presentó menor error fue el controlador PD+W.

En la figura 4.37, se muestran las señales de control generadas por cada controlador.

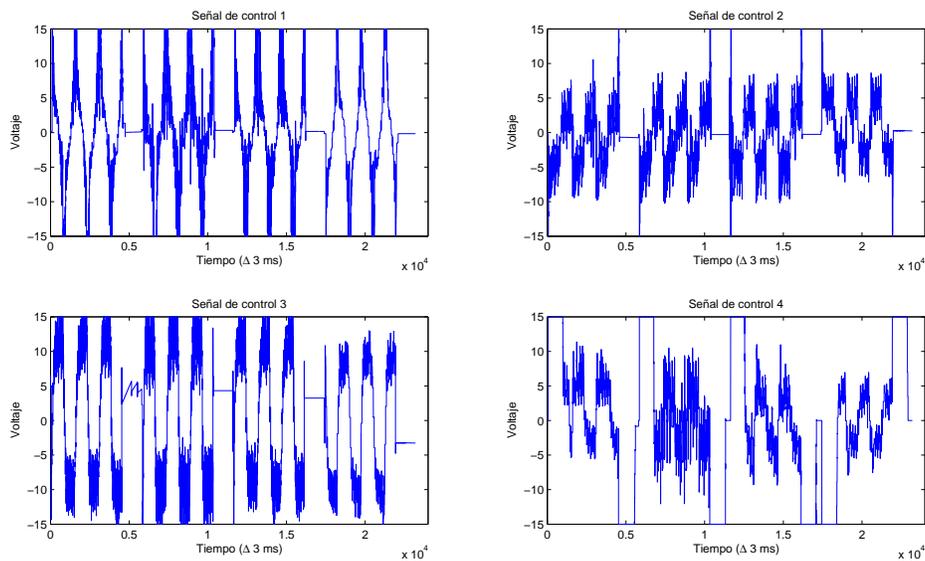


Figura 4.19: Comparativa de señal de control para los controladores en vacío.

Como puede apreciarse, las señales del controlador PD + W, en ambas modalidades; son las señales que presentan mas ruido, lo cual podría llegar a desgastar más

4. RESULTADOS Y DISCUSIÓN

rápida las componentes del sistema. Mientras que el controlador PD+G obtuvo las señales de control más limpias. Esto nos da oportunidad de elegir, entre un controlador que produzca una menor desviación en la consigna pero con una señal de control ruidosa(PD+W), un controlador con menor ruido en su señal de control pero con una desviación mayor con respecto a la consigna(PD+G) o un controlador que tenga la ventaja de tener una fácil implementación sacrificando la señal de error y de control (PD+I).

4.2. Resultados con carga

Una vez que fueron realizadas las pruebas de movimiento en vacío, se colocó una carga de 2.3kg al efector final, con la finalidad de observar el comportamiento de los controladores y determinar que controlador presenta ventajas.

4.2.1. PD+I

La figura 4.20, muestra la comparación entre la posición de referencia y la posición obtenida por cada una de las articulaciones del manipulador.

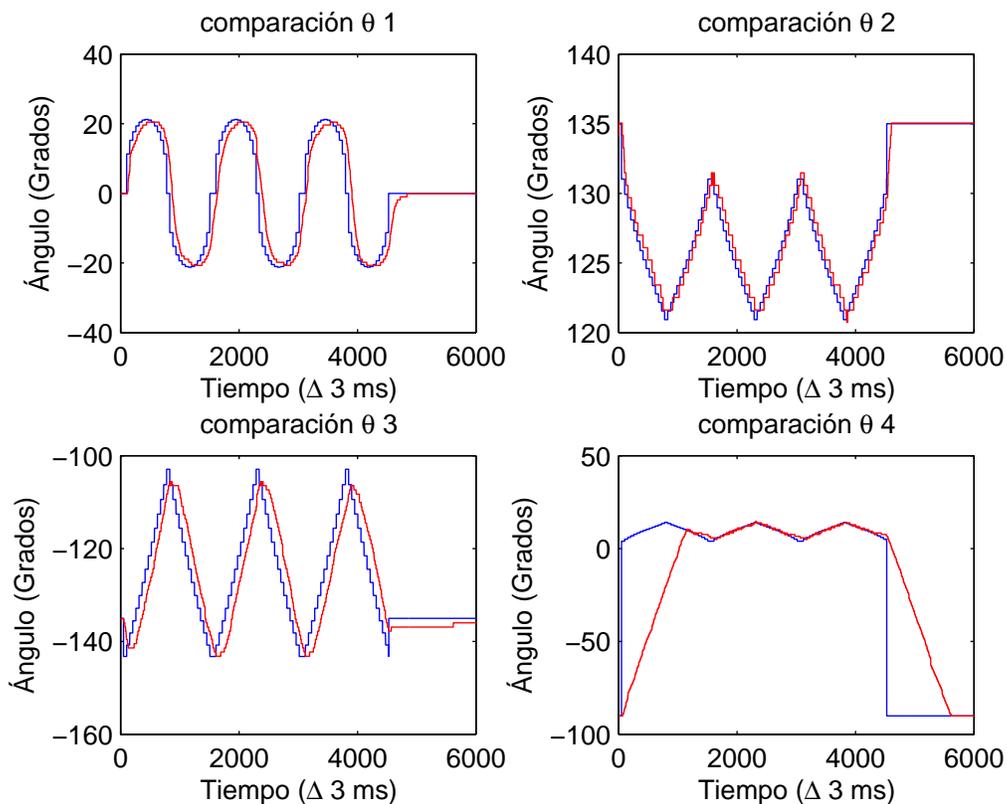


Figura 4.20: Posición articular obtenida (PD+I) con peso

4. RESULTADOS Y DISCUSIÓN

Como puede apreciarse el seguimiento sigue siendo aceptable aún con la presencia de carga.

En la figura 4.21, se muestra la señal de error articular, cuya amplitud puede apreciarse en grados.

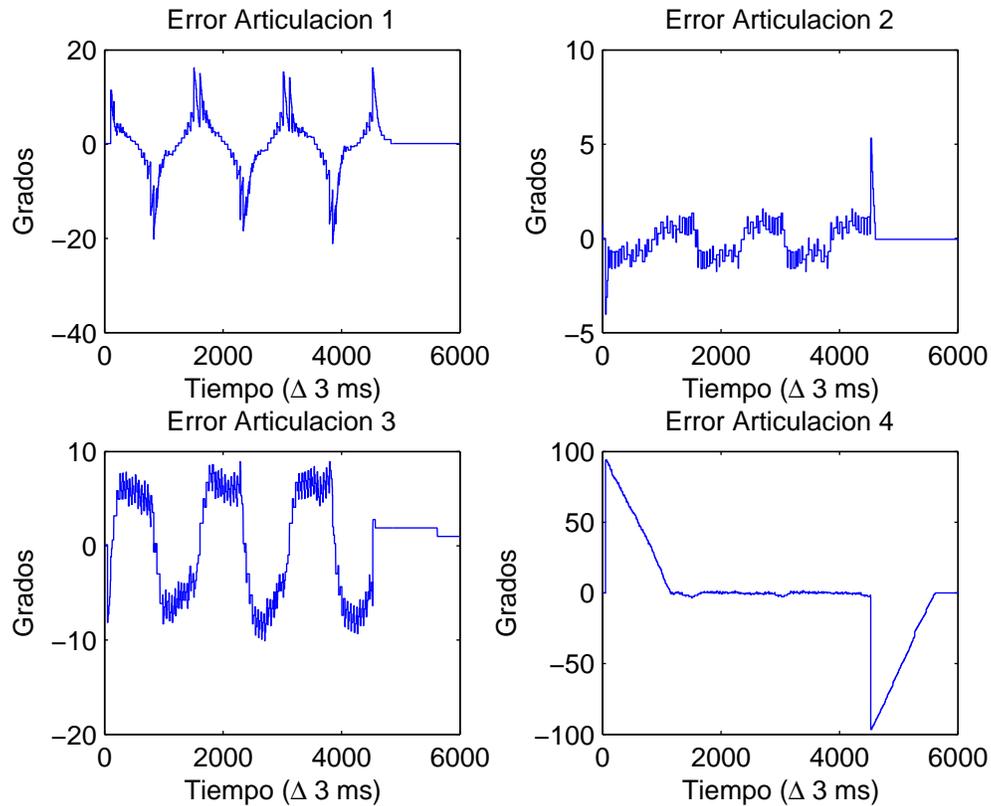


Figura 4.21: Señal de error articular (PD+I) con peso

La figura 4.22, muestra la señal de control requerida para alcanzar la posición deseada, como puede apreciarse la señal de control para la articulación 3, presenta una saturación en la region positiva.

4. RESULTADOS Y DISCUSIÓN

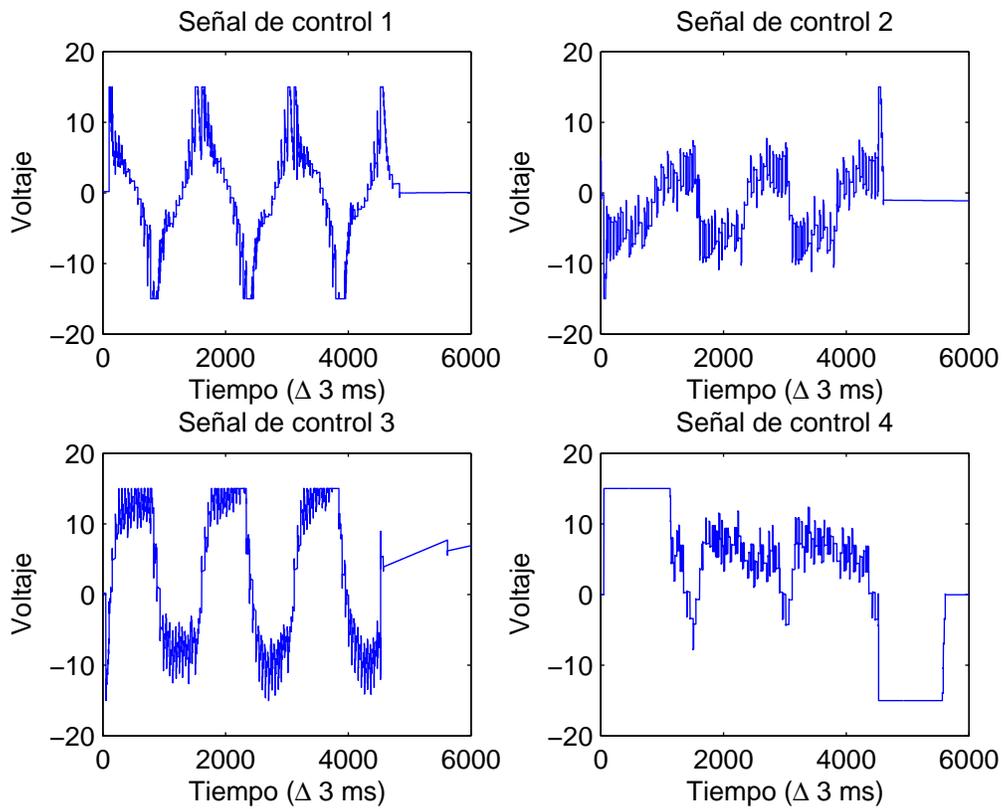


Figura 4.22: Señal de control (PD+I) con peso

Finalmente en la figura 4.23, se muestra la posición del efector final obtenida al realizar la cinemática inversa con respecto a las posiciones articulares generadas.

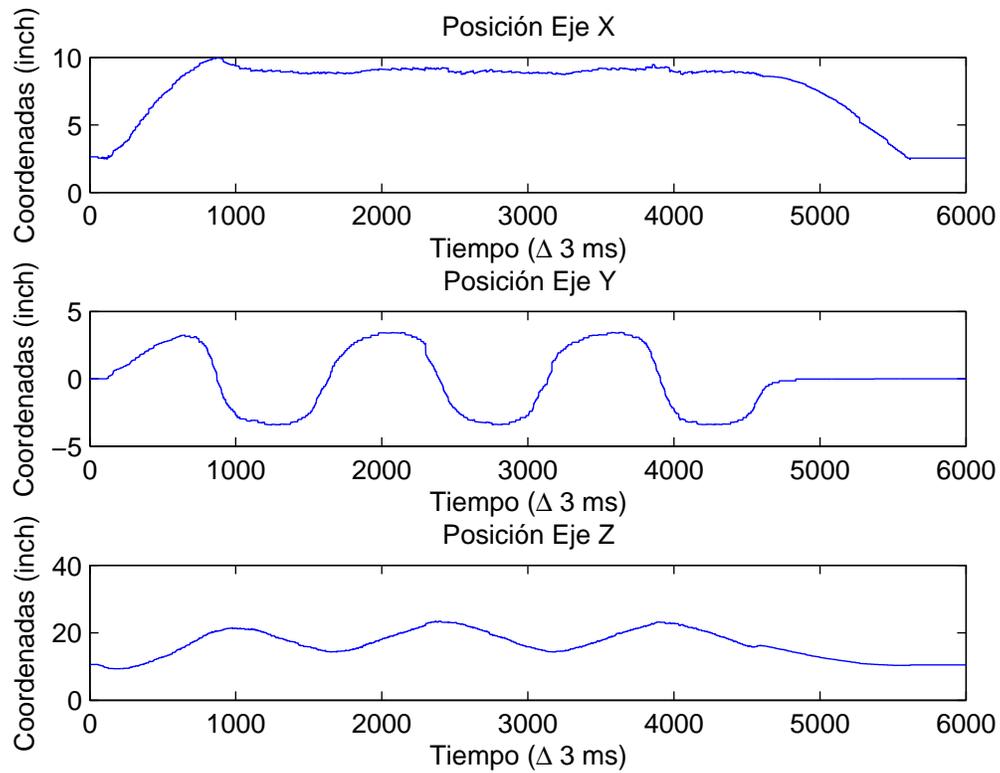


Figura 4.23: Posición final obtenida (PD+I) con peso

4.2.2. PD+W

A continuación se presentan los resultados mostrados por el manipulador con el controlador PD+w con la presencia de carga.

La figura 4.24, muestra la comparación entre la posición de referencia y la posición obtenida por cada una de las articulaciones del manipulador.

4. RESULTADOS Y DISCUSIÓN

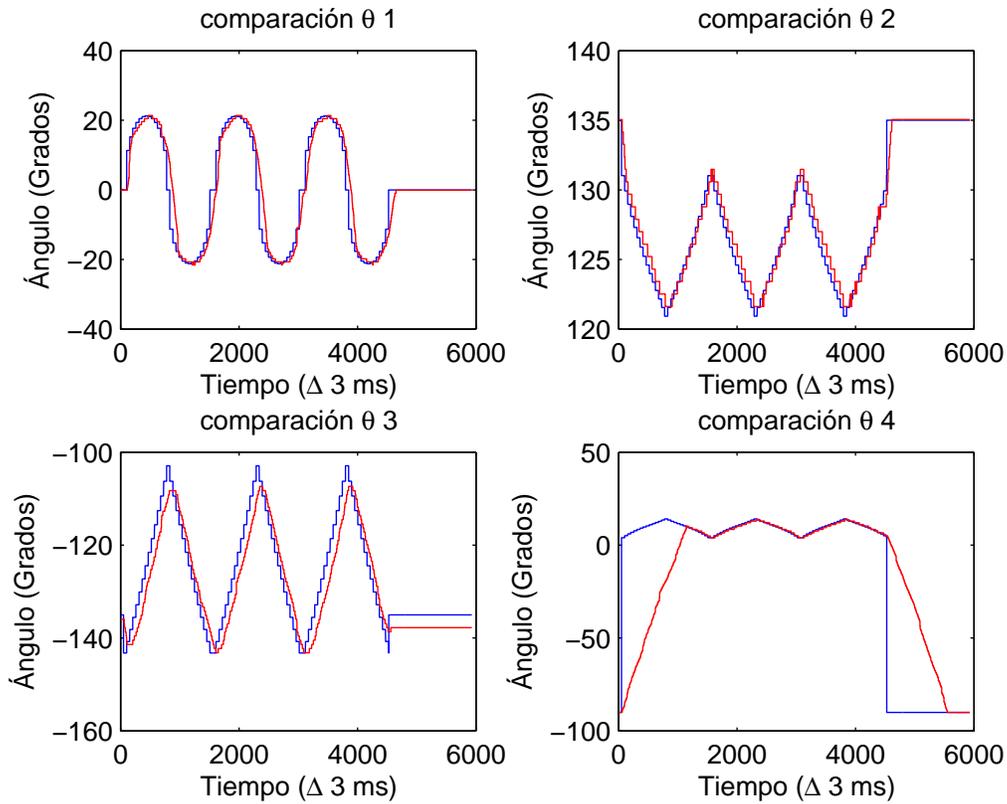


Figura 4.24: Posición articular obtenida (PD+W) con peso

Como puede apreciarse el seguimiento es aceptable, pero presenta una desviación apreciable en la articulación 3.

En la figura 4.25, se muestra la señal de error articular, cuya amplitud puede apreciarse en grados.

En esta figura, se puede apreciar que la articulación 4, redujo su señal de error, mientras que las articulaciones restantes no presentan una desviación significativa.

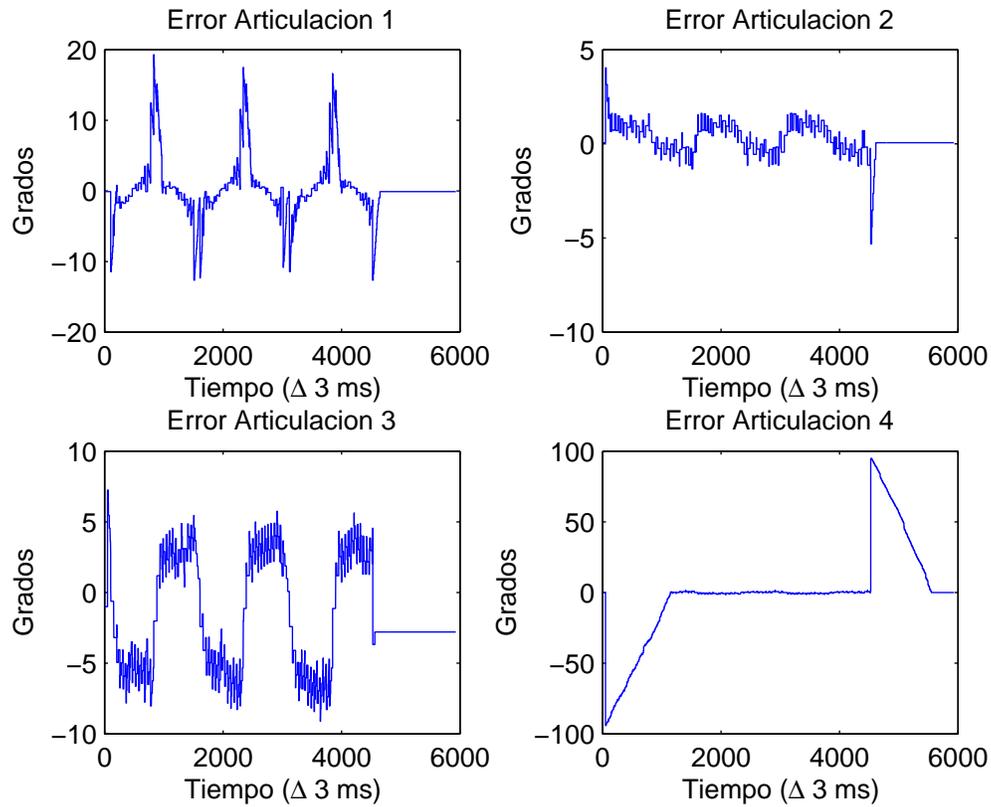


Figura 4.25: Señal de error articular (PD+W) con peso

La figura 4.26, muestra la señal de control requerida para alcanzar la posición deseada, como puede apreciarse al igual que la respuesta del controlador en vacío (figura 4.8), éste genera una señal de control con ruido excesivo lo cual se vería reflejado en el desgaste prematuro de los componentes del sistema.

4. RESULTADOS Y DISCUSIÓN

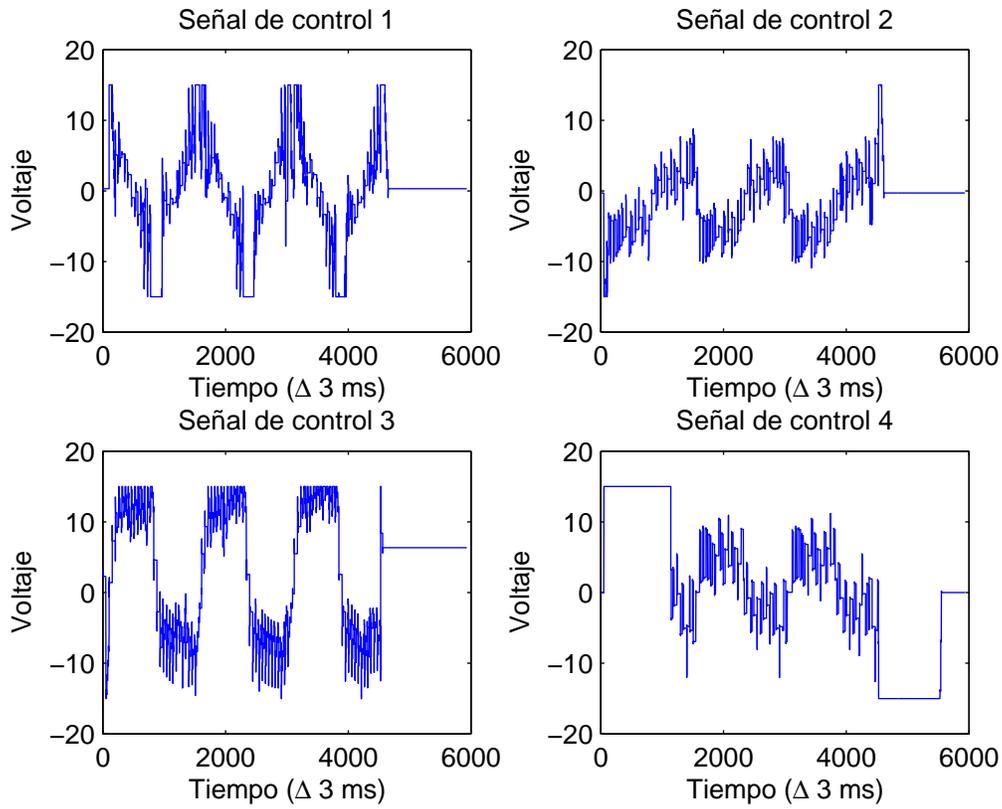


Figura 4.26: Señal de control (PD+W) con peso

Finalmente en la figura 4.27, se muestra la posición del efector final obtenida al realizar la cinemática inversa con respecto a las posiciones articulares generadas.

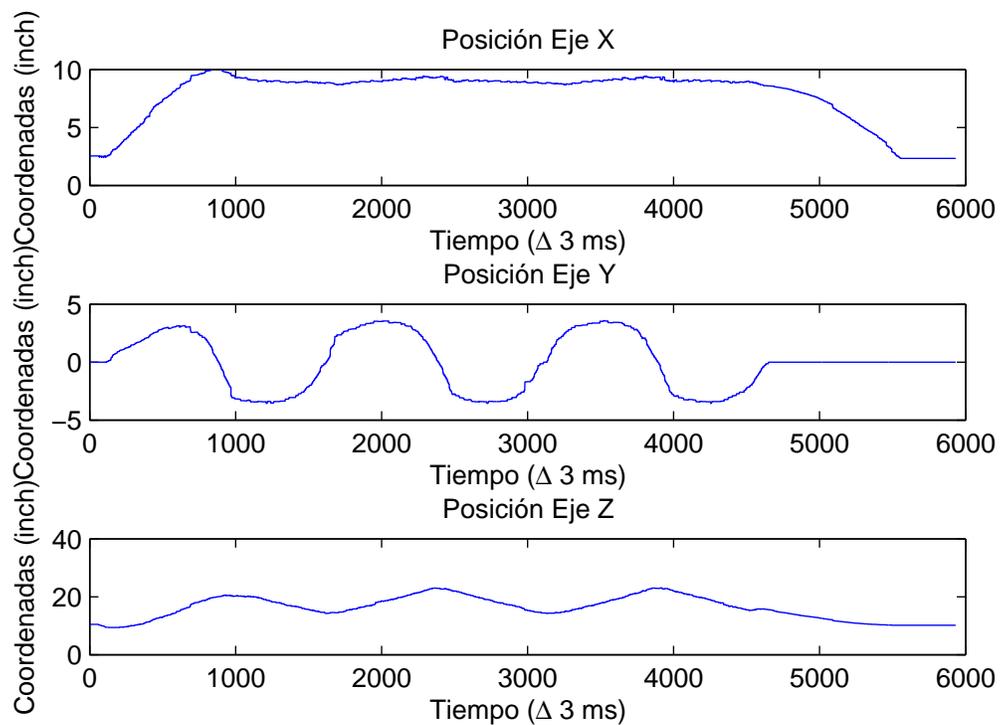


Figura 4.27: Posición final obtenida (PD+W) con peso

4.2.3. PD+W (Adaptable)

Con los resultados obtenidos por el controlador PD+W, se procedió a realizar pruebas con el controlador PD+W en la modalidad de adaptable, obteniendo los resultados que a continuación se describen.

La figura 4.28, muestra la comparación entre la posición de referencia y la posición obtenida por cada una de las articulaciones del manipulador.

4. RESULTADOS Y DISCUSIÓN

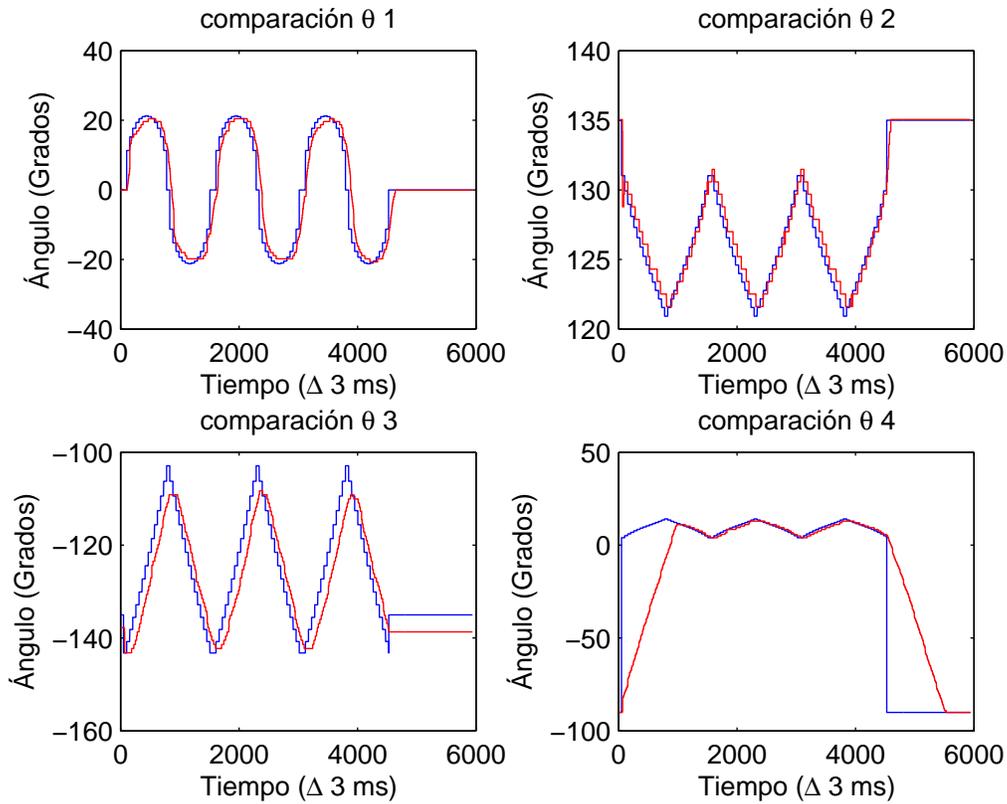


Figura 4.28: Posición articular obtenida (PD+W Adaptable) con peso

Como puede apreciarse, el seguimiento es aceptable para la mayoría de las articulaciones, a excepción de la articulación 3, la cuál presenta una desviación apreciable.

En la figura 4.29, se muestra la señal de error articular, cuya amplitud puede apreciarse en unidades de grados.

4. RESULTADOS Y DISCUSIÓN

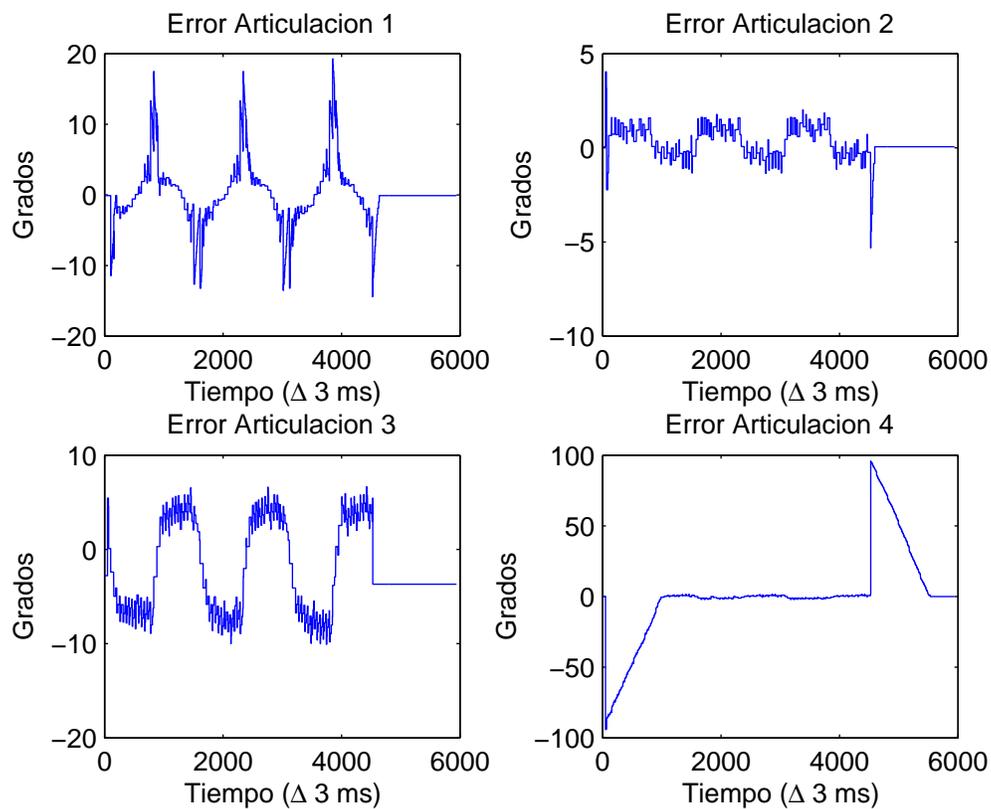


Figura 4.29: Señal de error articular (PD+W Adaptable) con peso

La figura 4.30, muestra la señal de control requerida para alcanzar la posición deseada, como puede apreciarse, se obtuvo una señal de control ruidosa, y la articulación 2 presenta saturación lo cual concuerda con la respuesta obtenida en la figura 4.29.

4. RESULTADOS Y DISCUSIÓN

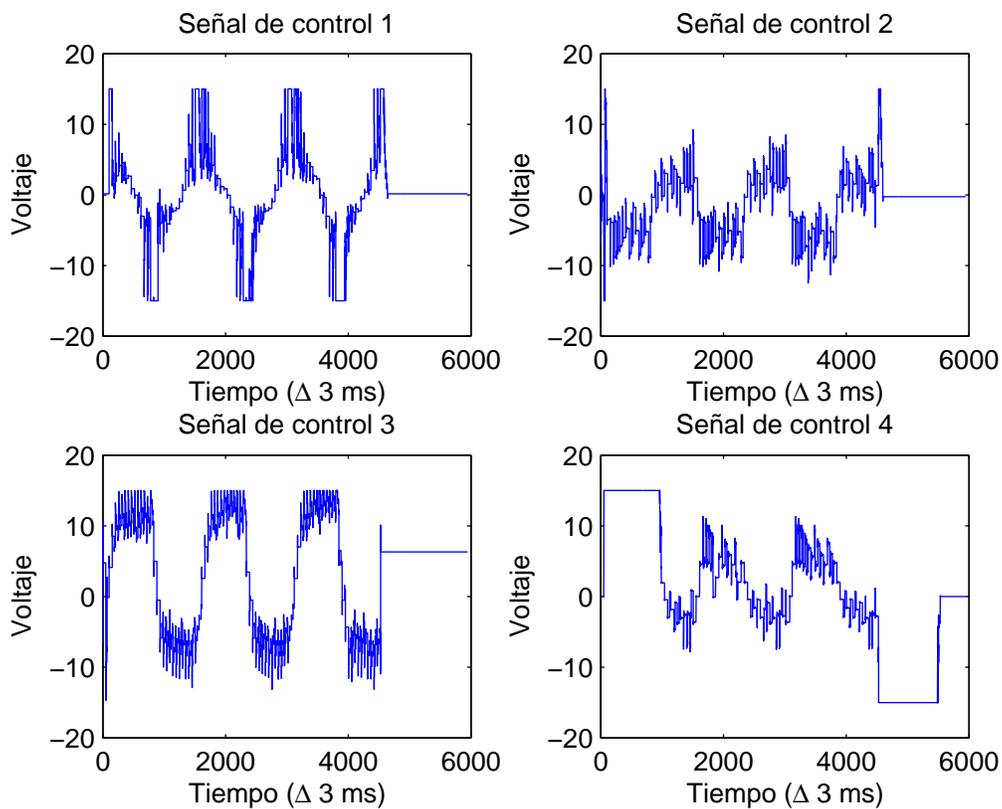


Figura 4.30: Señal de control (PD+W Adaptable) con peso

Finalmente en la figura 4.31, se muestra la posición del efector final obtenida al realizar la cinemática inversa con respecto a la posiciones articulares generadas. como puede apreciarse la señal es suave lo cual provocaría un buen desempeño en aplicaciones industriales.

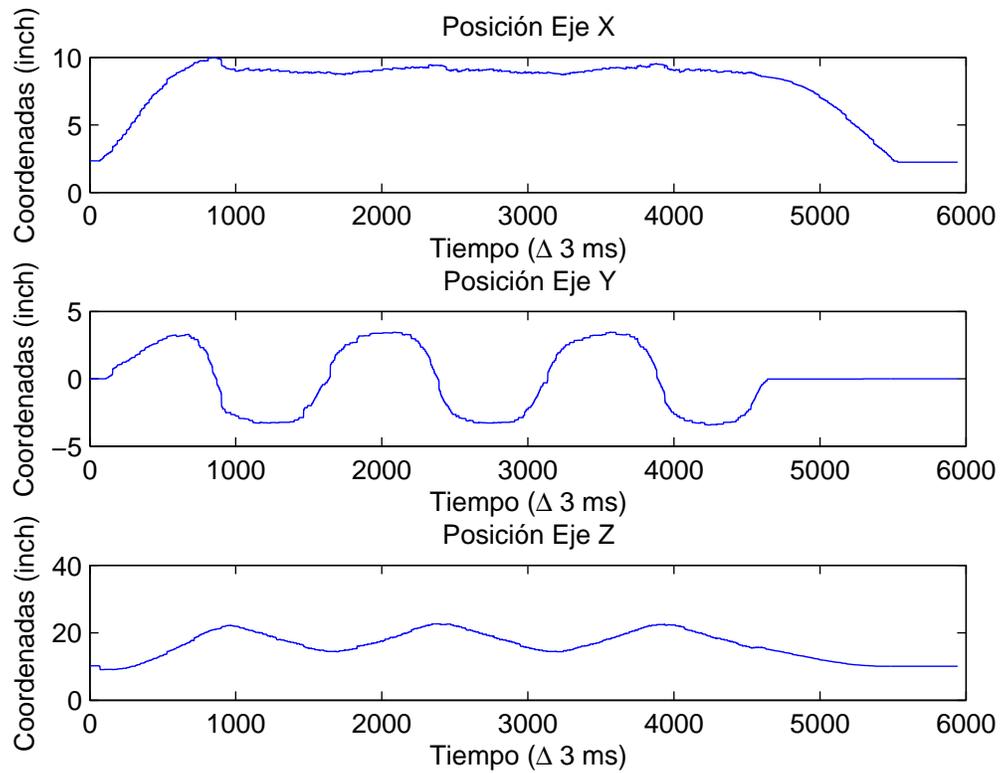


Figura 4.31: Posición final obtenida (PD+W Adaptable) con peso

4.2.4. PD+G

Finalmente se realizó la prueba con el controlador con compensación de gravedad, el cual arrojó los siguientes resultados.

La figura 4.32, muestra la comparación entre la posición de referencia y la posición obtenida por cada una de las articulaciones del manipulador.

4. RESULTADOS Y DISCUSIÓN

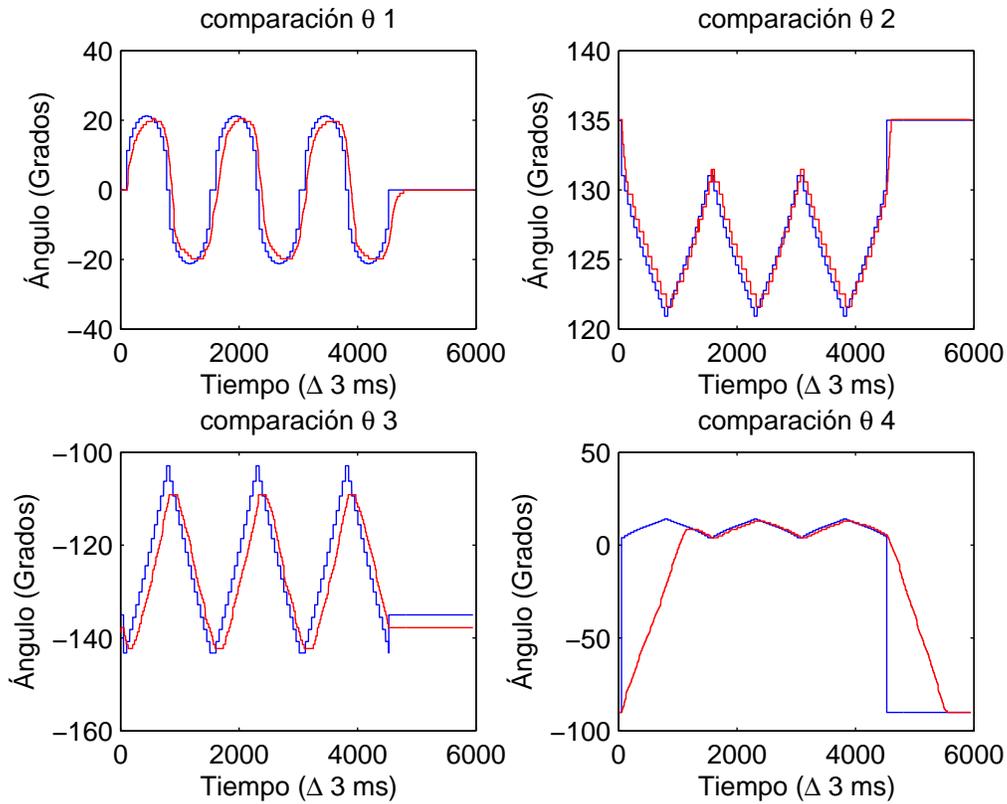


Figura 4.32: Posición articular obtenida (PD+G) con peso

Como puede apreciarse el seguimiento es aceptable. Las articulaciones que obtuvieron una desviación mayor con respecto a sus consignas fueron la articulación 1 y 3.

En la figura 4.33, se muestra la señal de error articular, cuya amplitud puede apreciarse en unidades de grados. Como puede apreciarse aun con la presencia de carga, las articulaciones que se ven afectadas por la compensación del controlador presentan un error pequeño en comparación con la articulación 1, en la cuál la señal de compensación no actúa.

4. RESULTADOS Y DISCUSIÓN

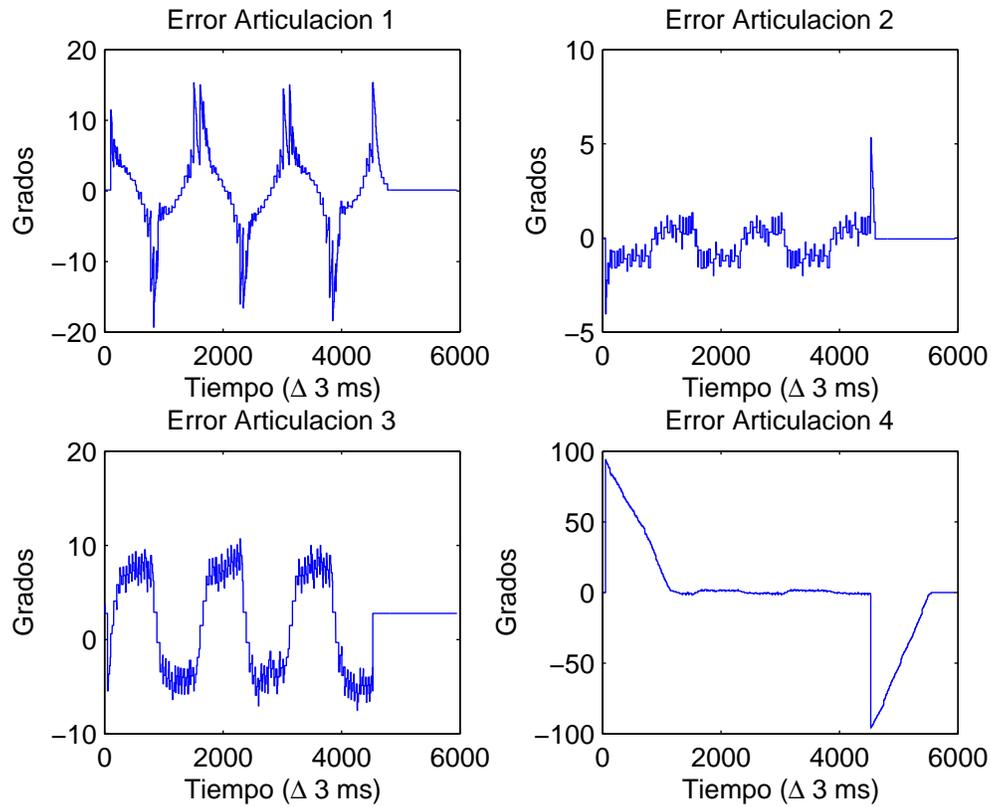


Figura 4.33: Señal de error articular (PD+G) con peso

La figura 4.34, muestra la señal de control requerida para alcanzar la posición deseada, como puede apreciarse debido a que para la articulación 1 se tiene solamente un controlador de tipo PD, presenta una señal de control con saturación, mientras que las articulaciones 2,3 y 4 en las que actúa la compensación de gravedad, se obtuvieron señales de control sin saturación.

4. RESULTADOS Y DISCUSIÓN

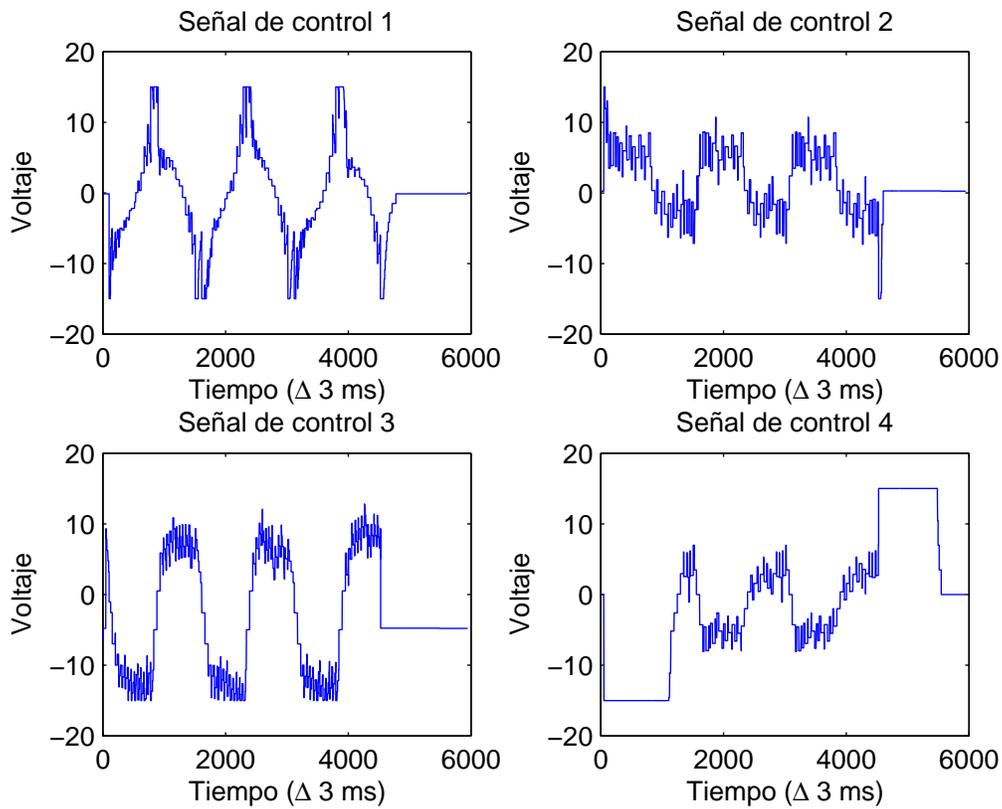


Figura 4.34: Señal de control (PD+G) con peso

Finalmente en la figura 4.35, se muestra la posición del efector final obtenida al realizar la cinemática inversa con respecto a la posiciones articulares generadas.

4. RESULTADOS Y DISCUSIÓN

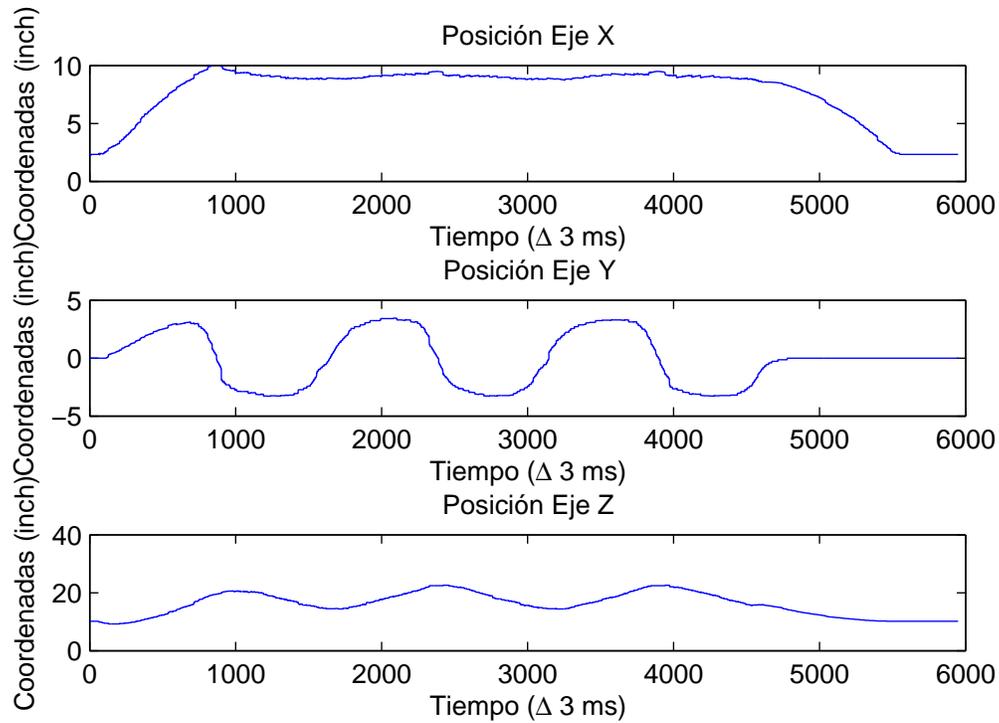


Figura 4.35: Posición final obtenida (PD+G) con peso

Una vez finalizadas las pruebas y con los resultados obtenidos, se realizó una comparación entre los diferentes controladores que fueron caso de estudio en esta tesis.

se presentan dos figuras, en las cuales se muestran las señales obtenidas por cada controlador. Dichas figuras se encuentran divididas en 4 cuadrantes, donde cada cuadrante muestra la señal generada por los controladores en el siguiente orden: PD+I, PD+W, PD+W (adaptable) y PD+G.

En la figura 4.36, se muestran las señales de error generadas por cada uno de los controladores.

4. RESULTADOS Y DISCUSIÓN

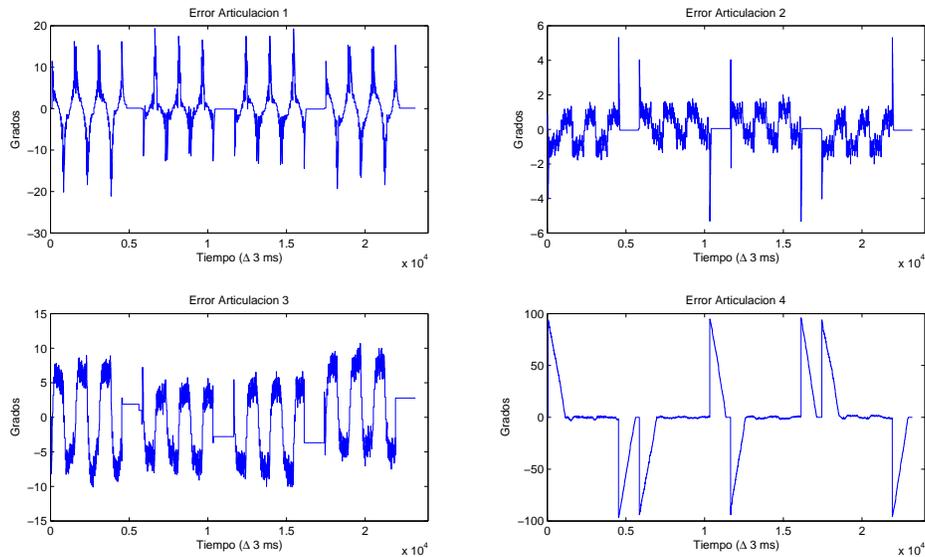


Figura 4.36: Comparativa de error para los controladores en vacío.

Como puede apreciarse, aunque el controlador PD+I, presenta un buen rendimiento, las variaciones del controlador PD, presentaron un mayor rendimiento. En este punto se, observó que el controlador que sigue presentando una menor desviación con respecto al punto de consigna fue el controlador PD+W.

En la figura 4.37, se muestran las señales de control generadas por cada controlador.

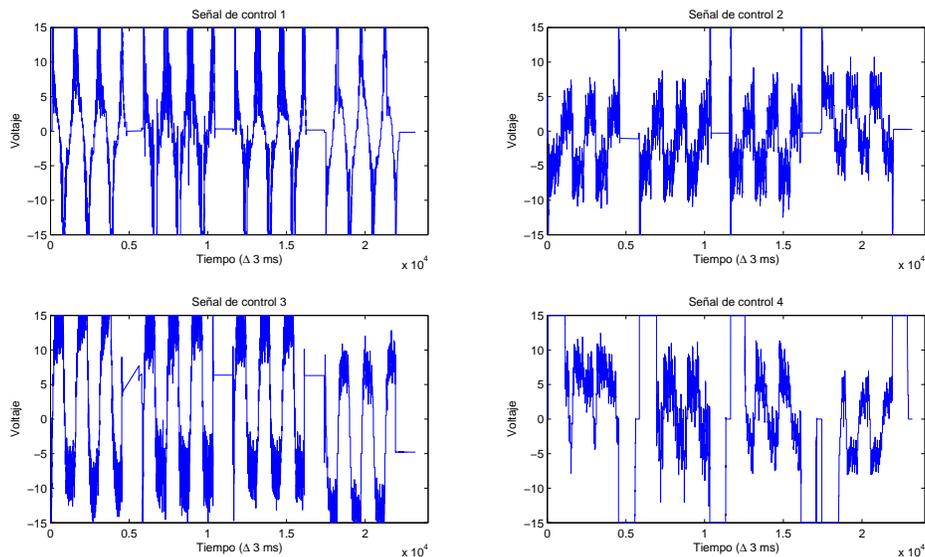


Figura 4.37: Comparativa de señal de control para los controladores en vacío.

Como puede apreciarse, las señales del controlador PD + W, en ambas modalidades; son las señales que presentan mas ruido, lo cual podría llegar a desgastar más

rápidamente las componentes del sistema. Mientras que el controlador PD+G obtuvo las señales de control más limpias. En ambos casos, tanto en vacío como en carga, el controlador PD+G presenta una señal de control sin saturación y con menos ruido.

Las conclusiones obtenidas son presentadas en la siguiente sección.

4.3. Discusión

El uso de un sistema de simulación virtual 3D para el caso de un manipulador serial, permite realizar el análisis de comportamiento fuera de línea, ahorrando costos y tiempo en la industria, mientras que en la rama académica proporciona una potente herramienta para el estudio de sistemas dinámicos.

En esta tesis se presentó el desarrollo de una interfaz gráfica y la implementación de variaciones de controladores PD más utilizados actualmente. La interfaz fue desarrollada bajo el lenguaje Java, lo cual permite que sea implementado como interfaz de control o podría llegar a ser utilizada como laboratorio virtual. Expandiendo sus capacidades tanto en lo académico como en la industria al poder manipular al controlador a distancia a través de internet.

El uso de ecuaciones de cinemática, permite obtener las configuraciones requeridas por el manipulador con el fin de alcanzar un punto en el espacio, o viceversa.

Finalmente con la implementación de los controladores, se pudo apreciar que aún con la existencia de algoritmos cada vez más sofisticados que incrementan las operaciones de computo, el hacer uso de un simple controlador tipo PID, sigue siendo una solución viable para el problema de seguimiento de trayectoria de un manipulador del tipo tratado en esta tesis, obteniendo resultados satisfactorios cuando la velocidad de cambio en la trayectoria es relativamente baja.

Con los resultados obtenidos, se demostró que el uso de controladores PID tendrán un tiempo de vida muy grande en la industria, debido a sus resultados obtenidos y su simplicidad de implementación. Sin embargo, los controladores presentados en esta tesis mostraron ventajas con respecto al controlador PID, al momento de efectuar movimientos en presencia de cargas, siendo en este caso los controladores PD+W y el controlador PD+G los que presentaron mejores resultados, finalmente se pudo apreciar que el controlador PD+G genera una señal de control más estable mientras que a través del controlador PD+W se obtuvieron señales de control con mayor ruido lo cual puede verse reflejado en el desgaste prematuro de los componentes del sistema.

Bibliografía

- Akan B., B. Curuklu, G. Spampinato, and L. Asplund. 2010. Towards robust human robot collaboration in industrial environments, *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on* 71-72.
- Alavandar S., and M.J. Nigam. 2009. *New hybrid adaptative neuro-fuzzy algorithms for manipulator control with uncertainties - Comparative study*. ISA Transactions 48 497-502.
- Andres, J., L. Gracia, and J. Tornero. 2009. Inverse kinematics of a redundant manipulator for cam integration. An industrial perspective of implementation, *Mechatronics, 2009. ICM 2009. IEEE International Conference on*, 1-6.
- Bao, X., W. Peng, X. Yin, X. Fang, and H. Zhang. 2009. With a human touch - how robots have learned to polish. *ABB Review special report*.
- Behrang M.A., E. Assareh, A. Ghanbarzadeh, and A.R. Noghrehabadi. 2010. *The potential of different artificial neural network (ANN) techniques in daily global solar radiation modeling based on meteorological data*, Solar Energy 84 (8) 1468-1480.
- Blomdell A., G. Bolmsjo, T. Brogardh, P. Cederberg, M. Isaksson, R. Johansson, M. Haage, K. Nilsson, M. Olsson, A. Robertsson, and J. Wang. 2005. Extending an industrial root controller implementation and applications of a fast open sensor interface. *IEEE Robotics & Automation Magazine*, 12 (3) 1070-9932.
- Caccavale, F., P. Cilibrizzi, F. Pierri, and L. Villani. 2009. *Actuators fault diagnosis for robot manipulators with uncertain model*. Control Engineering Practice. 17(1) 146-157.
- Chaoui, H., P. Sicard, and Gueaieb. 2009. *W. ANN-Based Adaptive Control of Robotic Manipulators With Friction and Joint Elasticity*. IEEE Transactions on Industrial Electronics. 56(8) 3174-3187.
- Chen K.Z., Y. Leung, and K.S. Leung. 2002. *A neural network for solving nonlinear programming problems*, Neural Computing and Applications 11. 103-111.
- Daeinabi K., and M. Teshnehlab. 2007. Industrial Arc Welding Robot Defect Tracking System in Automotive Industry, *Mechatronics and Automation, 2007. ICMA 2007. International Conference on* 3937-3941.

- Fauadi H.F., and M.S. Jumali. 2008. Modeling and simulation of programmable universal machine for assembly (PUMA) industrial robot for automotive-related assembly process, *Information Technology* 3. 26-28.
- Gao X.B. 2004. *A novel neural network for nonlinear convex programming*, *IEEE Transactions on Neural Networks* 15 (3) 613-621.
- Hazel B. and J. Cote. 2010. Robotic penstock welding, *Applied Robotics for the Power Industry (CARPI), 2010 1st International Conference on* 1-2.
- Guo Y., and T. Hu. 1990. *Neural network for multiobjective dynamic programming*, *Acta Electronica Sinica* 27 (10) 70-72.
- Hasan A.T, and N. Ismail. 2010. *Artificial neural network-based kinematics Jacobian solution for serial manipulator passing through singular configurations*, *Advances in Engineering Software* 41 359-367.
- Hasan A.T, AMS. Hamouda, N. Ismail, and HMAA. Al-Assadi. 2007. *A new adaptive learning algorithm for robot manipulator control*. *Proc I Mech E, Part I: J Syst ControlEng* 221 (4) 663-72.
- IFR(International Federation of Robotics) departamento de estadística. 2010. *The robotics industry is getting back on track! Sales slump in 2009 - Strong recovery in 2010 - Further growth expected in 2011 and 2012*. World Robotics 2009 Industrial Robots. First News. <http://www.worldrobotics.org>.
- Jose A. Ramirez, I. Cervantes, and R. Kelly. 2000. *PID regulation of robot manipulators: stability and performance*. *Systems & Control Letters*. 41(2) 73-83
- Karami, A., H. Karimi, P. Maralani, and B. Moshiri. 2008. *Intelligent optimal control of robotic manipulators using wavelets*. *international journal of wavelets, multiresolution & information processing*. 6(4) 575-592.
- Kennedy M.P., and L.O. Chua. 1998. *Neural network for nonlinear programming*, *IEEE Transaction on Circuits and Systems* 35 554-562.
- Kosa G., M. Shoham, and M. Zaaroor. 2005. Propulsion of a Swimming Micro Medical Robot, *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on* 1327-1331.
- Kucuk S. and Z. Bingul. 2004. The inverse kinematics solutions of industrial robot manipulators, *Mechatronics, 2004. ICM '04. Proceedings of the IEEE International Conference on*.274-279.
- Liangyong W., C. Tianyou, and Z. Lianfei. 2009. *Neural-Network-Based Terminal Sliding-Mode Control of Robotic Manipulators Including Actuator Dynamics*. *IEEE Transactions on Industrial Electronics*. 56(9) 3296-3304.
- Lin J., and R. Lian. 2009. *Hybrid fuzzy-logic and neural-network controller for MIMO systems*. *Mechatronics*. 19(6) 972-986.

- Merriam-Webster. 2006. *Diccionario Merriam-Webster*.
- Moon J.W, and k. Jong-Jin. 2010. *ANN-based thermal control models for residential buildings*, Building and Environment 45 1612-1625
- Ogata, K. 1998. *Ingeniería de control moderna*, Editorial Prentice Hall. Tercera edición.
- Owena, W.S., E. A. Croftb, and B. Benhabiba 2008. *A multi-arm robotic system for optimal sculpting*, Robotics and Computer-Integrated Manufacturing 24 92-104.
- Qinjun D. 2008. Study on distributed control system for a medical robot, *Automation and Logistics, 2008. ICAL 2008. IEEE International Conference on* 1678-1682.
- RAE 2010. *Diccionario de la real academia española*.
- Radaschin A. 2010. Sliding mode control of an autonomous welding robot, *Automation Quality and Testing Robotics (AQTR), IEEE International Conference on* 1. 1-5.
- Rocco P. 1996. Robotics and Automation, IEEE Transactions on , *IEEE Robotics and Automation Society* 12(4) 606-614.
- Sadati, N., and R. Ghadami. 2008. *Adaptive multi-model sliding mode control of robotic manipulators using soft computing*. Neurocomputing. 71 13-15 2702-2710.
- SEDECO (Secretaria de Desarrollo Económico). *Economía con Equidad. Capital en crecimiento*. Número 6. Marzo-Abril de 2009. www.sedeco.df.gob.mx 2345.
- Shi J., H. Liuab, and N. Bajcincac. 2008. *Robust control of robotic manipulators based on integral sliding mode.*, Internacional Journal of control. 81 (10) 1537-1548.
- Siljak, D.D., 1991. *Decentralized control of complex systems.*, San diego, CA: Academic.
- Solteiro E.J., P.B. de Moura, and J.A. Tenreiro. 2007. *Manipulator trajectory planning using a MOGA*, Applied Soft Computing.
- Tae-Yong K., and W. Han. Montemayor. 2000. *An adaptive PID learning control of robot manipulatorsq*, Automatica 36 (2000) 717-725.
- Tang Y., M. Tomizuka, G. Guerrero, and G. Montemayor. 2000. *Decentralized robust control of mechanical systems*, IEEE Transaction on Automatic Control. 45(4) 771-776.
- Tank D.W., and J.J. Hopfield. 1986. *Simple neural optimization networks: An A/D convert, siginal decision circuit, and a linear programming circuit*, IEEE Transactions on Circuits and Systems 33 533-541.
- Ting Y., G. Yuxian, and L. Huiying. 2010. Development of a Robot System for Pipe Welding, *Measuring Technology and Mechatronics Automation (ICMTMA), International Conference on*. 1 1109-1112.

- Tomei, P. 1991. Adaptive PD controller for robot manipulators, *M Robotics and Automation, IEEE Transactions on*. 7(4) 556-570.
- Troncoso, R. 2007. *Electrónica digital y lógica programable*, Ed. Universidad de Guanajuato. Primera edición.
- Wallén J., Norrlöf and S. Gunnarsson. 2008. *Arm-side evaluation of ILC applied to a six-degrees-of-freedom industrial robot*, International federation of automatic control. Proceedings of 17th IFAC World Congress 2008, Seoul, Korea 13450-13455.
- Xia Y.S. and J. Wang. 1998. *A general methodology for designing globally convergent optimizatin neural networks*, *IEEE Transactions on Neural Networks* 9 (6) 1331-1343.
- Yangmin L. and X. Qingsong. 2007. Design and Development of a Medical Parallel Robot for Cardiopulmonary Resuscitation, *Mechatronics, IEEE/ASME Transactions on* 12 (3) 265-273.
- Zhang S. and A.G. Constantinides. 1992. *Lagrange programming neural networks*, *IEEE Transactions on Circuits and Systems* 39 (7) 441-452.
- Zuo, Y., Y. Wang, X. Liu, S. Yang, L. Huang, and X. Wu. 2010. *Neural network robust tracking control strategy for robot manipulators*. *Applied Mathematical Modelling*. 34(7) 1823-1838.

Capítulo 5

ANEXOS

5.1. Control PD+I

```
public int controladorPD(int nArt,float kp,float kd,float ki){

    float de=0.0f;
    float Int=0.0f;
    float control=0.0f;
    e[nArt] = this.pd[nArt]-pos[nArt];
    de = (pos[nArt] - ep[nArt])/Tm;
    Inte[nArt] =Inte[nArt] + e[nArt]*Tm;
    this.ep[nArt] = pos[nArt];
    System.out.println("" + nArt + ": \t" + e[nArt]*kp + "\t" + de*kd + "\t"
        + Inte[nArt]*ki);
    control = e[nArt]*kp + de*kd + Inte[nArt]*ki ; //Controlador PID

    if (e[nArt]<0){ //Indicamos la direccion de giro del motor
    dir[nArt] = 8;
    }else{
    dir[nArt]=0;
    }

    if (control > vo){ //Normalizamos la señal de control
        control = vo;
    }
    if (control < -vo){
        control = -vo;
    }

    Scontrol[nArt]=control; //Almacenamos la señal de control de la junta

    control=java.lang.Math.abs(control);
    int pwmC = (int) java.lang.Math.round(control*7/vo);
```

```
    pwmC = dir[nArt] | pwmC;
    return pwmC;
}
```

5.2. Control PD+W

```
public int controladorPDW(int nArt,float k,float lambda,float delta,float epsilon)

    float de=0.0f;
    float s=0.0f;
    float abss=0.0f;
    float S=0.0f;
    float w=0.0f;
    float deltaS=0.0f;
    float control=0.0f;
    e[nArt] = pos[nArt]-pd[nArt];
    de = (e[nArt]-ep[nArt])/Tm;
    ep[nArt] = e[nArt];
    s = de + lambda*e[nArt];
    abss = java.lang.Math.abs(s);
    S = 1 + abss;
    deltaS=delta*S;
    w=(float) (-(deltaS*deltaS)*(s/(deltaS*abss + epsilon)));
    control = -k*s + w; //Controlador PID

    if (e[nArt]<0){ //Indicamos la direccion de giro del motor
    dir[nArt] = 0;
    }else{
    dir[nArt]=8;
    }

    if (control > vo){ //Normalizamos la señal de control
        control = vo;
    }
    if (control < -vo){
        control = -vo;
    }

    Scontrol[nArt]=control; //Almacenamos la señal de control de la junta

    control=java.lang.Math.abs(control);
    int pwmC = (int) java.lang.Math.round(control*7/vo);
    pwmC = dir[nArt] | pwmC;
    return pwmC;
```

```
}
```

5.3. Control PD+W (adaptable)

```
public int controladorPDWadap(int nArt,float k,float lambda,
float epsilon){

    float de=0.0f;
    float s=0.0f;
    float abss=0.0f;
    float S=0.0f;
    float w=0.0f;
    float deltaS=0.0f;
    float control=0.0f;
    e[nArt] = pos[nArt]-pd[nArt];
    de = (e[nArt]-ep[nArt])/Tm;
    ep[nArt] = e[nArt];
    s = de + lambda*e[nArt];
    abss = java.lang.Math.abs(s);
    S = 1 + abss;
    //System.out.println(nArt + ": " + e[nArt] + ": " + s +
    //": " + S + ": " + delta[nArt]);
    ///adaptabilidad
    delta[nArt]=-sigma[nArt]*delta[nArt] + S*abss*gamma[nArt];
    //System.out.println(nArt + ": " + -sigma[nArt]*delta[nArt] +
    ": " + gamma[nArt]*S*abss);// +
    //      ": " + S);
    //System.out.println(nArt + ": " + delta[nArt]);
    deltaS=delta[nArt]*S;
    w=(float) (-(deltaS*deltaS)*(s/(deltaS*abss + epsilon)));
    control = -k*s + w; //Controlador PID
    //System.out.println(e[nArt] + " " + de*kd + " " + control +
    //" " + pwmC);
    //System.out.println(nArt + ": " + w);
    System.out.println(nArt + ": " + delta[nArt]);

    if (e[nArt]<0){ //Indicamos la direccion de giro del motor
    dir[nArt] = 0;
    }else{
    dir[nArt]=8;
    }

    if (control > vo){ //Normalizamos la señal de control
        control = vo;
    }
}
```

```
    }
    if (control < -vo){
        control = -vo;
    }

    Scontrol[nArt]=control; //Almacenamos la señal de control de la junta

    control=java.lang.Math.abs(control);
    int pwmC = (int) java.lang.Math.round(control*7/vo);
    pwmC = dir[nArt] | pwmC;
    return pwmC;
}
```

5.4. Control PD+G

```
public int controladorPDG(int nArt,float kp,float kd){

    float de=0.0f;
    float Int=0.0f;
    float control=0.0f;
    float G=0.0f;
    float Y=0.0f;
    //pos [nArt]=myPort.getPos(nArt);
    e[nArt] = this.pd[nArt]-pos[nArt];
    de = (e[nArt]-ep[nArt])/Tm;
    Inte[nArt] =Inte[nArt] + e[nArt]*Tm;
    this.ep[nArt] = e[nArt];
    //adaptacion

    float teta2=(float) (pd[1] * gTor);
    float teta3=(float) (pd[2] * gTor);
    float teta4=(float) (pd[3] * gTor);
    //Calculo de parametros de posición
    float c2 = (float) (java.lang.Math.cos(teta2));
    float c23 = (float) (java.lang.Math.cos(teta2 + teta3));
    float c234 = (float) (java.lang.Math.cos(teta2 + teta3 + teta4));

    switch(nArt){
        case 0 : Y= 0;          break;
        case 1 : Y= c2 + c23 + c234;break;
        case 2 : Y= c23 + c234; break;
        default : Y= c234;      break;
    }
}
```

```
ThetaParam[nArt] = -beta[nArt]*Y*(gammaG[nArt]*de+
(2*e[nArt]/(1+2*(e[nArt]*e[nArt]))));

G = Y*ThetaParam[nArt];
//System.out.println(G);
control = e[nArt]*kp + de*kd + G; //Controlador PID

if (e[nArt]<0){ //Indicamos la direccion de giro del motor
dir[nArt] = 8;
}else{
dir[nArt]=0;
}

if (control > vo){ //Normalizamos la señal de control
control = vo;
}
if (control < -vo){
control = -vo;
}

Scontrol[nArt]=control; //Almacenamos la señal de control de la junta

control=java.lang.Math.abs(control);
int pwmC = (int) java.lang.Math.round(control*7/vo);
//System.out.println(e + " " + ep + " " + (ep-e));
//System.out.println(e[nArt] + " " + de*kd + " " + control +
//" " + pwmC);
pwmC = dir[nArt] | pwmC;
return pwmC;
}
```

5.5. Puente H (L298n)



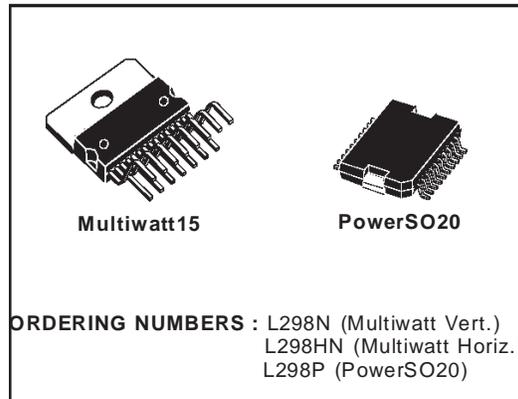
L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

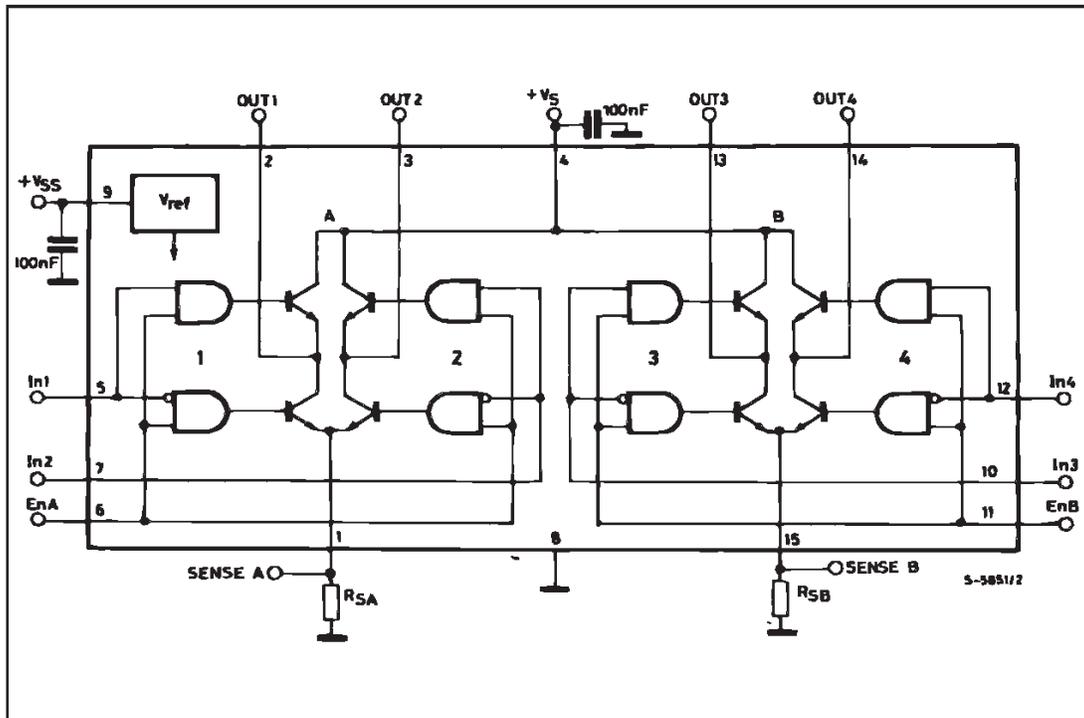
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM

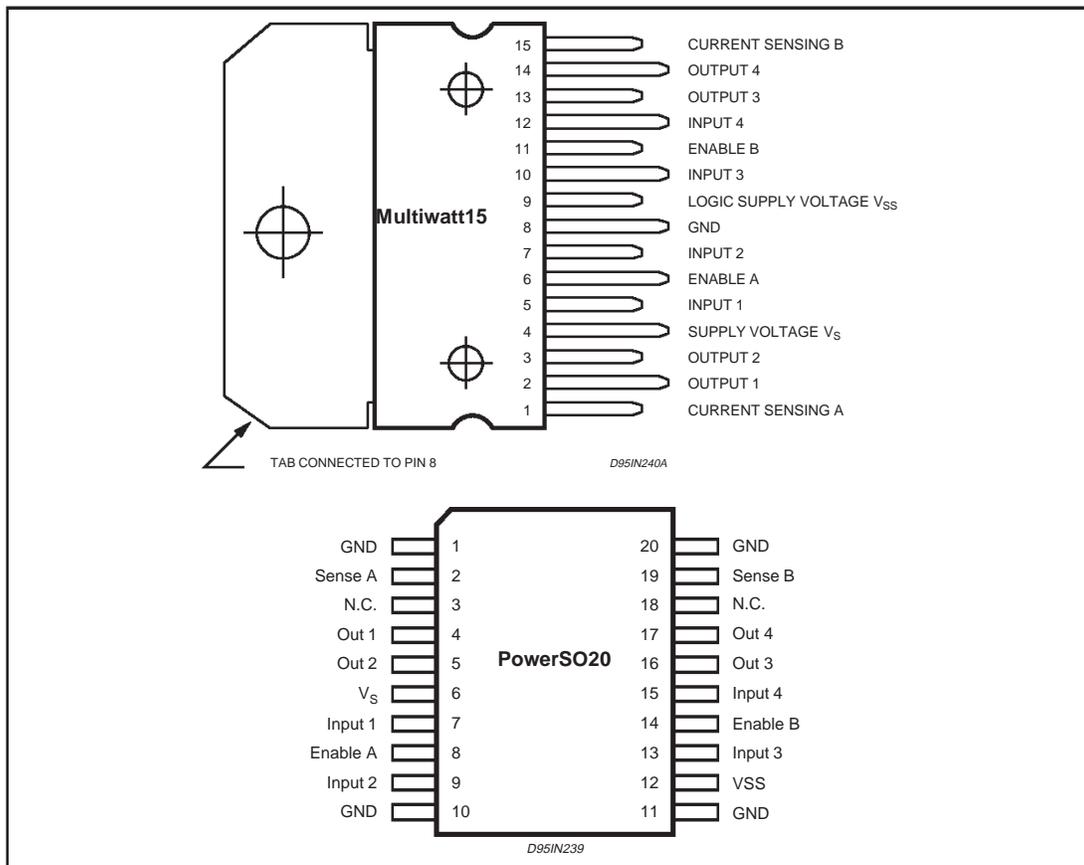


L298

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_I, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive (t = 100 μ s)	3	A
	- Repetitive (80% on -20% off; t _{on} = 10ms)	2.5	A
	-DC Operation	2	A
V_{sens}	Sensing Voltage	-1 to 2.3	V
P_{tot}	Total Power Dissipation (T _{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	-25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	-40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter	PowerSO20	Multiwatt15	Unit
$R_{th\ j-case}$	Thermal Resistance Junction-case	Max. -	3	°C/W
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient	Max. 13 (*)	35	°C/W

(*) Mounted on aluminum substrate

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	V _{SS}	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V _S	Supply Voltage (pin 4)	Operative Condition	V _{IH} +2.5		46	V
V _{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I _S	Quiescent Supply Current (pin 4)	V _{en} = H; I _L = 0	V _i = L	13	22	mA
			V _i = H	50	70	mA
		V _{en} = L	V _i = X		4	mA
I _{SS}	Quiescent Current from V _{SS} (pin 9)	V _{en} = H; I _L = 0	V _i = L	24	36	mA
			V _i = H	7	12	mA
		V _{en} = L	V _i = X		6	mA
V _{iL}	Input Low Voltage (pins 5, 7, 10, 12)		-0.3		1.5	V
V _{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		V _{SS}	V
I _{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V _i = L			-10	μA
I _{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	V _i = H ≤ V _{SS} -0.6V		30	100	μA
V _{en} = L	Enable Low Voltage (pins 6, 11)		-0.3		1.5	V
V _{en} = H	Enable High Voltage (pins 6, 11)		2.3		V _{SS}	V
I _{en} = L	Low Voltage Enable Current (pins 6, 11)	V _{en} = L			-10	μA
I _{en} = H	High Voltage Enable Current (pins 6, 11)	V _{en} = H ≤ V _{SS} -0.6V		30	100	μA
V _{CEsat(H)}	Source Saturation Voltage	I _L = 1A	0.95	1.35	1.7	V
		I _L = 2A		2	2.7	V
V _{CEsat(L)}	Sink Saturation Voltage	I _L = 1A (5)	0.85	1.2	1.6	V
		I _L = 2A (5)		1.7	2.3	V
V _{CEsat}	Total Drop	I _L = 1A (5)	1.80		3.2	V
		I _L = 2A (5)			4.9	V
V _{sens}	Sensing Voltage (pins 1, 15)		-1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T ₁ (V _i)	Source Current Turn-off Delay	0.5 V _i to 0.9 I _L (2); (4)		1.5		μs
T ₂ (V _i)	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		0.2		μs
T ₃ (V _i)	Source Current Turn-on Delay	0.5 V _i to 0.1 I _L (2); (4)		2		μs
T ₄ (V _i)	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.7		μs
T ₅ (V _i)	Sink Current Turn-off Delay	0.5 V _i to 0.9 I _L (3); (4)		0.7		μs
T ₆ (V _i)	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.25		μs
T ₇ (V _i)	Sink Current Turn-on Delay	0.5 V _i to 0.9 I _L (3); (4)		1.6		μs
T ₈ (V _i)	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.2		μs
f _c (V _i)	Commutation Frequency	I _L = 2A		25	40	KHz
T ₁ (V _{en})	Source Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (2); (4)		3		μs
T ₂ (V _{en})	Source Current Fall Time	0.9 I _L to 0.1 I _L (2); (4)		1		μs
T ₃ (V _{en})	Source Current Turn-on Delay	0.5 V _{en} to 0.1 I _L (2); (4)		0.3		μs
T ₄ (V _{en})	Source Current Rise Time	0.1 I _L to 0.9 I _L (2); (4)		0.4		μs
T ₅ (V _{en})	Sink Current Turn-off Delay	0.5 V _{en} to 0.9 I _L (3); (4)		2.2		μs
T ₆ (V _{en})	Sink Current Fall Time	0.9 I _L to 0.1 I _L (3); (4)		0.35		μs
T ₇ (V _{en})	Sink Current Turn-on Delay	0.5 V _{en} to 0.9 I _L (3); (4)		0.25		μs
T ₈ (V _{en})	Sink Current Rise Time	0.1 I _L to 0.9 I _L (3); (4)		0.1		μs

- 1) 1)Sensing voltage can be -1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ -0.5 V.
- 2) See fig. 2.
- 3) See fig. 4.
- 4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

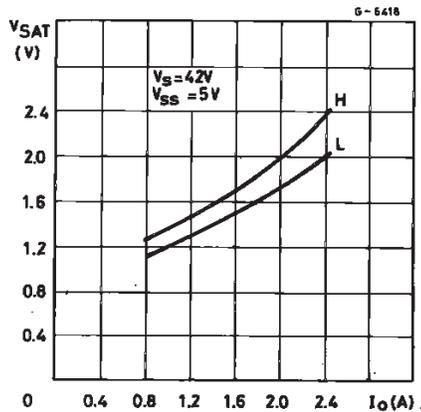
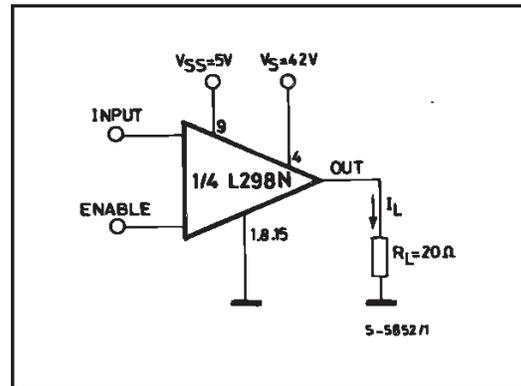


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

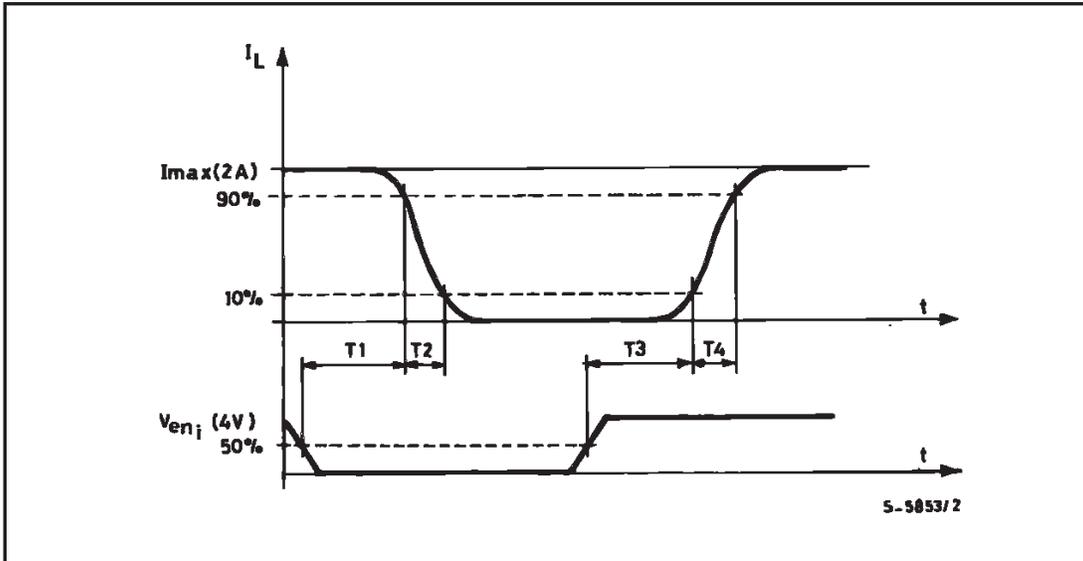
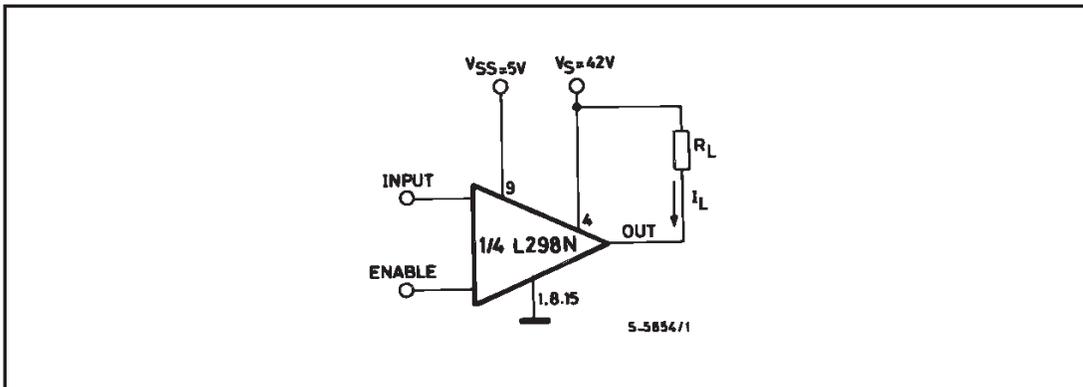


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
 For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

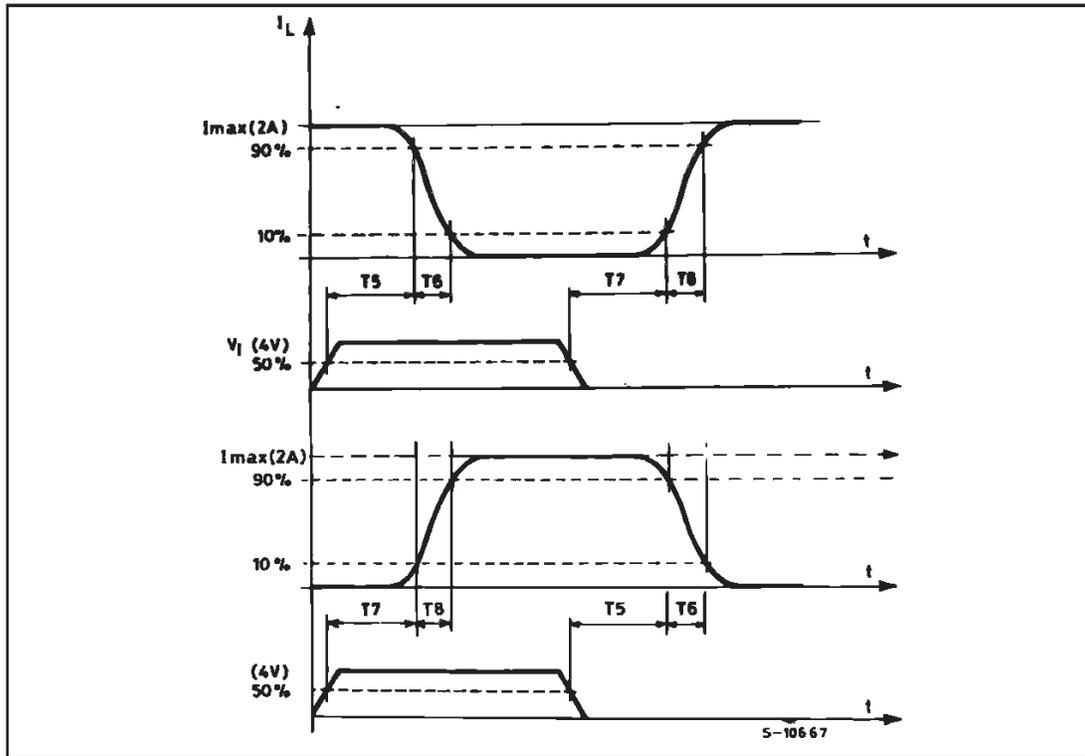


Figure 6 : Bidirectional DC Motor Control.

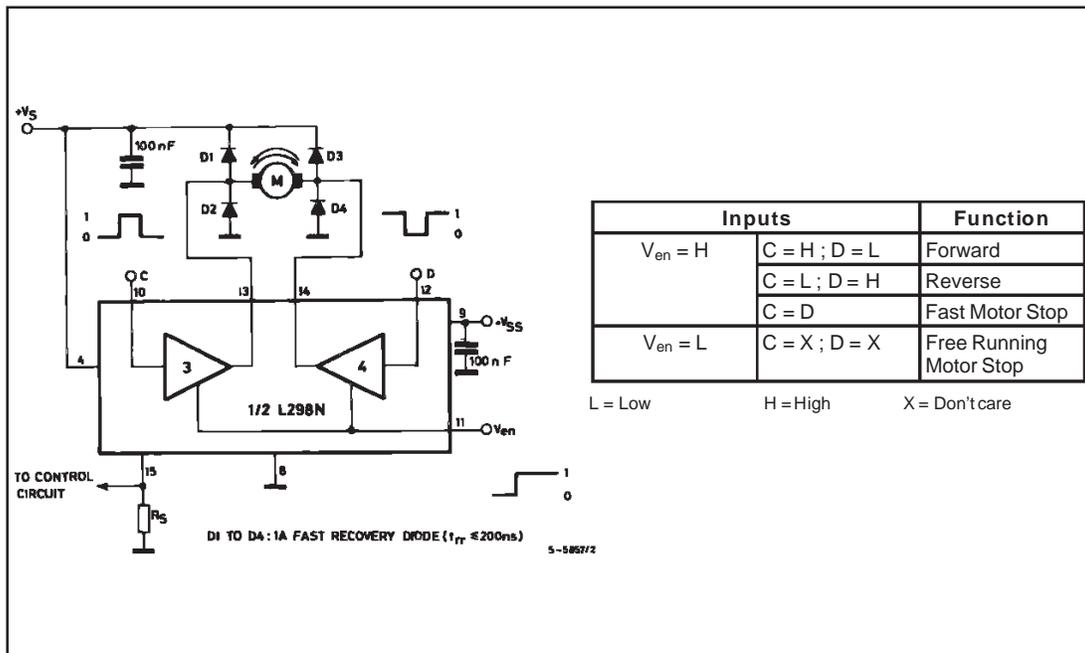
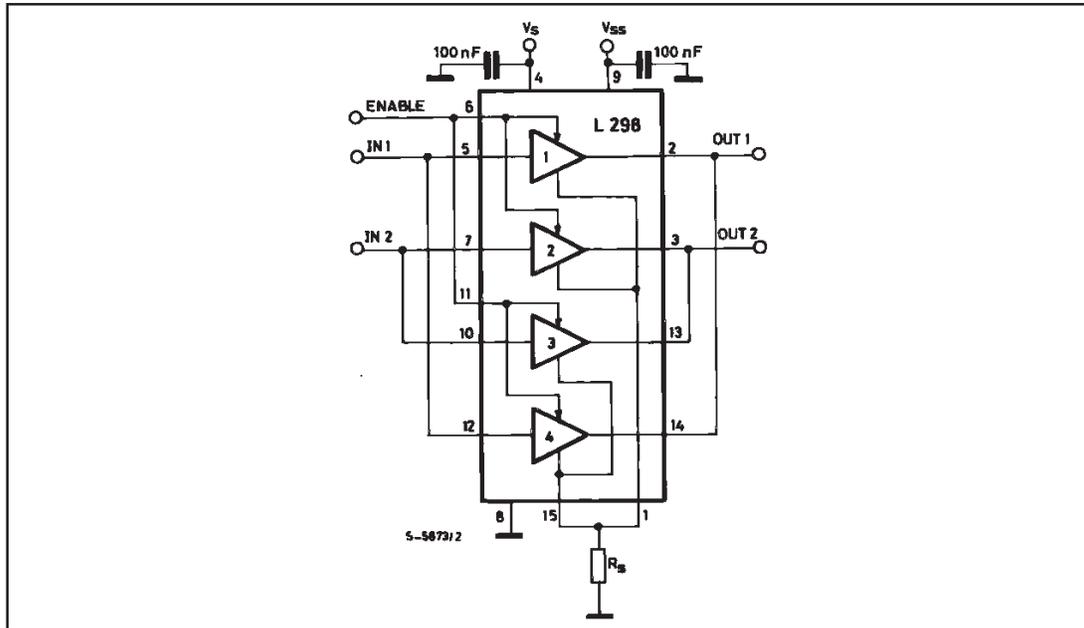


Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output: an external resistor (R_{SA} ; R_{SB} .) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are $In1$; $In2$; EnA and $In3$; $In4$; EnB . The In inputs set the bridge state when The En input is high; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both V_s and V_{ss} , to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of V_s that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turnit OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes $D1$ to $D4$ is made by four fast recovery elements ($trr \leq 200$ nsec) that must be chosen of a V_F as low as possible at the worst case of the load current.

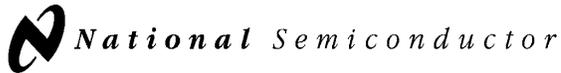
The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Schottky diodes would be preferred.

5.6. Compuerta Not (74ls14)



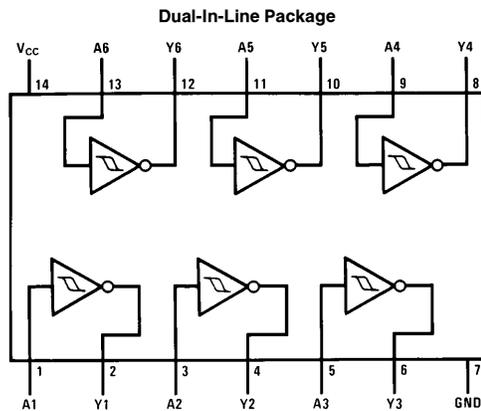
June 1989

54LS14/DM74LS14 Hex Inverters with Schmitt Trigger Inputs

General Description

This device contains six independent gates each of which performs the logic INVERT function. Each input has hysteresis which increases the noise immunity and transforms a slowly changing input signal to a fast changing, jitter free output.

Connection Diagram



TL/F/6353-1

Order Number 54LS14DMQB, 54LS14FMQB,
54LS14LMQB, DM74LS14M or DM74LS14N
See NS Package Number E20A, J14A, M14A, N14A or W14B

Function Table

$$Y = \bar{A}$$

Input	Output
A	Y
L	H
H	L

H = High Logic Level
L = Low Logic Level

54LS14/DM74LS14 Hex Inverters with Schmitt Trigger Inputs

Absolute Maximum Ratings (Note)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	
54LS	−55°C to +125°C
DM74LS	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the "Electrical Characteristics" table are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	54LS14			DM74LS14			Units
		Min	Nom	Max	Min	Nom	Max	
V _{CC}	Supply Voltage	4.5	5	5.5	4.75	5	5.25	V
V _{T+}	Positive-Going Input Threshold Voltage (Note 1)	1.5	1.6	2.0	1.4	1.6	1.9	V
V _{T−}	Negative-Going Input Threshold Voltage (Note 1)	0.6	0.8	1.1	0.5	0.8	1	V
HYS	Input Hysteresis (Note 1)	0.4	0.8		0.4	0.8		V
I _{OH}	High Level Output Current			−0.4			−0.4	mA
I _{OL}	Low Level Output Current			4			8	mA
T _A	Free Air Operating Temperature	−55		125	0		70	°C

Electrical Characteristics over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V _I	Input Clamp Voltage	V _{CC} = Min, I _I = −18 mA			−1.5	V
V _{OH}	High Level Output Voltage	V _{CC} = Min, I _{OH} = Max V _{IL} = Max	54LS	2.5	3.4	V
			DM74	2.7	3.4	
V _{OL}	Low Level Output Voltage	V _{CC} = Min, I _{OL} = Max V _{IH} = Min	54LS		0.25	V
			DM74		0.35	
		V _{CC} = Min, I _{OL} = 4 mA	DM74		0.25	
I _{T+}	Input Current at Positive-Going Threshold	V _{CC} = 5V, V _I = V _{T+}	DM74		−0.14	mA
I _{T−}	Input Current at Negative-Going Threshold	V _{CC} = 5V, V _I = V _{T−}	DM74		−0.18	mA
I _I	Input Current @ Max Input Voltage	V _{CC} = Max, V _I = 7V	DM74		0.1	mA
		V _{CC} = Max, V _I = 10.0V	54LS			
I _{IH}	High Level Input Current	V _{CC} = Max, V _I = 2.7V			20	μA
I _{IL}	Low Level Input Current	V _{CC} = Max, V _I = 0.4V			−0.4	mA
I _{OS}	Short Circuit Output Current	V _{CC} = Max (Note 3)	54LS	−20	−100	mA
			DM74	−20	−100	
I _{CCH}	Supply Current with Outputs High	V _{CC} = Max		8.6	16	mA
I _{CCL}	Supply Current with Outputs Low	V _{CC} = Max		12	21	mA

Note 1: V_{CC} = 5V.

Note 2: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

Switching Characteristics at $V_{CC} = 5V$ and $T_A = 25^\circ C$ (See Section 1 for Test Waveforms and Output Load)

Symbol	Parameter	$R_L = 2\text{ k}\Omega$				Units
		$C_L = 15\text{ pF}$		$C_L = 50\text{ pF}$		
		Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time Low to High Level Output	5	22	8	25	ns
t_{PHL}	Propagation Delay Time High to Low Level Output	5	22	10	33	ns