



U

P

T

“ANÁLISIS, DISEÑO, CONSTRUCCIÓN Y CONTROL
DE UN ROBOT PARALELO TIPO DELTA DE TRES
GRADOS DE LIBERTAD”

Tesis que presenta

JONATAN MARTÍN ESCORCIA HERNÁNDEZ

Para obtener el grado de:

MAESTRO EN AUTOMATIZACIÓN Y CONTROL

Director de Tesis:

Dr. Omar Aguilar Mejía

Tulancingo de Bravo, Hidalgo

OCTUBRE 2016

©UPT 2016

Derechos reservados

El autor otorga a UPT el permiso de reproducir
y distribuir copias de esta tesis en su totalidad
o en partes.





U

P

T

“ANÁLISIS, DISEÑO, CONSTRUCCIÓN Y CONTROL
DE UN ROBOT PARALELO TIPO DELTA DE TRES
GRADOS DE LIBERTAD”

Tesis de Maestría en
Automatización y Control

Por:

JONATAN MARTÍN ESCORCIA HERNÁNDEZ

Ingeniero en Robótica

Universidad Politécnica de Tulancingo 2010-2013

Becario de CONACYT, expediente no. 5938804

Director de Tesis:

Dr. Omar Aguilar Mejía

Tulancingo de Bravo, Hidalgo, octubre 2016



Dedicatoria

*Dedico esta tesis a Dios quien me ha dado la vida y ha estado conmigo en todo tiempo
dándome fuerza, sabiduría, e inteligencia para seguir adelante.
A mis padres Martín Escorcía Portillo y Rosalba Hernández Ramírez quienes me han
educado e instruido con amor, afecto y paciencia.*

*¡Adquirir sabiduría es lo más sabio que puedes hacer!
Y en todo lo demás que hagas, desarrolla buen juicio.
Engrandécela, y ella te engrandecerá;
Si valoras la sabiduría, ella te engrandecerá.
Abrázala, y te honrará.*

Proverbios 4:7-8

Agradecimientos

Primeramente quiero darle las gracias a Dios por la vida que me ha dado, porque Él siempre ha estado conmigo todo tiempo para instruirme, corregirme a fin de tomar las mejores decisiones en la vida; además de permitirme concluir una etapa más en mi vida.

A mis padres Martín y Rosalba por apoyarme en todo momento y estar pendiente de mi en tiempos buenos y en en tiempos difíciles. Gracias por el amor que me han dado, por sus consejos y sus palabras de aliento.

A mi asesor el Dr. Omar Aguilar Mejía por ayudarme a desarrollar el presente trabajo, por sus consejos que me han ayudado a mejorar profesionalmente.

A la Universidad Politécnica de Tulancingo, por el apoyo brindado durante el tiempo que desarrolle mi trabajo de investigación.

A CONACYT, por el apoyo económico brindado, el cual me permitió construir el prototipo de robot delta y solventar mis gastos personales.

A mis familiares y amigos quienes también me han apoyado en momentos buenos y difíciles; los cuales son parte de mi vida y me han dado hermosos momentos llenos de alegría.

Resumen

En esta tesis se presenta el análisis cinemático, dinámico, la construcción y el control de posición de un Robot Paralelo tipo Delta (RPD) de tres grados de libertad. El análisis cinemático se enfoca en la solución de dos problemas: el problema cinemático inverso, y el problema cinemático directo. El primer problema se resuelve por un método geométrico llamado *Loop Closure Method*, y el segundo problema se soluciona aplicando un método de trilateración empujando tres esferas virtuales cuyo centro y radio se obtienen mediante parámetros geométricos del robot.

El modelo dinámico del robot se aproxima empleando el método de Lagrange. El sistema de ecuaciones dinámicas se emplea para realizar simulaciones en lazo abierto, y en lazo cerrado aplicando leyes de control, las cuales deben regular la posición del elemento final del robot.

En este trabajo se compara la respuesta en estado transitorio y estacionario de dos controladores: el primer controlador es un PD+G, y un controlador PD adaptativo por redes neuronales B-Spline BSNN.

En el trabajo se detalla la forma en que se fabricaron las piezas que conforman el mecanismo del RPD, partiendo de los diseños en CAD, la selección de los materiales, y el proceso de mecanizado.

La instrumentación es un tema relevante en el presente trabajo debido a que enlaza los algoritmos de cinemática y control con los actuadores y sensores empleados en el prototipo del robot Delta. Para la adquisición y el envío de datos al robot, se hace uso de un sistema embebido en tiempo real, el cual contiene un chip FPGA en el cual se programa los algoritmos de control de posición.

Índice general

Dedicatoria	I
Agradecimientos	III
Resumen	V
Índice de Figuras	XI
Índice de Tablas	XV
1. Reseña de los robots paralelos	1
1.1. Introducción	1
1.1.1. ¿Qué es un Robot?	1
1.1.2. Tipos de robots manipuladores	2
1.1.3. Reseña histórica de los robots paralelos	2
1.2. El Robot Delta	5
1.2.1. Diseño del RPD	6
1.2.2. El Robot Delta en el mercado	6
1.3. Planteamiento del problema	9
1.4. Objetivos del trabajo	9
1.4.1. Objetivo general	9
1.4.2. Objetivos particulares	9
1.5. Justificación	9
1.6. Metas	10
1.7. Hipótesis	10
1.8. Conclusiones	10
Referencias	12
2. Análisis Cinemático del RPD	13
2.1. Introducción	13
2.2. Análisis de la cadena cinemática	14
2.3. Solución del problema cinemático inverso	17
2.3.1. Ejemplos de solución de algoritmo de cinemática inversa	18
2.4. Solución del problema cinemático directo	20
2.4.1. Ejemplos de solución de cinemática directa	23

2.5. Cinemática diferencial	25
2.5.1. Singularidad de cinemática inversa	26
2.5.2. Singularidad de cinemática directa	27
2.5.3. Singularidad combinada	27
2.6. Espacio de trabajo	27
2.7. Conclusiones	28
Referencias	29
3. Dinámica del RPD	31
3.1. Introducción	31
3.2. Formulación Lagrangiana	32
3.2.1. Energía cinética	34
3.2.2. Energía potencial	34
3.2.3. Función de Lagrange	35
3.2.4. Simulación	37
3.3. Conclusiones	39
Referencias	39
4. Control de movimiento del RPD	41
4.1. Introducción	41
4.2. Consideraciones para implementar un controlador	41
4.3. Implementación de un controlador PD+G	42
4.4. Propuesta de control adaptativo por redes neuronales	44
4.4.1. Regla de aprendizaje	46
4.5. Comparación entre controladores	47
4.6. Seguimiento de trayectoria circular	49
4.7. Conclusiones	52
Referencias	53
5. Manufactura del RPD	55
5.1. Introducción	55
5.2. Diseño mecánico en SOLIDWORKS	56
5.3. Programación en la máquina CNC	58
5.3.1. Sistema de coordenadas	58
5.3.2. Interpolación lineal (G01)	59
5.3.3. Interpolación circular en sentido horario (G02)	60
5.3.4. Interpolación circular en sentido anti horario (G03)	60
5.4. Fabricación de componentes del RPD	61
5.5. Selección de articulaciones esféricas	66
5.6. Ensamblaje total	67
5.7. Conclusiones	68
Referencias	69

6. Instrumentación	71
6.1. Introducción	71
6.2. Programación en LabVIEW	71
6.2.1. Entorno gráfico	72
6.2.2. Cables	73
6.2.3. Estructuras	74
6.3. Descripción del FPGA	75
6.3.1. Partes de un FPGA	76
6.4. Motores del RPD	77
6.5. Sensores	77
6.5.1. Sensor de proximidad	77
6.5.2. Encoder e instrumentación	78
6.6. Implementación de un controlador PD en FPGA	80
6.7. Resultados experimentales	83
6.8. Conclusiones	86
Referencias	87
A. Creación de un proyecto en LabVIEW FPGA	89
B. Diagramas eléctricos	93
B.1. Conexión de encoder	93
B.2. Conexión de los sensores de proximidad	94
B.3. Etapa de potencia	95
Trabajos futuros	99

Índice de figuras

1.1. Plataforma espacial patentada por James E. Gwinnett [1.4]	2
1.2. Robot paralelo de Willard LV Pollard [1.5]	3
1.3. Plataforma de Gough [1.6]	4
1.4. Plataforma de Stewart	4
1.5. Simulador de vuelo de Klaus Cappel [1.8]	5
1.6. Robot Delta de R. Clavel. Patente No. 4,976,582 [1.10]	6
1.7. Robot Delta de la empresa Demarex en la industria alimenticia [1.3] .	7
1.8. Sistema SurgiScope en acción [1.3]	7
1.9. Robots comerciales basados en el Robot Delta	8
2.1. Robot Delta en su concepto CAD	13
2.2. Esquema cinemático del Robot Delta	14
2.3. Detalles de las plataformas fija y móvil	14
2.4. Representación gráfica del primer ejemplo de solución de cinemática inversa	19
2.5. Representación gráfica del segundo ejemplo de solución de cinemática inversa	20
2.6. Diagrama para la obtención de la cinemática directa	21
2.7. Representación gráfica del primer ejemplo de solución de cinemática directa	24
2.8. Representación gráfica del segundo ejemplo de solución de cinemática directa	25
2.9. Distintas representaciones del espacio de trabajo	28
3.1. Esquema de una cadena cinemática para la obtención del modelo dinámico del RPD	33
3.2. Diagrama de bloques de la dinámica directa	37
3.3. Desplazamiento angular del RPD sin control	38
3.4. Desplazamiento cartesiano del RPD sin control	38
4.1. Diagrama de control PD+G en lazo cerrado para regular la posición del RPD	43
4.2. Respuesta del sistema con control coordenadas cartesianas	43
4.3. Respuesta transitoria del Robot Delta con control valores de desplazamiento angular	44
4.4. Lazo de control adaptativo por BSNN	46

4.5. Comparación de la respuesta transitoria del desplazamiento cartesiano entre el PD y BSNN	47
4.6. Comparación de la respuesta transitoria del desplazamiento angular entre el PD y BSNN	48
4.7. Trayectoria deseada a seguir por el RPD	49
4.8. Seguimiento de trayectoria	50
4.9. Comparación de la respuesta de seguimiento de trayectoria en coordenadas cartesianas	50
4.10. Comparación de la respuesta de seguimiento de trayectoria en desplazamiento angular	51
4.11. Comparación del error de seguimiento de trayectoria en coordenadas cartesianas	51
4.12. Comparación del error de seguimiento de trayectoria en desplazamiento angular	52
5.1. RDDCA con bastidor y componentes de control	56
5.2. Herramientas de posicionamiento estándar para ensamblajes	56
5.3. Mecanismo del RPD ensamblado totalmente	57
5.4. Traslación del sistema de coordenadas	59
5.5. Ejemplo de aplicación del G01	59
5.6. Ejemplo de aplicación del G02	60
5.7. Ejemplo de aplicación del G03	60
5.8. Sujeta-motor de la plataforma fija	61
5.9. Simulación de mecanizado de una pieza para la base de los motores del RPD	61
5.10. Centro de maquinado VDL-1000 de DMTG	62
5.11. Material en bruto empleado para fabricar las piezas	63
5.12. Cortador vertical de 10mm	63
5.13. Editor de programa del controlador FANUC	64
5.14. Maquinado de pieza sujeta-motor	64
5.15. Etapas de fabricación de un brazo del RDDCA	65
5.16. Diseño en SOLIWORKS	65
5.17. Producto final	66
5.18. Aurora CW-5S	66
5.19. Ensamble de rotulas con sus respectivos antebrazos	67
5.20. Ensamble total del RDDCA	68
6.1. Panel frontal y diagrama de bloques de un programa en LabVIEW	72
6.2. Paletas de funciones y controles	73
6.3. Cables de datos	74
6.4. 1) Ciclo While 2) Diagrama de flujo 3) Pseudocódigo	74
6.5. 1) Ciclo For 2) Diagrama de flujo 3) Pseudocódigo	75
6.6. Estructura Case	75

6.7. Partes que conforman un chip FPGA	76
6.8. Sensor de proximidad empleado	78
6.9. Código de conteo de pulsos del encoder	79
6.10. Posición angular en radianes	79
6.11. Controlador PD+G en LabVIEW FPGA	80
6.12. Derivación en FPGA	81
6.13. Conector DB25 para programar fuente remota	82
6.14. Condición para inversión de giro	82
6.15. Trayectoria deseada para el RPD	83
6.16. Desplazamiento en coordenadas cartesianas de la trayectoria	84
6.17. Referencia de desplazamiento angular de los actuadores	84
6.18. Referencia tiempo real en LabVIEW	85
6.19. Seguimiento de trayectoria en tiempo real LabVIEW	85
6.20. Comparación entre referencia y posición real	86
A.1. Ventana de inicio en LabVIEW 2012	89
A.2. Selección de tipo de proyecto en LabVIEW 2012	90
A.3. Ventana para encontrar dispositivos FPGA LabVIEW 2012	90
A.4. Selección de controlador en LabVIEW 2012	91
A.5. Detalles de Hardware del cRIO	91
A.6. Proyecto nuevo en LabVIEW FPGA	92
B.1. Conexión de los codificadores ópticos con el NI 9403	93
B.2. Conexión de los sensores inductivos con el NI 9423	94
B.3. Etapa de potencia del primer motor	95
B.4. Etapa de potencia del segundo motor	96
B.5. Etapa de potencia del tercer motor	96

Índice de Tablas

1.1. Empresas fabricantes de robots basados en el Robot Delta	8
2.1. Parámetros geométricos del RPD	15
3.1. Parámetros empleados para la simulación de la dinámica del Robot Delta	37
5.1. Códigos G elementales.	58
5.2. Códigos M elementales.	58
6.1. Características del MTE-3363-BLBCN	77
B.1. Pines de conexión del cable LD5481A20	94

1

Reseña de los robots paralelos

1.1. Introducción

Los robots son recursos importantes para la industria de hoy en día. El uso de robots es vital en la automatización industrial para preservar la calidad y cantidad de producción mientras introduce flexibilidad en la línea de fabricación. Debido a la necesidad de incrementar la cantidad de producción de ciertos elementos consumibles por los seres humanos, algunos de manera directa tales como, medicamentos y alimentos; y otros los cuales se emplean para la vida cotidiana como componentes electrónicos, electrodomésticos, automóviles etc; se ha recurrido a la automatización, lo cual agiliza la producción y mejora o preserva la calidad de los productos finales.

Las primeras compañías que emplearon robots en sus líneas de fabricación, fueron las relacionadas a la industria automotriz, en años recientes, otro tipo de industrias como la alimenticia y materiales farmacéuticos, han adoptado sistemas robóticos en sus líneas de producción. La mayor motivación del crecimiento de el uso de robots industriales en diferentes líneas de producción, es la reducción de costos de fabricación. En años recientes, los precios de los robots se han reducido considerablemente, mientras los costos de labor humana han incrementado. A medida en que avanza la tecnología los robots se vuelven más efectivos, más rápidos, más inteligentes, más precisos, y más flexibles.

1.1.1. ¿Qué es un Robot?

Existen varias definiciones para describir a un robot, según la definición del Robot Institute of America (RIA) establece que:

Un robot es un manipulador multifuncional reprogramable diseñado para mover materiales, partes, herramientas o dispositivos especializados a través de movimientos programados para la ejecución de una variedad de tareas [1.1, págs. 9-10].

La mayoría de los robots, pero no necesariamente todos tendrán las siguientes propiedades.

- No es natural, ha sido creado artificialmente
- Puede sensor su entorno

Una década mas tarde en 1938, Willard LV Pollard, diseña el primer robot paralelo industrial, destinado a operaciones de pintura con spray. El diseño consiste de un sistema de cinco grados de libertad, y con tres brazos de dos eslabones cada uno, los eslabones estában unidos mediante articulaciones universales. Los tres actuadores de la base comandaban la posición de la herramienta, mientras que la orientación era proporcionada por otros dos actuadores situados en la base y que transmitían el movimiento a la herramienta mediante la rotación proporcional de cables flexibles, el robot paralelo de Willard LV Pollard se muestra en la Figura 1.2 [1.5].

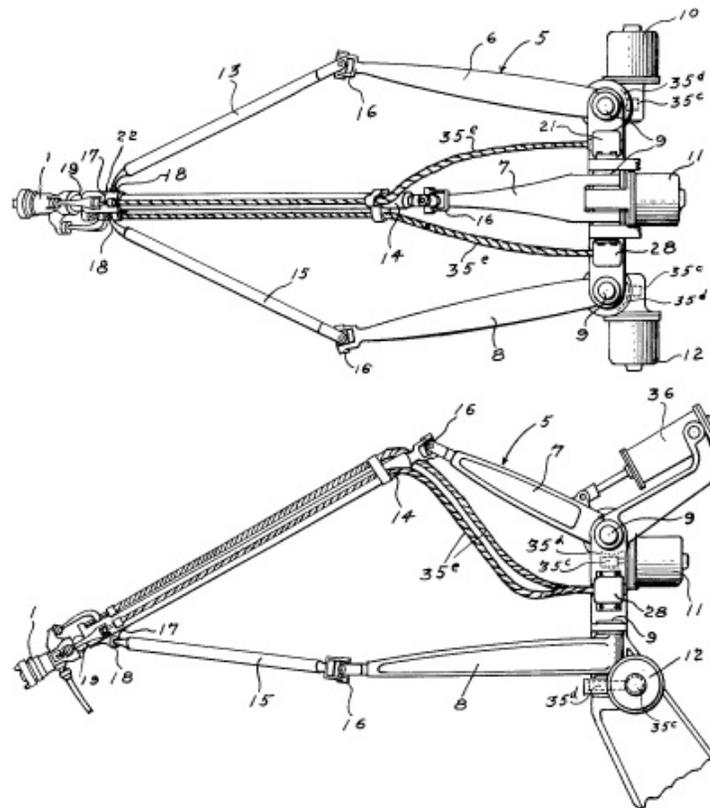


Figura 1.2: Robot paralelo de Willard LV Pollard [1.5]

En el año de 1947, el Dr. Eric Gough diseño una plataforma tipo octaedro hexápodo con lados de longitud variable, el Dr. Gough la llamó máquina de pruebas de cubiertas universal, o plataforma universal, este sistema, se inventó con el fin de responder a los problemas de cargas aero-aterrizaje, es decir, la plataforma de Gough intentaba simular el proceso de aterrizaje de un avión. Esta máquina se usó para la comprobación de neumáticos de la casa Dunlop para la cual trabajaba, bajo cargas aplicadas en diferentes ejes. La plataforma de Gough ha sido uno de los mecanismos que han logrado un mayor reconocimiento en el área de la robótica paralela, la Figura 1.3 es un ejemplo de la plataforma de Gough [1.6].



Figura 1.3: Plataforma de Gough [1.6]

En 1965 el Sr. Stewart, publicó un artículo, en el cual describe una plataforma de seis grados de libertad destinada a ser usada como simulador de vuelo; la unión de varias cadenas cinemáticas en este mecanismo, proveían varios movimientos complejos a la cabina del piloto en entrenamiento, la Figura 1.4 muestra un bosquejo de la plataforma de Stewart [1.7].

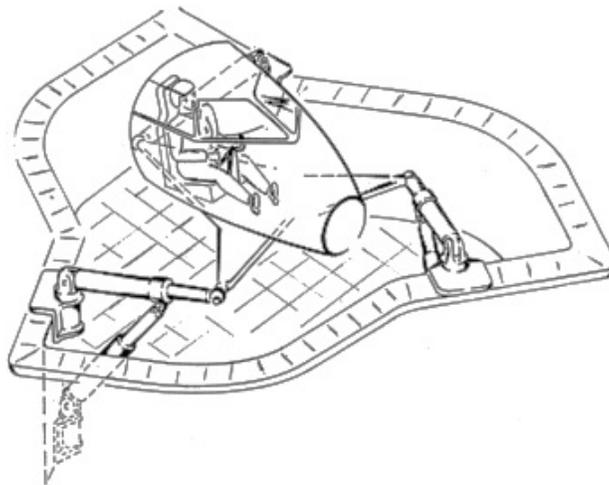


Figura 1.4: Plataforma de Stewart

El ingeniero Klaus Cappel, en el año de 1967 patentó un simulador de movimiento basado en un hexápodo, de esta forma se construyó el primer simulador de vuelo, en el año de 1967. En la Figura 1.5 se muestra el primer simulador de vuelo construido por el Ing. Klaus Cappel [1.8].

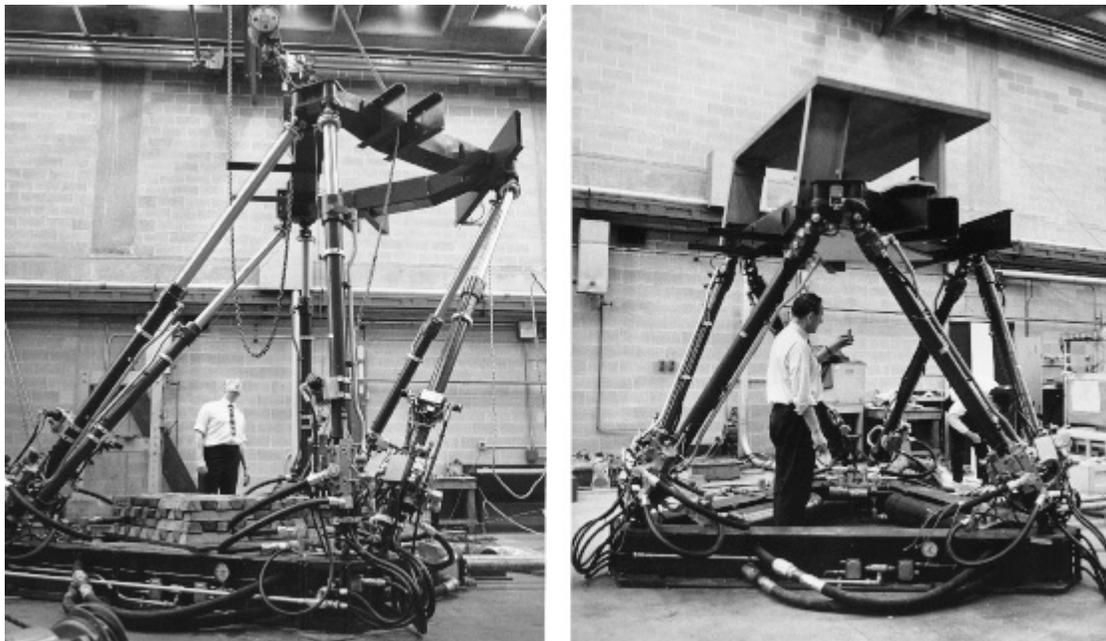


Figura 1.5: Simulador de vuelo de Klaus Cappel [1.8]

Debido a los avances de la investigación, los robots paralelos se están perfeccionando rápidamente, y gracias al desarrollo de nuevas tecnologías hacen que el control automático de estas máquinas sea más preciso, y puedan integrarse una gran variedad de dispositivos externos como sensores, interfaces gráficas de usuario, cámaras digitales, etc.

1.2. El Robot Delta

En los años 80, Reymond Clavel, profesor del École Polytechnique Fédérale de Lausanne (EPFL) irrumpe con la idea de usar paralelogramos para construir una estructura paralela robusta con tres grados de libertad de traslación y un rotatorio. A esta estructura se le dio el nombre de Robot Delta, sin sospechar que al final del siglo, sería uno de los diseños paralelos más acertados. Con varios cientos de estructuras activas en todo el mundo, el Dr. Clavel es premiado con el “Golden Robot Award” (patrocinado por ABB) por su trabajo innovador en el Robot paralelo Delta (Clavel, 1988) [1.9, pág. 17].

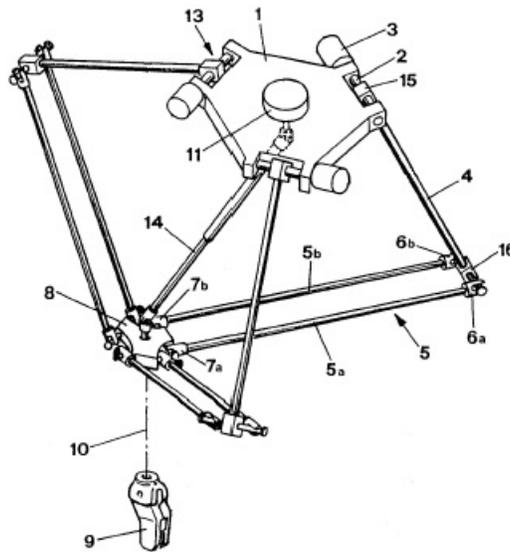


Figura 1.6: Robot Delta de R. Clavel. Patente No. 4,976,582 [1.10]

1.2.1. Diseño del RPD

La idea básica del diseño del Robot Delta, es el uso de paralelogramos, que permiten que un eslabón de salida tenga una orientación fija con respecto a un eslabón de entrada. Al hacer uso de tres paralelogramos la orientación de la plataforma móvil queda totalmente restringida, haciendo que se traslade únicamente sobre los tres ejes del plano cartesiano [1.10].

El Robot Delta tiene tres grados de libertad traslacionales, y en algunos casos un grado de libertad rotacional. Este tipo de robot consta de una plataforma fija que esta conectada a una plataforma móvil donde se coloca el efector final. Los eslabones consisten en una articulación rotacional actuada, conectada al efector a través de una articulación universal o esférica pasiva. Por lo anterior dos eslabones sólo reciben una fuerza de tensión compresión (sin deslizamiento), lo que incrementa la precisión en el posicionamiento y permite la construcción de una estructura muy ligera. Dos eslabones cualesquiera forman un circuito cerrado, con lo que se obtiene alta resistencia estructural, debido a que el efector final está soportado en varios puntos al mismo tiempo.

1.2.2. El Robot Delta en el mercado

La historia del robot Delta en el mercado comenzó en 1983, cuando los hermanos suizos Marc-Olivier y Pascal Demareux crearon la compañía Demareux con base en Laussane Suiza [1.11]. Después de obtener una licencia para el robot Delta en 1987, su mayor objetivo era llegar a comercializar el robot paralelo para la industria del empaquetado. Después de varios años, Demareux ocupaba un importante lugar en este mercado. Cuatro versiones eran comercializadas bajo los nombres de: Pack-Place, Line-

Placer, Top-Placer, y Presto. En la Figura 1.7 se muestra una línea de montaje con robots tipo delta de la empresa Demarex.



Figura 1.7: Robot Delta de la empresa Demarex en la industria alimenticia [1.3]

El Robot Delta se usa para la toma y traslación de pequeños objetos con una gran velocidad y precisión, que poco a poco ha adquirido su lugar en industrias como la alimenticia o farmacéutica, por lo cual el Robot Delta ha incursionando lentamente en el área médica. Un ejemplo es el sistema SurgiScope fabricado por ISIS Robotics, que es un robot empleado para sostener y controlar la posición de un microscopio de 20 Kg; como se puede apreciar en Figura 1.8.



Figura 1.8: Sistema SurgiScope en acción [1.3]

Actualmente algunas de las empresas más importantes que fabrican este tipo de robots se muestran en la Tabla 1.1. En la Figura 1.9 se muestran físicamente algunos modelos de robots tipo delta construidos por algunas empresas.

Tabla 1.1: Empresas fabricantes de robots basados en el Robot Delta

Robots basados en el Robot Delta	
Compañía	Modelo
ABB Group	IRB360 Flexpicker
Yaskawa	Motoman MPP3S
Fanuc	M-1 , M-2 y M-3 Series
Robert Bosch GmbH	Paloma D-2 Top Loader Robot
Adept Technology, Inc	Adep Quattro s650H parallel robot
Codian Robotics	D4 y D5 Series



(a) ABB Flexpicker IRB360 [1.12]



(b) Yaskawa Motoman MPP3S [1.13]



(c) Codian D5 [1.14]



(d) Fanuc M-11A [1.15]

Figura 1.9: Robots comerciales basados en el Robot Delta

1.3. Planteamiento del problema

Actualmente en materia de automatización el país tiene una gran dependencia de empresas extranjeras las cuales proporcionan tecnología para controlar procesos en diversas industrias. Los elementos tecnológicos que cada vez más se emplean son los robots; debido a que se adaptan a diversos campos de trabajos, y tienen mayor productividad. Sin embargo los costos para adquirir un robot son altos, y para algunas pequeñas empresas nacionales resulta muy difícil poder obtener este tipo de tecnología. Por este motivo se ha decidido construir un robot manipulador, en este caso el RPD. ¿Por qué el robot paralelo tipo delta?. El RPD, tiene una construcción mecánicamente simple y además simétrica, estas características son útiles para construir un prototipo en el menor tiempo posible, además, este modelo de robot, ha tenido un gran impacto en industrias del *Pick and Place*, como farmacéutica, o control de calidad y alimentos, bajo este contexto se plantean los siguientes objetivos del trabajo.

1.4. Objetivos del trabajo

1.4.1. Objetivo general

Analizar, construir y controlar un robot paralelo tipo delta RPD de tres grados de libertad.

1.4.2. Objetivos particulares

- Analizar y resolver los algoritmos de cinemática del Robot Delta de tres grados de libertad, para conocer la posición y orientación del robot.
- Determinar la configuraciones singulares del mecanismo, para conocer el espacio de trabajo del robot.
- Obtener el modelo dinámico de el robot delta, para simular el comportamiento en estado estable y transitorio.
- Diseñar un esquema de control para que el robot paralelo siga una trayectoria de referencia.
- Diseñar y construir un robot paralelo tipo delta de tres grados de libertad para conocer el desempeño del sistema planta controlador.

1.5. Justificación

Cada vez más empresas incorporan robots industriales en sus líneas de producción. Sin embargo una gran parte de los robots industriales que se usan en México son de

fabricación extranjera; por tal motivo los costos de importación de estos dispositivos son altos. El desarrollar un prototipo de un robot manipulador basado en un robot comercial; (En este caso el robot Delta), permitirá reducir los costos para adquirir un manipulador de esta naturaleza y su vez ir reduciendo la dependencia tecnológica que tiene el país.

1.6. Metas

- Simulación de la respuesta transitoria y en estado estacionario del RPD empleando los algoritmos de cinemática y dinámica con el control de posición
- Construcción del prototipo de RPD
- Programar el control de posición del RPD dentro de un sistema embebido con la finalidad de que el elemento final del robot pueda seguir distintas trayectorias deseadas

1.7. Hipótesis

Construir un robot industrial no es una tarea sencilla, debido la cantidad de elementos que influyen al momento de su construcción por ejemplo; la selección de los actuadores que se deben utilizar, los materiales empleados para su construcción, la electrónica de potencia que se debe emplear para el control de motores, y el sistema encargado para la adquisición y el envío de datos. En el caso de querer construir un robot industrial tipo serial se debe considerar en número de eslabones que contendrá el robot; y debido a la naturaleza de este tipo de robots, cada actuador debe poseer distintas características respecto a los demás, por ejemplo el motor ubicado en la base del robot debe ser de mayor potencia que el que se ubica al final de la cadena cinemática debido a la masa y el momento de inercia que este debe soportar; por lo cual la construcción del mecanismo resulta complicada y por consiguiente el control de variables como posición velocidad y par. Sin embargo un robot tipo paralelo como es el robot Delta, al ser un robot simétrico, cada cadena cinemática es igual una de la otra y de la misma manera los motores que son empleados; lo cual facilita la construcción del mecanismo del robot y consecuentemente, se reduce el tiempo empleado en la fabricación. Por estos motivos se ha decidido construir un robot paralelo tipo Delta.

1.8. Conclusiones

En este capítulo de introducción se abrió un panorama sobre los robots paralelos de manera general, en el cual se explicaron algunas diferencias que existen entre robots seriales y paralelos. Profundizando más en el concepto de robots paralelos; se menciona

en una reseña histórica, los avances que esta rama de la robótica ha tenido a lo largo del tiempo hasta llegar al RPD. Por la descripción del RPD presentada en el presente capítulo, se observa que sus mayores virtudes son: precisión y velocidad; por lo cual ha tenido un gran impacto en industrias como la alimenticia y farmacéutica; tales virtudes han llevado a muchas empresas fabricantes de robots industriales a desarrollar su propia versión del RPD basándose en el robot creado por Reymond Clavel. Sin embargo la gran mayoría de las empresas que proporcionan estos robots a la industria mexicana son de origen extranjero; lo cual eleva los costos del producto debido a la importación; además de generar una dependencia tecnológica de otros países. Por tales motivos se ha decidido analizar, diseñar y construir y controlar un prototipo basado en el RPD el cual podrá ser empleado en tareas *Pick and Place*.

Referencias

- [1.1] A. Barrientos, L. F. Peñin, C. Balaguer, and R. Aracil, *Fundamentos de robótica*. McGraw-Hill, Interamericana de España, 2007.
 - [1.2] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators*. John Wiley & Sons, 1999.
 - [1.3] I. Bonev, “The true origins of parallel robots <http://www.parallemic.org/Reviews/Review007.html>,” 2003.
 - [1.4] J. E. Gwinnett, “Amusement device,” Jan. 20 1931. US Patent 1,789,680.
 - [1.5] W. Pollard, “Spray painting machine: Usa,” *US2213108*, 1940.
 - [1.6] V. Gough and S. Whitehall, “Universal tyre test machine,” in *Proc. FISITA 9th Int. Technical Congress*, pp. 117–137, 1962.
 - [1.7] D. Stewart, “A platform with six degrees of freedom,” *Proceedings of the institution of mechanical engineers*, vol. 180, no. 1, pp. 371–386, 1965.
 - [1.8] K. Cappel, “Motion simulator,” Feb. 9 1971. US Patent RE27,051.
 - [1.9] H. D. Taghirad, *Parallel robots: mechanics and control*. CRC press, 2013.
 - [1.10] R. Clavel, “Delta, a fast robot with parallel geometry,” in *Proc. 18th Int. Symp. on Industrial Robots, Lausanne, 1988*, pp. 91–100, 1988.
 - [1.11] J. D. R. Florez, “Metodología para el diseño de un robot paralelo industrial tipo delta,” pp. 29–32, 2008.
 - [1.12] ABB, “Irb360 flexpicker. <http://new.abb.com/products/robotics/industrial-robots/irb-360>.”
-

- [1.13] Yaskawa, “Mpp3s. <https://www.motoman.com/industrial-robots/mpp3s>.”
 - [1.14] Codian, “D5 pick and place robot. <http://www.codian-robotics.com/en/d5-robots/>.”
 - [1.15] Fanuc, “Fanuc m-1ia. <http://www.fanuc.eu/fr/en/robots/robot-filter-page/m1-series/m-1ia-1h>.”
-

2

Análisis Cinemático del RPD

2.1. Introducción

En cualquier rama de la robótica, al comenzar a realizar un estudio detallado de un robot es necesario conocer como se relaciona su construcción, articulaciones y eslabones, y los movimientos del robot en el área de trabajo. El análisis cinemático estudia la geometría del movimiento del robot, sin considerar las fuerzas y torques que lo producen.

El análisis cinemático resuelve el problema de la cinemática inversa y el problema de la cinemática directa. La cinemática inversa permite encontrar los valores de las coordenadas articulares del robot en base a la posición y orientación del efector final en coordenadas cartesianas. La cinemática directa permite calcular la posición y orientación del efector final del robot en función de las coordenadas articulares dadas [2.1]. La Figura 2.1 muestra la estructura del robot delta que se pretende construir físicamente, por lo cual se tomará como referencia para los análisis matemáticos posteriores.

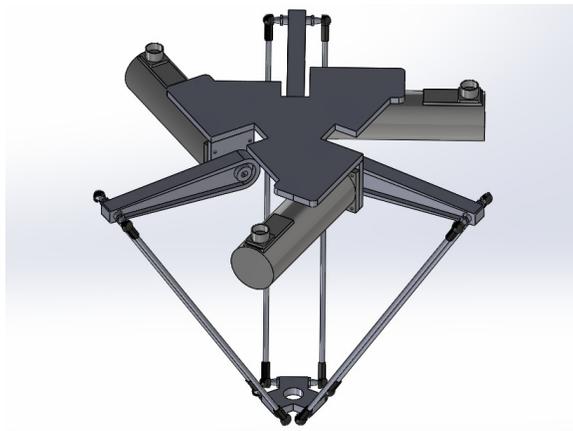


Figura 2.1: Robot Delta en su concepto CAD

2.2. Análisis de la cadena cinemática

El RPD está compuesto de tres cadenas cinemáticas cerradas paralelas, cada cadena consiste de un brazo y un antebrazo, cada brazo está conectado a una base fija mediante una articulación rotacional que genera el movimiento por medio de un motor. El antebrazo se conecta al brazo a través de una junta esférica y el efector final está unido al antebrazo por medio de otra articulación esférica. La configuración articular para cada cadena cinemática del robot queda definida como RSS (Revolute-Spherical-Spherical). La figura 2.2 muestra los elementos que componen las tres cadenas cinemáticas cerradas y en la Figura 2.3 se muestran los detalles de la plataforma fija y móvil [2.2].

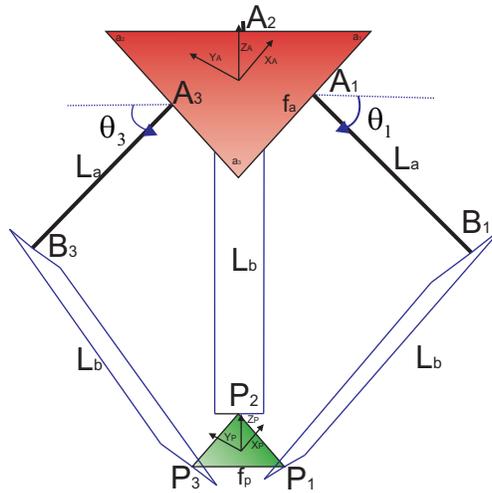


Figura 2.2: Esquema cinemático del Robot Delta

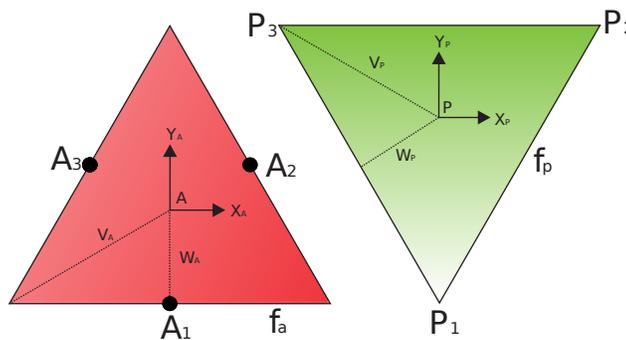


Figura 2.3: Detalles de las plataformas fija y móvil

Del diagrama cinemático del Robot Delta mostrado en la Figura 2.2, se plantea la ecuación que describe la geometría de las cadenas cinemáticas cerradas [2.3].

$$\{^A \mathbf{A}_i\} + \{^A \mathbf{L}_{a_i}\} + \{^A \mathbf{L}_{b_i}\} = \{^A \mathbf{P}_P\} + \{^P \mathbf{P}_i\} \quad (2.1)$$

La referencia de la base fija $\{A\}$ es donde se localiza el centro del triángulo equilátero; A_i , $i = 1, 2, 3$, son los puntos donde se localizan las articulaciones rotacionales. La longitud de lado del triángulo equilátero de la base fija está dada por f_a ; B_i , para $i = 1, 2, 3$ son los puntos donde se conectan el brazo de longitud L_a , con el antebrazo de longitud L_b , por medio de las articulaciones esféricas; P_i , $i = 1, 2, 3$, son los puntos donde se unen, la plataforma móvil con las barras del antebrazo del robot, por medio de articulaciones esféricas, la longitud de lado de la plataforma móvil está dada por f_p . Las variables cartesianas de RPD son:

$${}^A\mathbf{P}_P = \{P_x \ P_y \ P_z\}^T \quad (2.2)$$

Las variables angulares del RPD están dadas por la siguiente expresión:

$$\Theta = \{\theta_1 \ \theta_2 \ \theta_3\}^T \quad (2.3)$$

En la Tabla 2.1 se indican las constantes geométricas del RPD:

Tabla 2.1: Parámetros geométricos del RPD

Nombre	Descripción	Valor(mm)
f_A	Longitud de lado de plataforma fija	439
f_P	Longitud de lado de plataforma móvil	86
L_a	Longitud del brazo del Robot	300
L_b	Longitud del antebrazo del Robot	624

Cabe mencionar que $\{{}^A\mathbf{A}_i\}$ son vectores de constantes cuyos valores están dados por:

$${}^A\mathbf{A}_1 = \begin{Bmatrix} 0 \\ -w_A \\ 0 \end{Bmatrix} \quad {}^A\mathbf{A}_2 = \begin{Bmatrix} \frac{\sqrt{3}}{2}w_A \\ \frac{1}{2}w_A \\ 0 \end{Bmatrix} \quad {}^A\mathbf{A}_3 = \begin{Bmatrix} -\frac{\sqrt{3}}{2}w_A \\ \frac{1}{2}w_A \\ 0 \end{Bmatrix} \quad (2.4)$$

De la misma manera las coordenadas cartesianas de $\{{}^P\mathbf{P}_i\}$ están dadas por

$${}^P\mathbf{P}_1 = \begin{Bmatrix} 0 \\ -v_P \\ 0 \end{Bmatrix} \quad {}^P\mathbf{P}_2 = \begin{Bmatrix} \frac{f_P}{2} \\ -w_P \\ 0 \end{Bmatrix} \quad {}^P\mathbf{P}_3 = \begin{Bmatrix} -\frac{f_P}{2} \\ w_P \\ 0 \end{Bmatrix} \quad (2.5)$$

Los vértices de la plataforma fija están determinados como:

$${}^A\mathbf{a}_1 = \begin{Bmatrix} \frac{f_P}{2} \\ -w_A \\ 0 \end{Bmatrix} \quad {}^A\mathbf{a}_2 = \begin{Bmatrix} 0 \\ v_A \\ 0 \end{Bmatrix} \quad {}^A\mathbf{a}_3 = \begin{Bmatrix} -\frac{f_P}{2} \\ -w_A \\ 0 \end{Bmatrix} \quad (2.6)$$

Para comprender mejor el análisis cinemático se hace uso de distancias auxiliares, mostradas en la Figura 2.3, las cuales son:

- w_a es la distancia planar desde $\{A\}$ hasta la arista f_a de la base fija
- v_A es la distancia planar del origen de la base fija $\{A\}$ hasta el vértice \mathbf{a}_i
- w_P es la distancia planar desde el origen de la base móvil $\{P\}$ hasta la arista f_p
- v_P es la distancia entre $\{P\}$ y el vértice \mathbf{P}_i .

Para calcular las constantes anteriores se emplean las siguientes transformaciones para relacionar las variables:

$$w_A = \frac{\sqrt{3}}{6} f_a \quad v_A = \frac{\sqrt{3}}{3} f_a \quad (2.7)$$

$$w_P = \frac{\sqrt{3}}{6} f_p \quad v_P = \frac{\sqrt{3}}{3} f_p \quad (2.8)$$

Los vectores $\{{}^A\mathbf{L}_{\mathbf{a}_i}\}$ que son dependientes de $\Theta = \{\theta_1 \ \theta_2 \ \theta_3\}^T$ están definidos por:

$${}^A\mathbf{L}_{\mathbf{a}_1} = \begin{Bmatrix} 0 \\ -L_a \cos \theta_1 \\ -L_a \sin \theta_1 \end{Bmatrix} \quad {}^A\mathbf{L}_{\mathbf{a}_2} = \begin{Bmatrix} \frac{\sqrt{3}}{2} L_a \cos \theta_2 \\ \frac{1}{2} L_a \cos \theta_2 \\ -L_a \sin \theta_2 \end{Bmatrix} \quad {}^A\mathbf{L}_{\mathbf{a}_3} = \begin{Bmatrix} -\frac{\sqrt{3}}{2} L_a \cos \theta_3 \\ \frac{1}{2} L_a \cos \theta_3 \\ -L_a \sin \theta_3 \end{Bmatrix} \quad (2.9)$$

La longitud virtual L_b es determinada a partir de (2.1) donde se obtiene la norma vectorial de la ecuación geométrica de las cadenas cinemáticas del robot.

$$L_b = \|\{{}^A\mathbf{L}_{\mathbf{b}_i}\}\| = \|\{{}^A\mathbf{P}_P\} + \{{}^P\mathbf{P}_i\} - \{{}^A\mathbf{A}_i\} - \{{}^A\mathbf{L}_{\mathbf{a}_i}\}\| \quad (2.10)$$

Elevando al cuadrado (2.10), para evitar raíces cuadradas en las normas euclidianas, se obtiene que:

$$L_b^2 = \|\{{}^A\mathbf{L}_{\mathbf{b}_i}\}\|^2 = L_{bix}^2 + L_{biy}^2 + L_{biz}^2 \quad (2.11)$$

Substituyendo (2.2), (2.4), (2.5), y (2.9) en (2.10) resulta que:

$${}^A\mathbf{L}_{\mathbf{b}_1} = \begin{Bmatrix} P_x \\ P_y + L_a \cos \theta_1 + c \\ P_z + L_a \sin \theta_1 \end{Bmatrix} \quad {}^A\mathbf{L}_{\mathbf{b}_2} = \begin{Bmatrix} P_x - \frac{\sqrt{3}}{2} L_a \cos \theta_2 + d \\ P_y - \frac{1}{2} L_a \cos \theta_2 + e \\ P_z + L_a \sin \theta_2 \end{Bmatrix} \quad (2.12)$$

$${}^A\mathbf{L}_{\mathbf{b}_3} = \begin{Bmatrix} P_x + \frac{\sqrt{3}}{2} L_a \cos \theta_3 - d \\ P_y - \frac{1}{2} L_a \cos \theta_3 + e \\ P_z + L_a \sin \theta_3 \end{Bmatrix}$$

donde:

$$c = w_A - v_P \quad d = \frac{f_p}{2} - \frac{\sqrt{3}}{2} w_A \quad e = w_P - \frac{1}{2} w_A \quad (2.13)$$

desarrollando (2.11), para obtener las tres ecuaciones geométricas para cada cadena cinemática del RPD, se expresan como:

$$2L_a(P_y + c) \cos \theta_1 + 2P_z L_a \sin \theta_1 + P_x^2 + P_y^2 + P_z^2 + c^2 + L_a^2 + 2cP_y - L_b^2 = 0 \quad (2.14)$$

$$-L_a[\sqrt{3}(P_x + d) + P_y + e] \cos \theta_2 + 2P_z L_a \sin \theta_2 + P_x^2 + P_y^2 + P_z^2 + d^2 + e^2 + L_a^2 + 2dP_x + 2eP_y - L_b^2 = 0 \quad (2.15)$$

$$-L_a[\sqrt{3}(P_x - d) - P_y - e] \cos \theta_3 + 2P_z L_a \sin \theta_3 + P_x^2 + P_y^2 + P_z^2 + d^2 + e^2 + L_a^2 - 2dP_x + 2eP_y - L_b^2 = 0 \quad (2.16)$$

2.3. Solución del problema cinemático inverso

El problema cinemático del robot comienza dando las posiciones deseadas de la plataforma móvil, en coordenadas cartesianas ${}^A\mathbf{P}_P = \{P_x \ P_y \ P_z\}^T$, este algoritmo matemático, calcula los desplazamientos angulares que deben tener los motores, $\Theta = \{\theta_1 \ \theta_2 \ \theta_3\}^T$, para llevar a la plataforma móvil a la posición deseada.

La solución puede ser encontrada usando métodos geométricos cómo analíticos, en este caso la solución se encuentra analíticamente por medio de las ecuaciones geométricas de las cadenas cinemáticas del Robot Delta, (2.14-2.16). Las tres ecuaciones se pueden agrupar de la siguiente forma:

$$L_i \cos \theta_i + M_i \sin \theta_i + N_i = 0 \quad \text{para} \quad i = 1, 2, 3 \quad (2.17)$$

donde:

$$\begin{cases} L_1 = 2L_a(P_y + c) \\ M_1 = 2P_z L_a \\ N_1 = P_x^2 + P_y^2 + P_z^2 + c^2 + L_a^2 + 2cP_y - L_b^2 \end{cases} \quad (2.18)$$

$$\begin{cases} L_2 = -L_a[\sqrt{3}(P_x + d) + P_y + e] \\ M_2 = 2P_z L_a \\ N_2 = P_x^2 + P_y^2 + P_z^2 + d^2 + e^2 + L_a^2 + 2dP_x + 2eP_y - L_b^2 \end{cases} \quad (2.19)$$

$$\begin{cases} L_3 = -L_a[\sqrt{3}(P_x - d) - P_y - e] \\ M_3 = 2P_z L_a \\ N_3 = P_x^2 + P_y^2 + P_z^2 + d^2 + e^2 + L_a^2 - 2dP_x + 2eP_y - L_b^2 \end{cases} \quad (2.20)$$

definiendo:

$$t_i = \tan \frac{\theta_i}{2} \quad \text{entonces} \quad \cos \theta_i = \frac{1 - t_i^2}{1 + t_i^2} \quad \text{y} \quad \sin \theta_i = \frac{2t_i}{1 + t_i^2} \quad (2.21)$$

Substituyendo (2.21) en (2.17) y realizando una serie de despejes, se encuentra que:

$$L_i \left(\frac{1 - t_i^2}{1 + t_i^2} \right) + M_i \left(\frac{2t_i}{1 + t_i^2} \right) + N_i = 0 \quad (2.22)$$

$$L_i(1 - t_i^2) + M_i(2t_i) + N_i(1 + t_i^2) = 0 \quad (2.23)$$

$$(N_i - L_i)t_i^2 + (2M_i)t_i + (N_i + L_i) = 0 \quad (2.24)$$

Utilizando la formula cuadrática, para calcular t_i de la siguiente forma:

$$t_i = \frac{-M_i \pm \sqrt{L_i^2 + M_i^2 - N_i^2}}{N_i - L_i} \quad (2.25)$$

Despejamos θ_i de (2.21) y sustituimos (2.25) en (2.21), se obtiene la solución para la cinemática inversa del Robot Delta.

$$\theta_i = 2 \tan^{-1}(t_i) \quad (2.26)$$

La expresión (2.25) tiene dos soluciones; al momento de proponer los desplazamientos de la base móvil del robot en coordenadas x, y, z , la coordenada z que se va a proponer siempre deberá ser negativa, por lo tanto de las dos soluciones obtenidas en (2.25), se escogerá la de signo negativo.

2.3.1. Ejemplos de solución de algoritmo de cinemática inversa

En esta sección se proponen algunos ejemplos en los cuales se desea desplazar la plataforma móvil del RPD a un determinado punto, el punto esta dado en coordenadas cartesianas, por lo tanto se necesita conocer el desplazamiento angular que deben proporcionar los actuadores para poder llegar al punto propuesto, para ello se emplea la cinemática inversa (2.26); los parámetros empleados en esta simulaciones se encuentran en la Tabla 2.1.

Caso 1

Utilizando en algoritmo de la cinemática inversa, se desea calcular el desplazamiento angular de los tres actuadores para que la plataforma móvil llegue a la posición $\{^A\mathbf{P}_P\} = \{0 \ 0 \ -500\}^T$. Introduciendo en punto deseado se obtiene que los actuadores deben de desplazarse $\Theta = \{0,53 \ 0,53 \ 0,53\}^T$

En las gráficas de la Figura 2.4 se observa el posicionamiento del RPD con 4 perspectivas diferentes.

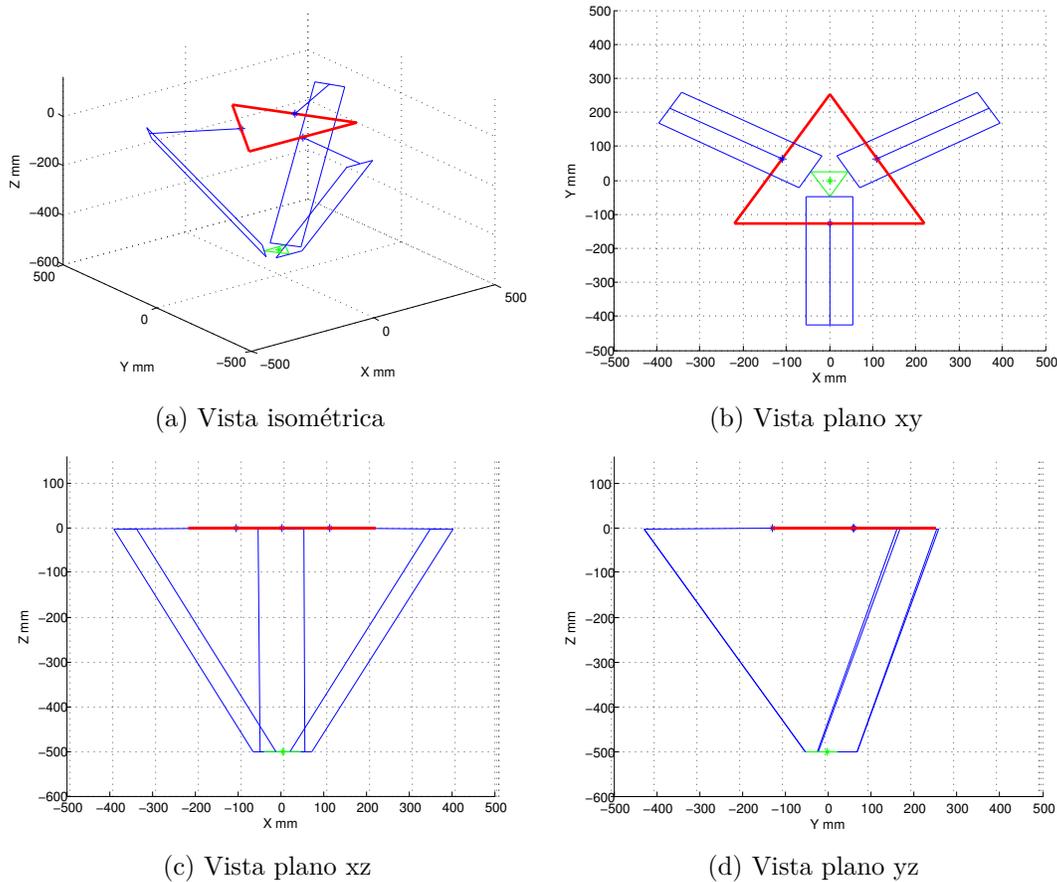


Figura 2.4: Representación gráfica del primer ejemplo de solución de cinemática inversa

En el ejemplo de la Figura 2.4, se puede apreciar que debido a que la posición deseada únicamente requiere un desplazamiento sobre el eje z , por lo cual los ángulos de desplazamiento que entrega el algoritmo de la cinemática inversa son exactamente iguales. En la vista del plano xz se visualiza el desplazamiento sobre el eje z , y en la vista del plano xy se observa que el centro de la base móvil se encuentra en origen del plano al igual que el centro de la base fija.

Caso 2

Igual que en caso anterior, se desea conocer el desplazamiento de los actuadores (motores de CD), para que el RPD se posicione en $\{^A\mathbf{P}_P\} = \{100 \ 200 \ -480\}^T$. Los valores que entrega el algoritmo de la cinemática inversa son: $\Theta = \{33,06 \ -20,98 \ 4,57\}^T$

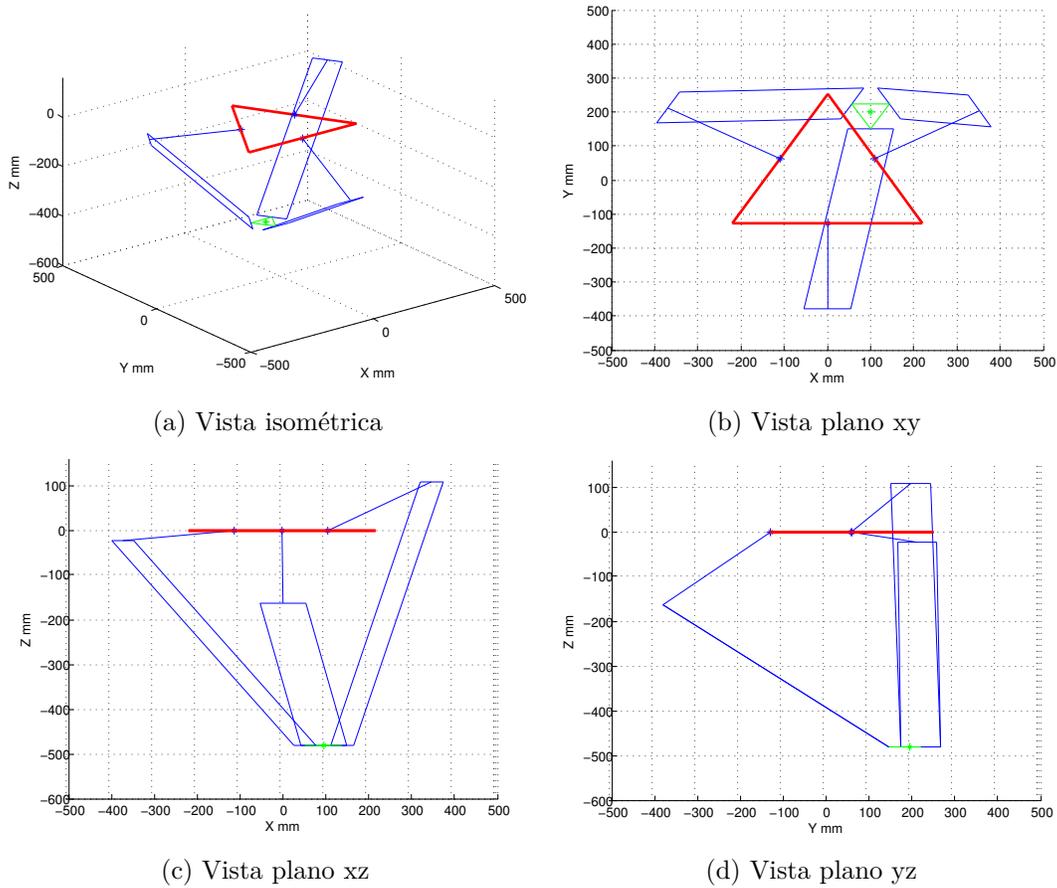


Figura 2.5: Representación gráfica del segundo ejemplo de solución de cinemática inversa

Los resultados del segundo caso se muestran en la Figura 2.5, donde se aprecia el posicionamiento de la base móvil del RPD y el conjunto de eslabonamientos de una manera diferente al del caso 1.

2.4. Solución del problema cinemático directo

El problema cinemático directo del Robot Delta comienza a partir de los valores de los ángulos de las articulaciones rotacionales $\Theta = \{\theta_1 \ \theta_2 \ \theta_3\}^T$, para calcular la posición de la plataforma móvil en sus coordenadas cartesianas, $\{^A\mathbf{P}_P\} = \{P_x \ P_y \ P_z\}^T$. El método empleado para conocer la cinemática directa del Robot Delta consiste en encontrar la intersección de tres esferas virtuales [2.4]. La Figura 2.6 hace referencia a este método para encontrar la cinemática directa del Robot Delta.

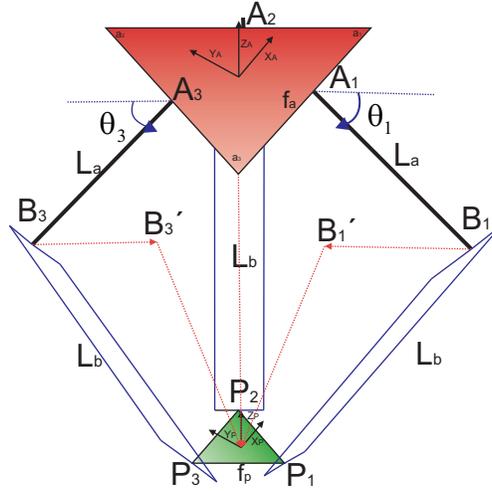


Figura 2.6: Diagrama para la obtención de la cinemática directa

Cada esfera virtual es de radio escalar L_b cuyos centros se encuentran dados por el vector ${}^A\mathbf{B}'_i$, para conocer las coordenadas de los centros de las esferas virtuales, se tiene la siguiente ecuación:

$${}^A\mathbf{B}'_i = {}^A\mathbf{B}_i - {}^P\mathbf{P}_i \quad \text{para} \quad i = 1, 2, 3 \quad (2.27)$$

donde ${}^A\mathbf{B}_i$, esta dada por:

$${}^A\mathbf{B}_1 = \begin{Bmatrix} 0 \\ w_A - L_a \cos \theta_1 \\ -L_a \sin \theta_1 \end{Bmatrix} \quad {}^A\mathbf{B}_2 = \begin{Bmatrix} \frac{\sqrt{3}}{2}(w_A + L_a \cos \theta_2) \\ \frac{1}{2}(w_A + L_a \cos \theta_2) \\ -L_a \sin \theta_2 \end{Bmatrix} \quad (2.28)$$

$${}^A\mathbf{B}_3 = \begin{Bmatrix} -\frac{\sqrt{3}}{2}(w_A + L_a \cos \theta_3) \\ \frac{1}{2}(w_A + L_a \cos \theta_3) \\ -L_a \sin \theta_3 \end{Bmatrix}$$

Substituyendo los valores de (2.28) en (2.27), se obtienen las coordenadas de los centros de las esferas virtuales.

$${}^A\mathbf{B}'_1 = \begin{Bmatrix} 0 \\ w_A - L_a \cos \theta_1 + v_P \\ -L_a \sin \theta_1 \end{Bmatrix} \quad {}^A\mathbf{B}'_2 = \begin{Bmatrix} \frac{\sqrt{3}}{2}(w_A + L_a \cos \theta_2) - \frac{f_P}{2} \\ \frac{1}{2}(w_A + L_a \cos \theta_2) - w_A \\ -L_a \sin \theta_2 \end{Bmatrix} \quad (2.29)$$

$${}^A\mathbf{B}'_3 = \begin{Bmatrix} -\frac{\sqrt{3}}{2}(w_A + L_a \cos \theta_3) + \frac{f_P}{2} \\ \frac{1}{2}(w_A + L_a \cos \theta_3) - v_P \\ -L_a \sin \theta_3 \end{Bmatrix}$$

Para encontrar la intersección de las esferas virtuales, $(\{{}^A\mathbf{B}'_1\}, L_b)$, $(\{{}^A\mathbf{B}'_2\}, L_b)$, $(\{{}^A\mathbf{B}'_3\}, L_b)$

, se plantea el siguiente sistema de expresiones matemáticas:

$$P_x^2 + (P_y - y_1)^2 + (P_z - z_1)^2 = L_b^2 \quad (2.30)$$

$$(P_x - x_2)^2 + (P_y - y_2)^2 + (P_z - z_2)^2 = L_b^2 \quad (2.31)$$

$$(P_x - x_3)^2 + (P_y - y_3)^2 + (P_z - z_3)^2 = L_b^2 \quad (2.32)$$

Existen muchos métodos algebraicos para resolver sistemas de ecuaciones, para este caso se empleara el método de suma y resta; expandiendo (2.30-2.32) se obtiene el siguiente sistema de ecuaciones:

$$P_x^2 + P_y^2 + P_z^2 - 2y_1P_y - 2z_1P_z + y_1^2 + z_1^2 - L_b^2 = 0 \quad (2.33)$$

$$P_x^2 + P_y^2 + P_z^2 - 2x_2P_x - 2y_2P_y - 2z_2P_z + x_2^2 + y_2^2 + z_2^2 - L_b^2 = 0 \quad (2.34)$$

$$P_x^2 + P_y^2 + P_z^2 - 2x_3P_x - 2y_3P_y - 2z_3P_z + x_3^2 + y_3^2 + z_3^2 - L_b^2 = 0 \quad (2.35)$$

Introduciendo una nueva variable para simplificar las relaciones anteriores

$$w_i = x_i^2 + y_i^2 + z_i^2 \quad \text{para} \quad i = 1, 2, 3 \quad (2.36)$$

Restando (2.34) de (2.33), y restando (2.35) de (2.33), se obtiene un nuevo sistema de ecuaciones:

$$2x_2P_x + 2P_y(y_2 - y_1) + 2P_z(z_2 - z_1) = w_2 - w_1 \quad (2.37)$$

$$2x_3P_x + 2P_y(y_3 - y_1) + 2P_z(z_3 - z_1) = w_3 - w_1 \quad (2.38)$$

del sistema (2.37-2.38) se despeja la variable y , con el fin de poder ser eliminada de manera momentánea en una resta, obteniendo una ecuación con dos variables P_x, P_z ; restando (2.38) de(2.37), se obtiene la siguiente ecuación para poder determinar P_x :

$$P_x = \frac{a_1P_z + b_1}{d} \quad (2.39)$$

donde:

$$a_1 = (z_2 - z_1)(y_3 - y_1) - (z_3 - z_1)(y_2 - y_1) \quad (2.40)$$

$$b_1 = \frac{1}{2}(w_2 - w_1)(y_3 - y_1) - (w_3 - w_1)(y_2 - y_1) \quad (2.41)$$

$$d = (y_2 - y_1)x_3 - (y_3 - y_1)x_2 \quad (2.42)$$

Para encontrar la variable y sustituimos (2.39) en (2.38), manipulando las ecuaciones, obtenemos la siguiente expresión para encontrar la variable y :

$$P_y = \frac{a_2P_z + b_2}{d} \quad (2.43)$$

donde:

$$a_2 = -((z_2 - z_1)x_3 - (z_3 - z_1)x_2) \quad (2.44)$$

$$b_2 = \frac{1}{2}((w_2 - w_1)x_3 - (w_3 - w_1)x_2) \quad (2.45)$$

Para determinar la variable P_z , se sustituirán las ecuaciones (2.39) y (2.43) en (2.30), dando lugar a la siguiente ecuación:

$$a_z^2 + b_z + c = 0 \quad (2.46)$$

donde:

$$a = a_1^2 + a_2^2 + d^2 \quad (2.47)$$

$$b = 2(a_1b_1 + a_2(b_2 - y_1d) - z_1d^2) \quad (2.48)$$

$$c = b_1^2 + d^2(z_1^2 - L_b^2) + (b_2 - y_1d)^2 \quad (2.49)$$

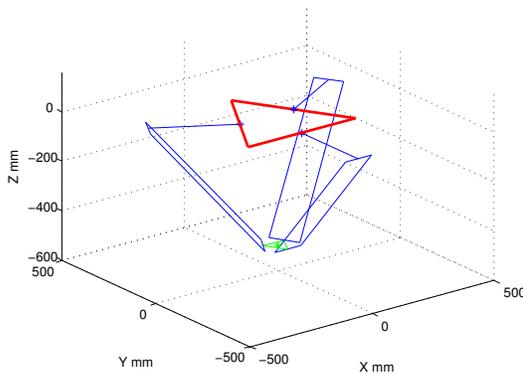
(2.46) es una ecuación algebraica de segundo grado, que entrega dos resultados, ya que las esferas virtuales se intersectan en más de un punto, la solución para P_z será el valor mas negativo de la solución de (2.46).

2.4.1. Ejemplos de solución de cinemática directa

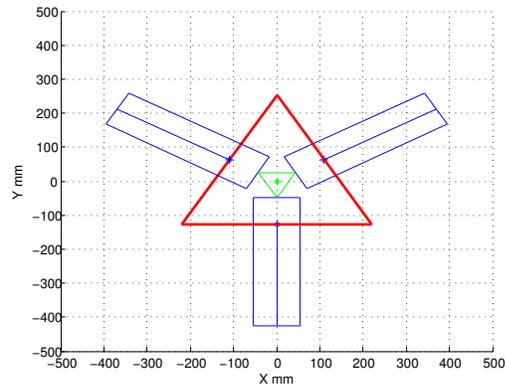
Dentro de esta sección se proponen 2 casos en los cuales se hace uso del algoritmo de cinemática directa, en ambos casos se desea conocer la posición de la plataforma móvil del RPD en coordenadas cartesianas cuando los actuadores se desplazan a determinadas posiciones angulares propuestas.

Caso 1

Se desea conocer la posición de la plataforma móvil cuando los actuadores se desplazan $\Theta = \{0 \ 0 \ 0\}^T$, empleando el algoritmo de cinemática inversa se obtiene: $\{^A\mathbf{P}_P\} = \{0 \ 0 \ -497\}^T$. Las unidades del desplazamiento cartesiano están dadas en milímetros.



(a) Vista isométrica



(b) Vista plano xy

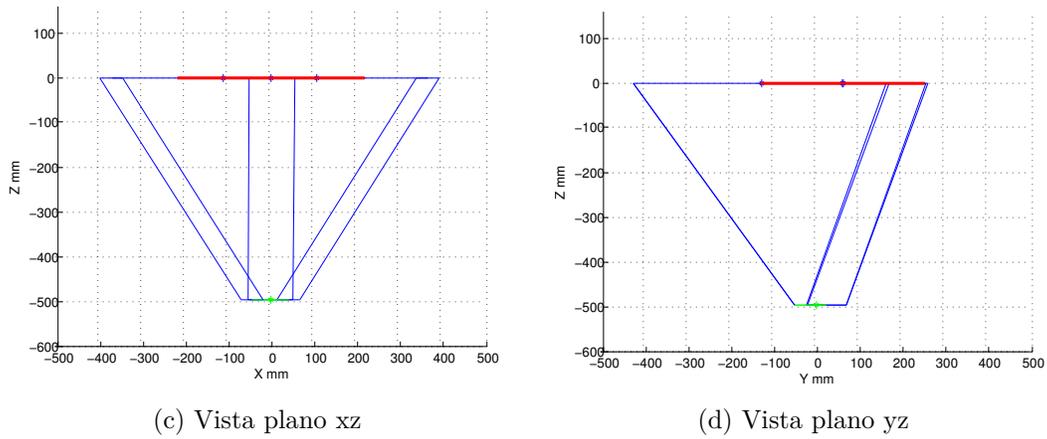
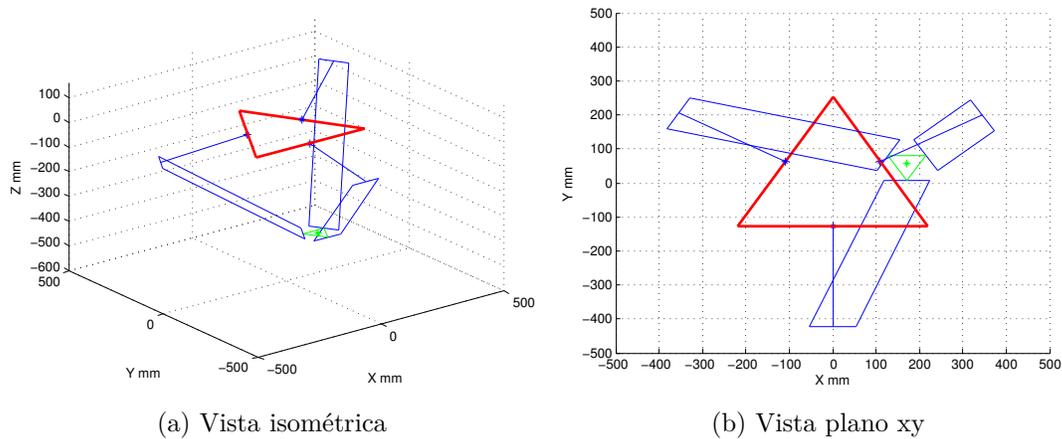


Figura 2.7: Representación gráfica del primer ejemplo de solución de cinemática directa

La Figura 2.7 muestra la posición de inicio del RPD, esto debido a que los valores de desplazamiento propuestos para los tres actuadores son iguales a cero, y al aplicar el algoritmo de cinemática directa se puede conocer cual es la posición inicial del robot en coordenadas cartesianas.

Caso 2

Similar al caso anterior, se desea conocer la posición de la plataforma móvil cuando los actuadores se desplazan $\Theta = \{10 \quad -25 \quad 19\}^T$, empleando el algoritmo de cinemática inversa se obtiene: $\{^A\mathbf{P}_P\} = \{171,3 \quad 56,2 \quad -471,78\}^T$



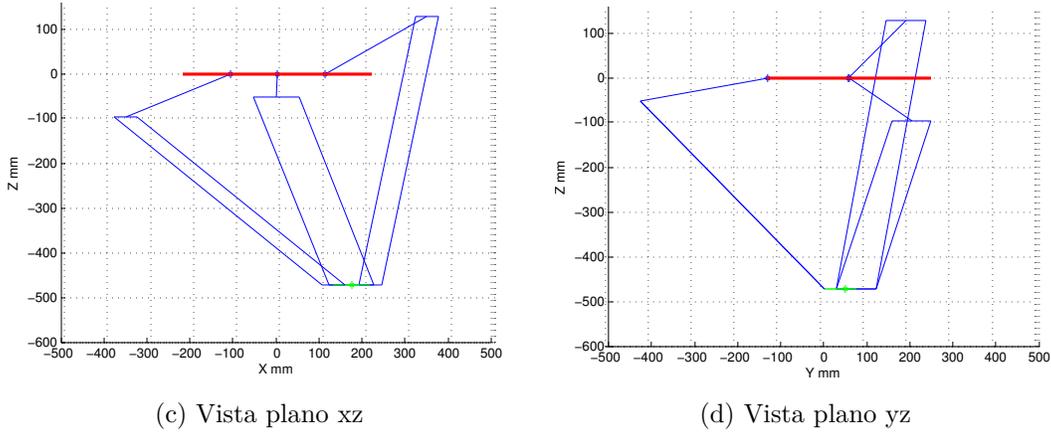


Figura 2.8: Representación gráfica del segundo ejemplo de solución de cinemática directa

El ejemplo del segundo caso simplemente se proponen de manera aleatoria los desplazamientos de los tres actuadores, y como se aprecia en la Figura 2.8, la plataforma móvil es llevada a la posición calculada en coordenadas cartesianas.

2.5. Cinemática diferencial

En esta sección se muestra la manera en que se obtiene la matriz jacobiana del Robot delta [2.5], la matriz jacobiana se emplea para distintos análisis matemáticos entre los cuales destaca el análisis de singularidades. La matriz jacobiana está dada por la siguiente ecuación:

$$\mathbf{J} = \mathbf{J}_q^{-1} \mathbf{J}_x \quad (2.50)$$

donde J_x es la matriz jacobiana para la velocidad lineal del RPD, y J_q es la matriz jacobiana para la velocidad angular de los actuadores del RPD. Es necesario derivar el sistema de expresiones matemáticas conformado por: (2.14), (2.15), (2.16) con el fin de obtener las matrices jacobianas del RPD. El sistema de ecuaciones resultantes es el siguiente:

$$\begin{aligned} & 2L_a \dot{P}_y \cos\theta_1 - 2L_a(P_y - c)\dot{\theta}_1 \sin\theta_1 + 2L_a \dot{P}_z \sin\theta_1 + \\ & 2L_a P_z \dot{\theta}_1 \cos\theta_1 + 2P_x \dot{P}_x + 2(P_y + c)\dot{P}_y + 2P_z \dot{P}_z = 0 \end{aligned} \quad (2.51)$$

$$\begin{aligned} & -L_a(\sqrt{3}\dot{P}_x + \dot{P}_y)\cos\theta_2 + L_a(\sqrt{3}(P_x + d) + P_y + e)\dot{\theta}_2 \sin\theta_2 + \\ & 2L_a \dot{P}_z \sin\theta_2 + 2L_a P_z \dot{\theta}_2 \cos\theta_2 + 2(P_x + d)\dot{P}_x + 2(P_y + e)\dot{P}_y + 2P_z \dot{P}_z = 0 \end{aligned} \quad (2.52)$$

$$\begin{aligned} & L_a(\sqrt{3}\dot{P}_x - \dot{P}_y)\cos\theta_3 - L_a(\sqrt{3}(P_x - d) - P_y - e)\dot{\theta}_3 \sin\theta_3 + \\ & 2L_a \dot{P}_z \sin\theta_3 + 2L_a P_z \dot{\theta}_3 \cos\theta_3 + 2(P_x - d)\dot{P}_x + 2(P_y + e)\dot{P}_y + 2P_z \dot{P}_z = 0 \end{aligned} \quad (2.53)$$

Las ecuaciones anteriores, deben ser ordenadas de tal forma que la velocidad lineal pueda ser representada en una matriz y los términos de velocidad angular en otra

matriz. El conjunto de ecuaciones diferenciales anteriores se pueden ordenar de forma matricial como sigue:

$$\mathbf{J}_x \dot{\mathbf{p}} = \mathbf{J}_q \dot{\boldsymbol{\theta}} \quad (2.54)$$

Realizando la sustitución de (2.51), (2.52) y (2.53) en (2.54) se tiene:

$$\begin{aligned} \begin{bmatrix} P_x & P_y + c + L_a \cos \theta_1 & P_z + L_a \sin \theta_1 \\ 2(P_x + d) - \sqrt{3}L_a \cos \theta_2 & 2(P_y + e) - L_a \cos \theta_2 & 2(P_z + L_a \sin \theta_2) \\ 2(P_x - d) + \sqrt{3}L_a \cos \theta_3 & 2(P_y + e) - L_a \cos \theta_3 & 2(P_z + L_a \sin \theta_3) \end{bmatrix} \begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \end{bmatrix} \\ = \begin{bmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \end{aligned} \quad (2.55)$$

donde:

$$\begin{aligned} b_{11} &= L_a[(P_y + c) \sin \theta_1 - P_z \cos \theta_1] \\ b_{22} &= -L_a[(\sqrt{3}(P_x + d) + P_y + e) \sin \theta_2 + 2P_z \cos \theta_2] \\ b_{33} &= L_a[(\sqrt{3}(P_x - d) - P_y - e) \sin \theta_3 - 2P_z \cos \theta_3] \end{aligned}$$

Por lo tanto se tiene que:

$$\mathbf{J}_x = \begin{bmatrix} P_x & P_y + c + L_a \cos \theta_1 & P_z + L_a \sin \theta_1 \\ 2(P_x + d) - \sqrt{3}L_a \cos \theta_2 & 2(P_y + e) - L_a \cos \theta_2 & 2(P_z + L_a \sin \theta_2) \\ 2(P_x - d) + \sqrt{3}L_a \cos \theta_3 & 2(P_y + e) - L_a \cos \theta_3 & 2(P_z + L_a \sin \theta_3) \end{bmatrix} \quad (2.56)$$

$$\mathbf{J}_q = \begin{bmatrix} b_{11} & 0 & 0 \\ 0 & b_{22} & 0 \\ 0 & 0 & b_{33} \end{bmatrix} \quad (2.57)$$

Las singularidades en robots paralelos se pueden clasificar en tres categorías las cuales se detallan de la siguiente manera [2.6].

2.5.1. Singularidad de cinemática inversa

Haciendo referencia a (2.54), la singularidad de cinemática inversa ocurre cuando la matriz \mathbf{J}_q es de rango deficiente, debido a que todos los manipuladores paralelos tienen la matriz \mathbf{J}_q de tamaño $m \times m$, esta singularidad ocurrirá cuando:

$$\det(\mathbf{J}_q) = 0 \quad (2.58)$$

Las configuraciones correspondientes están localizadas en el límite del espacio de trabajo del manipulador o en límites internos entre sub regiones del espacio de trabajo donde el número de soluciones del problema cinemático inverso no es el mismo. En otras palabras el movimiento infinitesimal de la plataforma móvil a través de ciertas direcciones no puede ser realizado.

2.5.2. Singularidad de cinemática directa

Tomando como referencia a (2.54), la singularidad de cinemática directa ocurre cuando la matriz \mathbf{J}_x es de rango deficiente, en robots paralelos, la matriz \mathbf{J}_x es de tamaño $m \times n$ donde $m > n$, la singularidad ocurrirá cuando:

$$\det(\mathbf{J}_x^T \mathbf{J}_x) = 0 \quad (2.59)$$

Sin embargo si la matriz \mathbf{J}_x fuera cuadrada, es decir $m = n$, entonces la singularidad de cinemática directa ocurrirá cuando se cumpla el siguiente determinante.

$$\det(\mathbf{J}_x) = 0 \quad (2.60)$$

Opuesto a la singularidad de cinemática inversa, este tipo de degeneración puede ocurrir dentro del espacio de trabajo cartesiano, en robots paralelos esta singularidad es única debido a que este tipo de singularidad no puede ocurrir en robots seriales, donde el problema cinemático directo siempre tiende a una única solución.

2.5.3. Singularidad combinada

La singularidad combinada ocurre cuando \mathbf{J}_x y \mathbf{J}_q son de rango deficiente, y corresponde a una degeneración de las ecuaciones de posición y orientación, es decir:

$$\det(\mathbf{J}_q) = 0 \quad \text{y} \quad \det(\mathbf{J}_x) = 0 \quad (2.61)$$

Algunas condiciones en los parámetros geométricos del robot son requeridas para que este tipo de singularidad ocurra.

2.6. Espacio de trabajo

El espacio de trabajo, es la colección de puntos que pueden ser alcanzados por el efector final del robot, al estar directamente relacionado con los eslabonamientos del robot y su longitud. Para calcular el espacio de trabajo del RPD se utiliza el algoritmo de cinemática inversa, en el cual, se propone un conjunto de desplazamientos lineales en coordenadas x, y, z . El algoritmo evalúa diversos puntos para determinar si son alcanzables, si los resultados de los valores de desplazamiento angular son complejos, significa que la coordenada es inalcanzable. El espacio de trabajo se construye mediante las soluciones reales que arroja el algoritmo de la cinemática inversa. Los parámetros geométricos usados para el cálculo del espacio de trabajo son los siguientes:

$$f_a = 439\text{mm}, f_p = 86\text{mm}, L_a = 300, L_b = 624.$$

En la Figura 2.9 se muestra el espacio de trabajo calculado, las gráficas son muy útiles para conocer que puntos pueden ser alcanzados por el robot [2.7].

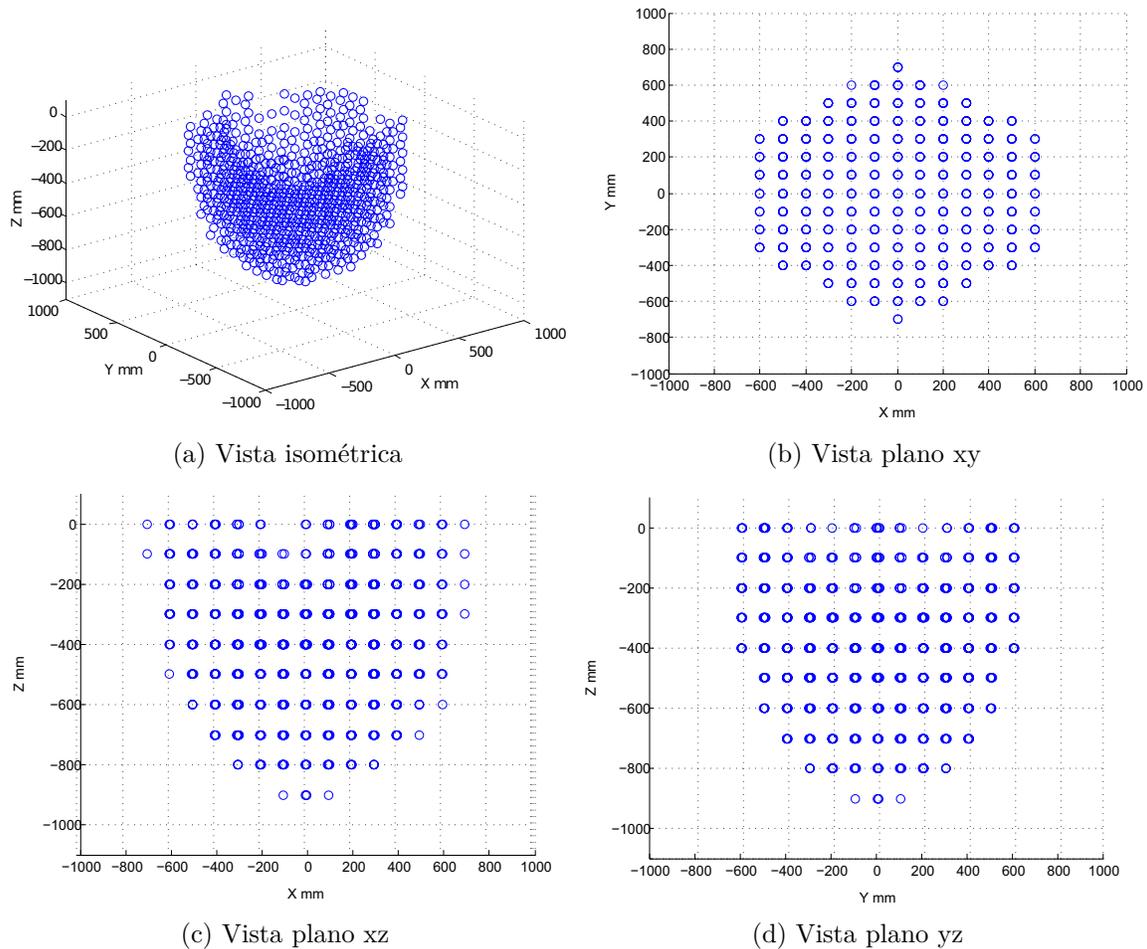


Figura 2.9: Distintas representaciones del espacio de trabajo

2.7. Conclusiones

En el presente capítulo se detalla un método geométrico que permite establecer las ecuaciones que relacionan cadenas cinemáticas del RPD y las plataformas fija y móvil. Estableciendo las ecuaciones que describen la geometría del RPD es posible dar solución al problema cinemático inverso el cual es fundamental para conocer el desplazamiento angular que deben proporcionar los actuadores del robot en base a la posición deseada del efector final. Sin embargo para dar solución al problema cinemático directo es necesario emplear un método de trilateración. Se establecen ciertos criterios los cuales determinan si existe una singularidad en el robot, una singularidad representa una posición a la cual el robot no puede llegar debido a las restricciones de las cadenas cinemáticas cerradas y en base al análisis de singularidades es posible establecer el espacio de trabajo para el RPD.

Referencias

- [2.1] K. Rosquist, “Modelling and control of a parallel kinematic robot,” 2013.
 - [2.2] R. Clavel, “Delta, a fast robot with parallel geometry,” in *Proc. 18th Int. Symp. on Industrial Robots, Lausanne, 1988*, pp. 91–100, 1988.
 - [2.3] R. L. W. II, “The delta parallel robot: Kinematics solutions,” *Mechanical Engineering, Ohio University*, pp. 1–21, 2015.
 - [2.4] J. M. García, “Inverse - forward kinematics of a delta robot,” pp. 5–7, 2010.
 - [2.5] H. A. H. Alashqar, “Modeling and high precision motion control of 3 dof parallel delta robot manipulator,” pp. 22–26, 2007.
 - [2.6] H. D. Taghirad, *Parallel robots: mechanics and control*. CRC press, 2013.
 - [2.7] M. Maya, E. Castillo, A. Lomelí, E. González-Galván, and A. Cárdenas, “Workspace and payload-capacity of a new reconfigurable delta parallel robot,” *International Journal of Advanced Robotic Systems*, vol. 10, 2013.
-

3

Dinámica del RPD

3.1. Introducción

A diferencia del análisis cinemático, la dinámica estudia el movimiento de un sistema involucrando las fuerzas que producen el movimiento.

Determinar el modelo dinámico del RPD es muy importante por varios motivos; destacan el poder realizar una simulación de sistema bajo ciertas condiciones operativas sin tener que construir un prototipo, el siguiente motivo es proponer una ley de control y ver la respuesta transitoria del sistema. La ventaja de realizar simulaciones es que permite someter el sistema a diversas condiciones operativas, y de esta manera observar su comportamiento mediante la respuesta transitoria del sistema.

Las principales formulaciones propuestas para aproximar el modelo dinámico de los robots paralelos son: [3.1]:

- Formulación Newton-Euler
- El principio de los trabajos virtuales
- Formulación Lagrangiana

Mediante el modelo dinámico por Newton-Euler, es necesario formular las ecuaciones de movimiento de cada elemento del robot y debido a la naturaleza del RPD las ecuaciones resultantes serán muchas, esto conllevará a un alto consumo de recursos computacionales para poder simular la dinámica del sistema [3.2].

El principio de los trabajos virtuales implica que el trabajo debido a fuerzas externas que corresponden a cualquier conjunto de desplazamiento virtual es cero. Aplicando este principio para analizar la dinámica de un sistema, no implica fuerzas y momentos de restricción desconocidos. Por lo tanto es más eficiente que Newton-Euler y la formulación Lagrangiana. Sin embargo al calcular los torques de los actuadores, se debe considerar la matriz inversa de la matriz jacobiana transpuesta; lo cual hace que este método se vuelva inestable para configuraciones singulares [3.1].

3.2. Formulación Lagrangiana

La formulación Lagrangiana es una reformulación de la mecánica clásica introducida por Joseph-Louis de Lagrange en 1788. Esta formulación combina la conservación del momento con la conservación de la energía. Las Ecuaciones de Lagrange (también conocidas como Ecuaciones de Euler-Lagrange) nos permiten contar con un sistema analítico para llegar a las ecuaciones que describen el comportamiento físico de un sistema, en este caso el RPD

La formulación Lagrangiana puede escribirse de la siguiente manera [3.3]:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} = Q_j + \sum_{i=1}^k \lambda_i \frac{\partial \Gamma_i}{\partial q_j} \quad \text{para } j = 1, \dots, n \quad (3.1)$$

donde:

L = Función de Lagrange

q_j = Coordenada generalizada j

λ_i = Multiplicador de Lagrange

Γ_i = Ecuación de restricción cinemática

Q_j = Fuerza externa generalizada

i = Índice de la restricción

j = Índice de la coordenada generalizada

n = Número de coordenadas generalizadas

El análisis dinámico se desarrolla usando solo tres coordenadas generalizadas. Sin embargo, debido a la complejidad cinemática, se incluyen las tres coordenadas redundantes (p_x, p_y, p_z) , con lo cual se obtienen las 6 coordenadas generalizadas que se definen en el siguiente vector [3.4]:

$$\mathbf{q} = [p_x, p_y, p_z, \theta_1, \theta_2, \theta_3] \quad (3.2)$$

La Figura 3.1 muestra el esquema de una cadena cinemática, el cual sirve de apoyo para formular las ecuaciones dinámicas del Robot Delta.

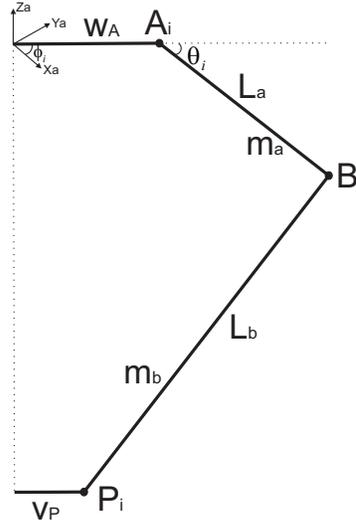


Figura 3.1: Esquema de una cadena cinemática para la obtención del modelo dinámico del RPD

Para determinar la dinámica del sistema es necesario acoplar dos sistemas de ecuaciones basados en (3.1), el primer sistema contiene los multiplicadores de Lagrange como variables desconocidas, el primer sistema es descrito como:

$$\sum_{i=1}^k \lambda_i \frac{\partial \Gamma}{\partial q_j} = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} - \hat{Q}_j \quad \text{para} \quad j = 1, 2, 3 \quad (3.3)$$

donde \hat{Q}_j denota la fuerza generalizada contribuida por una fuerza externa aplicada. El segundo sistema de ecuaciones relaciona las fuerzas de actuación y puede ser escrito como:

$$Q_j = \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_j} \right) - \frac{\partial L}{\partial q_j} - \sum_{i=1}^k \lambda_i \frac{\partial \Gamma}{\partial q_j} \quad \text{para} \quad j = 4, 5, 6 \quad (3.4)$$

donde Q_j es el torque del actuador. Las ecuaciones de restricción están dadas por:

$$\begin{aligned} \Gamma_i &= \overline{B_i P_i}^2 - L_b^2 = 0 \\ &= (p_x + v_P \cos \phi_i - w_A \cos \phi_i - L_a \cos \phi_i \cos \theta_i)^2 \\ &\quad + (p_y + v_P \sin \phi_i - w_A \sin \phi_i - L_a \sin \phi_i \sin \theta_i)^2 \\ &\quad + (p_z - L_a \sin \theta_i)^2 - L_b^2 \quad \text{para} \quad i = 1, 2, 3 \end{aligned} \quad (3.5)$$

donde ϕ_i es el ángulo de separación entre cada cadena cinemática del Robot Delta, donde: $\phi_1 = 0^\circ$, $\phi_2 = 120^\circ$, $\phi_3 = 240^\circ$

3.2.1. Energía cinética

La energía cinética total del Robot Delta se obtiene mediante la sumatoria de las energías de cada uno de las articulaciones por medio de la siguiente formula:

$$K = K_m + \sum_{i=1}^3 (K_{ai} + K_{bi}) \quad (3.6)$$

donde K_m es la energía cinética de la plataforma móvil, K_{ai} representa la energía cinética para cada uno de los brazos, y K_{bi} es la energía cinética para cada antebrazo del Robot Delta. La energía cinética de la plataforma móvil, está dada por medio de la siguiente expresión:

$$K_p = \frac{1}{2} m_p (\dot{p}_x^2 + \dot{p}_y^2 + \dot{p}_z^2) \quad (3.7)$$

donde m_p es la masa de la plataforma móvil; la energía cinética para cada brazo del RPD se calcula por a través de la siguiente ecuación.

$$K_{ai} = \frac{1}{2} \left(I_m + \frac{1}{3} m_a L_a^2 \right) \dot{\theta}_i^2 \quad (3.8)$$

donde I_m es el momento de inercia del motor; la energía cinética de cada antebrazo del RPD se obtiene mediante la siguiente ecuación:

$$K_{bi} = \frac{1}{2} m_b (\dot{p}_x^2 + \dot{p}_y^2 + \dot{p}_z^2) + \frac{1}{6} m_b L_b^2 \dot{\theta}_i^2 \quad (3.9)$$

Debido a la presencia de articulaciones esfericas pasivas, el antebrazo tiene un movimiento general en el espacio, por lo que se tiene movimiento traslacional y rotacional.

3.2.2. Energía potencial

Debido a que el robot delta es un robot tipo espacial y tiene movimiento vertical en sus miembros, es necesario el planteamiento de la ecuación que describa la energía potencial del sistema. La energía potencial esta relacionada directamente con la aceleración de la gravedad, para el caso de robots planares cuyo desplazamiento se encuentre sobre un plano horizontal, el valor de esta energía siempre será cero. La energía potencial del Robot Delta esta dada por la siguiente ecuación:

$$U = U_p + \sum_{i=1}^3 (U_{ai} + U_{bi}) \quad (3.10)$$

donde U_p es la energía potencial de la plataforma móvil, U_{ai} es la energía potencial del brazo, y U_{bi} es la energía potencial del antebrazo. Las expresiones para determinar el valor de la magnitud de las energías que requiere (3.10), se pueden obtener como sigue:

$$U_p = -m_p g_c p_z \quad (3.11)$$

$$U_{ai} = -m_a g_c L_a \sin \theta_i \quad (3.12)$$

$$U_{bi} = -m_b g_c (p_z + L_b \sin \theta_i) \quad (3.13)$$

3.2.3. Función de Lagrange

La función de Lagrange está definida por la siguiente expresión:

$$L = K - U \quad (3.14)$$

donde K es la energía cinética del RPD y U , es la energía potencial. Substituyendo (3.6) y (3.10) en (3.14) se obtiene la siguiente ecuación de Lagrange:

$$L = \frac{1}{2}(m_p + 3m_b)(\dot{p}_x^2 + \dot{p}_y^2 + \dot{p}_z^2) + \frac{1}{2} \left(I_m + \frac{1}{3}m_a L_a^2 + m_b L_b^2 \right) \sum_{i=1}^3 \dot{\theta}_i^2 \quad (3.15)$$

$$+ (m_p + 3m_b)g_c p_z + g_c L_a (m_a + m_b) \sum_{i=1}^3 \sin \theta_i$$

Como se planteo anteriormente, se tienen dos sistemas de ecuaciones basados en la ecuación 3.1, por lo que se realizaran las derivadas parciales del primer sistema de ecuaciones, el cual es el sistema de las restricciones con respecto a las seis variables generalizadas, Γ_i , con lo que se obtiene:

$$\frac{\partial \Gamma_i}{\partial p_x} = 2(p_x + v_P \cos \phi_i - w_A \cos \phi_i - L_a \cos \phi_i \cos \theta_i) \quad \text{para } i = 1, 2, 3$$

$$\frac{\partial \Gamma_i}{\partial p_y} = 2(p_y + v_P \cos \phi_i - w_A \cos \phi_i - L_a \cos \phi_i \cos \theta_i) \quad \text{para } i = 1, 2, 3$$

$$\frac{\partial \Gamma_i}{\partial p_z} = 2(p_z - L_a \sin \theta_i) \quad \text{para } i = 1, 2, 3$$

$$\frac{\partial \Gamma_1}{\partial \theta_1} = 2L_a((p_x \cos \phi_1 + p_y \sin \phi_1 - w_A + v_P) \sin \theta_1 - p_z \cos \theta_1)$$

$$\frac{\partial \Gamma_2}{\partial \theta_2} = 2L_a((p_x \cos \phi_2 + p_y \sin \phi_2 - w_A + v_P) \sin \theta_2 - p_z \cos \theta_2)$$

$$\frac{\partial \Gamma_3}{\partial \theta_3} = 2L_a((p_x \cos \phi_3 + p_y \sin \phi_3 - w_A + v_P) \sin \theta_3 - p_z \cos \theta_3)$$

$$\frac{\partial \Gamma_i}{\partial \theta_1} = 0 \quad \text{para } i = 2, 3$$

$$\frac{\partial \Gamma_i}{\partial \theta_2} = 0 \quad \text{para } i = 1, 3$$

$$\frac{\partial \Gamma_i}{\partial \theta_3} = 0 \quad \text{para } i = 1, 2$$

Realizando las derivadas parciales para cada una de las seis coordenadas generalizadas en el segundo sistema se obtiene:

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{x}_p} \right) &= (m_p + 3m_b)\ddot{x}_p & \frac{\partial L}{\partial x_p} &= 0 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{y}_p} \right) &= (m_p + 3m_b)\ddot{y}_p & \frac{\partial L}{\partial y_p} &= 0 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{z}_p} \right) &= (m_p + 3m_b)\ddot{z}_p & \frac{\partial L}{\partial z_p} &= (m_p + 3m_b)g_c \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_1} \right) &= \left(I_m + \frac{1}{3}m_a L_a^2 + \frac{1}{6}m_b L_b^2 \right) \ddot{\theta}_1 & \frac{\partial L}{\partial \theta_1} &= (m_a + m_b)g_c L_a \cos \theta_1 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_2} \right) &= \left(I_m + \frac{1}{3}m_a L_a^2 + \frac{1}{6}m_b L_b^2 \right) \ddot{\theta}_2 & \frac{\partial L}{\partial \theta_2} &= (m_a + m_b)g_c L_a \cos \theta_2 \\ \frac{d}{dt} \left(\frac{\partial L}{\partial \dot{\theta}_3} \right) &= \left(I_m + \frac{1}{3}m_a L_a^2 + \frac{1}{6}m_b L_b^2 \right) \ddot{\theta}_3 & \frac{\partial L}{\partial \theta_3} &= (m_a + m_b)g_c L_a \cos \theta_3 \end{aligned}$$

Realizando la sustitución de las derivadas parciales en (3.3), se tiene como resultado:

$$2 \sum_{i=1}^3 \lambda_i (p_x + v_P \cos \phi_i - w_A \cos \phi_i - L_a \cos \phi_i \cos \theta_i) = (m_p + 3m_b)\ddot{p}_x - f_{px} \quad (3.16)$$

$$2 \sum_{i=1}^3 \lambda_i (p_y + v_P \sin \phi_i - w_A \sin \phi_i - L_a \sin \phi_i \cos \theta_i) = (m_p + 3m_b)\ddot{p}_y - f_{py} \quad (3.17)$$

$$2 \sum_{i=1}^3 \lambda_i (p_z - L_a \sin \theta_i) = (m_p + 3m_b)\ddot{p}_z + (m_p + 3m_b)g_c - f_{pz} \quad (3.18)$$

donde f_{px} , f_{py} , y f_{pz} , son las componentes x , y , z de una fuerza externa en la plataforma móvil. Los coeficientes λ_1 , λ_2 , y λ_3 , son los multiplicadores de Lagrange y se obtienen al solucionar el sistema de ecuaciones (3.16) -(3.18).

Una vez encontrados los multiplicadores de Lagrange estos se sustituyen en el siguiente sistema de ecuaciones, el cual es empleado para calcular los torques de los actuadores τ_i .

$$\begin{aligned} \tau_1 &= \left(I_m + \frac{1}{3}m_a L_a^2 + \frac{1}{3}m_b L_b^2 \right) \ddot{\theta}_1 + (m_a + m_b)g_c L_a \cos \theta_1 \\ &\quad - 2L_a \lambda_1 ((p_x \cos \varphi_1 + p_y \sin \varphi_1 - w_A + v_P) \sin \theta_1 - p_z \cos \theta_1) \end{aligned} \quad (3.19)$$

$$\begin{aligned} \tau_2 &= \left(I_m + \frac{1}{3}m_a L_a^2 + \frac{1}{3}m_b L_b^2 \right) \ddot{\theta}_2 + (m_a + m_b)g_c L_a \cos \theta_2 \\ &\quad - 2L_a \lambda_2 ((p_x \cos \varphi_2 + p_y \sin \varphi_2 - w_A + v_P) \sin \theta_2 - p_z \cos \theta_2) \end{aligned} \quad (3.20)$$

$$\begin{aligned} \tau_3 &= \left(I_m + \frac{1}{3}m_a L_a^2 + \frac{1}{3}m_b L_b^2 \right) \ddot{\theta}_3 + (m_a + m_b)g_c L_a \cos \theta_3 \\ &\quad - 2L_a \lambda_3 ((p_x \cos \varphi_3 + p_y \sin \varphi_3 - w_A + v_P) \sin \theta_3 - p_z \cos \theta_3) \end{aligned} \quad (3.21)$$

Para poder realizar la simulación de la dinámica del sistema, se emplean los parámetros mostrados en la siguiente Tabla:

Tabla 3.1: Parámetros empleados para la simulación de la dinámica del Robot Delta

Nombre	Descripción	Valor
f_A	Longitud de lado de plataforma fija	439 mm
f_P	Longitud de lado de plataforma móvil	86 mm
L_a	Longitud del brazo del Robot	300 mm
L_b	Longitud del antebrazo del Robot	624 mm
m_a	Masa de cada brazo del Robot	0.29 Kg
m_b	Masa de cada antebrazo del Robot	0.28 Kg
m_p	Masa de la plataforma móvil	0.19 Kg
I_m	Momento de inercia del motor	$3.8E^{-6} \text{ Kg}m^2$

3.2.4. Simulación

Para realizar la simulación a lazo abierto del sistema, se proponen los siguientes pares para los actuadores de los motores $\tau = \{0,98 \quad 1 \quad 0,99\}^T$, con las perturbaciones externas al sistema son iguales a cero; las unidades del par de los motores están dadas en Newton-metro (Nm).

El diagrama mostrado en la Figura 3.2 es empleado para realizar la simulación del sistema a lazo abierto sin control.

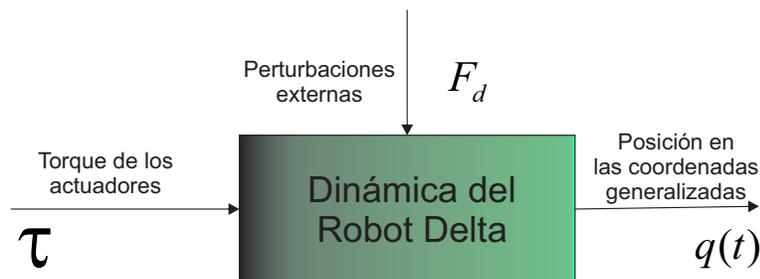


Figura 3.2: Diagrama de bloques de la dinámica directa

La Figura 3.3 muestra la respuesta del sistema a lazo abierto cuando se aplica un par directamente a los actuadores del robot.

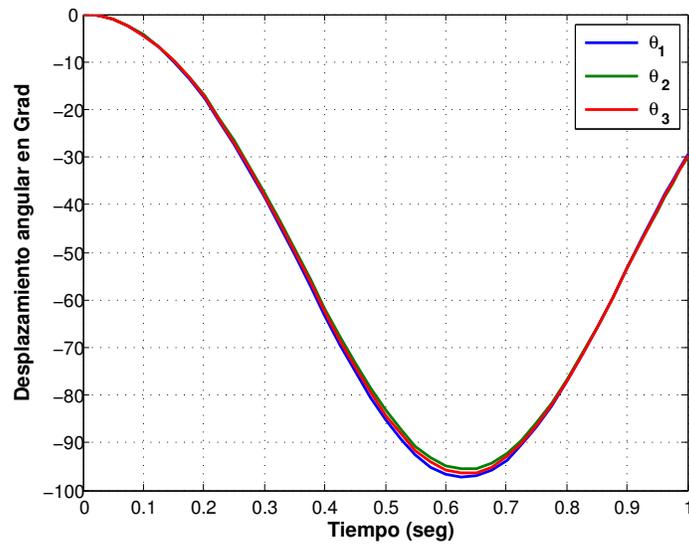


Figura 3.3: Desplazamiento angular del RPD sin control

Como se aprecia en la Figura 3.3 el la respuesta de los ángulos θ_i presentan oscilaciones, esto se debe a que no existe ninguna referencia a seguir por parte de los actuadores además de que no hay ninguna ley de control.

La Figura 3.4 muestra el desplazamiento de la plataforma móvil durante la simulación, para lo cual se hace uso del algoritmo de cinemática directa.

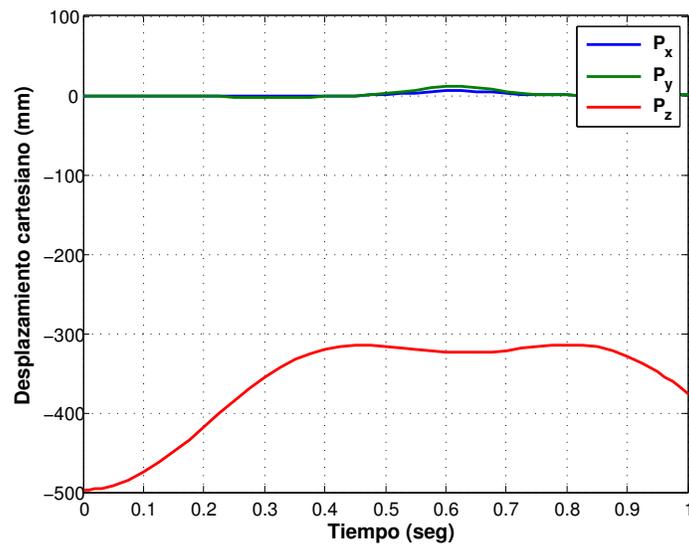


Figura 3.4: Desplazamiento cartesiano del RPD sin control

En la Figura 3.4 se puede observar que la plataforma móvil se desplaza sobre el eje z , este movimiento se debe a que el par aplicado a los actuadores es similar, por lo tanto

el desplazamiento sobre los otros 2 ejes x y y es casi nulo.

3.3. Conclusiones

En el presente capítulo se describen las 3 principales aproximaciones que permiten obtener el modelo dinámico del robot delta las cuales son Newton-Euler, principio de los trabajos virtuales, y formulación Lagrangiana. En este caso para obtener la dinámica del RPD, se eligió la formulación Lagrangiana, debido a que esta formulación se basa en energías; por lo tanto basta únicamente conocer la energía cinética y potencial de cada cadena cinemática para poder hacer una aproximación de la dinámica. Este método simplifica el cálculo a diferencia del método de Newton-Euler por ejemplo, en el cual hay que desarrollar ecuaciones basadas la primer ley de Newton las cuales describan el movimiento para cada componente del RPD. Dentro de este capítulo se presento una simulación del sistema a lazo abierto aplicando un par a cada actuador y obteniendo una respuesta oscilatoria; sin embargo es necesario implementar un controlador para mejorar la respuesta transitoria del sistema, el cual deberá seguir una referencia deseada.

Referencias

- [3.1] J. Brinker, B. Corves, and M. Wahle, “A comparative study of inverse dynamics based on clavel,” in *Proceedings of the 14th IFToMM World Congress*, pp. 89–98, IEEE, 2015.
 - [3.2] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators*. John Wiley & Sons, 1999.
 - [3.3] S. B. Park, H. S. Kim, C. Song, and K. Kim, “Dynamics modeling of a delta-type parallel robot (ISR 2013),” in *Robotics (ISR), 2013 44th International Symposium on*, pp. 1–5, IEEE, 2013.
 - [3.4] M. Ahmadi, M. Dehghani, M. Eghtesad, and A. R. Khayatian, “Inverse dynamics of hexa parallel robot using lagrangian dynamics formulation,” in *2008 International Conference on Intelligent Engineering Systems*, pp. 145–149, IEEE, 2008.
-

4

Control de movimiento del RPD

4.1. Introducción

Los robots paralelos esencialmente están diseñados para dos tipos de aplicaciones. En el primer tipo de aplicación, la plataforma móvil del robot sigue una posición y orientación deseada. El segundo tipo de aplicación, la plataforma móvil esta en contacto con objetos para poder ser desplazados de un punto a otro para lo cual es necesario generar suficiente fuerza/torque para poder mover las masas. En este capítulo se pondrán algoritmos de control para hacer la primer aplicación, primeramente controlar el desplazamiento de un punto a otro y seguir una trayectoria

Usando el algoritmo de cinemática inversa, es posible calcular el desplazamiento angular de los tres motores, en base al desplazamiento cartesiano de la plataforma móvil; sin embargo debido a la complejidad de la dinámica del sistema, el robot no puede alcanzar las posiciones deseadas, en lazo abierto por lo cual es necesario el implementar una ley de control.

4.2. Consideraciones para implementar un controlador

En el control de movimiento de un robot paralelo se asume que el controlador calcula los torques requeridos para los actuadores, los cuales llevaran al robot a seguir una trayectoria deseada.

De las ecuaciones diferenciales que describen la dinámica del robot, se pueden reescribir de la siguiente forma: [4.1]:

$$\mathbf{M} \frac{d^2 \boldsymbol{\Theta}(t)}{dt^2} + \mathbf{G}(\boldsymbol{\Theta}(t)) + \mathbf{R}(\boldsymbol{\Theta}(t), \mathbf{P}(t)) \boldsymbol{\lambda}(t) = \boldsymbol{\tau}(t) \quad (4.1)$$

donde $\boldsymbol{\Theta}(t) = [\theta_1 \ \theta_2 \ \theta_3]^T$, es el desplazamiento angular de los actuadores, $\mathbf{P}(t) = [p_x(t) \ p_y(t) \ p_z(t)]^T$, es la posición del efector final, $\boldsymbol{\lambda}(t) = [\lambda_1 \ \lambda_2 \ \lambda_3]^T$, es el vector

de los multiplicadores de Lagrange, $\boldsymbol{\tau}(t) = [\tau_1 \ \tau_2 \ \tau_3]^T$, es el vector de entrada de los pares; M es la matriz de inercia, $\mathbf{G}(\boldsymbol{\Theta})$, es el vector de pares gravitacionales, $\mathbf{R}(\boldsymbol{\Theta}, \mathbf{P})$, es el vector de restricciones del robot delta. Las matrices M , $\mathbf{G}(\boldsymbol{\Theta})$, $\mathbf{R}(\boldsymbol{\Theta}, \mathbf{P})$, están dadas de la siguiente forma:

$$\mathbf{M} = I_m + \frac{1}{3}m_a L_a^2 + \frac{1}{3}m_b L_b^2 \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.2)$$

$$\mathbf{G}(\boldsymbol{\Theta}) = (m_a + m_b)gL_a \begin{bmatrix} \cos\theta_1 \\ \cos\theta_2 \\ \cos\theta_3 \end{bmatrix} \quad (4.3)$$

$$\mathbf{R}(\boldsymbol{\Theta}, \mathbf{P}) = \begin{bmatrix} -2L_a^2\lambda_1((p_x\cos\varphi_1 + p_y\sin\varphi_1 + v_P - w_A)\sin\theta_1 - p_z\cos\theta_1) \\ -2L_a^2\lambda_2((p_x\cos\varphi_2 + p_y\sin\varphi_2 + v_P - w_A)\sin\theta_2 - p_z\cos\theta_2) \\ -2L_a^2\lambda_3((p_x\cos\varphi_3 + p_y\sin\varphi_3 + v_P - w_A)\sin\theta_3 - p_z\cos\theta_3) \end{bmatrix} \quad (4.4)$$

Con motivo de hacer un análisis y después proponer una ley de control, la ecuación 4.1 puede ser escrita de la siguiente forma:

$$\frac{d^2\boldsymbol{\Theta}(t)}{dt^2} = \mathbf{M}^{-1}\boldsymbol{\tau}(t) + \mathbf{M}^{-1}[\mathbf{G}(\boldsymbol{\Theta}(t)) + \mathbf{R}(\boldsymbol{\Theta}(t), \mathbf{P}(t))\boldsymbol{\lambda}(t)] \quad (4.5)$$

4.3. Implementación de un controlador PD+G

Debido a que se desea alcanzar determinadas posiciones de la plataforma móvil del robot, es necesario la sintonización de las ganancias del controlador. En este caso se empleara un Control Proporcional Derivativo con compensación a la gravedad (PD+G). El algoritmo de control PD+G es el más conocido y empleado en el área de control automático, este esquema de control se ha empleado desde 1950 en los comienzos de la robótica. La expresión del controlador PD+G es la siguiente [4.2]:

$$\boldsymbol{\tau} = \mathbf{K}_P\mathbf{e}(\mathbf{t}) - \mathbf{K}_v\dot{\mathbf{q}}(\mathbf{t}) + \mathbf{G}(\mathbf{q}) \quad (4.6)$$

donde $\mathbf{e}(\mathbf{t})$ es la señal de error, obtenida por la diferencia de $\mathbf{q}_d(\mathbf{t}) - \mathbf{q}(\mathbf{t})$, $\dot{\mathbf{q}}(\mathbf{t})$ es la velocidad angular del actuador. Tanto K_P como K_v son matrices definidas positivas. La acción proporcional $K_P\mathbf{e}(\mathbf{t})$, da una salida al controlador que es directamente proporcional al error de posición, la acción derivativa $K_v\dot{\mathbf{q}}(\mathbf{t})$, tiene el efecto de amortiguamiento o freno mecánico a través de la inyección de la velocidad articular. La acción derivativa tiene la finalidad de mejorar el desempeño del control proporcional. Debido a que el robot que se desea controlar tiene movimientos verticales, se emplea en el controlador PD un termino parcial de la dinámica del robot, el par gravitacional $\mathbf{g}(\mathbf{q})$. En los robots cuyo movimiento se encuentra únicamente en el plano horizontal, su energía potencial es cero y por lo tanto el par gravitacional el cero.

El diagrama de control para regular la posición del RPD se muestra en la Figura 4.1.

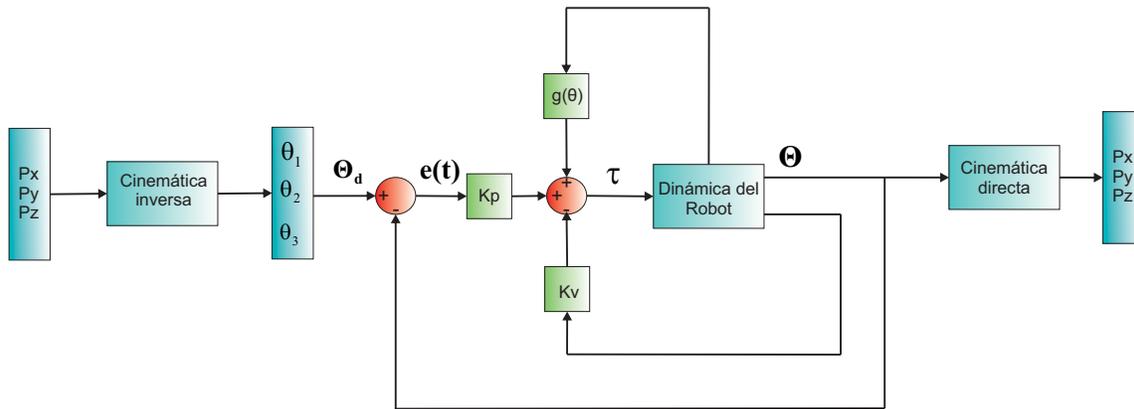


Figura 4.1: Diagrama de control PD+G en lazo cerrado para regular la posición del RPD

Al proponer un desplazamiento de la plataforma móvil del robot, es necesario aplicar el algoritmo de cinemática inversa; en este caso se propone el siguiente desplazamiento en coordenadas cartesianas: $\{^A\mathbf{P}_P\} = \{100 \quad -50 \quad -450\}^T$, al calcular el algoritmo de cinemática inversa se obtienen los siguientes desplazamientos angulares: $\Theta = \{-15,12 \quad -17,02 \quad 10,67\}^T$.

De manera que, las coordenadas generalizadas deseadas para este problema son: $q_d = [100, -50, -450, -15,02, -17,02, 10,67]$.

Los valores de las ganancias del controlador PD+G son: $\mathbf{K}_P = 6080[\mathbf{I}]$, $\mathbf{K}_D = 950[\mathbf{I}]$, donde \mathbf{I} es una matriz identidad de 3×3 .

La respuesta transitoria de las variables de salida se muestran en las Figuras 4.2 y 4.3.

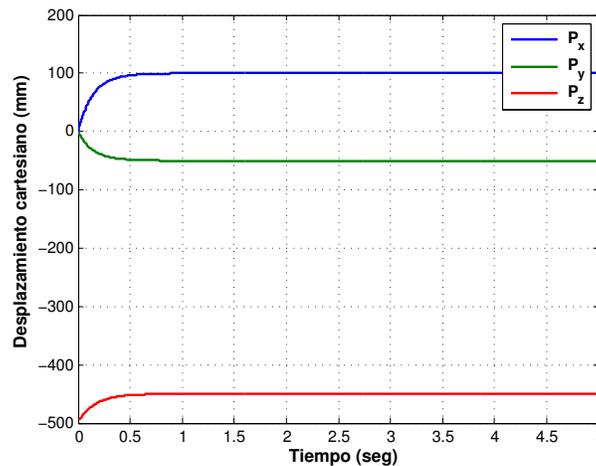


Figura 4.2: Respuesta del sistema con control coordenadas cartesianas

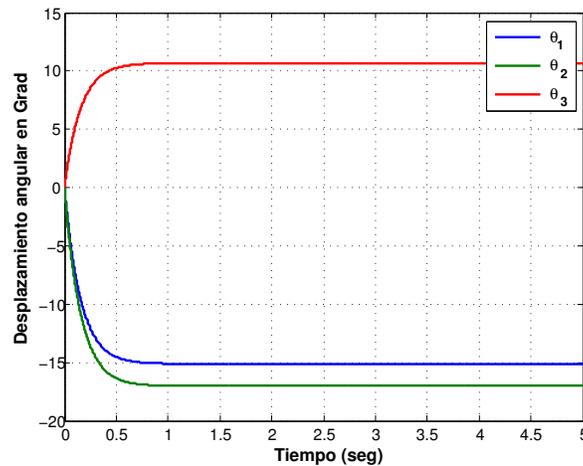


Figura 4.3: Respuesta transitoria del Robot Delta con control valores de desplazamiento angular

En las Figuras 4.2 y 4.3 se puede observar que el tiempo de asentamiento de las señales es aproximadamente a 1 segundo hasta llegar a la referencia deseada. En las gráficas se observa la respuesta transitoria del sistema, que no presenta sobre impulso en alguna de las señales, esto se debe a la contribución de la ganancia derivativa la cual tiene la función de amortiguamiento.

4.4. Propuesta de control adaptativo por redes neuronales

Debido a la no linealidad y complejidad de la dinámica de los robots manipuladores, incluyendo los robots paralelos, resulta difícil poder sintonizar las ganancias de un controlador clásico, para resolver este problema se han desarrollado distintos métodos matemáticos que auto ajustan las ganancias del controlador. Una forma de lograr obtener las ganancias del controlador es por medio de redes neuronales; los esquemas neuronales tienen la habilidad de identificar el modelo de sistemas desconocidos mediante un conjunto de datos que relacionan las entradas y las salidas [4.3]. En años recientes se ha incrementado el uso y diseño de sistemas de control usando redes neuronales artificiales RNA.

Las RNA se definen como sistemas de mapeos no lineales cuya estructura se basa en principios observados en los sistemas nerviosos de humanos y animales. Las RNA constan de un número grande de procesadores simples ligados por conexiones con pesos; las unidades de procesamiento se denominan neuronas [4.4].

Las neuronas artificiales simples fueron introducidas por McCulloch y Pitts en 1943. Una red neuronal se caracteriza por los siguientes elementos:

- Un conjunto de unidades de procesamiento o neuronas.

- Un estado de activación para cada unidad, equivalente a la salida de la unidad.
- Conexiones entre las unidades, generalmente definidas por un peso que determina el efecto de una señal de entrada en la unidad.
- Una regla de propagación la cual determina la entrada efectiva de una unidad a partir de las entradas externas.
- Una función de activación que actualiza el nuevo nivel de activación basándose en la entrada efectiva y la activación anterior.

Las redes neuronales de entrenamiento instantáneo B-Spline BSNN por sus siglas en inglés, permiten modelar y controlar sistemas en línea con una gran simplicidad en los cálculos de la señal de control. Una ventaja de usar BSNN es que reducen la complejidad de cálculos matemáticos a diferencia de otros controladores no lineales, además de que el desempeño es superior comparado con los controladores diseñados con técnicas clásicas, inclusive estas RNA no requieren calculos exesivos para su entrenamiento [4.5].

La estructura de una BSNN se compone por tres capas, las cuales son:

- Un espacio l dimensional de entradas normalizado
- Un conjunto de funciones base p , y
- La salida de la red

La función de salida de la RNA es una combinación lineal de funciones base y generalmente continuas. Los coeficientes lineales son los pesos ajustables (w_i) y debido a que la salida es linealmente dependiente del conjunto de pesos, el entrenamiento es simplemente un problema de optimización lineal. Por lo cual la salida de la BSNN se puede escribir como [4.6]:

$$\hat{y} = \sum_{i=1}^P a_i w_i = \mathbf{a}^T \mathbf{w} \quad (4.7)$$

donde \mathbf{a} es un vector P -dimensional que contiene las salidas de las funciones base, y \mathbf{w} es el vector correspondiente de pesos. En este caso, se utilizan solamente dos factores de peso para cada RNA, para disminuir el esfuerzo y el tiempo computacional.

El controlador adaptativo que se propone consiste actualizar las ganancias de tres reguladores PD empleados para controlar el posicionamiento del robot a través de la BSNN. La actualización se realiza cuando el error de la variable de salida sale de una norma acotada. De esta forma se asegura que la trayectoria de la respuesta del sistema planta-controlador converge a un conjunto compacto de solución cuando la perturbación es acotada. El esquema del control adaptativo propuesto, se muestra en la Figura 4.4.

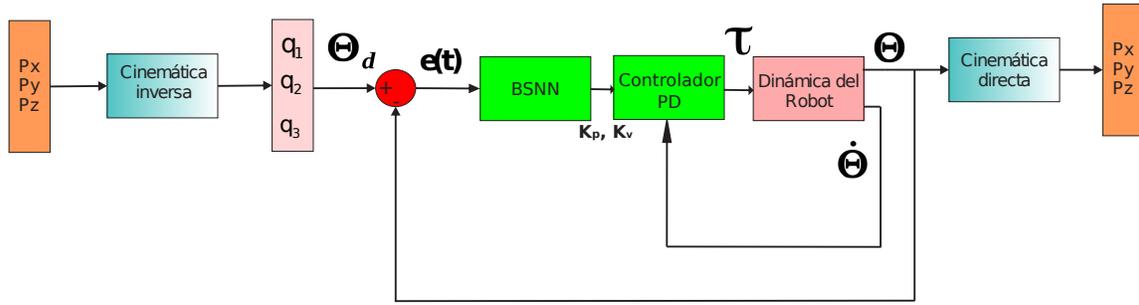


Figura 4.4: Lazo de control adaptativo por BSNN

Se hace uso de BSNN para obtener cada ganancia tanto proporcional como derivativa del controlador, por lo tanto las BSNN pueden ser escritas como:

$$k = NN_i[e(t), w_i] \quad (4.8)$$

donde NN_i para $i = 1, 2$ especifica la red BSNN que se emplea para el calculo de k , y w_i , es el vector de pesos correspondiente.

4.4.1. Regla de aprendizaje

Las RNA se configuran para obtener una o varias salidas deseadas, en base a las entradas de la red. Existen distintos métodos para ajustar los valores óptimos de las conexiones de la RNA con la finalidad de producir las salidas deseadas. Una forma es ajustando los pesos de la RNA hasta conseguir la salida deseada, sin embargo es necesario tener un conocimiento previo del sistema. La otra forma de entrenar la RNA es proporcionando patrones para el aprendizaje los cuales realizaran cambios en los pesos usando una regla de adaptación, está es la técnica que se emplea en el entrenamiento de las BSNN [4.7].

Se adopta una función de desempeño del error cuadrático medio (por sus siglas en inglés mean square error, MSE) debido a que proporciona un resultado satisfactorio en la mayoría de los casos, y las reglas de aprendizaje resultantes son simples para su implementación. Las reglas de aprendizaje instantáneo se formulan minimizando la estimación instantánea de una función de desempeño del MSE de la salida, y los parámetros se actualizan empleando reglas del gradiente descendente [4.5].

Para la actualización de los pesos $\Delta \mathbf{W}(t - 1)$, se utiliza la siguiente ecuación la cual es la regla de aprendizaje instantáneo [4.8].

$$\Delta \mathbf{W}(t - 1) = \frac{\gamma e_x(t)}{\|\mathbf{a}(t)\|_2^2} \mathbf{a}(t) \quad (4.9)$$

donde γ es la relación de aprendizaje, \mathbf{a} es el vector que contiene las salidas de las funciones base \mathbf{w} es el vector de pesos y $e_x = m(t) - \hat{m}(t)$ son las salidas deseada de

la red y la salida actual de la red B-spline, respectivamente. La relación de aprendizaje puede estar entre $0 < \gamma < 1$, esta relación se ajusta a manera de obtener una respuesta suave.

4.5. Comparación entre controladores

Dentro de este trabajo se realizó la simulación del controlador PD+G simple con ganancias fijas y un controlador PD adaptativo mediante el uso de la BSNN para la sintonización de las ganancias.

Las siguientes gráficas de la respuesta transitoria del sistema se realizaron con el siguiente conjunto de valores deseados; para el desplazamiento cartesiano se proponen las siguientes referencias: $\{^A\mathbf{P}_P\} = \{100 \quad -100 \quad -400\}^T$, en la Figura 4.5 se muestra la respuesta transitoria del sistema.

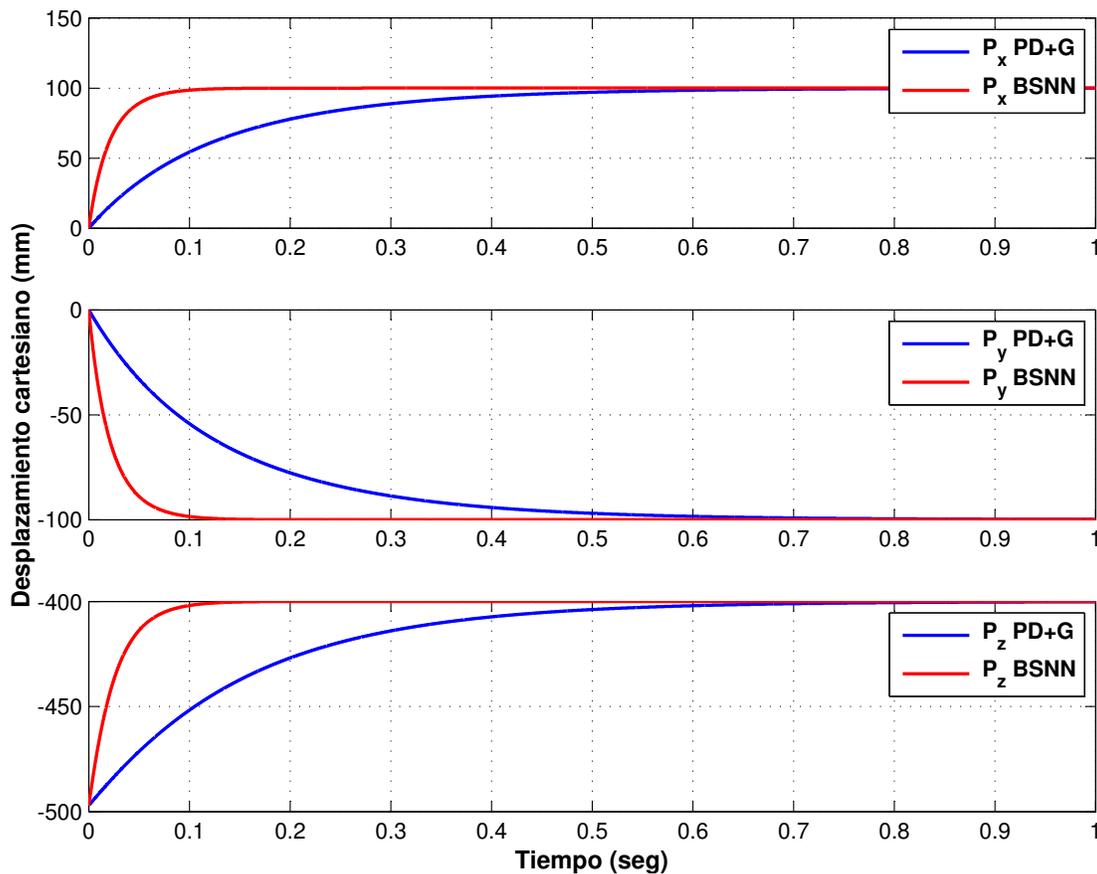


Figura 4.5: Comparación de la respuesta transitoria del desplazamiento cartesiano entre el PD y BSNN

Como puede observarse en la respuesta transitoria mostrada en la Figura 4.5, las

posiciones deseadas en coordenadas cartesianas, son alcanzadas de manera más rápida haciendo uso del control PD+G con la BSNN, el tiempo de establecimiento se logra cerca de los 0.2 segundos, este control responde más rápido en comparación de un simple controlador PD+G con ganancias fijas.

Para conocer el desplazamiento angular de los actuadores del RPD, se hace uso de la cinemática inversa para observar su respuesta transitoria; aplicando el algoritmo se obtiene los siguientes valores de desplazamiento angular, $\Theta = \{-35,69 \quad -23,80 \quad 8,26\}^T$, el resultado de la simulación se muestra en la Figura 4.6.

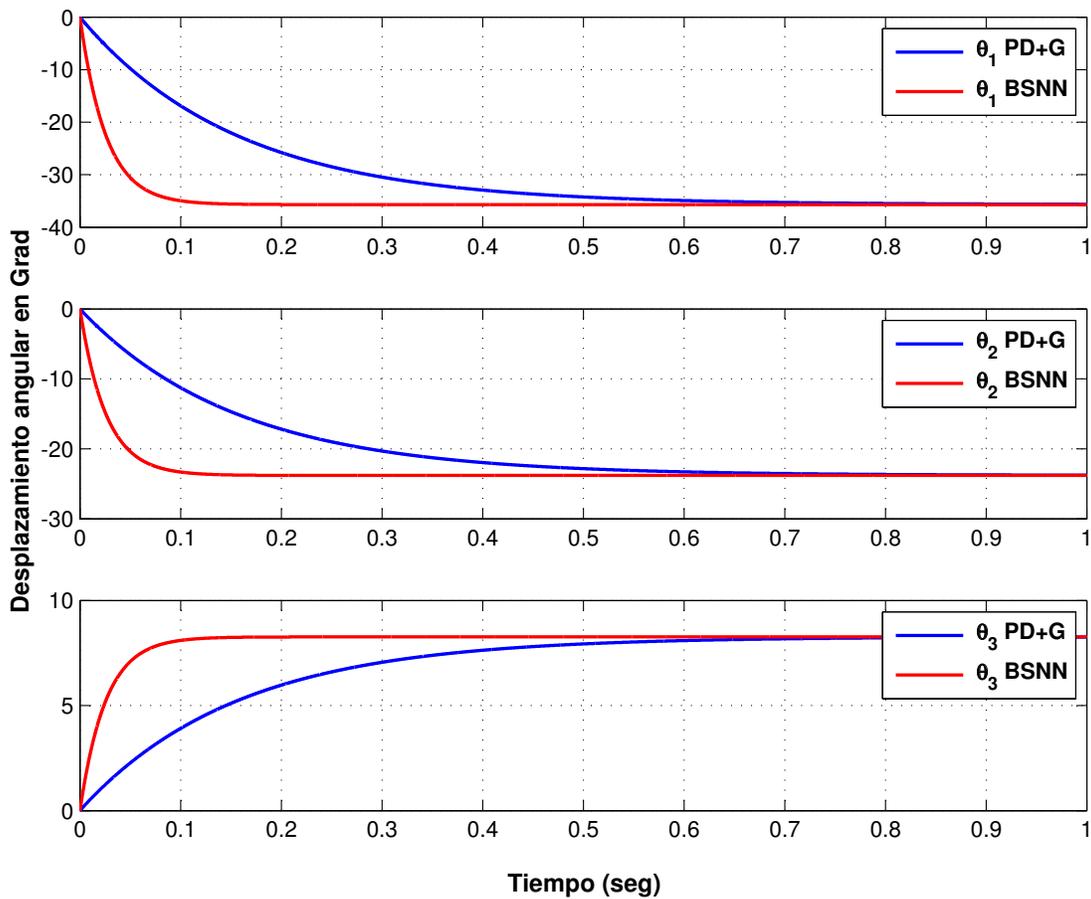


Figura 4.6: Comparación de la respuesta transitoria del desplazamiento angular entre el PD y BSNN

El comportamiento es similar al del desplazamiento cartesiano, y como puede observarse en la Figura 4.6, el desempeño del control por BSNN es mejor al realizado con una técnica de control clásica. La respuesta de los controladores no presenta sobre impulso, algo que es muy bueno para este tipo de robots. Los compensadores entregan una respuesta suave y sin oscilaciones, de manera que, a través de las simulaciones realizadas, se demuestra que un controlador adaptativo mejora aún más el desempeño

de un sistema dinámico, comparado con un controlador con ganancias estáticas.

4.6. Seguimiento de trayectoria circular

Se propone que el RPD siga una trayectoria circular, por lo que el robot únicamente se desplazará sobre un plano, en este caso el plano $x y$, por tal motivo el valor de z es constante, este caso $z = -400mm$. Las expresiones que definen la trayectoria son las siguientes:

$$\begin{aligned} x &= r \sin \theta_r t \\ y &= r \cos \theta_r t \end{aligned} \quad (4.10)$$

donde r es el radio del círculo, en este caso se propone un radio de 100 mm, y $\theta_r t$ es el valor del ángulo del círculo que varía a través del tiempo, el cual va adquiriendo valores de 0 a 7 radianes. La trayectoria deseada junto con el RPD son mostrados en la Figura 4.7.

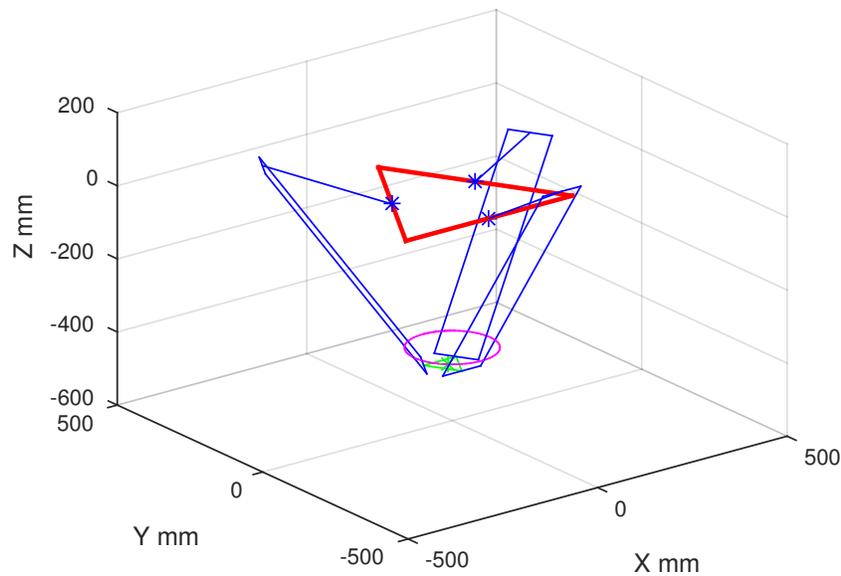


Figura 4.7: Trayectoria deseada a seguir por el RPD

Se realiza la comparación de seguimiento de trayectoria empleando los controladores BSNN y PD+G. La Figura 4.8 muestra el seguimiento de trayectoria del RPD empleando ambos controladores y la referencia deseada.

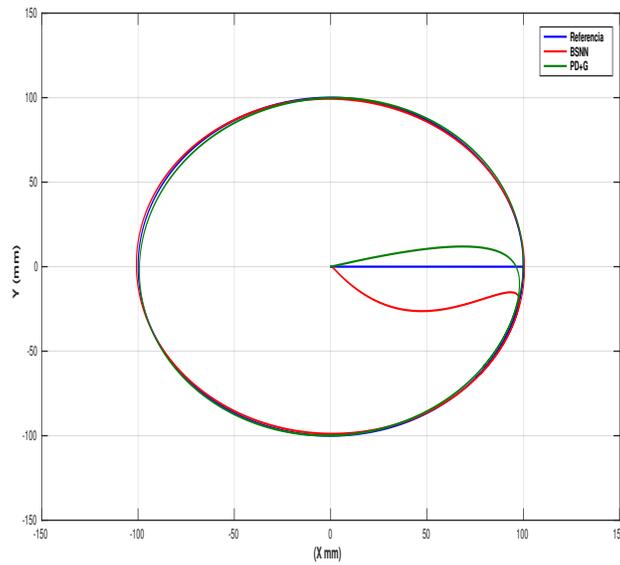


Figura 4.8: Seguimiento de trayectoria

Se observa en la Figura 4.8 que el RPD parte de su posición inicial $\Theta = \{0 \ 0 \ 0\}^T$ y comienza a seguir la trayectoria en el punto $\{^A\mathbf{P}_P\} = \{100 \ 0 \ -400\}^T$. Las Figuras 4.9 y 4.10 muestran la respuesta transitoria del sistema en coordenadas cartesianas, y la respuesta transitoria del sistema en desplazamiento angular respectivamente.

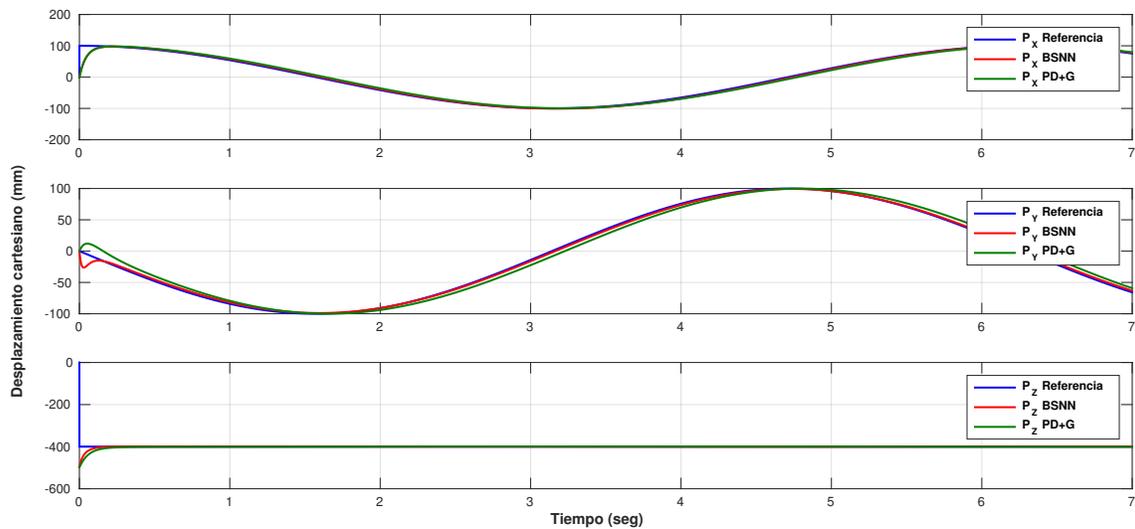


Figura 4.9: Comparación de la respuesta de seguimiento de trayectoria en coordenadas cartesianas

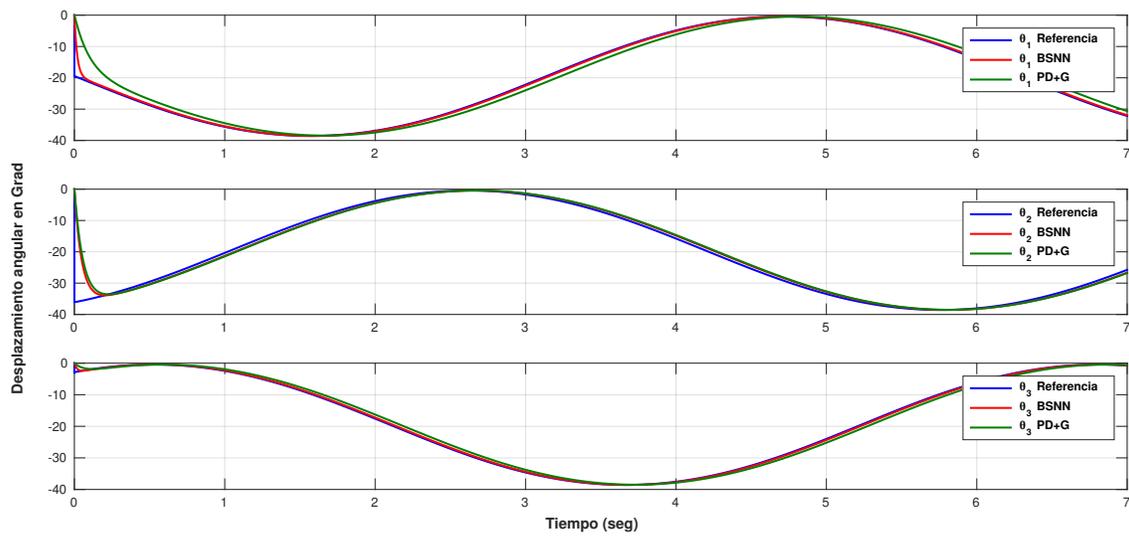


Figura 4.10: Comparación de la respuesta de seguimiento de trayectoria en desplazamiento angular

Las Figuras 4.9 y 4.10 muestran un seguimiento de la referencia con poca variación en cada una de las variables y empleando distintos controladores. Sin embargo para determinar que controlador presenta mejor desempeño se necesita calcular el error absoluto, el cual es la resta de la referencia menos la posición real. La gráficas del error de seguimiento para las variables cartesianas se muestran en la Figura 4.11, y las gráficas de error de seguimiento en desplazamiento angular son mostradas en la Figura 4.12.

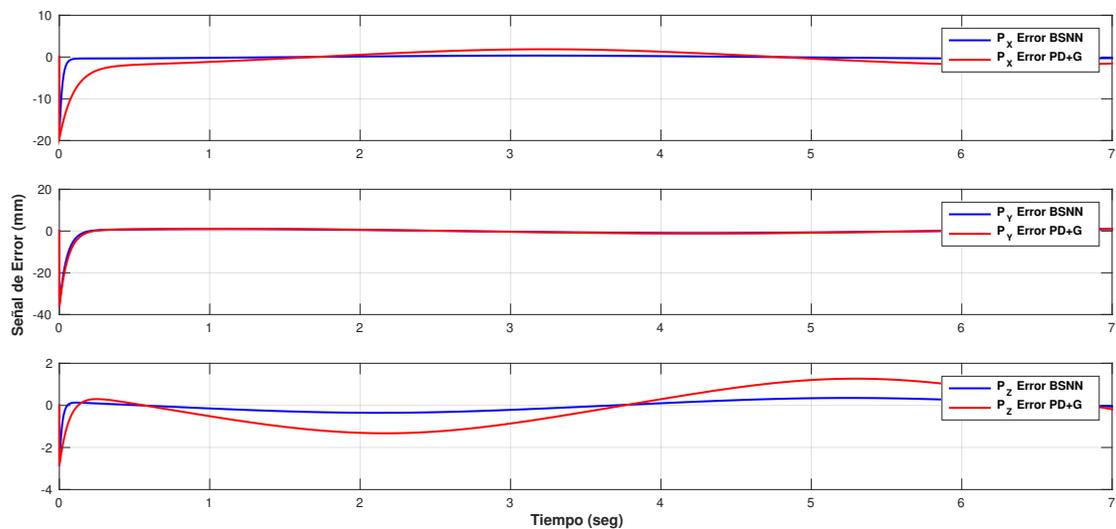


Figura 4.11: Comparación del error de seguimiento de trayectoria en coordenadas cartesianas

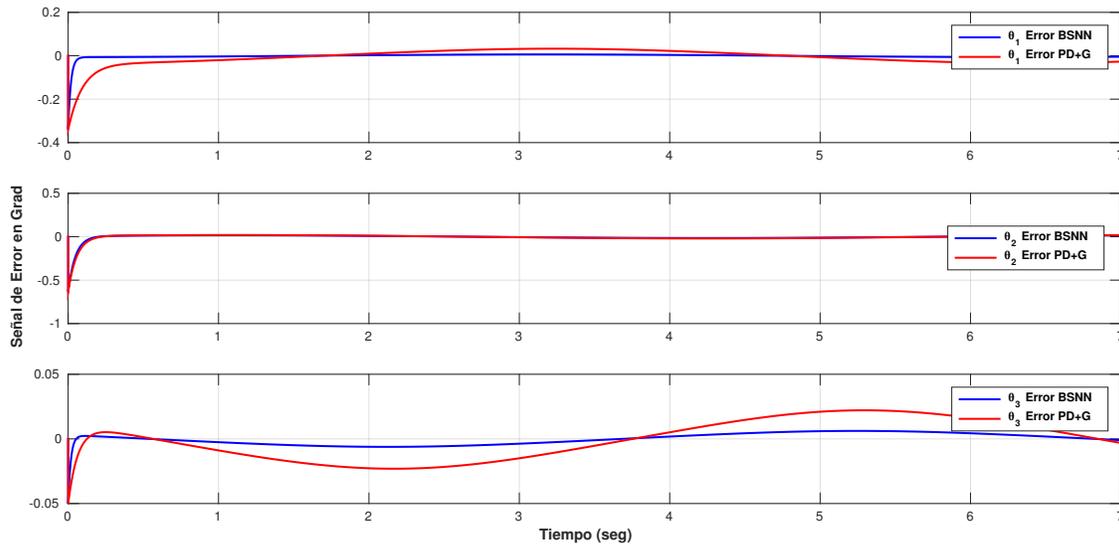


Figura 4.12: Comparación del error de seguimiento de trayectoria en desplazamiento angular

Se aprecia en las gráficas anteriores mostradas en las Figuras 4.11 y 4.12 que es muy poco el error de seguimiento de ambos controladores; sin embargo el controlador PD+G en algunos casos llega a tener un margen de error de 1mm en coordenadas cartesianas y 0.025° en desplazamiento angular mientras que el error de seguimiento empleando la BSNN es prácticamente nulo, por lo cual el control BSNN se ajusta mejor a la referencia.

4.7. Conclusiones

Dentro de este capítulo se desarrollaron dos algoritmos para controlar la posición y orientación del RPD. El primer algoritmo desarrollado y expuesto fue un control PD+G, el cual es un controlador que se ha empleado mucho en control de robots manipuladores, debido a la complejidad de la dinámica del RPD, las ganancias proporcional y derivativa del controlador fueron obtenidas mediante prueba y error, una desventaja de este tipo de controlador es que el error en estado estacionario nunca llegara totalmente a cero debido a la falta de una componente integral en el controlador; sin embargo la compensación a la gravedad reduce de manera significativa el error en estado estacionario. Debido a la no linealidad del sistema, existen regiones de operación las cuales no pueden ser alcanzadas con un controlador con ganancias fijas; por tal motivo se propuso implementar un algoritmo de redes neuronales B-Spline para actualizar las ganancias del controlador PD+G con el fin de mejorar la respuesta del sistema. Al realizar la simulación y comparar el desempeño de ambos controladores para regular la posición del RPD se observa que la respuesta de un controlador auto ajustable es más rápida que la de un controlador con ganancias fijas. Además en la prueba de seguimiento de una

trayectoria circular, se demostró que el control BSNN presenta un menor margen de error de seguimiento que el PD+G.

Referencias

- [4.1] L. A. Castañeda, A. Luviano-Juárez, and I. Chairez, “Robust trajectory tracking of a delta robot through adaptive active disturbance rejection control,” *IEEE Transactions on Control Systems Technology*, vol. 23, no. 4, pp. 1387–1398, 2015.
 - [4.2] F. Cortéz, “Robótica control de robots manipuladores,” *Alfaomega, México, DF*, 2012.
 - [4.3] R. T. Olvera, *Utilización del Neurocontrolador B-spline para Regular el StatCom*. CINVESTAV Guadalajara, 2006.
 - [4.4] P. P. Cruz, *Inteligencia artificial con aplicaciones a la ingeniería*. Alfaomega, 2011.
 - [4.5] O. Aguilar, R. Tapia, A. Valderrabano, and H. Minor, “Design and performance comparison of pi and adaptive current controllers for a wecs,” *IEEE Latin America Transactions*, vol. 13, no. 5, pp. 1361–1368, 2015.
 - [4.6] O. C. P. Myers, *Un Neurocontrolador para Dispositivos de Electrónica de Potencia*. CINVESTAV Guadalajara, 2009.
 - [4.7] P. Z. Haro, *Analysis and control of a series compensator*. A dissertation for the degree of Doctor of Science, 2006.
 - [4.8] M. Brown and C. J. Harris, “Neurofuzzy adaptive modelling and control,” 1994.
-

5

Manufactura del RPD

5.1. Introducción

Como se estableció en los objetivos particulares del proyecto, la construcción del RPD es una parte fundamental para poder implementar los análisis matemáticos mostrados en los capítulos anteriores.

Para construir cada pieza del robot, se emplean máquinas herramientas. Una máquina herramienta es un dispositivo que utiliza fuerzas mecánicas para dar forma y tamaño a un producto retirando material excedente, en forma de virutas, con la ayuda de instrumentos de corte [5.1].

Debido a que el RPD se conforma de 3 conjuntos de eslabonamientos idénticos, es necesario que las piezas fabricadas sean iguales. Para manufacturar las piezas se necesita un control preciso y exacto en las máquinas que darán forma a las piezas a partir de material en bruto.

Existen máquinas herramientas de control numérico computarizado CNC, las cuales son accionadas automáticamente, las máquinas herramientas CNC presentan varias ventajas, frente a las máquinas herramientas convencionales, entre las cuales se destacan [5.2]:

- Buena producción
- Precisión de lo que se produce
- Facilidad para el operario
- Confiabilidad de las máquinas herramientas
- Seguridad contra accidentes

Por estos motivos se ha decidido realizar las piezas que conforman el robot en un centro de maquinado CNC, para poder conseguir el resultado final, se tendrá que emplear software de diseño asistido por computadora para diseñar las piezas que conforman el RPD.

5.2. Diseño mecánico en SOLIDWORKS

Mediante software se hizo todo el ensamble del mecanismo del RPD, especificando el tipo de material del cual esta hecha la pieza que se esta diseñando. Conocer el tipo de material es muy importante para conocer sus propiedades físicas como son: masa, momentos de inercia, área, volumen, etc.

El diseño total del RPD incluyendo su bastidor se muestra en la Figura 5.1.

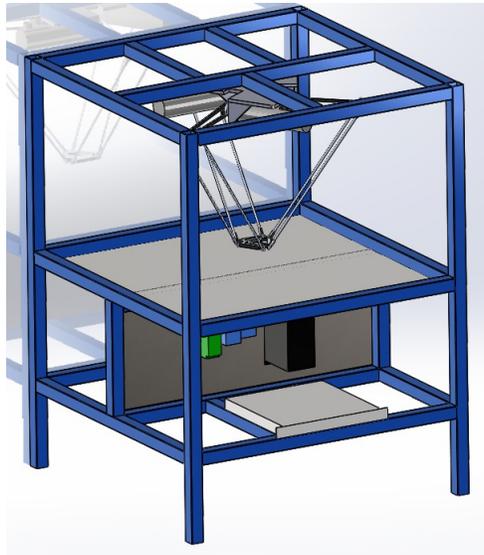


Figura 5.1: RDDCA con bastidor y componentes de control

Una parte importante del diseño asistido por computadora es que permite dar movimiento a los ensamblajes en los mecanismos [5.3], gracias a que se cuenta con distintas relaciones de posicionamiento de los elementos que conforman el mecanismo, dichas herramientas se muestran en la Figura 5.2.



Figura 5.2: Herramientas de posicionamiento estándar para ensamblajes

Aplicando de manera adecuada las herramientas mostradas en la Figura 5.2 se logra obtener ensamblado el mecanismo con movimiento del RPD, como se muestra en la Figura 5.3.



Figura 5.3: Mecanismo del RPD ensamblado totalmente

Obtenido el diseño del RPD, el siguiente paso es fabricar o adquirir los componentes que conforman la estructura del robot, los cuales son, sin incluir a los motores:

- Plataforma fija hecha de aluminio
- Plataforma móvil hecha de aluminio
- Tres acoples para unir el eje de los motores a los brazos L_a
- Tres barras de aluminio que serán los brazos del robot L_a
- Seis barras huecas de aluminio que conformaran los antebrazos L_b
- Doce rotulas, las cuales serán las articulaciones esféricas de RPD

De los elementos en listados anteriormente, los primeros cuatro puntos fueron fabricados en un centro de maquinado CNC, partiendo del diseño mostrado en la figura 5.3.

El siguiente paso es la generación del programa para el CNC, el cual permitirá el maquinado de cada pieza desde un material en bruto. Al no tener en los motores un reductor de engranajes que incremente el par del motor, es necesario que cada elemento del mecanismo del robot sea lo más ligero posible para ello las piezas son fabricadas en aluminio.

5.3. Programación en la máquina CNC

La programación nativa de la mayoría de las máquinas de Control Numérico Computarizado se efectúa mediante un lenguaje de bajo nivel llamado G & M. Se trata de un lenguaje de programación vectorial mediante el que se describen acciones simples y entidades geométricas sencillas (básicamente segmentos de recta y arcos de circunferencia) junto con sus parámetros de maquinado (velocidades de husillo y de avance de herramienta). A continuación se muestran en las siguientes Tablas los códigos G & M empleados en la fresadora CNC [5.2].

Tabla 5.1: Códigos G elementales.

Códigos G	
G00	Posicionamiento rápido (sin maquinar)
G01	Interpolación lineal (maquinando)
G02	Interpolación circular (horaria)
G03	Interpolación circular (antihoraria)
G54	Traslado del origen

Tabla 5.2: Códigos M elementales.

Códigos M	
M00	Parada
M01	Parada opcional
M02	Reset del programa
M03	Hacer girar el husillo en sentido horario
M06	Cambiar de herramienta
M04	Hacer girar el husillo en sentido antihorario
M08	Abrir el paso del refrigerante
M09	Cerrar el paso de los refrigerantes
M30	Finalizar programa

Aunque existen más códigos G & M, únicamente fueron empleados los códigos mostrados en las Tablas 5.1 y 5.2, a partir de los códigos G, es posible maquinar toda una pieza, debido a que en ellos se programan los desplazamientos de la herramienta de corte en distintos sistemas de coordenadas.

A continuación se muestran ejemplos para realizar la interpolación con los principales códigos G.

5.3.1. Sistema de coordenadas

Estas máquinas establecen sistemas de coordenadas cartesianas para posicionarse dentro de un espacio de trabajo, se establecen dos orígenes, el primer origen es conocido

como origen máquina, este sistema de coordenadas no puede ser modificado, el segundo origen se conoce como origen pieza, la programación de este origen es libre, y para ser trasladado, se orienta en las coordenadas del origen máquina, para poder mover el origen de la pieza, dentro del programa se debe incluir la instrucción G54.

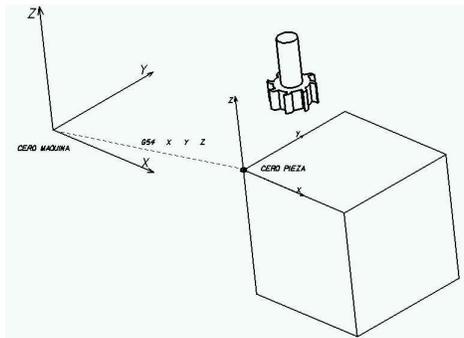


Figura 5.4: Traslación del sistema de coordenadas

5.3.2. Interpolación lineal (G01)

Mediante interpolación lineal se consigue una trayectoria recta y se necesitan al menos dos o tres coordenadas en X, Y, Z para que esta instrucción se pueda aplicar un ejemplo de esta interpolación en la fresadora seria [5.4]

G01 Z-5
G01 X40 Y20.1

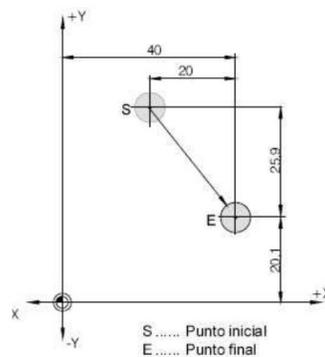


Figura 5.5: Ejemplo de aplicación del G01

Como se observo en el ejemplo anterior mostrado en la Figura 5.4 el desvaste realizado fue una diagonal, en el plano XY, con una profundidad en Z de 5mm, en caso que se quisiera una línea recta el valor de X o Y, según sea la dirección del trazo valdría cero.

5.3.3. Interpolación circular en sentido horario (G02)

Este código sirve para trazar arcos en sentido horario, dentro de la línea se deben incluir las coordenadas en XYZ. La función CR requiere el valor del radio, estas coordenadas tienen que ser el punto final del arco ya que se entiende que la coordenada anterior es el primer punto del arco. Un ejemplo de como realizar esta interpolación, se muestra a continuación:

G02 X40 Y20.1 CR36

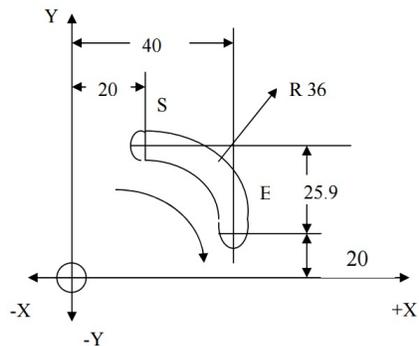


Figura 5.6: Ejemplo de aplicación del G02

5.3.4. Interpolación circular en sentido anti horario (G03)

Este código sirve para hacer trazos en sentido anti horario, se deben incluir nuevamente los caracteres mostrados en el caso anterior, por ejemplo:

G03 X40 Y20 CR36

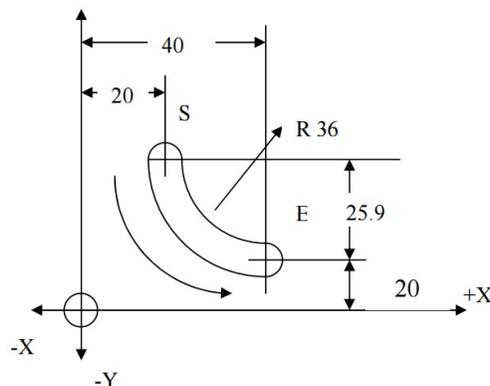


Figura 5.7: Ejemplo de aplicación del G03

5.4. Fabricación de componentes del RPD

Para realizar el programa para hacer las piezas que conforman el RPD, debido a la complejidad geométrica de algunas piezas, es necesario emplear una nueva herramienta de software que permita obtener el código G & M directamente de una pieza modelada en CAD. La herramienta es llamada CAM, (por sus siglas en ingles), representa la manufactura asistida por computadora, existen distintos software CAM que permiten simular el mecanizado de piezas y más importante, la obtención de los códigos G & M. Se presenta como ejemplo la fabricación de la pieza que sostiene los motores a la base fija, la pieza en SOLIDWORKS se muestra en la Figura 5.8:

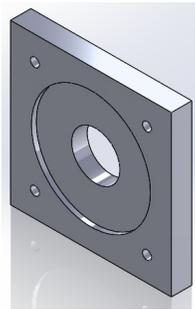


Figura 5.8: Sujeta-motor de la plataforma fija

Una vez adquirido el código, se puede simular el maquinado de la pieza. La simulación en este programa se acerca más al entorno real, cabe mencionar que también es necesario analizar el código generado para poder omitir ciertas líneas que no son necesarias en la fabricación de la pieza. La siguiente figura muestra la simulación del mecanizado del sujeta-motor.

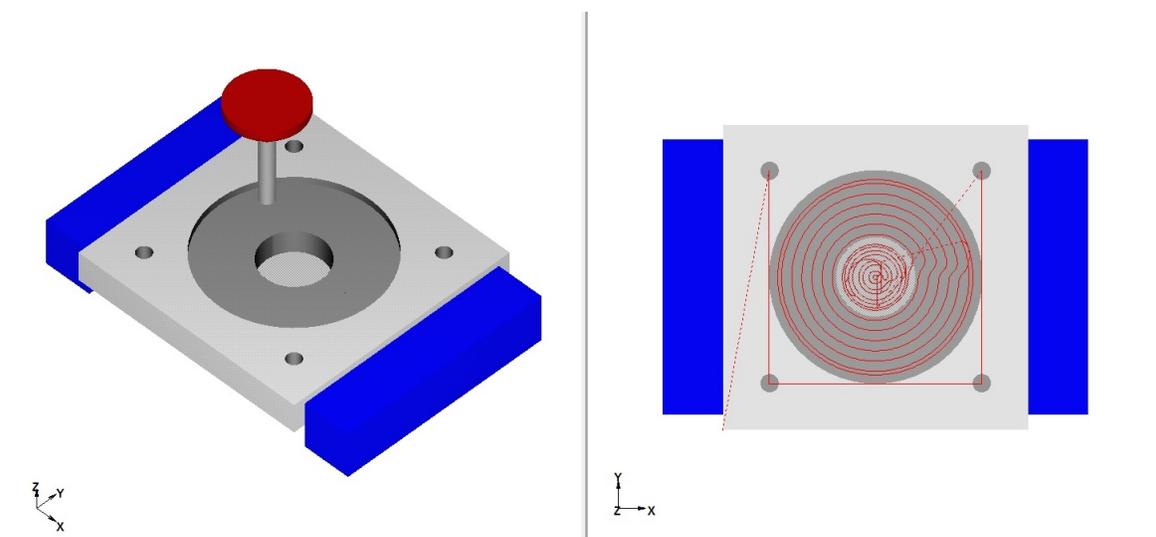


Figura 5.9: Simulación de mecanizado de una pieza para la base de los motores del RPD

La Figura 5.9 muestra dos perspectivas del maquinado de una pieza en fresadora CNC. La primera perspectiva es una vista isométrica, donde se puede apreciar el material en bruto ya maquinado, las modrazas que sujetan la pieza y, la herramienta de corte, en este caso un cortador vertical de 5mm. La segunda perspectiva es una vista en el plano XY, y se puede apreciar en líneas rojas la trayectoria que siguió la herramienta de corte. La pieza que se está simulando su fabricación es parte de la plataforma fija, es donde se colocan los motores, todas las piezas maquinadas en el CNC son planas. Una vez realizada la simulación del maquinado y después de haber verificado de que todas las líneas de código G & M que fueron generadas en SOLIDCAM son correctas, es momento de realizar la programación en un centro de maquinado real. El centro de maquinado de la universidad es el modelo "VDL-1000" de la marca DALIAN MACHINE TOOL GROUP (DMTG), este es un centro de maquinado vertical de tres ejes (XYZ), posee un controlador FANUC de la serie Oi Mate-MD, el centro de maquinado se muestra en la Figura 5.10:



Figura 5.10: Centro de maquinado VDL-1000 de DMTG

El centro de maquinado mostrado en la Figura 5.10 es muy robusto, algunas características de esta máquina son [5.5].

- Traslación máxima de X 40.16 pulgadas
- Traslación máxima de Y 22 pulgadas
- Traslación máxima de Z 23.6 pulgadas
- Torque de husillo 104 ft/lb
- Velocidad estándar de husillo 8000 RPM

Como se comentó anteriormente, el material utilizado es de distintos tipos de aleaciones de aluminio. Para la plataforma fija, donde se sostendrán los motores, se usará placa

de aluminio de 1/2 pulgada de espesor aleación 1100. La densidad de masa de esta aleación es de $2710\text{Kg}/\text{m}^3$, sin embargo para el resto de componentes como son los brazos y antebrazos del robot se optó por la aleación 6063-T5. La densidad de masa es de $2700\text{Kg}/\text{m}^3$, esto permitirá que el resto de los componentes sean más ligeros que las plataformas. La Figura 5.11 muestra el material en bruto con el que se fabricaran los distintos componentes.

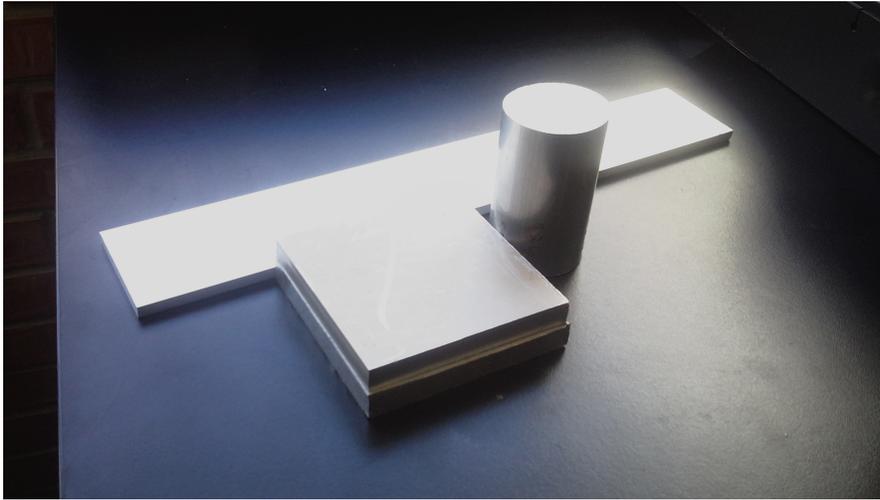


Figura 5.11: Material en bruto empleado para fabricar las piezas

Cabe mencionar que al momento de generar el código G & M es necesario especificar el tipo y medidas de las herramientas de corte que van a ser usadas para la fabricación de la pieza. En este caso se emplearán cortadores verticales de 5mm y 10mm de carburo sólido, en la Figura 5.12 se muestra un cortador vertical.



Figura 5.12: Cortador vertical de 10mm

Teniendo el material con el que desea trabajar dentro del centro de maquinado, es momento de pasar el código con el que se realizó la simulación en WinUnisoft al contro-

lador FANUC del centro de maquinado [5.6]. La única manera para poder transferir el código es mediante el uso del teclado integrado en el controlador, la Figura 5.13 muestra la interfaz del controlador FANUC.

```
PROGRAM 00001 N00000
00001 (ROUGH) (TOOL=D12FEM) ;
T1 M06 ;
G92 X0. Y0. Z0. ;
G91 G00 X-33.023 Y47.94 ;
F4000. ;
Z-95. ;
Z-3.323 ;
G01 Z-2. ;
Z-2.54 ;
Y-8.54 ;
X-39.377 ;
) _
S 0 T0000
EDIT ***** ** 12:14:26
(CANCEL SEL-ALL COPY CUT PASTE)
```

Figura 5.13: Editor de programa del controlador FANUC

La Figura 5.13 muestra la pantalla del editor de programa del centro de maquinado. En esta parte se escribe cada línea de código, además existen otros comandos importantes que no se mencionaron anteriormente en los códigos G & M debido a que son algunos complementos como son: La velocidad de avance, que es la velocidad de desplazamiento de la herramienta de corte a través de los ejes XYZ, esta velocidad se indica con la letra F y enseguida la velocidad en RPM: ejemplo **F4000**, la velocidad de corte, representada con la letra S, indica la velocidad de la herramienta de corte, la sintaxis por ejemplo seria: **S1500**.

Antes de ejecutar el programa se debe establecer el nuevo origen pieza, este nuevo origen debe ser colocado de acuerdo al origen que se tomo en la generación de los códigos del programa CAM.

Finalmente es momento de ejecutar el programa en el centro de maquinado, en la Figura 5.14 se observa la pieza sujeta-motor.



Figura 5.14: Maquinado de pieza sujeta-motor

Una vez realizada la primer pieza es posible hacer replicas de la misma las veces que se requiera, como se comento anteriormente una ventaja de la manufactura en un centro de maquinado CNC es la precisión al momento de hacer los cortes por eso se consiguió una buena calidad en las partes del robot y con muy poco margen de error respecto al diseño de SOLIDWORKS. La Figura 5.15 muestra el desarrollo de la fabricación de un brazo del RPD hecho con aluminio 6063 T-5.



Figura 5.15: Etapas de fabricación de un brazo del RDDCA

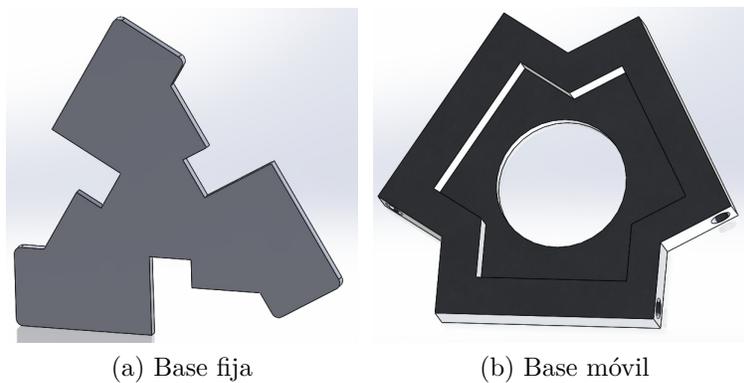


Figura 5.16: Diseño en SOLIWORKS



Figura 5.17: Producto final

Las Figuras 5.16 y 5.17 muestran el resto de piezas que fueron fabricadas en el centro de maquinado, las cuales son: la plataforma fija y la plataforma móvil. Como puede apreciarse, son muy similares casi en su totalidad los diseños y los productos finales.

5.5. Selección de articulaciones esféricas

Al principio de este capítulo se comentó que las únicas piezas que no se iban a fabricar son las articulaciones esféricas, en parte debido a la complejidad para su fabricación, haciendo una revisión de distintos fabricantes de rodamientos, se decidió adquirir el modelo de articulación esférica CW-5S del fabricante AURORA BEARING COMPANY, el modelo se muestra en la Figura 5.18 [5.7].



Figura 5.18: Aurora CW-5S

Esta articulación se compone de dos partes, un perno con cuerda y una rotula, algunas características son:

- Cuerpo hecho de acero bajo en carbono, con revestimiento protector resistente a la corrosión

- Aleación de acero, con tratamiento térmico, cromado duro.
- Rosca giro a la derecha
- Desplazamiento angular de 45°

Debido a que las barras huecas de aluminio que conforman los antebrazos del RPD tienen un diámetro de 3/8", es necesario una articulación con diámetro de rosca interna ligeramente más pequeño. La medida de la rosca interna del modelo CW-5S fue la más apropiada para las barras, cuyo valor es de 0.312 pulgadas. La Figura 5.19 muestra la unión de las barras que conforman los antebrazos con las articulaciones esféricas.



Figura 5.19: Ensamble de rotulas con sus respectivos antebrazos

5.6. Ensamblaje total

Teniendo todas las piezas que conforma en RPD, lo único que falta es ensamblar todas las partes que conforman el mecanismo, el resultado final se muestra en la Figura 5.20.



Figura 5.20: Ensamble total del RDDCA

5.7. Conclusiones

En este capítulo se describió la forma en que se materializó estructuralmente el mecanismo del RPD, partiendo de diseños concebidos en software y con el material disponible en el laboratorio. Los puntos clave de este capítulo fueron, el diseño en CAD, generación de código G & M, simulación del proceso de maquinado, selección de materiales para fabricación de piezas, operación del centro de maquinado CNC, y ensamble de los componentes del RPD. Antes de empezar a realizar los diseños en SolidWorks es necesario contemplar si se cuenta con la maquinaria necesaria para poder manufacturar las piezas diseñadas en CAD. La universidad cuenta con un centro de maquinado de tres ejes, donde es posible realizar piezas de alta complejidad, aunque se optó por realizar piezas con poca complejidad, esto en parte porque el centro de maquinado no dispone de una interfaz para poder transferir el programa generado en el software CAM a la máquina mediante un dispositivo como una memoria USB. Esta desventaja resulta complicado el escribir cada línea de código en el editor del controlador del CNC y más aún cuando se tiene un programa con muchas líneas de código.

Referencias

- [5.1] H. Bawa, *Pocesos de manufactura*. Mc Graw Hill, 2007.
 - [5.2] P. Smid, *CNC programming handbook*. Industrial Press, 2007.
 - [5.3] S. Corp, “Conceptos básicos de solidworks piezas y ensamblajes,” pp. 376–409, 2006.
 - [5.4] J. G. F. Pérez, “Manual de operación y prácticas didácticas en torno y fresadora cnc,” pp. 30–33, 2009.
 - [5.5] Dalian Machine Tool Group Corporation (DMTG), “Vdl-1200p vertical machining center. http://en.dmtg.com/product_details.php?type1=&id=158.”
 - [5.6] Fanuc Corporation, “Fanuc series 0i/0i mate-model d. [http://www.fanuc.co.jp/en/product/catalog/pdf/cnc/Series%00i-D\(E\)_v05a.pdf](http://www.fanuc.co.jp/en/product/catalog/pdf/cnc/Series%00i-D(E)_v05a.pdf).”
 - [5.7] Aurora Bearing Company, “Cw-s & cg-s series female rod ends with studs, general purpose - economy. <http://cad.aurorabearing.com/viewitems/-products-commercial-female-rod-ends-english-units/female-rod-ends-with-studs-general-purpose-economy>.”
-

6

Instrumentación

6.1. Introducción

La instrumentación es el grupo de sistemas y dispositivos que sirven para medir, convertir y registrar variables de un proceso y, luego, transmitirlos, evaluarlos y controlarlos. Los aparatos de medición se emplean para medir variables físicas y eléctricas, como velocidad, posición, tensión, corriente, etc.

Además, existen productos dedicados a la adquisición de datos y automatización para obtener procesos de medición y control cada vez más rápidos y eficientes, a costos reducidos. La instrumentación puede formar estructuras complejas para medir, controlar y monitorear todos los elementos de un sistema industrial con gran exactitud, además de automatizar tales procesos y, a la vez, garantizar la repetibilidad de las medidas y resultados [6.1].

Para poder controlar las variables de posición velocidad y par del RPD es necesario emplear sensores y transductores los cuales deberán proporcionar lecturas confiables y precisas de las variables de interés, además se necesita un sistema embebido que permita controlar las variables medidas en base a una referencia deseada.

En el presente capítulo se detalla la manera en que se procesan los datos que permiten el control del RPD, después se explican algunos detalles de programación en LabVIEW. Se detalla la instrumentación de los sensores y actuadores que conforman el RPD y los resultados obtenidos de manera experimental.

6.2. Programación en LabVIEW

LabVIEW es un entorno de programación gráfico con el cual se pueden crear aplicaciones rápidas y eficientes con interfaces de usuario profesionales.

Ya que LabVIEW imita la apariencia y operación de instrumentos físicos, como osciloscopios y multímetros, los programas de LabVIEW son llamados instrumentos virtuales o VIs. Los VIs tienen panel frontal y diagrama de bloques. El panel frontal es la interfaz de usuario, el diagrama de bloques es el programa detrás de la interfaz de usuario. Tras crear la ventana del panel frontal, se puede añadir código usando representaciones gráficas de funciones para controlar objetos del panel frontal. El código en el diagrama

de bloques es un código gráfico, también conocido como código G o código de diagrama de bloques. A diferencia de lenguajes de programación basados en texto como C++ y Visual Basic, LabVIEW usa iconos en vez de líneas de texto para crear aplicaciones. En la programación basada en texto, las instrucciones determinan el orden de ejecución. LabVIEW utiliza flujo de datos gráfico. En programación mediante flujo de datos gráfico, los datos fluyen a través de nodos en el diagrama de bloques y eso determina la orden de ejecución. [6.2].

Dentro de este capítulo se explicará la forma de programar el algoritmo de control de posición del RPD empleando el software LabVIEW. En la Figura 6.1 se muestra un simple programa en LabVIEW donde se aprecian las pantallas de panel frontal y el diagrama de bloques.

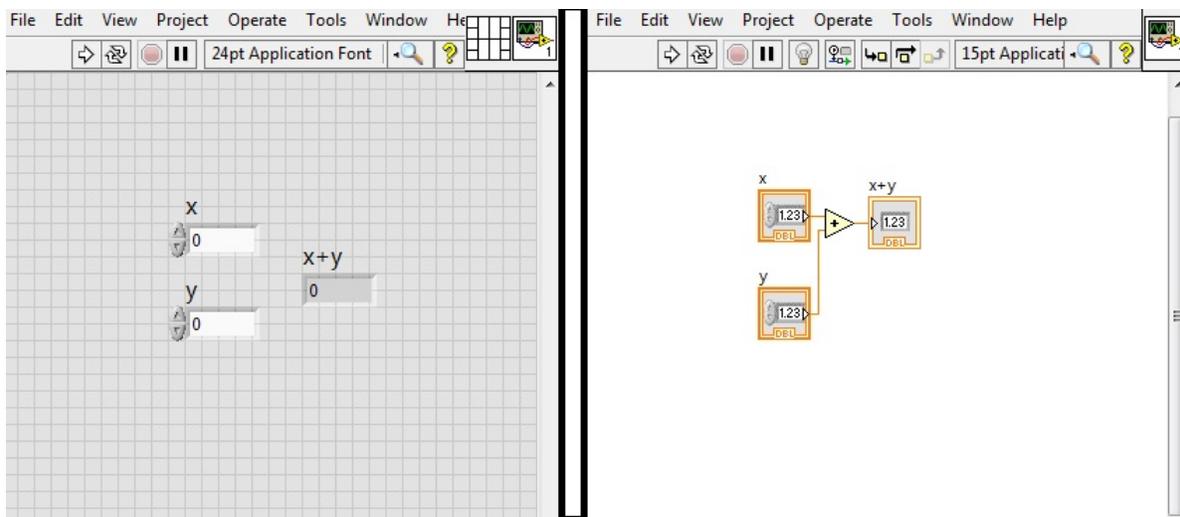


Figura 6.1: Panel frontal y diagrama de bloques de un programa en LabVIEW

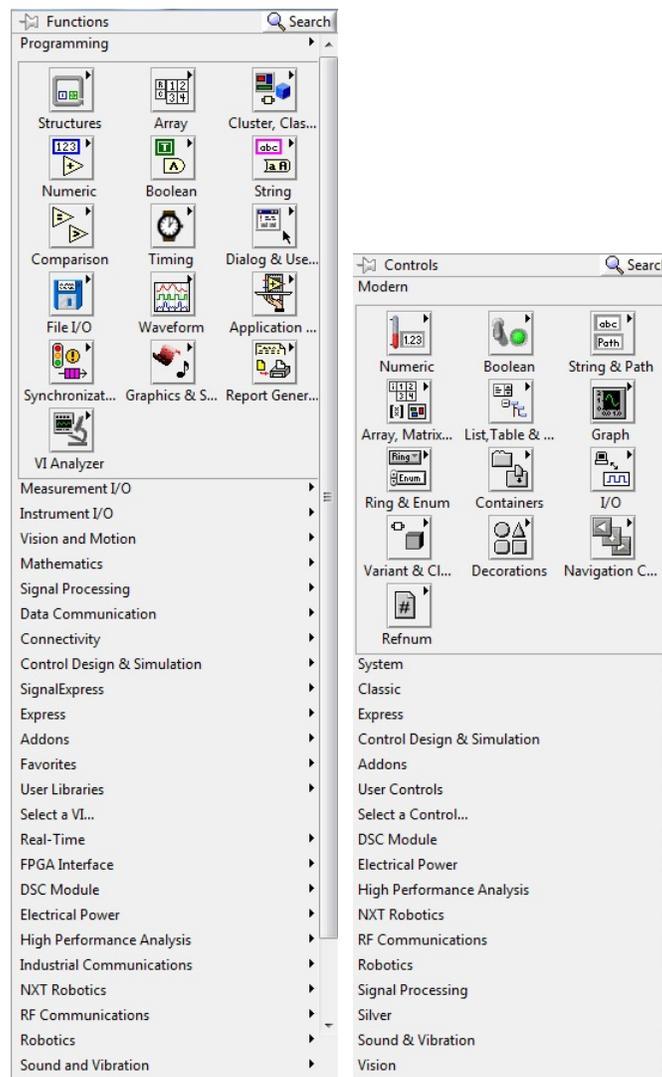
6.2.1. Entorno gráfico

Paleta de funciones

La paleta de funciones, mostrada en la Figura 6.2 (a), provee VIs o bloques para la creación de programas en código G, este menú es mostrado dando click derecho sobre cualquier parte del diagrama de bloques.

Paleta de controles

La paleta de controles, mostrada en la Figura 6.2 (b), provee controles e indicadores para el panel frontal, estas herramientas son con las que el usuario puede visualizar y manipular datos en tiempo real, el menú se muestra dando click derecho en cualquier parte del panel frontal.



(a) Paleta de funciones (b) Paleta de controles

Figura 6.2: Paletas de funciones y controles

6.2.2. Cables

En LabVIEW se usan cables para conectar varios terminales entre sí y pasar datos a un VI. Se deben conectar los cables a las entradas y salidas que sean compatibles con los datos que se transfieren por el cable. Por ejemplo no se puede cablear una salida de array a una entrada numérica. Además la dirección de los cables debe ser correcta. Se debe conectar los cables sólo a una entrada y al menos a una salida. Por ejemplo, no se puede cablear dos indicadores entre sí. Los componentes que determinan la compatibilidad del cableado son el tipo de datos del control o el indicador y el tipo de datos del terminal. La Figura 6.3 muestra una tabla con los cables principales en LabVIEW.

Tipo de Cable	Escalar	1D Array	2D Array	Color
Punto Flotante				Naranja
Númérico				Azul
Booleano				Verde
Cadena de caracteres				Rosa

Figura 6.3: Cables de datos

6.2.3. Estructuras

Las estructuras de ejecución contienen secciones de código gráfico y controlan la manera en que se ejecuta el código. Las estructuras de ejecución más comunes son Ciclos While, Ciclos For y Estructuras de Casos los cuales puede usar para ejecutar la misma sección del código varias veces o para ejecutar una sección diferente del código basada en alguna condición.

Ciclo While

Similar a un Ciclo Do o a un Ciclo Repeat-Until en lenguajes de programación basados en texto, un Ciclo While, que se muestra en la Figura 6.4, ejecuta el código que contiene hasta que ocurre una condición.

El Ciclo While ejecuta el código que contiene hasta la terminal condicional, una terminal de entrada, recibe un valor Booleano específico. La terminal de iteración es una terminal de salida que contiene el número de iteraciones terminadas. La cantidad de iteraciones para el Ciclo While siempre comienza en cero.

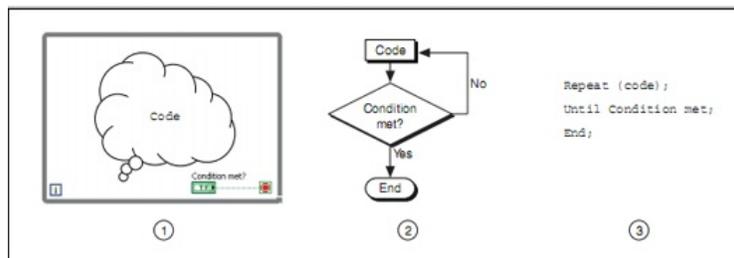


Figura 6.4: 1) Ciclo While 2) Diagrama de flujo 3) Pseudocódigo

Ciclo For

Un Ciclo For ejecuta un subdiagrama un número de veces establecido. La Figura 6.5 muestra un Ciclo For en LabVIEW, un diagrama de flujo equivalente a la funcionalidad de Ciclo For y un código de ejemplo pseudo de la funcionalidad del Ciclo For.

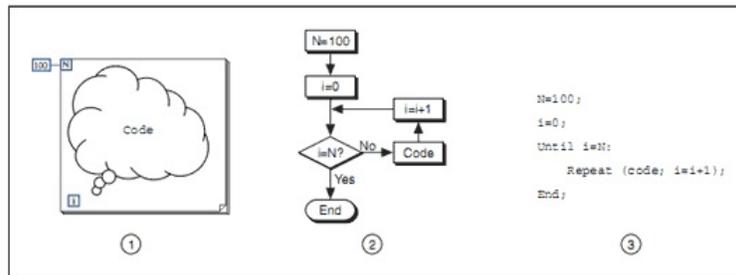


Figura 6.5: 1) Ciclo For 2) Diagrama de flujo 3) Pseudocódigo

La terminal de conteo es una terminal de entrada cuyos valores indican cuantas veces se repite el subdiagrama. La terminal de iteración es una terminal de salida que contiene el número de iteraciones terminadas, la cantidad de iteraciones para el Ciclo For siempre comienza en cero.

Case Structure

Una estructura de Casos tiene dos o más subdiagramas o casos. Solamente un subdiagrama es visible a la vez y la estructura ejecuta solamente un caso a la vez. Un valor de entrada determina cual subdiagrama se ejecuta. La estructura de Caso es similar a las instrucciones del interruptor o las instrucciones si...después... en lenguajes de programación basados en texto, la Figura 6.6 muestra la estructura case.

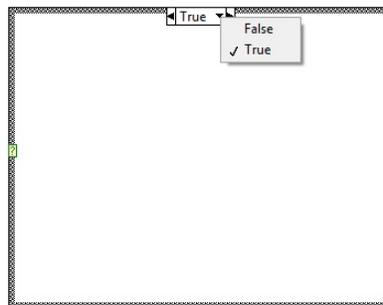


Figura 6.6: Estructura Case

La etiqueta del selector de caso en la parte superior de la estructura de Caso contiene el nombre del valor del selector que corresponde al caso en el centro y a las flechas de incremento y reducción a cada lado.

6.3. Descripción del FPGA

Los arreglos de compuertas programables denominados (FPGA's) son chips de silicio reprogramables. A diferencia de los procesadores que se encuentran en una PC, al programar un FPGA el chip se vuelve a cablear para implementar su funcionalidad en

lugar de ejecutar una aplicación de software [6.3]. Ross Freeman, el cofundador de Xilinx, inventó el FPGA en 1985. La adopción del chip FPGA en las industrias es guiada por el hecho que los FPGAs combinan lo mejor de los circuitos integrados de aplicación específica (ASICs) y sistemas basados en procesador. Los FPGAs ofrecen velocidades temporizadas por hardware y fiabilidad, pero sin requerir altos volúmenes de recursos para compensar el gran gasto que genera un diseño personalizado de ASIC [6.4].

El silicio reprogramable también tiene la misma flexibilidad que un software que se ejecuta en un sistema basado en procesador, pero no está limitado por el número de núcleos de procesamiento disponibles. A diferencia de los procesadores, los FPGAs son verdaderamente paralelos por naturaleza, así las diferentes operaciones de procesamiento no tienen que competir por los mismos recursos. Cada tarea de procesamiento independiente es asignada a una sección del chip y puede ejecutarse de manera autónoma sin ser afectada por otros bloques de lógica. Como resultado, el rendimiento de una parte de la aplicación no se ve afectado cuando se agregan otros procesos.

6.3.1. Partes de un FPGA

Cada chip de FPGA está hecho de un número limitado de recursos predefinidos con interconexiones programables para implementar un circuito digital reconfigurable y bloques de E/S para permitir que los circuitos tengan acceso al mundo exterior, la Figura 6.7, muestra a detalle las distintas partes que conforman un chip FPGA [6.4].

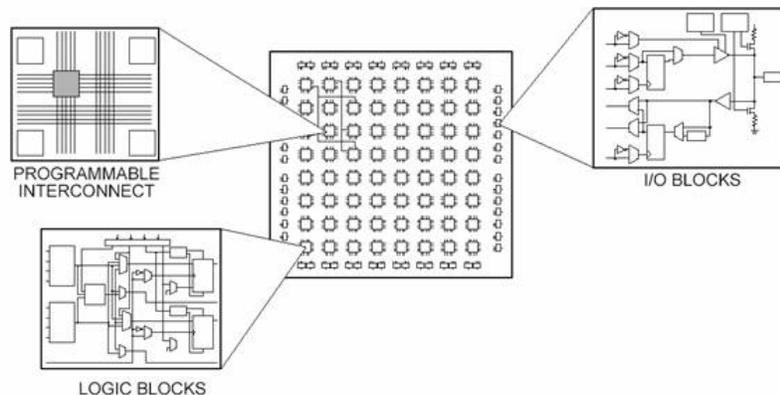


Figura 6.7: Partes que conforman un chip FPGA

Las especificaciones de recursos de FPGA a menudo incluyen el número de bloques de lógica configurables, número de bloques de lógica de función fijos como multiplicadores y el tamaño de los recursos de memoria como RAM en bloques embebidos. Los bloques de lógica configurables (CLB's) son la unidad de lógica básica de un FPGA. Algunas veces referido como segmentos o células de lógica, los CLB's están hechos de dos componentes básicos: flip-flops y tablas de consulta (LUT's). Es importante tomar esto en cuenta porque distintas familias de FPGA's se diferencian en la manera en que los flip-flops y las LUT's están empacados.

El Apéndice A, muestra el procedimiento para crear un nuevo proyecto en LabVIEW FPGA [6.3] [6.4].

6.4. Motores del RPD

Como parte vital en la construcción del RPD es importante seleccionar el tipo de actuador que producirá el par para cada cadena cinemática del robot. El modelo de motor empleado es el MTE-3363-BLBCN de Baldor Electric Company. Algunas características de este servomotor de corriente directa son [6.5]:

Tabla 6.1: Características del MTE-3363-BLBCN

Parámetro	Descripción
Torq Stall/Cont	180 OI
Velocidad Máxima	4000 RPM
Voltaje máximo	120 V
Constante Volt KE	31.1 V/KRPM
Constante corriente Stall	4.76 A RMS
Corriente Pico	28.5 A RMS
Tacometro	7V
Encoder	1000 L

6.5. Sensores

En esta sección se detalla los sensores empleados en el RPD, los cuales son el sensor de proximidad y el encoder incremental.

6.5.1. Sensor de proximidad

El sensor de proximidad es empleado dentro del robot para ubicar la posición inicial o de home. La ubicación del sensor debe estar alineada al eje de cada motor con la finalidad de detectar el cada brazo del RPD y de esta forma indicar el comienzo de conteo del encoder. El sensor de proximidad es el GAODE SN04-N el cual es un sensor de proximidad inductivo, la distancia máxima a detectar entre el sensor y un objeto metálico es de $5mm$. El voltaje de alimentación para el sensor puede ser entre $6V$ y $36V$, el consumo máximo de corriente del sensor son $5mA$. La Figura 6.8 muestra el sensor de proximidad empleado.



Figura 6.8: Sensor de proximidad empleado

El sensor envía un valor de $0V$ cada vez que detecta un objeto y cuando no detecta alguna presencia, continuamente esta enviando el voltaje con el que se está alimentando a través de la salida. En este caso se alimenta con un voltaje de $24V$ y el módulo que recibe la señal del sensor es el NI 9423 el cual es un módulo de entrada digital sinking de 8 Canales, a $24V$ [6.6].

6.5.2. Encoder e instrumentación

Para medir la posición y velocidad de cada motor se hace uso de un encoder tipo incremental. El encoder incremental proporciona normalmente dos formas de ondas cuadradas y desfasadas entre sí en 90° eléctricos, los cuales por lo general son "Canal A", "Canal B". Con la lectura de un solo canal se dispone la la frecuencia de la onda cuadrada es directamente proporcional a la velocidad de rotación; si se capta la señal "B", es posible discriminar el sentido de rotación en base a la secuencia de datos que producen ambas señales. El "Canal Z" proporciona la posición absoluta del cero del eje del encoder [6.7]. Para conocer el desplazamiento angular del motor, se emplea la siguiente formula:

$$R_{pos} = \frac{2\pi A}{N} \quad (6.1)$$

donde A es la cantidad de pulsos que se han contado, N es el número de pulsos por revolución del encoder, y R_{pos} es el desplazamiento angular en radianes. Haciendo uso de las herramientas de programación en LabVIEW FPGA; se hace uso del siguiente código mostrado en la Figura 6.9; es el empleado para obtener el conteo de pulsos del encoder.

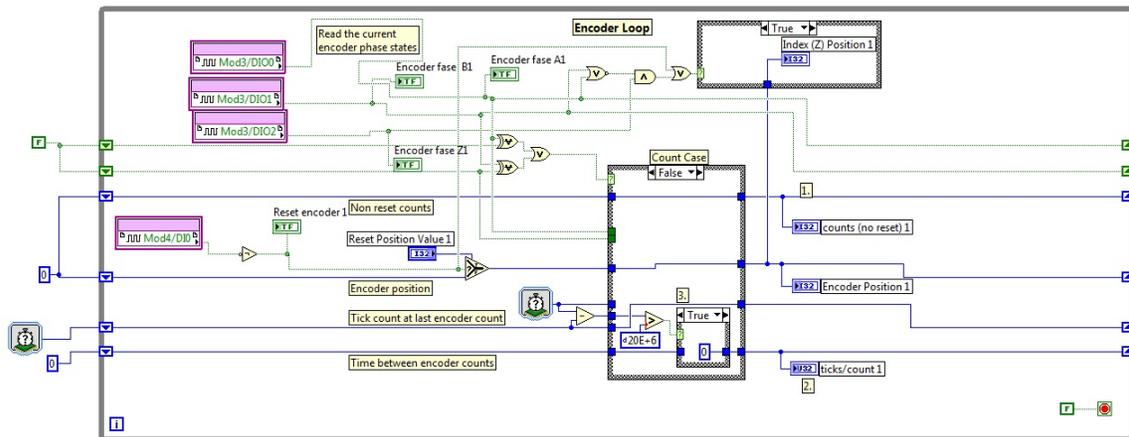


Figura 6.9: Código de conteo de pulsos del encoder

Para adquirir los trenes de pulsos enviados por el encoder se emplea el módulo NI 9403; el cual es un módulo de entradas y salidas digitales a 5V TTL. Este módulo posee 32 canales los cuales pueden ser configurados de manera individual como entrada o salida [6.8]. Para hacer el conteo de pulsos de los 3 encoder del RPD, se emplean tres ciclos while idénticos al que se muestra en la Figura 6.9. El ciclo anterior además de calcular la posición del motor, permite conocer la frecuencia de las ondas cuadradas que genera el encoder, Se asume que la velocidad de reloj del FPGA es 40MHZ. La expresión matemática (6.1) se programa dentro de un ciclo while en el FPGA para calcular el desplazamiento en radianes a partir de la posición dada en pulsos, la cual se obtiene del código de la Figura 6.9, el código G se muestra en la Figura 6.10.

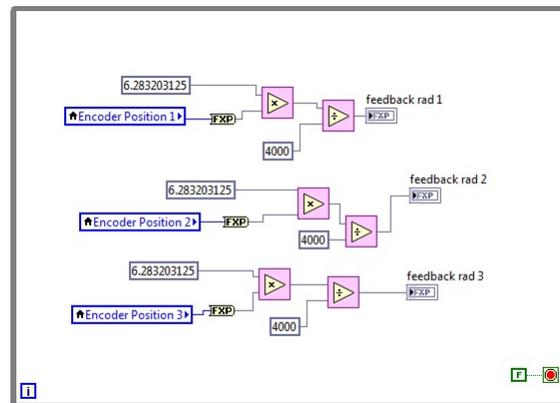


Figura 6.10: Posición angular en radianes

La figura anterior muestra el diagrama de bloques empleado para calcular la posición de los 3 motores del RPD, este código trabaja en paralelo con el que se muestra en la Figura 6.9. Se hace uso de variables locales dentro de la programación para desplazar distintos datos de un bucle a otro.

6.6. Implementación de un controlador PD en FPGA

Dentro de esta sección se explica la forma en que se programo el controlador PD+G propuesto en el capítulo 4 tomado de la ecuación (4.6), de la cual se pueden distinguir tres componentes elementales; la acción proporcional, la acción derivativa, y la compensación a la gravedad. El código G de la ley de control se muestra en la Figura 6.11.

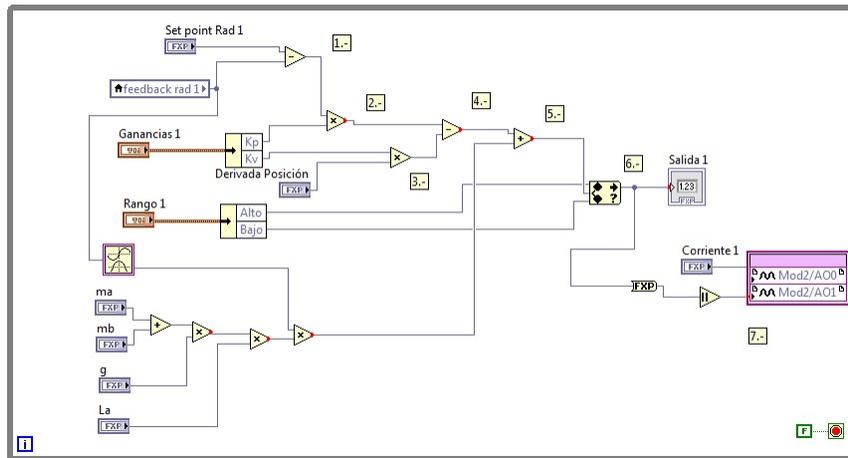


Figura 6.11: Controlador PD+G en LabVIEW FPGA

El diagrama de bloques mostrado anteriormente en la Figura 6.11, realiza las siguientes acciones.

1. Calcula el error de la posición el cual es la resta de el valor de referencia deseado menos la posición real, la unidades empleadas son radianes
2. Producto de la ganancia proporcional por el error de posición
3. Producto de la ganancia derivativa por la derivada de la posición
4. Sustracción de de la acción derivativa de la acción proporcional
5. Adición de la componente de compensación a la gravedad al controlador
6. Limitación del valor de salida del controlador
7. Salida de la señal de control a través del NI 9264

Es necesario emplear 3 códigos G como el de la figura 6.16 para poder controlar los 3 motores. Para estimar la velocidad angular del motor se hace una aproximación

obteniendo la derivada de la posición, por lo tanto se emplea la siguiente ecuación en diferencia, la cual permite obtener la derivada de una señal digital [6.9].

$$\dot{x}(k) \cong \frac{x(k+1) - x(k)}{T} \quad (6.2)$$

donde

- T es el periodo de muestreo
- $x(k)$ es la muestra anterior
- $x(k+1)$ es la muestra actual

La expresión (6.2) puede ser programada de la siguiente forma en LabVIEW ver Figura 6.12.

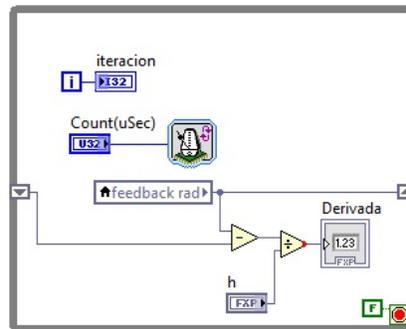


Figura 6.12: Derivación en FPGA

El ciclo while de la Figura 6.12 puede tomar muestras continuas o con un determinado periodo, el periodo de muestreo se establece en "Count(uSec)". Para conocer la muestra actual y la anterior se emplean registros de desplazamientos, cada vez que el ciclo while hace una iteración los datos guardados en los registros de desplazamiento son actualizados.

La señal de control sale a través del NI 9264 el cual es un módulo de salidas analógicas, el rango de voltaje de salida es de $\pm 10V$ con una resolución de 16 bits, este módulo posee 16 canales [6.10]. La señal analógica es empleada para programar la fuente conmutada Lambda; para que la fuente pueda ser usada en modo programable es necesario tener en ON los 2 primeros interruptores del DIP Switch. Una vez en modo programable es posible enviar 2 entradas de control a la fuente Lambda, una para controlar el voltaje de salida de la fuente y la otra para controlar la corriente de salida para ello se emplea un conector tipo DB25 mostrado en la Figura 6.13, el cual se configura de la siguiente forma [6.11].

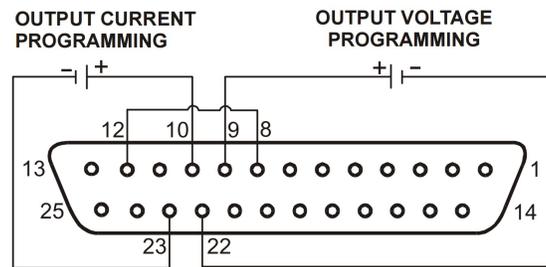


Figura 6.13: Conector DB25 para programar fuente remota

La Figura 6.13 muestra el diagrama de conexión del conector DB25 en el cual la programación del voltaje de salida es a través de los pines 9 y 22, y para controlar la corriente se emplean los pines 10 y 23; además es necesario conectar el pin 8 con el 12 para activar la función en modo programable. Se establece un valor fijo para la corriente controlada de la fuente.

Cuando el controlador envía un voltaje negativo es porque el valor de referencia es menor al de la variable con que se está comparando. El módulo de salidas analógicas NI 9264 puede enviar valores de voltaje negativos; sin embargo la fuente remota únicamente permite voltajes mayores a cero de manera que no es posible invertir la polaridad de la fuente de manera directa. Por lo tanto es necesario el uso de un Puente H el cual únicamente tendrá la función de invertir el giro de los motores de CD; en este caso se hará uso de un 6MBI50L-060 el cual es un Pack de 6 IGBT'S de la empresa Fuji electric. Los IGBT'S soportan un voltaje de colector-emisor de hasta 600V y una corriente de 50A, además este módulo tiene la capacidad de conmutación a altas velocidades [6.12]. Para realizar la inversión de giro de los motores de CD se emplea el siguiente código mostrado en la Figura 6.14.

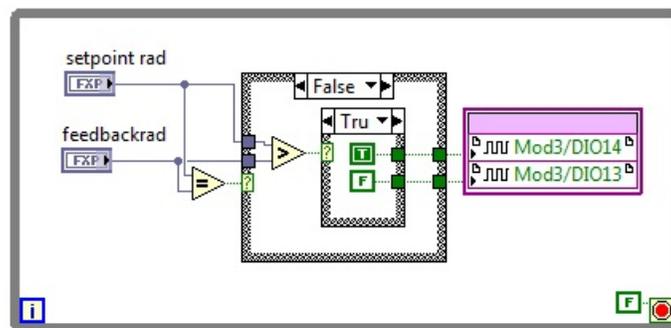


Figura 6.14: Condición para inversión de giro

La Figura 6.14 muestra una serie de reglas para hacer la inversión de giro de un motor que conforma el RPD, las reglas se determinan en base a los datos de valor de referencia y el valor actual de la variable medida, en este caso posición. Haciendo estas comparaciones se activan o se desactivan 2 salidas digitales según sea el caso. Las salidas

del NI 9403 activan el sentido horario o el sentido anti horario de los motores mediante el puente H de IGBT'S.

6.7. Resultados experimentales

Esta sección muestra la implementación de un seguimiento de trayectoria. El RPD necesita una secuencia de puntos para poder realizar una tarea específica; para este caso se propone el seguimiento de una trayectoria circular. Para generar una trayectoria circular es necesario hacer uso de las siguientes expresiones matemáticas las cuales se han mostrado en el capítulo 4.

$$\begin{aligned}x &= r \sin \theta_r t \\y &= r \cos \theta_r t\end{aligned}\tag{6.3}$$

donde r es el radio del círculo y $\theta_r t$ es el valor del ángulo del círculo que varia a través del tiempo. Debido a que para este ejemplo únicamente existirá desplazamiento sobre un plano horizontal, la coordenada z tendrá un valor fijo, de $z = -590mm$.

La referencia de trayectoria circular se muestra en la Figura 6.15.

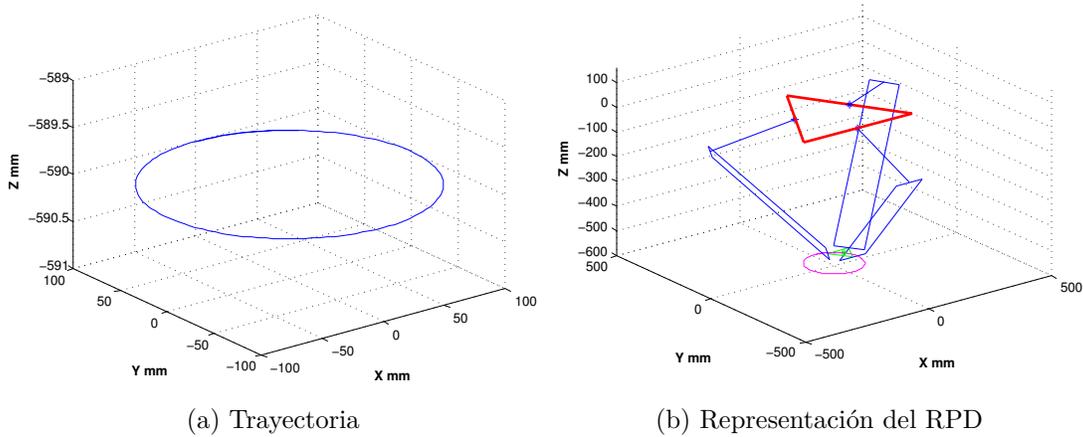


Figura 6.15: Trayectoria deseada para el RPD

La Figura 6.15 da un concepto de la trayectoria a seguir la cual se traza únicamente es sobre un plano horizontal y el posicionamiento que RPD debe tener para alcanzar un punto de la circunferencia deseada. Las coordenadas cartesianas que seguirá el RPD se muestran en la Figura 6.16.

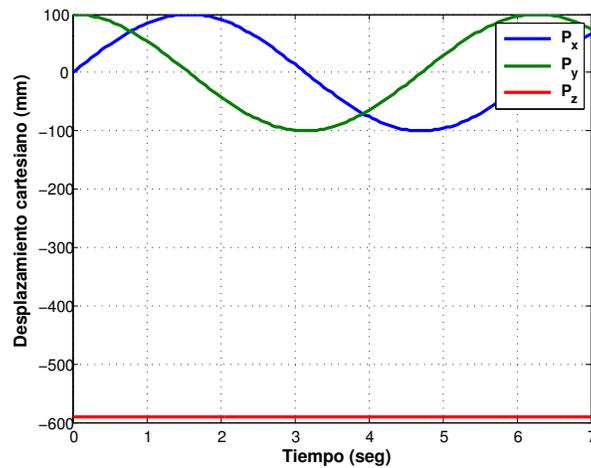


Figura 6.16: Desplazamiento en coordenadas cartesianas de la trayectoria

Es necesario realizar la transformación de la cinemática inversa para indicar los puntos de referencia que deben seguir los actuadores del RPD para lograr la trayectoria deseada. La Figura 6.17 muestra los valores de desplazamiento para los motores.

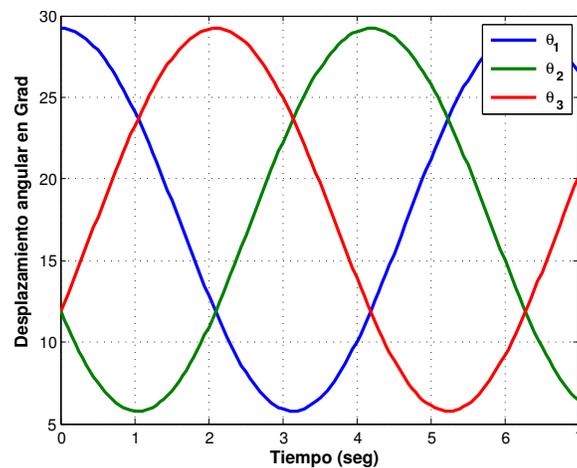


Figura 6.17: Referencia de desplazamiento angular de los actuadores

El ángulo es medido en radianes, la referencia comienza en un valor de cero y va incrementando en 0.05, el tiempo con el que se envía un nuevo valor es cada 50 milisegundos.

La Figura 6.18 muestra la referencia de la trayectoria a seguir en tiempo real programada en LabVIEW.

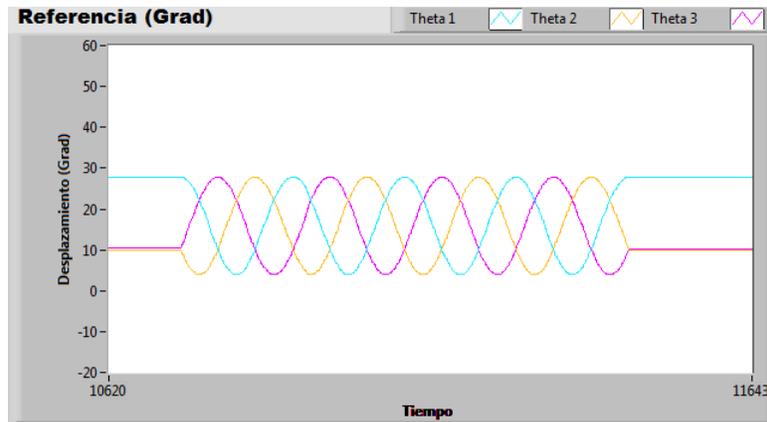


Figura 6.18: Referencia tiempo real en LabVIEW

Para que la trayectoria anterior pueda ser reproducida por el RPD es necesario implementar una ley de control para la posición del efector final. En el capítulo 4 se explicó a detalle dos tipos de controladores para regular la posición del RPD. En este ejemplo se hace uso del controlador PD+G debido a su simpleza para programar. El resultado de seguimiento de trayectoria del RPD se muestra en la Figura 6.19.

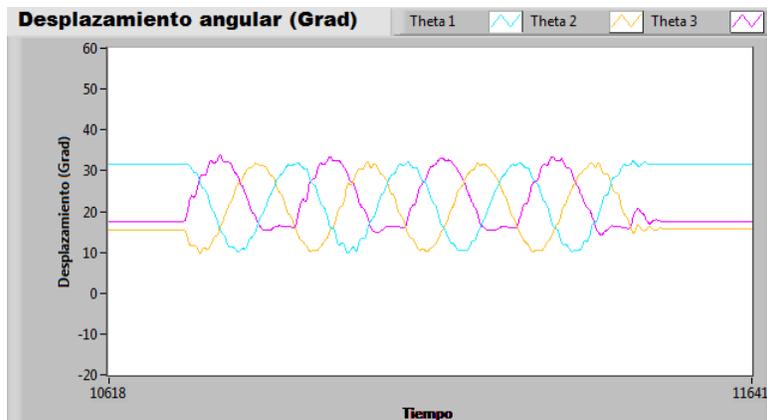


Figura 6.19: Seguimiento de trayectoria en tiempo real LabVIEW

En la Figura 6.19 se observa que la respuesta de la variable θ_1 no es suave, porque existen ligeros sobre impulsos que producen perturbaciones en el conjunto de eslabonamientos del robot. El controlador empleado únicamente controla la posición del RPD, de manera que para que la respuesta del sistema mejore es necesario implementar dos reguladores extras para controlar velocidad y par de carga de los motores del robot. Las gráficas mostradas en la Figura 6.20 comparan la señal de referencia generada y la posición real de cada actuador del RPD.

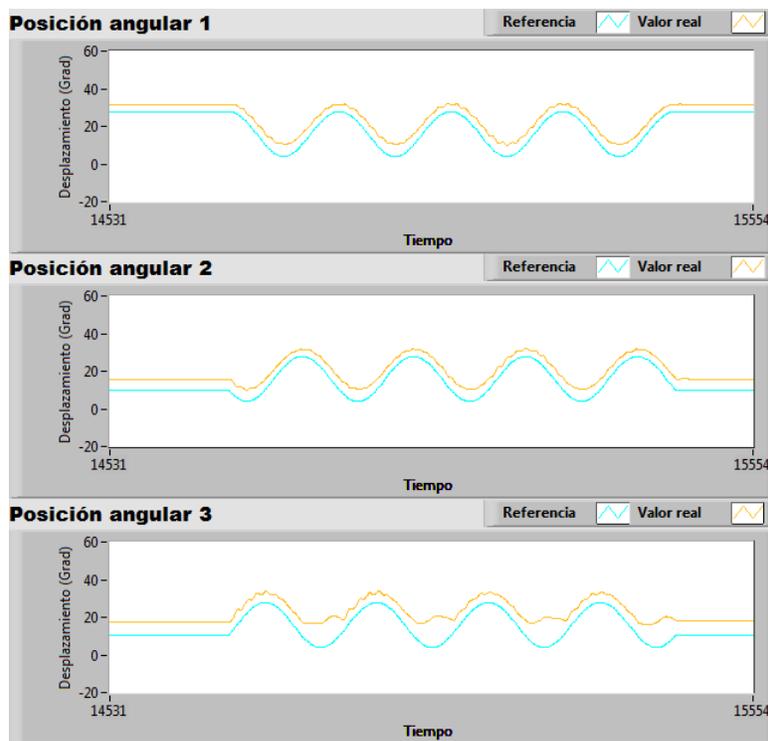


Figura 6.20: Comparación entre referencia y posición real

La Figura 6.20 detalla de mejor manera como es el comportamiento del sistema respecto a la señal de referencia. Se observa que el RPD sigue la referencia de manera paralela pero sin alcanzarla, existe aproximadamente 5 grados de error en el seguimiento de trayectoria.

En las Figuras 6.19 y 6.20 Se puede observar que para el tercer actuador hay determinadas posiciones a las cual no puede llegar esto se debe en parte a ligeras desproporciones de la carga que soporta este actuador respecto a los otros 2; y debido a que las ganancias del controlador para cada motor son idénticas. Algunas posiciones no pueden ser alcanzadas por el actuador 3 debido a que el controlador no puede proporcionar más energía con el fin de reducir el error de posicionamiento.

6.8. Conclusiones

La finalidad de este capítulo es dar a conocer a detalle la instrumentación de los actuadores y sensores empleados en RPD y la plataforma de desarrollo usada para la programación en este caso LabVIEW. Se puede observar que los sensores empleados únicamente proporcionan datos y mediciones acerca del estado del robot más no del entorno en que se desempeña el robot, esto debido a que únicamente se desea controlar el posicionamiento del robot. Los resultados de experimentación mostrados en este capítulo demuestran que es necesario realizar un control más robusto para mejorar el

comportamiento del RPD, para lo cual es necesario implementar controladores para regular las variables de velocidad y par de los actuadores.

Referencias

- [6.1] OMEGA Engineering inc., “La importancia de la instrumentación. <http://mx.omega.com/prodinfo/instrumentacion.html>.”
 - [6.2] L. Core, “Course manual,” *National Instruments*, 1.
 - [6.3] N. Kehtarnavaz and S. Mahotra, *Digital Signal Processing Laboratory: LabVIEW-Based FPGA Implementation*. Universal-Publishers, 2010.
 - [6.4] National Instruments, “Fpgas a fondo <http://www.ni.com/white-paper/6983/es/>.”
 - [6.5] B. Reliance, “Paquete de información de producto mte-3363-blbcn,” Baldor Electric Company.
 - [6.6] National Instruments, “Ni 9423. <http://sine.ni.com/nips/cds/view/p/lang/es/nid/208818>.”
 - [6.7] B. W. Edmister, “Industrial applications of optical shaft encoders,” in *1980 Huntsville Technical Symposium*, pp. 99–105, International Society for Optics and Photonics, 1980.
 - [6.8] National Instruments, “Ni 9403. <http://sine.ni.com/nips/cds/view/p/lang/es/nid/208811>.”
 - [6.9] G. F. Franklin, J. D. Powell, and M. L. Workman, *Digital control of dynamic systems*, vol. 3. Addison-wesley Menlo Park, 1998.
 - [6.10] National Instruments, “Ni 9264. <http://sine.ni.com/nips/cds/view/p/lang/es/nid/208807>.”
 - [6.11] L. Americas, “Technical manual for genesys 750w/1500w programable dc power supplies,” 405 Essex Road, Neptune NJ USA.
 - [6.12] Fuij Electric, “6mbi 50l-060 igbt module (1 series).”
-

A

Creación de un proyecto en LabVIEW FPGA

Para poder crear un nuevo proyecto en LabVIEW FPGA, primeramente hay que tener instalada en la computadora una versión completa de LabVIEW, en este caso, se trabajó con la versión 2012, al iniciar LabVIEW, se abrirá la siguiente ventana la cual es la Figura A.1, donde se selecciona la opción LabVIEW FPGA Project.

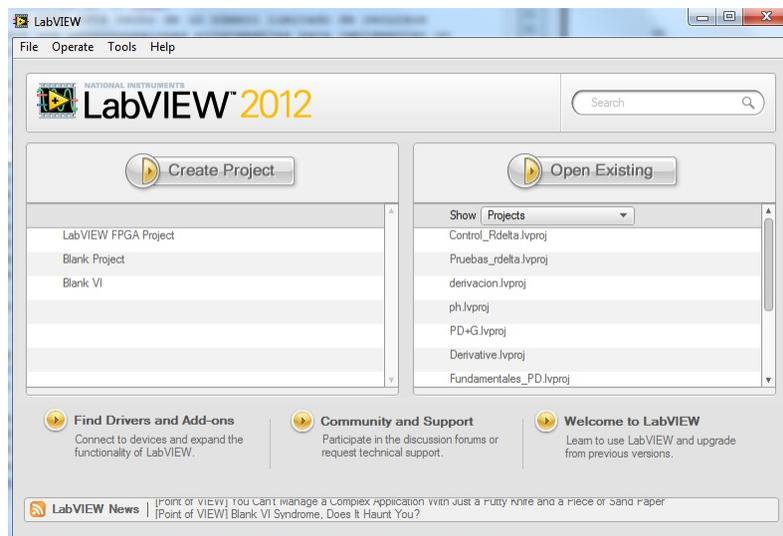


Figura A.1: Ventana de inicio en LabVIEW 2012

Al dar click en LabVIEW FPGA Project, se despliegan opciones para seleccionar el tipo de proyecto, se selecciona CompactRIO Reconfigurable Embedded System, y damos click en Next, ver Figura A.2.

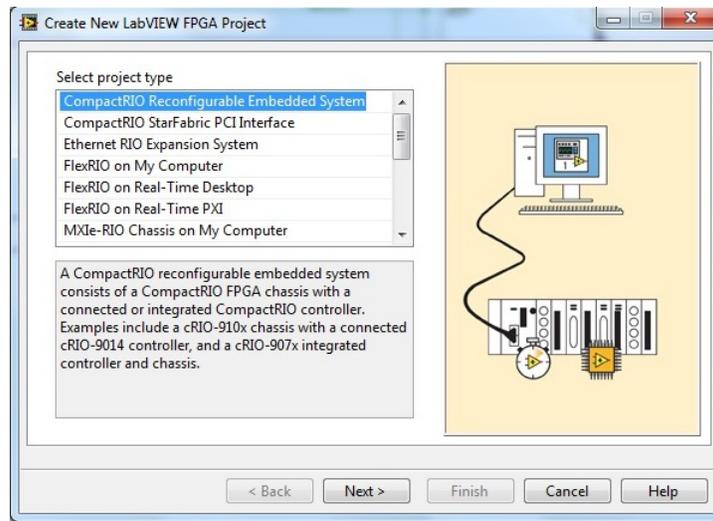


Figura A.2: Selección de tipo de proyecto en LabVIEW 2012

El siguiente paso simplemente es que la computadora busque si hay un dispositivo FPGA conectado al puerto ethernet, y nuevamente se da click en siguiente, ver Figura A.3.



Figura A.3: Ventana para encontrar dispositivos FPGA LabVIEW 2012

Al momento de detectar algún dispositivo, se mostrara una ventana como se observa en la Figura A.4, en la cual aparece el nombre del dispositivo, en este caso en un cRIO-9014.

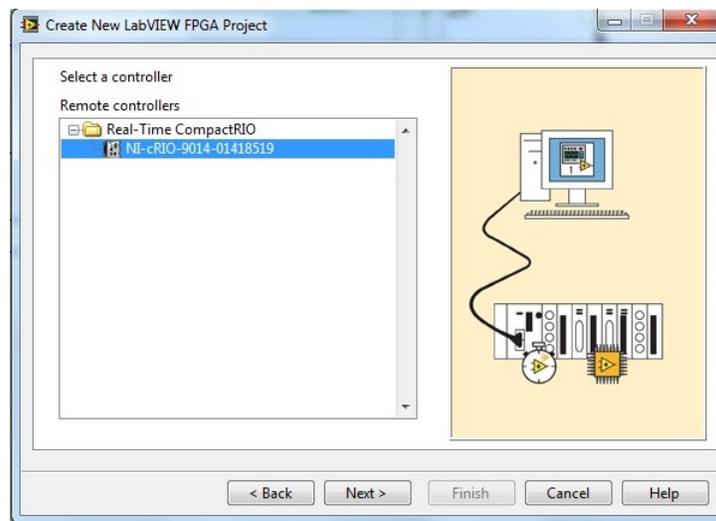


Figura A.4: Selección de controlador en LabVIEW 2012

Cuándo se presiona (next) en la pantalla anterior, se mostraran mas detalles de el hardware con el que se dispone, como es el chasis del cRIO, y la tarjeta FPGA, ver la Figura A.5.

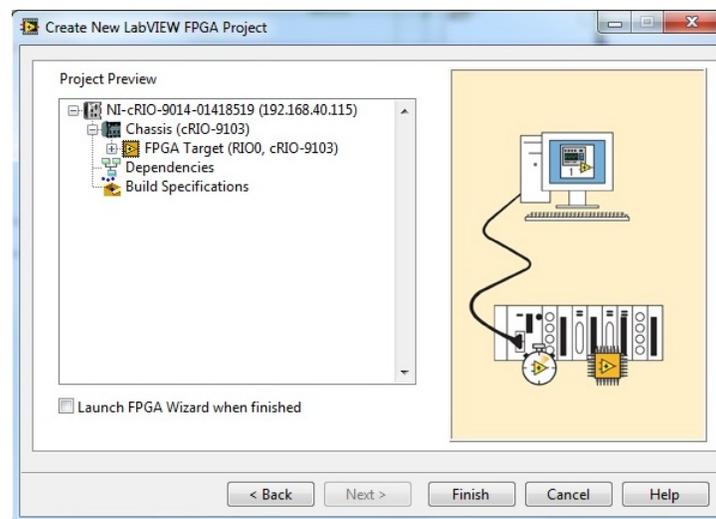


Figura A.5: Detalles de Hardware del cRIO

En este punto, se finaliza el proceso de creación de un proyecto en FPGA, la siguiente ventana que se abrirá será la del proyecto creado, esta ventana se muestra en la Figura A.6.

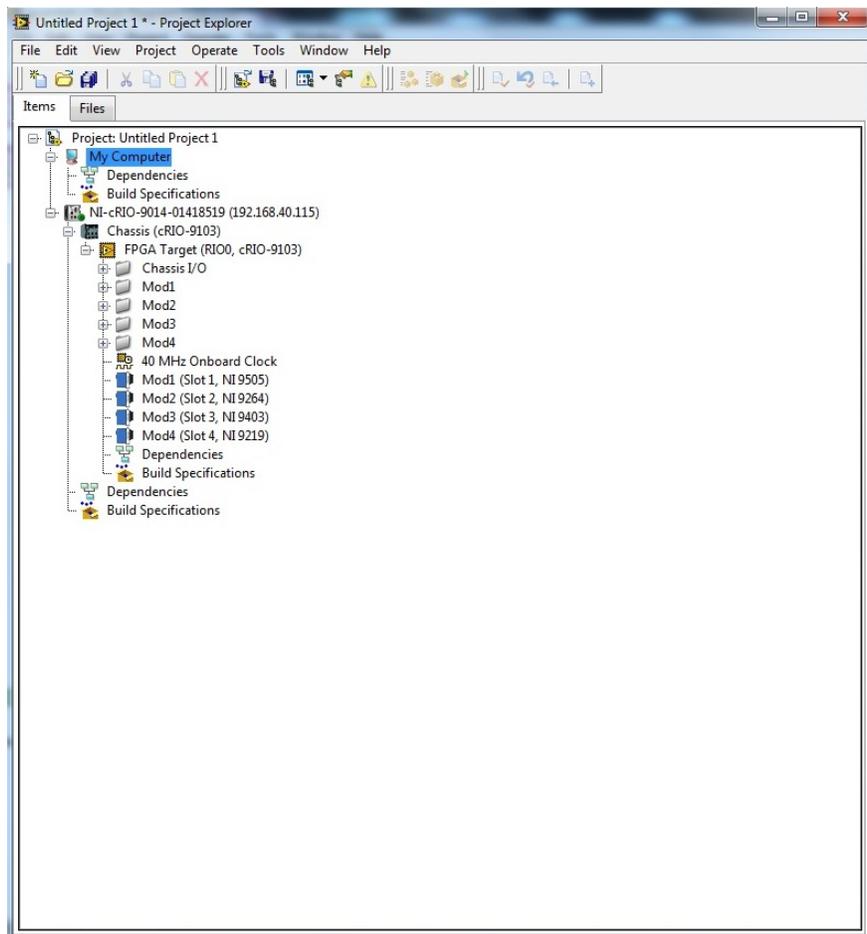


Figura A.6: Proyecto nuevo en LabVIEW FPGA

En la ventana principal del proyecto se pueden apreciar distintos componentes que conforman el árbol de proyecto, como es el modelo del cRio junto con el modelo del chasis. El chasis es importante porque es donde se colocan los distintos módulos de entradas y salidas del cRio. Desplegando el menú del chasis, aparece la tarjeta FPGA que esta dentro del cRio, además se observa la velocidad del reloj, en esta sección, se despliegan los módulos que tiene el chasis del cRio. El chasis del cRio con el que se trabajó, consta de 4 slots, y dentro de los slots, se encuentran los siguientes módulos:

- NI 9505, el cual es un controlador para motores de corriente directa con escobillas
- NI 9264 el cual son salidas analógicas (16 canales), el rango de voltaje de salida es de $\pm 10V$ con una resolución de 16 bits
- NI 9403 el cual son E/S digitales de 5V/TTL, este módulo incluye 32 canales
- NI 9219, el cual es un módulo de entradas analógicas de 4 canales

B

Diagramas eléctricos

B.1. Conexión de encoder

Para conectar los codificadores ópticos (Encoder) al cRio, se hace uso del módulo NI 9403 en el cual cada canal puede ser configurado como entrada o salida, en este caso se configuran 6 canales como entradas. El voltaje de alimentación es tomado del conector tipo serial del NI 9505 en el cual el pin 5 es el voltaje de 5V y el pin 9 el común. El diagrama de conexión se muestra en la Figura B.1.

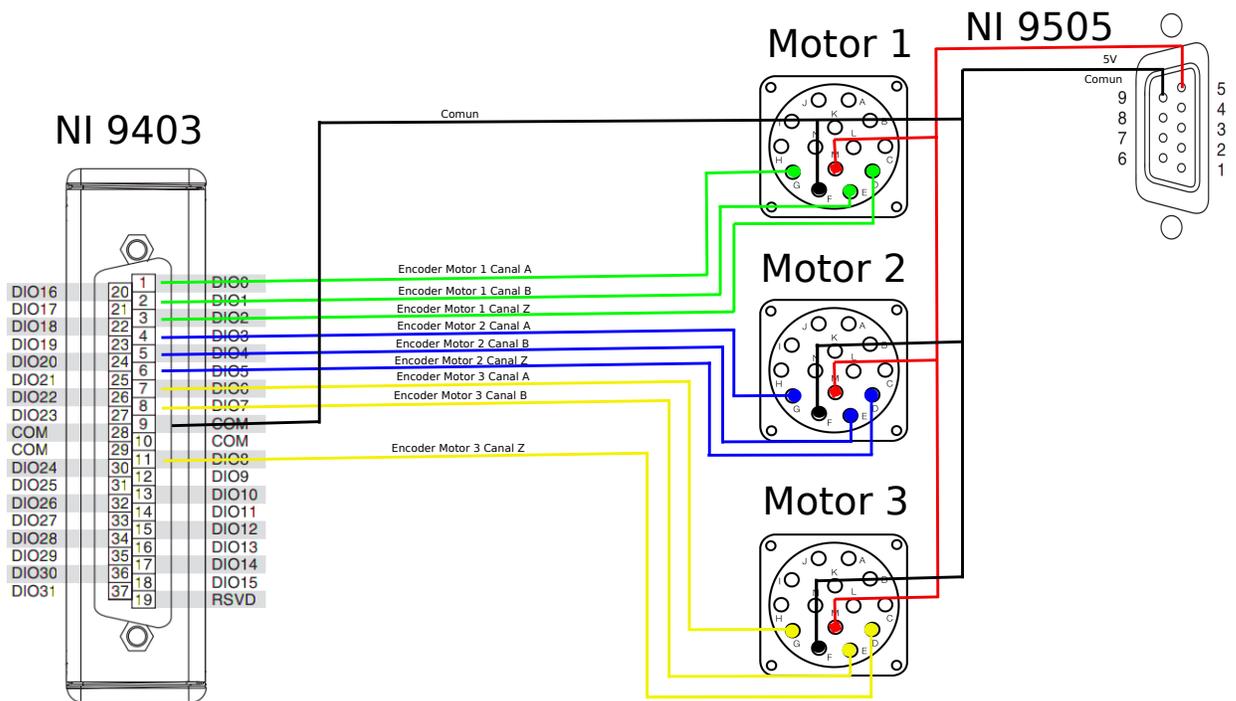


Figura B.1: Conexión de los codificadores ópticos con el NI 9403

El cable empleado para las conexiones es el LD5481A20 de Baldor Electric, el conector del cable tiene las siguientes terminales, las cuales se detallan en la Tabla B.1.

Tabla B.1: Pines de conexión del cable LD5481A20

Pin	Función	Color de cable	Calibre
A	Brake	Violeta	18
B	Tacómetro +	Café	20
C	Motor -	Azul	14
D	Encoder canal Z	Gris	22
E	Encoder canal B	Verde	22
F	Común	Azul	22
G	Encoder canal A	Blanco	22
H	Motor +	Rojo	14
I	Brake	Negro	18
J	Encoder canal B negado	Amarillo	22
K	Tacómetro -	Blanco	20
L	Encoder canal Z negado	Rosa	22
M	+5VDC	Rojo	22
N	Encoder canal A negado	Café	22

B.2. Conexión de los sensores de proximidad

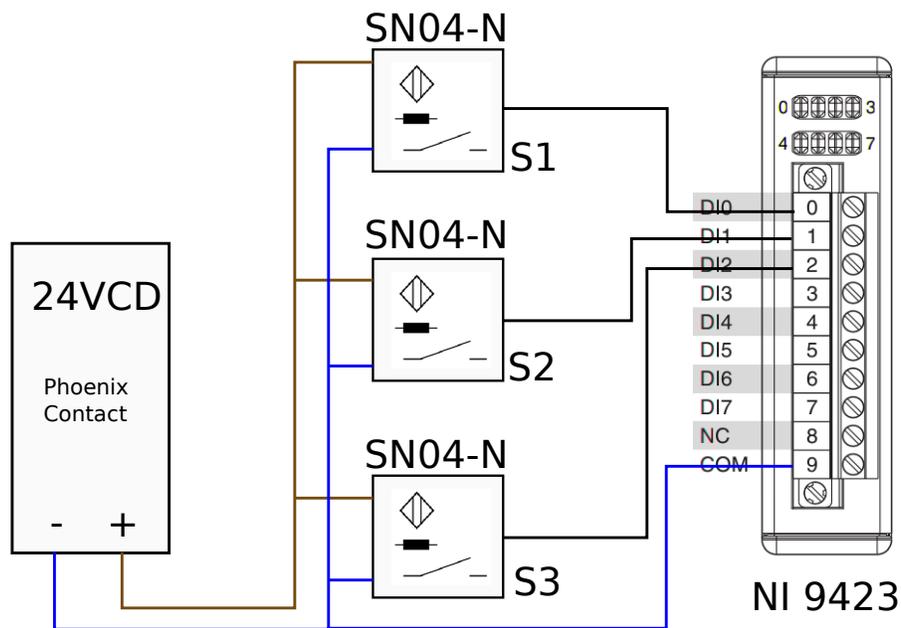


Figura B.2: Conexión de los sensores inductivos con el NI 9423

La Figura B.2 muestra la conexión de los sensores de proximidad inductivos con el módulo NI 9423. Los sensores se alimentan con la fuente Phoenix contact, la cual proporciona un voltaje de 24V.

B.3. Etapa de potencia

Los siguientes diagramas corresponden a la etapa de potencia de los actuadores; los cuales son alimentados mediante una fuente controlada de manera independiente, la señal de control se envía por medio del NI 9264 y entra en el pin 10 del conector DB25 de la fuente, además se necesita una señal fija para controlar la corriente la cual se envía de igual manera por medio del NI 9264 y entra en el pin 9 de la fuente controlada. El diagrama para el motor 1 se muestra en la Figura B.3.

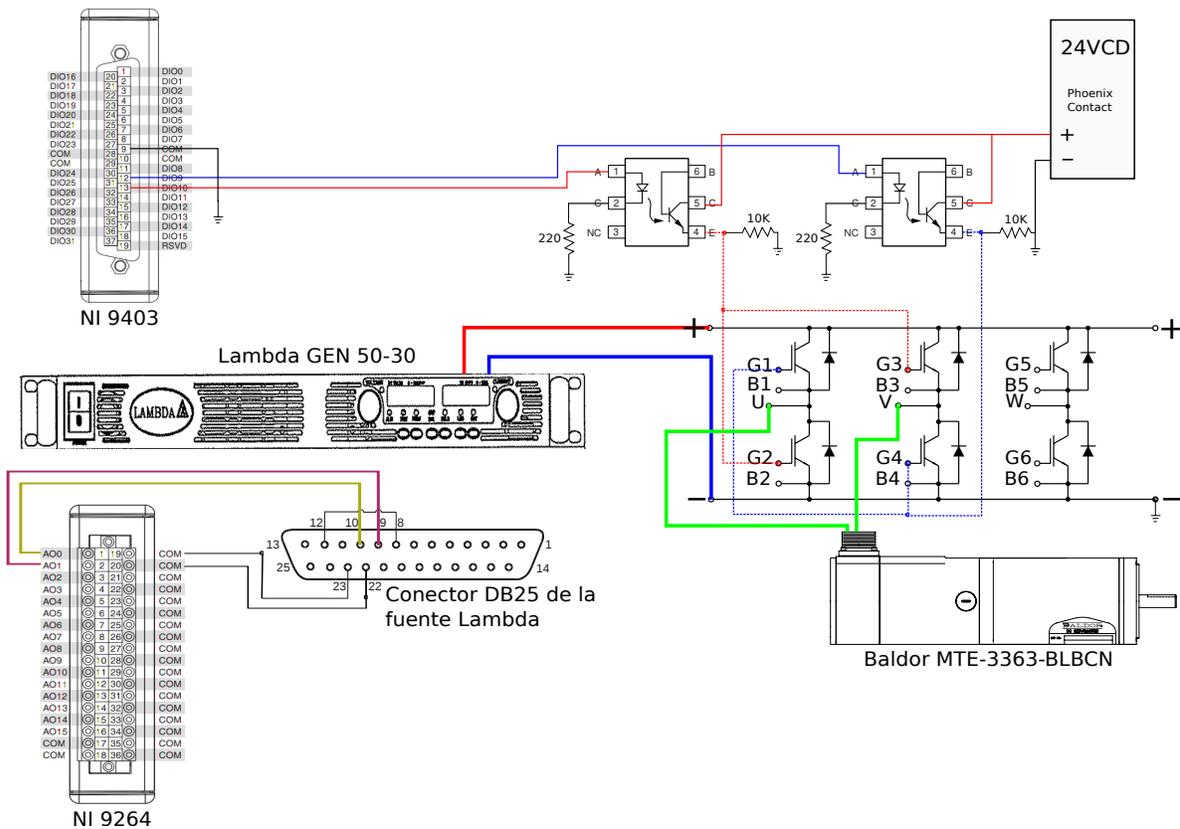


Figura B.3: Etapa de potencia del primer motor

Los diagramas de control para el segundo y tercer motor se muestran en las Figuras B.4 y B.5 respectivamente; los cuales son muy similares al diagrama de la Figura A.3. La diferencia son las salidas de control de los módulos NI 9403 y NI 9264 ya que deben ser distintas para controlar cada motor de manera independiente.

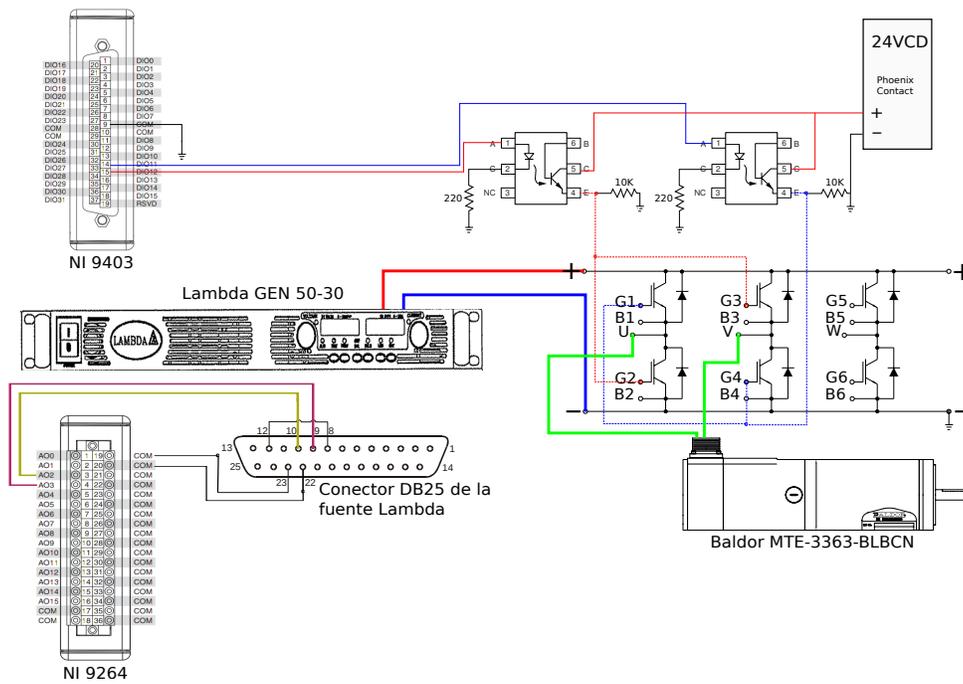


Figura B.4: Etapa de potencia del segundo motor

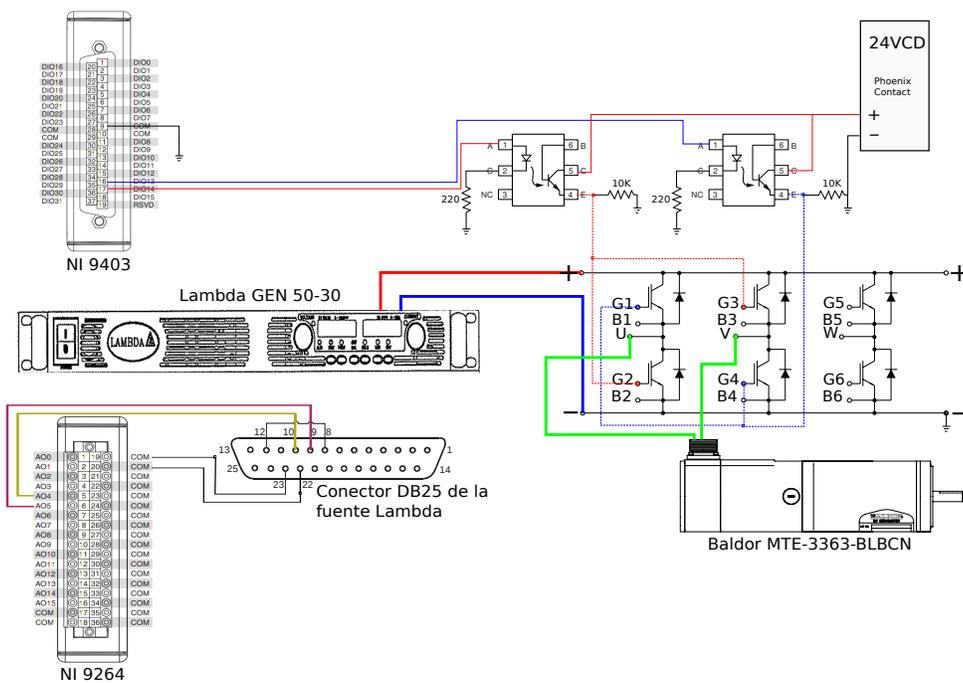


Figura B.5: Etapa de potencia del tercer motor

La inversión de giro del motor de CD se logra por medio de un arreglo de puente H usando IGBT'S, el módulo empleado de los IGBT'S es el 6MBI50J-120 de Fuji Electric el cual contiene 6 transistores; sin embargo únicamente se emplean 4 para la inversión de giro. Los IGBT'S son activados por medio de optocopladores, los cuales interrumpen 24V para accionar a los IGBT'S, el modelo empleado para los optocopladores es el 4N28. Para activar los optocopladores se emplea el NI 9403 en modo salida.

Trabajos futuros

La parte central de este trabajo consiste en controlar la posición de la plataforma móvil del RPD; sin embargo el controlar únicamente esta variable no es suficiente para que el prototipo se pueda desempeñar adecuadamente en un entorno real de trabajo. Por tal motivo se propone desarrollar los siguientes trabajos.

- Diseñar e implementar un control de velocidad y par de los motores de CD del RPD.
- Construir los drivers para cada motor de CD, esto debido a que momentáneamente se están empleando las fuentes controladas del laboratorio como etapa de potencia junto un puente H de IGBT's.
- Colocar un efector final sobre la plataforma móvil del RPD, en este caso se desea adquirir una ventosa de vacío, para tomar y dejar objetos de masa ligera dentro de su espacio de trabajo.