



**Universidad Autónoma
de Querétaro**

Facultad de ingeniería

Control de movimiento para un robot (ROMAT 56).

TESIS

Que como parte de los requisitos para obtener el grado
de Ingeniero en Automatización con Línea Terminal en
Sistemas Mecatrónicos.

Presenta

Salvador Ortiz Santos

Asesor de tesis

Dr. Juvenal Rodríguez Reséndiz

Querétaro, Qro., Mayo, 2012



Universidad Autónoma de Querétaro
Facultad de Ingeniería
Ingeniería en automatización

Control de movimiento para un robot (ROMAT 56).

TESIS

Que como parte de los requisitos para obtener el grado de Ingeniero en Automatización con Línea Terminal en Sistemas Mecatrónicos.

Presenta:

Salvador Ortiz Santos

Dirigido por:

Dr. Juvenal Rodríguez Reséndiz

SINODALES

Dr. Juvenal Rodríguez Reséndiz
Presidente



Firma

Dr. Edgar Alejandro Rivas Araiza
Secretario



Firma


Ing. Fernando Bonilla Arenas
Suplente

Firma


Dr. Roberto Valentín Carrillo Serrano
Suplente



Firma



Dr. Aurelio Domínguez González
Director de la Facultad de Ingeniería



Dr. Juvenal Rodríguez Reséndiz
Coordinador de Ingeniería en Automatización

Centro Universitario
Querétaro, Qro.
Mayo 2012
México

Resumen

El trabajo que se presenta a continuación es el diseño del control de movimiento de un robot industrial, el sistema de control lo conforman la parte de hardware y en el software. En la parte de software se presenta la cinemática del robot industrial Romat 56, para la cual se utilizaron las coordenadas de Denavit – Hartenberg para la cinemática inversa y directa, las cuales nos permiten conocer la posición y la orientación del órgano terminal que es donde se encuentra instalada la herramienta. Una parte importante del trabajo es el diseño de la interfaz gráfica de usuario que se realizó en LabView 10.0 que tiene un fácil manejo. El hardware consiste en un controlador PID que utiliza dispositivos de bajo costo, el cual permite que el robot tenga buena precisión.

Palabras claves: Robot industrial, Software, Tecnología, Controlador PID.

Summary

This work presents the design of motion control of an industrial robot as well as hardware and software. The theoretical part presents the industrial robot kinematics Romat 56 with the help of the coordinates of Denavit - Hartenberg to solve the inverse kinematics and direct kinematics where they help us to know the position and orientation of the end-effector, where the tool is installed. An important part of work is the design of graphical user interface which was performed in 10.0 LabView, an easy program to use. Hardware is designed with low cost devices but it will help the robot to reach a good accuracy with a PID controller system.

Keywords: Industrial Robot, Software, Technology, PID Controller

**A mis padres (Marina y Ricardo), a mis hermanos (Hilda, Adrian,
Imeldo), a mi familia, a mis amigos.
Por hacer esto posible.
Gracias por todo.**

Agradecimientos

Estoy muy contento de culminar con una carrera que siempre fue de exigencias y desvelos. Por lo que le estoy muy agradecido a toda mi familia por ese gran apoyo que me brindaron y sé que seguirán apoyándome en las decisiones de mi vida.

Agradezco al Dr. Juvenal Rodríguez Reséndiz por su orientación, su paciencia y por transmitir sus conocimientos. Así mismo, a la M. en C. Georgina del Carmen Mota Valtierra por su apoyo, sus consejos y por su colaboración. También agradezco al Dr. Edgar Alejandro Rivas, al Dr. Roberto Valentín Carrillo y al Ing. Fernando Bonilla por sus comentarios y sus críticas constructivas.

Gracias al Dr. Gilberto Herrera Ruiz por haber hecho de la facultad un lugar de formación de calidad. Al Dr. Roberto Augusto Gómez Loenzo por su apoyo y por incluirme en proyectos. También quiero agradecer a Eduardo Cabrera Martínez, a Humberto Aguirre Becerra y a Ramón García Cortés por estar en aquellos momentos difíciles, por su amistad, su ayuda y sus consejos.

Por último y no menos importante quiero agradecer a todos mis amigos y compañeros por todos los momentos compartidos.

Índice general

Resumen.....	II
Summary.....	III
Agradecimientos.....	V
Índice de figuras.....	VIII
Índice de cuadros.....	X
I. Introducción.....	1
1.1 Automatización y Robótica.....	1
1.1.1 Automatización Fija.....	1
1.1.2 Automatización programable.....	2
1.1.3 Automatización Flexible.....	3
1.2 Robótica.....	5
1.3 Antecedentes.....	7
1.3.1 Antecedentes de robots industriales realizados en la Universidad Autónoma de Querétaro.....	7
1.3.2 Estadísticas de suministro de robots industriales.....	8
1.4 Justificación.....	12
1.5 Hipótesis.....	14
1.6 Objetivo.....	14
2. Fundamentación teórica.....	15
2.1 Estructuras básicas.....	15
2.2 Notación de Denavit – Hartenberg.....	19
2.3 Posición y orientación.....	26
2.3.1 Orientación.....	26
2.3.2 Posición.....	28
2.4 Cinemática.....	30
2.4.1 Cinemática directa.....	30

2.4.2 Cinemática inversa.....	31
2.5 Controladores.....	31
2.5.1 Control proporcional.....	31
2.5.2 Control integral.....	32
2.5.3 Control proporcional más integral (P+I).	32
2.5.4 Control proporcional más derivativo (P+D).	33
2.5.5 Control Proporcional más integral más derivativo (P+I+D).	34
3. Metodología	35
3.1 Robot industrial Romat 56.....	35
3.2 Sistema de control de lazo cerrado para el robot industrial Romat 56.	37
3.3 Sistema de control en diagrama de bloques.	38
3.4 Sistema de control del robot Romat 56.....	39
3.5 Interfaz gráfica de usuarios.....	50
3.6 Librerías que permiten establecer la comunicación entre Labview 10.0 y el controlador de movimiento.....	58
4. Resultados y conclusiones.	59
4.1 Resultados.....	59
4.2 Conclusión	66
Bibliografía.....	69
5. APENDICE.....	71
5.1 Diagrama de conexión.....	72
5.2 Conectores.....	75
5.3 Partes de código para el control de movimiento.....	78

Índice de figuras

Figura 1. Relación de la automatización fija, automatización programable y automatización flexible.	4
Figura 2. Suministro anual a nivel mundial de robots industriales.	8
Figura 3. Suministro anual de robots industriales por regiones.	9
Figura 4. Suministro anual de robots industriales más grande de los mercados.	10
Figura 5. Suministro anual de robots industriales de mercados en crecimiento.	11
Figura 6. Configuración RRR.	16
Figura 7. Configuración esférica.	16
Figura 8. Configuración cilíndrica.	17
Figura 9. Configuración cartesiana.	17
Figura 10. Configuración SCARA.	18
Figura 11. Las tramas de los vínculos se unen de manera que la trama $\{i\}$ se una rígidamente al vínculo i	22
Figura 12. El sistema de coordenadas según la convención de Denavit – Hartenberg. ...	24
Figura 13. Se representa la posición y orientación del órgano terminal con respecto a un punto de referencia.	26
Figura 14. Rotación que nos lleva de $F(i)$ a $F(i+1)$	28
Figura 15. Robot industrial (Romat 56).	36
Figura 16. Estructura de un sistema de control de lazo cerrado (Jacob Tal, 1989).	37
Figura 17. Sistema de control de una articulación (Masatoshi Nakamura, 1998).	38
Figura 18. Control del robot Romat 56 (Masatoshi Nakamura, 1998).	39
Figura 19. Diagrama de conexiones de dispositivos para el Romat 56.	40
Figura 20. Controlador de movimiento DMC-1866 de Galil Motion Control.	41
Figura 21. Función elemental de un controlador de movimiento (Jacob Tal, 1989).	42
Figura 22. Terminal de conexiones ICM 2900.	43
Figura 23. Servo amplificador.	44
Figura 24. Elementos de un amplificador en modo de velocidad (Jacob Tal, 1989).	45
Figura 25. Servo amplificadores instalados para el robot Romat 56.	45
Figura 26. Motor DC.	46
Figura 27. Señal de encoder; señal 1 (canal -A), señal 2 (canal +A), señal 3 (canal +B) y señal 4 (canal -B).	47
Figura 28. Función de sensor de fin de carrera.	48
Figura 29. Tarjetas de control de los sensores de fin de carrera.	49
Figura 30. Página de inicio.	51
Figura 31. Página de control manual.	52
Figura 32. Página de control de punto a punto.	54
Figura 33. Página de archivos.	55

Figura 34. Página de configuración.....	57
Figura 35. Gráfica de posición, velocidad y torque (prueba 1).	67
Figura 36. Gráfica de posición, velocidad y torque (prueba 2).	68
Figura 37. Hoja1 diagrama de conexion.....	73
Figura 38. Hoja 2 diagrama de conexion.....	74
Figura 39. Conectores del gabinete.	75
Figura 40. Conector DB-9 para señales de encoder.	76
Figura 41. Conector para motor y sensores.	76
Figura 42. Conector para sensor de fin de carrera.....	77
Figura 43. Conector para motor y freno.	77

Índice de cuadros

Tabla 1.1: Antecedentes de robots industriales en la Universidad Autónoma de Querétaro.	7
Tabla 4.1: Prueba 1 respuesta del sistema.	60
Tabla 4.2: Prueba 2 respuesta del sistema con PID.	62
Tabla 4.3: Prueba 3 respuesta del sistema.	63
Tabla 4.4: Prueba 4 control del sistema con PID.	65
Tabla 5.a. Conexión en el conector de los señales de encoders de cada motor.	75
Tabla 5.b. Conexión en el conector de alimentación y señales de control.....	76

I. Introducción

La robótica hoy en día es una gran herramienta en el sector industrial dado que tiene una aplicación infinita por su flexibilidad para la realización de diversas tareas, razón por la cual se ha incrementado la utilización de esta tecnología en sus diversos procesos dentro de la industria.

En este capítulo se presenta la noción de la robótica, así como los antecedentes, la definición dando como pregunta ¿Qué es un robot industrial?, se expondrán los motivos del por qué es importante este trabajo de investigación, para finalmente concluir con el objetivo del trabajo.

1.1 Automatización y Robótica

La automatización y la robótica son dos tecnologías estrechamente relacionadas, en un contexto industrial podemos definir la automatización como una tecnología que está relacionada con el empleo de sistemas mecánicos electrónicos y basados en computadoras en la operación y control de la producción. Ejemplos de esta tecnología son; líneas de transferencia, máquinas de montaje mecanizado, sistemas de control de realimentación (aplicados a los procesos industriales), máquinas – herramientas con control numérico y robots. En consecuencia, la robótica es una forma de automatización industrial (Groover et al., 1990).

Hay tres clases amplias de automatización industrial; *automatización fija*, *automatización programable* y *automatización flexible*.

1.1.1 Automatización Fija

La automatización fija se utiliza cuando el volumen de producción es muy alto, y por tanto es adecuado para diseñar equipos especializados para procesar el producto (o un componente de un producto) con alto rendimiento y con elevadas tasas de producción. Un ejemplo de la automatización fija puede encontrarse en la

industria del automóvil, en donde líneas de transferencia muy integradas y constituidas por varias decenas de estaciones de trabajo se utilizan para operaciones de maquinado en componentes de motores y transmisiones. La economía de la automatización fija es tal que el costo de los equipos especiales puede dividirse entre un gran número de unidades y los costos unitarios resultantes son bajos en relación con los métodos de producción alternativos. El riesgo encontrado con la automatización fija es que al ser el costo de inversión inicial elevado, si el volumen de producción resulta ser más bajo de lo previsto, los costos unitarios se incrementarían más de lo previsto. Otro problema con la automatización fija es que el equipo está especialmente diseñado para obtener el producto, y una vez que se haya acabado el ciclo de vida del producto es probable que el equipo quede obsoleto. Para productos con cortos de ciclos de vida, el empleo de la automatización fija representa un gran riesgo (Groover et al., 1990).

1.1.2 Automatización programable

La automatización programable se emplea cuando el volumen de producción es relativamente bajo y hay una diversidad de producción a obtener. En este caso el equipo de producción está diseñado para ser adaptable a variaciones en la configuración del producto. Esta característica de adaptabilidad se realiza haciendo funcionar el equipo bajo el control de un “programa” de instrucciones que se preparó especialmente para el producto dado. El programa se introduce por lectura en el equipo de producción y este último realiza la secuencia particular de operaciones de procesamiento (o montaje) para obtener el producto. En términos de economía, el costo del equipo programable puede repartirse entre un gran número de productos, aun cuando sean diferentes. Gracias a la característica de programación y a la adaptabilidad resultante del equipo, muchos productos diferentes y únicos en su género pueden obtenerse económicamente en pequeños lotes (Groover et al., 1990).

1.1.3 Automatización Flexible

Existe una tercera categoría entre la automatización fija y automatización programable que se denomina “automatización flexible”. Otros términos utilizados para la automatización flexible incluyen los “sistemas de fabricación por computadora” y los “sistemas de fabricación integrados por computadora”. La experiencia adquirida hasta ahora con este tipo de automatización, indica que es más adecuado para el rango de producción de volumen medio, como se ilustra en la Figura 1, por su posición relativa con los otros dos tipos de automatización los sistemas flexibles tienen algunas de las características de la automatización fija y de la automatización programable; deben programarse para diferentes configuraciones de productos, pero la diversidad de las configuraciones suelen estar más limitadas que para el caso de la automatización programable, lo que permite que se produzca un cierto grado de integración en el sistema. Los sistemas automatizados flexibles suelen estar contruidos por una serie de estaciones de trabajo que están interconectadas por un sistema de almacenamiento y manipulación de materiales. Una computadora central se utiliza para controlar las diversas actividades que se producen en el sistema, encaminando las diversas piezas a las estaciones adecuadas y controladas las operaciones programadas en las diferentes estaciones (Groover et al., 1990).

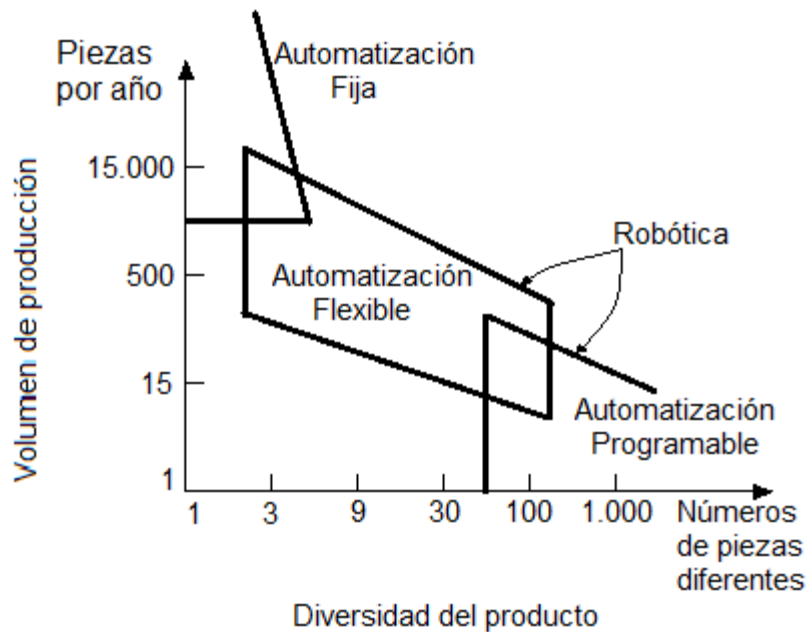


Figura 1. Relación de la automatización fija, automatización programable y automatización flexible.

Una de las características que distingue a la automatización programable de la automatización flexible es que con la automatización programable los productos se obtienen en lotes. Cuando se completa un lote, el equipo se reprograma para procesar el siguiente lote. Con la automatización flexible, diferentes productos pueden obtenerse al mismo tiempo en el mismo sistema de fabricación. Esta característica permite un nivel de versatilidad que no está disponible en la automatización programable pura, como se definió anteriormente. Esto significa que pueden obtenerse productos en un sistema flexible en lotes si ello fuera deseable, o varios estilos de productos diferentes pueden mezclarse en el sistema. La potencia de cálculo de la computadora de control es lo que posibilita esta versatilidad.

1.2 Robótica

De los tres tipos de automatización, la robótica coincide más estrechamente con la automatización programable, un robot industrial es una máquina programable de uso general que tiene algunas características antropomórficas o “humanoides”, la característica humanoide más típica de los robots actuales es de los brazos móviles. El robot puede programarse para desplazar un brazo a través de una secuencia de movimientos con el fin de realizar alguna tarea de utilidad. Repetirá este modelo de movimientos una y otra vez hasta que se re programe para ejecutar alguna otra tarea. Por consiguiente, la característica de programación permite que los robots se utilicen para una diversidad de operaciones industriales diferentes, muchas de las cuales implican el trabajo del robot junto con otros elementos del equipo automatizados o semi-automatizados. Estas operaciones incluyen la carga y descarga de máquinas, la soldadura por arcos y puntos, procesos de fundición, corte por agua/láser y la pintura por pulverización.

La definición “oficial” de un robot industrial se proporciona por Robotics Industries Association (RIA), anteriormente Robotics Institute of America (RIA):

“Un robot industrial es un manipulador multifuncional reprogramable diseñado para desplazar materiales, piezas, herramientas o dispositivos especiales mediante movimientos programados variables para la ejecución de tareas diversas.”

Aunque los propios robots son ejemplo de automatización programable, a veces se utilizan en la automatización flexible e incluso en sistemas de automatización fija. Estos sistemas están constituidos por varias máquinas y/o robots que trabajan juntos y se suelen controlar por una computadora o un controlador programable. Una línea de producción que realiza soldaduras por puntos en las carrocerías de los automóviles es un ejemplo de esta clase de sistema. La línea de soldadura podría estar constituida por una veintena de robots o más y es capaz de realizar centenares de soldaduras por puntos separadas en

dos o tres estilos de cuerpos diferentes. Los programas de robots están contenidos en la computadora o controlador programable y se cargan en cada robot para la carrocería de automóvil particular que ha de soldarse en cada estación de trabajo. Gracias a esta característica, dicha línea podría considerarse como un sistema de automatización flexible de alta producción (Ollero, 2001).

1.3 Antecedentes

1.3.1 Antecedentes de robots industriales realizados en la Universidad Autónoma de Querétaro.

Fecha	Descripción
2000	Tesis desarrollada por José Joaquín Castellanos Galindo titulado <i>“Diseño y construcción de una tarjeta de control de arquitectura abierta”</i> .
2001	Tesis desarrollada por Erika Martínez Ramírez titulado <i>“Simulador de manipuladores robóticos desacoplados 6R”</i>
2002	Tesis desarrollada por Carlos Guillermo López Guevara titulado <i>“Modelo y validación del modelo cinemática y de pares gravitacionales de un robot industrial”</i>
2003	Tesis desarrollada por José Felipe Aguilar Pereyra titulado <i>“Control de movimiento de un robot industrial”</i>
2003	Tesis desarrollada por Roberto Augusto Gómez Loenzo titulado <i>“Diseño e implementación de una celda robotizada de soldadura”</i>
2007	Tesis desarrollada por Miguel Ángel Martínez Prado titulado <i>“Sistema de Control para Robots Industriales”</i>
2009	Tesis desarrollada por Levy Yashin Bello León titulado <i>“Optimización de referencias y control de los actuadores de un robot PUMA para el mejoramiento de su dinámico”</i>
2010	Tesis desarrollada por J. Jesús Cano Morales titulado <i>“Identificación de servomotores en un robot PUMA, mediante redes neuronales implementadas en FPGA”</i>

Tabla 1.1: Antecedentes de robots industriales en la Universidad Autónoma de Querétaro.

1.3.2 Estadísticas de suministro de robots industriales

En la figura 2 se muestra el suministro de robots industriales donde se puede notar el incremento excepto en el año 2009 donde la crisis económica mundial impacto a todos los sectores productivos. También notamos que para los siguientes años la venta de robots industriales se pronostica que tendrá un gran incremento, esto se debe a la reactivación de la economía de cada país, ya que el uso de estos equipos hace que el sector industrial sea más eficiente y productivo. Por ello muchas empresas están adquiriendo esta tecnología para una mayor eficiencia, ayudando a la calidad, los tiempos y la cantidad de productos que a su vez puede llegar a generar para su crecimiento.

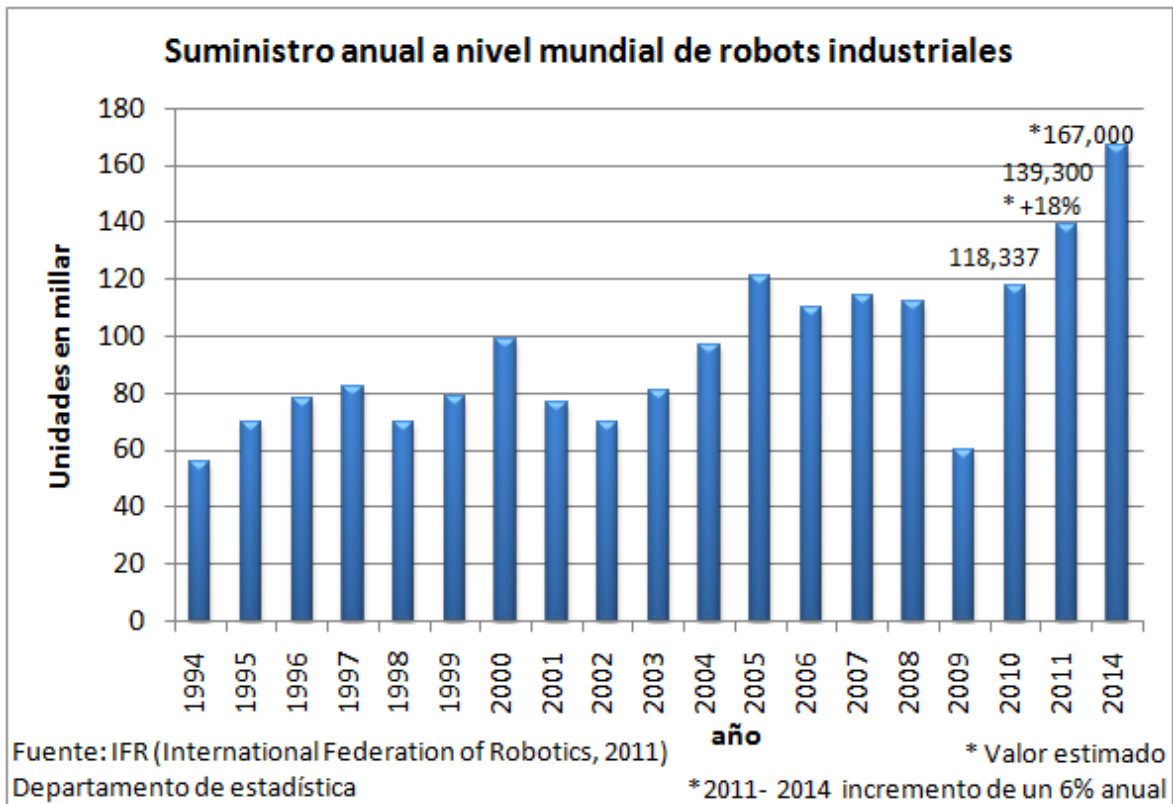


Figura 2. Suministro anual a nivel mundial de robots industriales.

En la figura 3 se muestra el suministro de robots industriales por regiones donde notamos que Asia tiene un gran número de robots industriales, así mismo podemos notar la crisis económica del 2009 que afectó a todas las regiones, sin embargo en estos últimos años sabemos que estos equipos van en incremento.

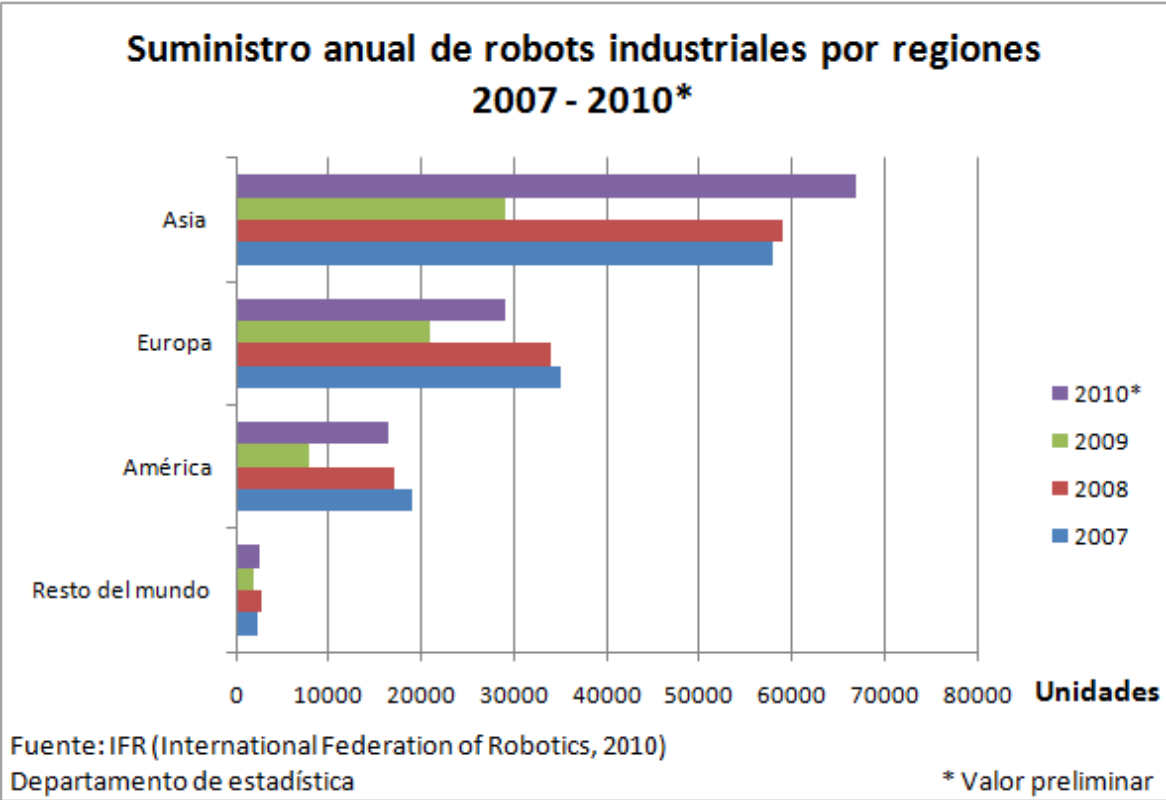


Figura 3. Suministro anual de robots industriales por regiones.

En la figura 4 se muestra el suministro de robots industriales más grande de los mercados y con esta gráfica podremos justificar la anterior donde notábamos que Asia tiene un número muy grande de robots industriales esto se debe a que Japón es consumidor y productor de esta tecnología pero también genera otros tipos de tecnología lo cual hace que este país sea de primer mundo. Esto permite que Japón cuente con una gran cantidad de industrias, por esto y por lo que hemos visto en estos últimos años que se habla mucho de China, porque este país va en crecimiento de forma exponencial y hoy en día es un país de primer mundo, estos factores hace que Asia sea una de las regiones que tiene un suministro muy alto de robots industriales.

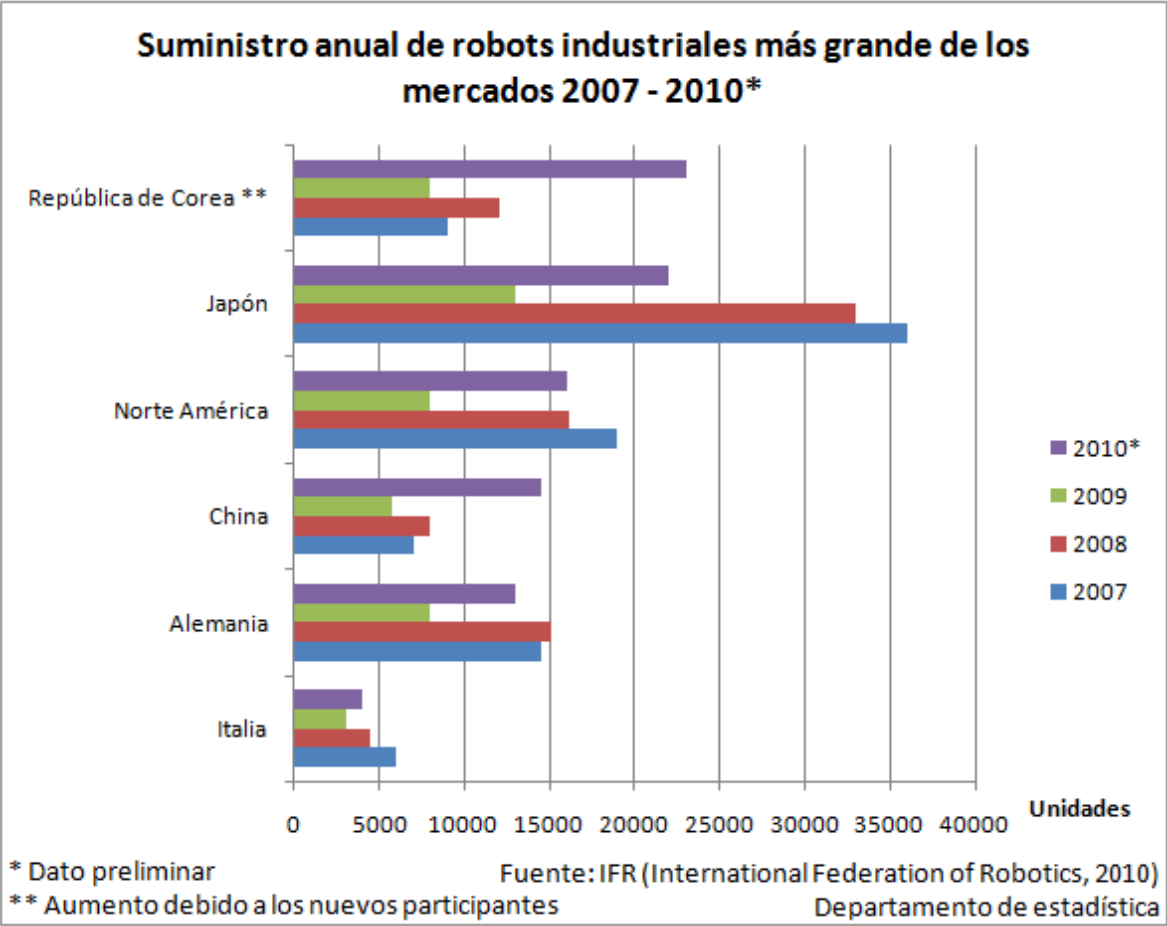


Figura 4. Suministro anual de robots industriales más grande de los mercados.

En la figura 5 se muestra el suministro de robots industriales de países en crecimiento, donde vemos nuevamente China que ocupa un número significativo de estos productos, pero también notamos a Brasil e India que en estos últimos años han ido en crecimiento. Estos países tienen un número importante de agricultura que les ayuda a crecer en la economía de su país pero que también notamos que están invirtiendo en el sector industrial donde crecieron en un número importante a causa de la crisis económica mundial que estos aprovecharon para su crecimiento.

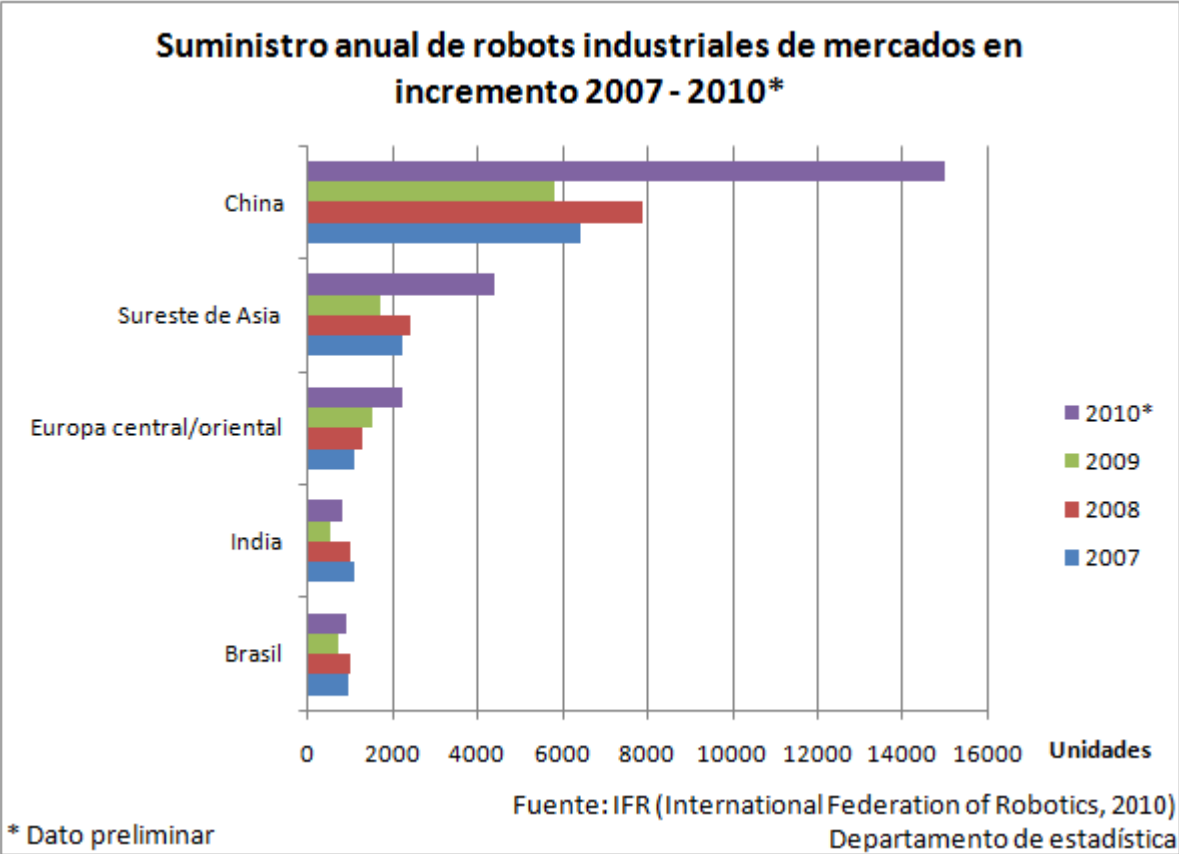


Figura 5. Suministro anual de robots industriales de mercados en crecimiento.

1.4 Justificación

De acuerdo a los antecedentes de la robótica, esta evoluciona directamente con las necesidades del ser humano o más específicamente con las necesidades de la industria, ya que la competitividad en el mercado es cada vez mayor.

Una de las principales razones del crecimiento en el uso de robots industriales es la reducción de costos y mejora de calidad, con esto deberíamos hacernos la siguiente pregunta ¿Cuáles son las ventajas en el uso de los robots industriales comparadas con máquinas especiales? Las ventajas de los robots es que son cada vez más efectivos, rápidos, precisos y flexibles, sin embargo una de sus grandes desventajas es que los robots tienen una inversión inicial costosa.

Sin embargo, sabemos que para cualquier industria la calidad, los costos, tiempos y seguridad son algunos de las principales razones para la competitividad hacia el mercado. Por lo que muchas industrias invierten en este tipo de tecnología. Desafortunadamente no todas las industrias cuentan con una economía sólida para la inversión de esta tecnología, podríamos decir que solo las grandes industrias pueden presumir de esta tecnología ya que los costos son elevados.

Por todo lo expuesto anteriormente es que se considera importante la realización de un sistema de control para un robot industrial que contribuya al desarrollo de las pequeñas y medianas. El sistema propuesto también permitirá una fácil manipulación lo cual significa que no es necesario contar con personal especializado para su manejo.

Otro de los grandes problemas que tienen los robots industriales comerciales es que usan tecnologías que solo son conocidas por la empresa desarrolladora, donde su reparación o mantenimiento suelen ser muy caros porque solo se puede acudir a estas empresas ya que solo ellos pueden realizarlas. Por ello en este trabajo fue muy importante considerar que el sistema

debería aplicar tecnología que es fácil de sustituir y conseguir con una gran variedad de precios según las necesidades o aplicaciones que este requiera.

Por otra parte la manipulación del robot es muy importante ya que la mayoría de los robots presentan una interfaz gráfica que no es nada familiar para el operador y en donde sus orígenes de desarrollo realmente se desconocen. Esto nos lleva a trabajar en una interfaz gráfica donde el usuario tenga todas las herramientas necesarias y que además sea muy fácil de usar. Es por esto que se decidió trabajar bajo el entorno de un software muy conocido por las industrias y las instituciones educativas LabView, todo ello con el propósito de dar más herramientas a las empresas para cualquier mejora o alguna pequeña modificación necesaria en el robot.

1.5 Hipótesis

Es posible generar un control de movimiento de un robot Romat 56 con algoritmos clásicos de control, y la implementación de una interfaz gráfica de usuario con la integración de dispositivos electrónicos comunes que permita que el usuario tenga una mayor interacción con el robot.

1.6 Objetivo

Desarrollar un sistema de control de movimiento para un robot industrial que sea accesible no solo a grandes empresas sino también a pequeñas y medianas empresas, contribuyendo al desarrollo de la región.

Desarrollar una interfaz grafica de usuario que ayude al operador interactuar con un robot industrial sin la necesidad de contar con un conocimiento amplio en el tema de la robótica.

2. Fundamentación teórica

2.1 Estructuras básicas

Las estructuras que comúnmente se aplican son brazos compuestos por elementos con articulaciones entre ellos, donde en la parte final se coloca el órgano terminal o efector final como puede ser una pinza o algún otro dispositivo especial para realizar operaciones.

Los robots industriales se diferencian por el tipo de estructura que cambian las propiedades en cuanto a espacio de trabajo y accesibilidad a posiciones determinadas. El espacio de trabajo es un conjunto de puntos en el espacio tridimensional en los que se puede situar el órgano terminal del robot industrial, cuenta con un número infinito de puntos en las que se puede situar el robot desde su estructura totalmente plegada y hasta su estructura totalmente extendida (Ollero, 2001).

Por otra parte los puntos en el espacio de trabajo no tiene la misma accesibilidad, por ello muchas de ellas se hace con la ayuda de la orientación.

Configuración articulada (RRR)

Esta configuración también se conoce como manipulador antropomorfo por su semejanza con el brazo humano (ver figura siguiente) posee tres articulaciones de revolución lo que le permite gran libertad de movimiento.

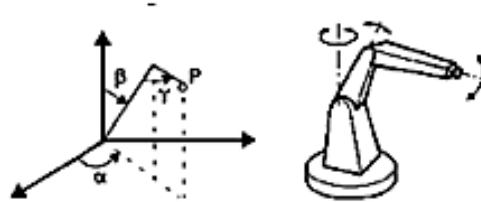


Figura 6. Configuración RRR.

Configuración esférica (RRP)

Las coordenadas esféricas definen la posición del órgano terminal con respecto al sistema de coordenadas de la base. En la siguiente figura se puede notar que cuenta con dos articulaciones angulares y un desplazamiento lineal.

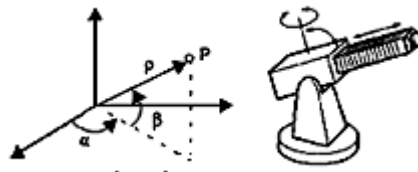


Figura 7. Configuración esférica.

Configuración cilíndrica (RPP)

Como su nombre lo indica la variable de la articulación son coordenadas cilíndricas del órgano terminal con respecto a su base, en la siguiente figura podemos notar que contamos con una articulación angular y contamos con dos desplazamiento.

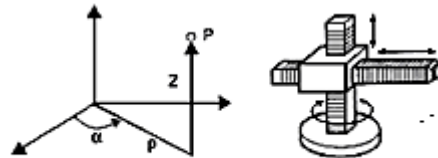


Figura 8. Configuración cilíndrica.

Configuración cartesiana (PPP)

La configuración cartesiana es la más sencilla ya que cuenta con articulaciones primaticas, es decir de desplazamiento lineal, son muy usados para la traslación de materiales, en la siguiente figura podemos ver que las tres articulaciones son prismáticas.

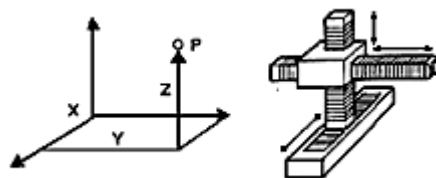


Figura 9. Configuración cartesiana.

Configuración SCARA (RRP)

El manipulador SCARA (*Selective Compliant Articulated Robot for Assembly*) como su nombre lo indica, está diseñada para operaciones de ensamble, este tiene una gran semejanza con la configuración cilíndrica la diferencia son en las articulaciones y sus aplicaciones.

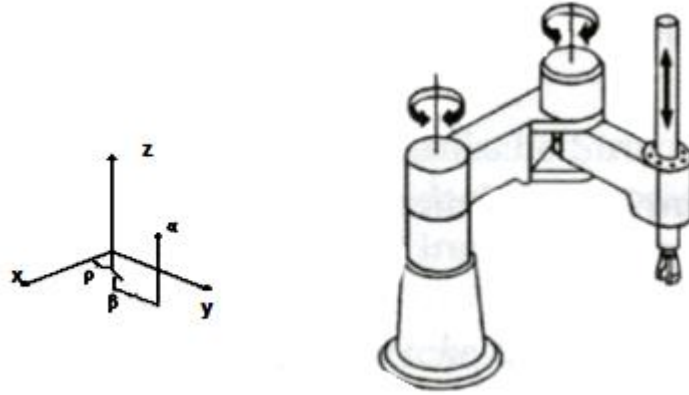


Figura 10. Configuración SCARA.

2.2 Notación de Denavit – Hartenberg.

Para relacionar dos elementos contiguos se puede realizar con cualquier sistema de referencia ligada a cada elemento, pero la forma más habitual de realizar en la robótica es con la representación de Denavit – Hartenberg (D – H). En 1955 Denavit y Hartenberg propusieron un método matricial para la obtención de la localización que este debe tomar para cada sistema de coordenada S_i ligada a cada eslabón i de una cadena articulada, para poder sistematizar la obtención de las ecuaciones cinemáticas de la cadena completa.

De acuerdo a los sistemas de coordenadas asociadas a cada eslabón según la representación propuesta por (D–H), este será posible pasar de uno al siguiente eslabón mediante 4 transformaciones básicas que dependen de la estructura geométrica del eslabón. Las 4 transformaciones básicas consisten de la rotación y la traslación que relaciona el sistema de referencia del elemento $i - 1$ al elemento i . Las 4 transformaciones están sujetas del sistema S_{i-1} al S_i , que son definidas de acuerdo a unas normas que se presentan posteriormente:

1. Rotación alrededor del eje z_{i-1} un ángulo θ_i .
2. Traslación a lo largo de z_{i-1} una distancia d_i , vector $\mathbf{d}_i(0,0,d_i)$.
3. Traslación a lo largo de x_i una distancia a_i , vector $\mathbf{a}_i(a_i,0,0)$.
4. Rotación alrededor del eje x_i un ángulo α_i .

Las transformaciones se refieren al sistema móvil.

Dado que el producto de matrices no es conmutativo, las transformaciones se realizan de la siguiente manera:

$$A_i^{i-1} = Rotz(\theta_i)T(0,0,d_i)T(a_i,0,0)Rotx(\alpha_i)$$

Y realizando el producto de matriz se tiene:

$$A_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_i^{i-1} = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Donde θ_i , d_i , a_i , α_i son parámetros Denavit – Hartenberg del eslabon i . Identificando estos parámetros podemos obtener las matrices A_i^{i-1} y relacionar así todos y cada uno de los eslabones del robot.

Para encontrar los parámetros de **Denavit – Hartenberg** se sigue el siguiente algoritmo:

1. Numerar los eslabones comenzando con 1 (primer eslabona móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
2. Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n .
3. Localizar el eje de cada articulación. Si esta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
4. Para i de 0 a $n - 1$ situar el eje z_i sobre el eje de la articulación $i + 1$.
5. Situar el origen del sistema de la base S_0 en cualquier punto del eje z_0 . Los ejes x_0 y y_0 se situarán de modo que formen in sistema dextrógiro con z_0 .
6. Para i de 1 a $n - 1$, situar el origen del sistema S_i (solidario al eslabón i) en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i . Si ambos ejes se contarán, S_i se situaría en el punto de corte. Si fuesen paralelos S_i se situaría en la articulación $i + 1$.

7. Situar x_i en la línea normal común a z_{i-1} y z_i .
8. Situar y_i de modo que forme un sistema dextrógiro con x_i y z_i .
9. Situar el sistema S_n en el extremo del robot de modo que z_n coincida con la dirección de z_{n-1} y x_n sea normal a z_{n-1} y z_n .
10. Obtener θ_i como el ángulo que hay que girar en torno a z_{n-1} para que x_{i-1} y x_i queden paralelos.
11. Obtener d_i como la distancia, medida a lo largo de z_{i-1} , que habría que desplazar S_{i-1} para que x_i y x_{i-1} quedasen alineados.
12. Obtener a_i como la distancia medida a lo largo de x_i (que ahora coincidiría con x_{i-1}) que habría de desplazar el nuevo S_{i-1} para que su origen coincidiese con S_i .
13. Obtener α_i como el ángulo que habría que girar entorno a x_i , para que el nuevo S_{i-1} coincidiese totalmente con S_i .
14. Obtener las matrices de transformación A_i^{i-1} .
15. Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot $T = A_1^0 \cdot A_2^1 \cdot A_n^{n-1}$.
16. La matriz T define la orientación (submatriz de rotación) y posición (submatriz de translación) del extremo referido a la base, en función de las n coordenadas articulares.

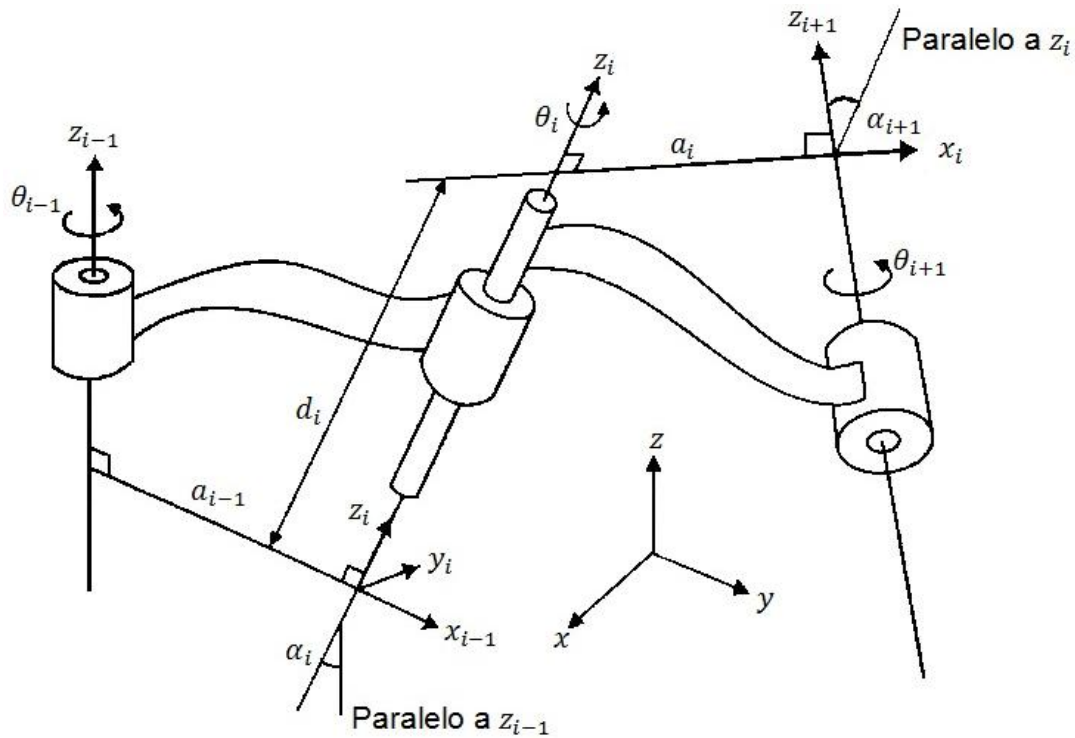


Figura 11. Las tramas de los vínculos se unen de manera que la trama $\{i\}$ se una rígidamente al vínculo i .

θ_i Es el ángulo que forman los ejes x_{i-1} y x_i medido en un plano perpendicular al eje z_{i-1} , utilizando la regla de la mano derecha. Se trata de un parámetro variable en articulaciones giratorias.

d_i Es la distancia a lo largo del eje z_{i-1} desde el origen del sistema de coordenadas $(i-1)$ -ésimo hasta la intersección del eje z_{i-1} con el eje x_i . Se trata de un parámetro variable en articulaciones prismáticas.

a_i Es a la distancia a lo largo del eje x_i que va desde la intersección del eje z_{i-1} con el eje x_i hasta el origen del sistema i -ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes z_{i-1} y z_i .

α_i Es el ángulo de separación del eje z_{i-1} y el eje z_i , medido en un plano perpendicular al eje x_i , utilizando la regla de la mano derecha.

Una vez obtenidos los parámetros DH, el cálculo de las relaciones entre los eslabones consecutivos del robot es inmediato, ya que vienen dadas por las matrices A, que se calcula según la expresión general.

Las relaciones entre eslabones no consecutivos vienen dadas por las matrices T que se obtienen como producto de un conjunto de matrices A.

Obtenida la matriz T, esta expresara la orientación (submatriz (3x3) de rotación) y posición (submatriz (3x1) de traslación) del extremo del robot en función de sus coordenadas articulares, con lo que quedara resuelto el problema de cinemática directa (*esta se expone en las páginas posteriores*).

En la figura 12 se muestra los sistemas de coordenados sujetas a la descripción de Denavit – Hartenberg.

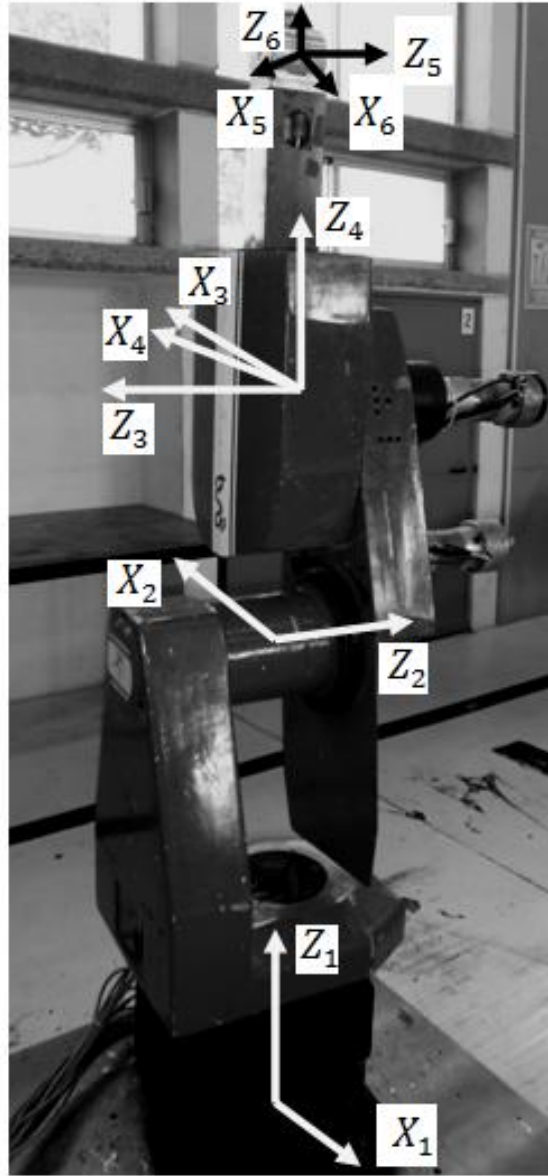


Figura 12. El sistema de coordenados según la convención de Denavit – Hartenberg.

A partir de los sistemas de coordenadas señalados anteriormente se establecen los parámetros estructurales y estos se muestran en la siguiente tabla 3.1, la importancia de estos parámetros es necesaria para modelar la posición y orientación de sus eslabones a partir de dichos marcos coordenados.

Unión i	$a_i(\text{mm})$	$b_i(\text{mm})$	α_i	$\lambda_i = \cos\alpha_i$	$\mu_i = \text{sen}\alpha_i$
1	0	895	90°	0	1
2	430	0	180°	-1	0
3	0	0	90°	0	1
4	0	430	90°	0	1
5	0	0	90°	0	1

Tabla 3.1: Parámetros de Denavit – Hartenberg para el Romat 56.

2.3 Posición y orientación

En los robots es necesario conocer la ubicación del órgano terminal en el espacio tridimensional por ello se habla mucho de encontrar la posición y la orientación del órgano terminal con una referencia que generalmente está situada en la base del robot donde se usan coordenadas absolutas o coordenadas relativas.

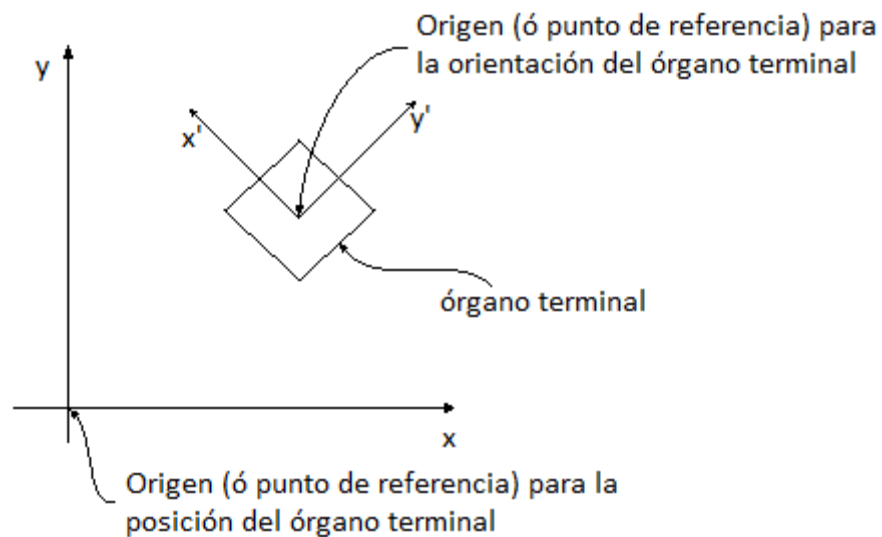


Figura 13. Se representa la posición y orientación del órgano terminal con respecto a un punto de referencia.

2.3.1 Orientación

Para el análisis de la orientación del órgano terminal (End Effector), podemos suponer que el origen de F_i y F_{i+1} es la misma. Como nota tener en cuenta que las rotaciones no son conmutativas.

En la figura 14 se muestra una rotación sobre el eje Z_i y posteriormente una rotación sobre el eje X_a . Para calcular la matriz de rotación Q_i que lleva a los ejes X_i, Y_i, Z_i en ejes $X_{i+1}, Y_{i+1}, Z_{i+1}$, nos apoyaremos en un marco coordenado auxiliar F_a , que nos ayuda a representar a Q_i como un producto de rotación simple.

En la figura 14(a) tenemos el marco F_a después de haber rotado el marco F_i alrededor del eje Z_i con un ángulo θ_i . En consecuencia, la figura 14(b) nos muestra el marco F_{i+1} a causa de haber rotado el marco F_a alrededor del eje X_a con un ángulo de α_i .

Por otra parte la matriz Θ_i describe la rotación de F_i a F_a , si está referido al marco F_i entonces tenemos:

$$[\Theta_i]_i = \begin{bmatrix} \cos \theta_i & -\text{sen } \theta_i & 0 \\ \text{sen } \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Si definimos

$$\lambda_i = \cos \alpha_i, \quad \mu_i = \text{sen } \alpha_i$$

Por lo tanto tenemos a la matriz R_i que describe la rotación de F_a a F_{i+1} , si lo referimos al marco F_a entonces tenemos que:

$$[R_i]_a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \lambda_i & -\mu_i \\ 0 & \mu_i & \lambda_i \end{bmatrix}$$

Como se explico anteriormente que Q_i es el producto de rotaciones simples con respecto al marco F_i donde tenemos que:

$$[Q_i]_i = [\Theta_i]_i [R_i]_a$$

Entonces

$$[Q_i]_i = \begin{bmatrix} \cos \theta_i & -\lambda_i \text{sen } \theta_i & \mu_i \text{sen } \theta_i \\ \text{sen } \theta_i & \lambda_i \cos \theta_i & -\mu_i \cos \theta_i \\ 0 & \mu_i & \lambda_i \end{bmatrix}$$

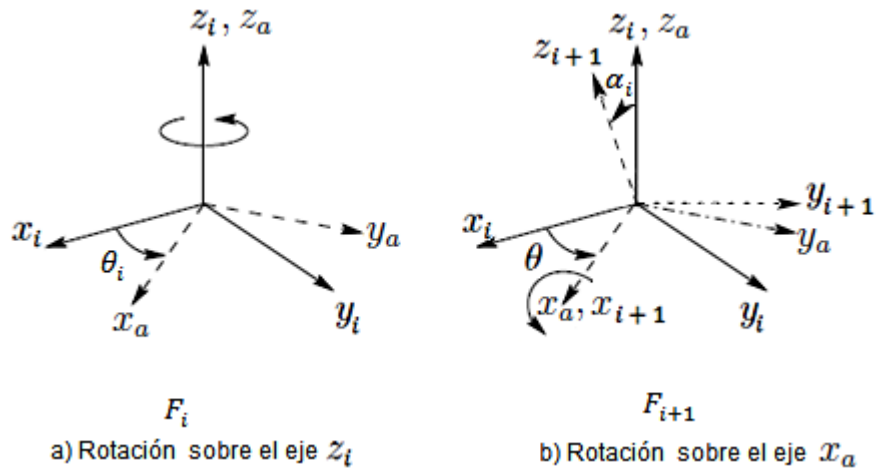


Figura 14. Rotación que nos lleva de $F(i)$ a $F(i+1)$.

2.3.2 Posición

Una de las tareas que se realizan en los robots es encontrar la posición del órgano terminal con respecto a una referencia con punto $(0,0,0)$ para un espacio tridimensional dando como resultado un punto en este espacio donde:

$$\mathbb{R}^3 = \{(x, y, z) \mid x, y, z \in \mathbb{R}\}$$

Por otro lado el origen de O_i del marco F_i con el origen O_{i+1} del marco F_{i+1} se conecta por un vector que denotaremos por T_i , este vector determina la posición relativa entre dos marcos coordenados. Para el cálculo de T_i tendremos que pasar por el marco F_a por lo tanto tenemos que:

$$T_i = \overrightarrow{O_i O_{i+1}} = \overrightarrow{O_i O_a} + \overrightarrow{O_a O_{i+1}}$$

Por definición, b_i es la coordenada sobre el eje Z_i de O_a (figura 14), intersección de Z_i, Z_a con X_a, X_{i+1} . Por lo tanto:

$$\overrightarrow{[O_i O_a]}_i = \begin{bmatrix} 0 \\ 0 \\ b_i \end{bmatrix}$$

Además, a_i por definición es la distancia entre Z_i y Z_{i+1} y que son perpendiculares con X_a, X_{i+1} , entonces tenemos

$$\overrightarrow{[O_a O_{i+1}]_{i+1}} = \begin{bmatrix} a_i \\ 0 \\ 0 \end{bmatrix}$$

En contraste para poder sumar los vectores $\overrightarrow{O_i O_a}$ con $\overrightarrow{O_a O_{i+1}}$ estos deben estar referidos al mismo marco de coordenadas. Entonces para sumar con respecto a F_i debemos calcular.

$$\overrightarrow{[O_a O_{i+1}]_i} = [Q_i]_i \overrightarrow{[O_a O_{i+1}]_{i+1}}$$

Por lo tanto

$$\overrightarrow{[O_a O_{i+1}]_i} = \begin{bmatrix} a_i \cos \theta_i \\ a_i \operatorname{sen} \theta_i \\ 0 \end{bmatrix}$$

Entonces

$$[T_i]_i = \overrightarrow{[O_i O_{i+1}]_i} = \overrightarrow{[O_i O_a]_i} + \overrightarrow{[O_a O_{i+1}]_i} = \begin{bmatrix} a_i \cos \theta_i \\ a_i \operatorname{sen} \theta_i \\ b_i \end{bmatrix}$$

Por consiguiente y recordando que $[Q_i]_i^{-1} = [Q_i]_i^T$ tenemos

$$\begin{aligned} [T_i]_{i+1} &= \overrightarrow{[O_i O_{i+1}]_{i+1}} = \overrightarrow{[O_i O_a]_{i+1}} + \overrightarrow{[O_a O_{i+1}]_{i+1}} = [Q_i]_i^{-1} \overrightarrow{[O_i O_a]_i} + \overrightarrow{[O_a O_{i+1}]_{i+1}} \\ &= \begin{bmatrix} \cos \theta_i & -\lambda_i \operatorname{sen} \theta_i & \mu_i \operatorname{sen} \theta_i \\ \operatorname{sen} \theta_i & \lambda_i \cos \theta_i & -\mu_i \cos \theta_i \\ 0 & \mu_i & \lambda_i \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ b_i \end{bmatrix} + \begin{bmatrix} a_i \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} a_i \\ b_i \mu_i \\ b_i \lambda_i \end{bmatrix} \end{aligned}$$

2.4 Cinemática

La *cinemática* es una rama de la mecánica que estudia las leyes del movimiento de los cuerpos sin tener en consideración las causas que lo producen, limitándose, esencialmente, al estudio de la trayectoria en función del tiempo es decir calcula las trayectorias, velocidades y aceleraciones estas dos últimas son las que describen como cambian su posición en función del tiempo.

2.4.1 Cinemática directa

La cinemática directa se usa para calcular la localización de un objeto en el espacio tridimensional con respecto a una referencia fija. Se sabe que un robot está formado por una cadena cinemática de objetos rígidos o eslabones unidos entre sí mediante articulaciones, donde se establece un sistema de referencia fijo lo cual está situado generalmente en la base del robot o en algún punto del espacio tridimensional la cual es llamado origen, por lo que esté se representa con coordenadas (0,0,0). La cinemática directa describe la localización de la posición y la orientación del órgano terminal con respecto a dicho sistema de referencia. Los cálculos se realizan a través de la localización de cada uno de los eslabones.

Para determinar la posición \mathbf{p} y la orientación \mathbf{Q} , conocido los valores de las variables de las uniones θ_i para $i = 1, \dots, 6$ esto se reduce a un cálculo algebraico.

$$\begin{aligned} [Q]_1 &= [Q_6]_1 [Q_5]_1 [Q_4]_1 [Q_3]_1 [Q_2]_1 [Q_1]_1 \\ [p]_1 &= [T_1]_1 + [T_2]_1 + [T_3]_1 + [T_4]_1 + [T_5]_1 + [T_6]_1 \end{aligned}$$

O también

$$\begin{aligned} [Q]_1 &= [Q_1]_1 [Q_2]_2 [Q_3]_3 [Q_4]_4 [Q_5]_5 [Q_6]_6 \\ [p]_1 &= [T_1]_1 + [Q_1]_1 [T_2]_2 + [Q_1]_1 [Q_2]_2 [T_3]_3 + [Q_1]_1 [Q_2]_2 [Q_3]_3 [T_4]_4 \\ &\quad + [Q_2]_2 [Q_3]_3 [Q_4]_4 [T_5]_5 + [Q_1]_1 [Q_2]_2 [Q_3]_3 [Q_4]_4 [Q_5]_5 [T_6]_6 \end{aligned}$$

2.4.2 Cinemática inversa

La cinemática inversa se describe de la siguiente manera, dada una posición y una orientación del órgano terminal del robot, esta encuentra todos los conjuntos posibles de ángulos de cada articulación que podrían usarse para alcanzar la posición y orientación que se desea.

La cinemática inversa cuenta con sistemas de ecuaciones cinemáticas no lineales, esto significa que no siempre cuenta con una solución puede existir múltiples soluciones o puede que la solución no exista.

2.5 Controladores

La principal función del controlador es comparar la respuesta real de la salida del sistema con la orden de entrada o más comúnmente llamado *referencia* para proporcionar una señal de control que reducirá el error a cero o lo más cercano posible del cero.

Existen 3 acciones básicas de control que se utilizan por separado o en combinación que son el control proporcional, control derivativo y el control integral. A continuación se describe la operación de cada tipo de controlador, los más comúnmente usados.

2.5.1 Control proporcional.

En algunos casos se desea tener un control suave y un error que sea casi nula en el estado estacionario, donde el controlador proporcional es apto para este tipo de trabajos. Dado que el controlador proporcional desarrolla una señal de control proporcional al error. Principalmente actúa como un controlador con una ganancia K_p , que es representado como:

$$m(t) = K_p e(t)$$

Obteniendo la función de transferencia aplicando la transformada de Laplace.

$$\frac{M(s)}{E(s)} = K_p$$

2.5.2 Control integral.

Al implementar un control integral, la señal de control se modifica a una velocidad proporcional a la señal de error. En otras palabras, si la señal de error es muy grande, entonces la señal de control se incrementa con gran rapidez, pero si la señal de error es pequeña, entonces la señal de control incrementa con lentitud. Donde este se representa como:

$$m(t) = K_i \int e(t) dt$$

Donde K_i es la constante de ganancia del integrador. Obteniendo la función de transferencia tenemos:

$$\frac{M(s)}{E(s)} = K_i/s = \frac{K_i}{s} = K_i \frac{1}{s}$$

Donde $1/s$ es el operador para la integración. Donde podemos notar que si el error tiende a cero, la salida del controlador permanecería constante. Por esta característica los controladores integrales permiten ser utilizados en algún tipo de carga constante en el sistema. Incluso cuando no exista ningún error, el controlador, permanecerá conservado una señal de salida, para neutralizar.

2.5.3 Control proporcional más integral (P+I).

La mayoría de las veces es necesario hacer alguna combinación de control. Esto se debe a que un controlador proporcional es incapaz de neutralizar una carga en el sistema sin ningún error. Pero también un controlador integral

puede proporcionar un error cero, pero este suele suministrar una respuesta lenta en el sistema. Para resolver este tipo de problemas se utiliza un controlador P + I. se representa como:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int e(t) dt , \quad K_i = \frac{K_p}{T_i}$$

Donde T_i regula la ganancia del integrador y K_p ajusta el integrador y la ganancia proporcional. Por lo que su función de transferencia se presenta como:

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right)$$

2.5.4 Control proporcional más derivativo (P+D).

El trabajo del control derivativo es proporcionar una señal de control proporcional a la velocidad de cambio de la señal de error. Por lo que la señal de salida presenta un cambio al menos que exista un cambio en el error. En otras palabras acción derivativa es mantener el error al mínimo corrigiéndolo proporcionalmente con la misma velocidad que se produce; de esta manera evita que el error se incremente. El controlador P+D se representa como:

$$m(t) = K_p e(t) + K_p T_d \frac{de(t)}{dt} , \quad K_d = K_p T_d$$

La función de transferencia se presenta como:

$$\frac{M(s)}{E(s)} = K_p (1 + T_d s)$$

La acción del control derivativo es anticipar cambios en el error y proporcionar una respuesta más amortiguada en los cambios.

2.5.5 Control Proporcional más integral más derivativo (P+I+D).

La combinación de los tres controles tenemos un controlador PID que se representa como:

$$m(t) = K_p e(t) + \frac{K_p}{T_i} \int e(t) dt + K_p T_d \frac{de(t)}{dt}$$

La función de transferencia es:

$$\frac{M(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} + T_d s \right)$$

El controlador PID es el control más completo que existe y el más utilizado, donde proporciona una respuesta rápida, un control de buena estabilidad en el sistema y un bajo error de régimen en el estado estacionario.

3. Metodología

En este capítulo se muestran las partes de un control de movimiento tanto en su forma de hardware como software, así como las descripciones de cada componente que integra el sistema.

3.1 Robot industrial Romat 56.

Los robots industriales son una herramienta esencial en una empresa ya que estos equipos pueden ser flexibles a realizar diversos trabajos y que además los realizan con una precisión significativa. Este último factor puede depender del tipo de tecnología que se usa, puede ser un controlador de movimiento de alta precisión pero si este no cuenta con dispositivos para realizar trabajos de esta magnitud de nada servirá contar con tecnología avanzada, lo importante de este proyecto es combinar todos los dispositivos adecuados para lograr una eficiencia significativa a bajo costo.

El robot que se trabajo es el modelo Romat 56 que es producido por CLOOS. Este robot cuenta con 6 grados de libertad donde sus uniones son rotativas. Este robot también es conocido como PUMA (Programmable Universal Manipulator For Assembly) por la semejanza en su estructura. Este robot usa motores de corriente directa que son acoplados a un juego de engranes que es comúnmente llamada transmisión mecánica. Los motores se alimentan con 24 Volts y tienen instalados encoders ópticos en la flecha con resolución de 1024 pulsos por revolución, los motores también cuentan con un freno mecánico que es controlado con una señal digital de 24 volts.

El robot Romat 56 tiene un peso aproximadamente de 110 kg con una altura de 2.1 metros extendido verticalmente, los motores que usan son controlados por servo amplificadores que posteriormente se hablara más a detalle,

para el control de cada uno de los motores se cuenta con un controlador de movimiento del modelo DMC – 18x6 de la serie Accelera producido por Galil Motion Control.



Figura 15. Robot industrial (Romat 56).

3.2 Sistema de control de lazo cerrado para el robot industrial Romat 56.

El control de movimiento del robot Romat 56 es un sistema de lazo cerrado en la figura 16 se muestra el diagrama a bloques del sistema propuesto, los cuales incluyen el control de un motor de DC y la forma de comunicación entre cada bloque.

El primer bloque muestra la comunicación necesaria entre la computadora y el controlador de movimiento para conocer la posición, las variables son medidas con el sensor de posición que se llama retroalimentación de posición C . El controlador de movimiento con la referencia de velocidad envía un voltaje V al bloque amplificador por lo que este aplica una corriente I que es directamente proporcional al voltaje.

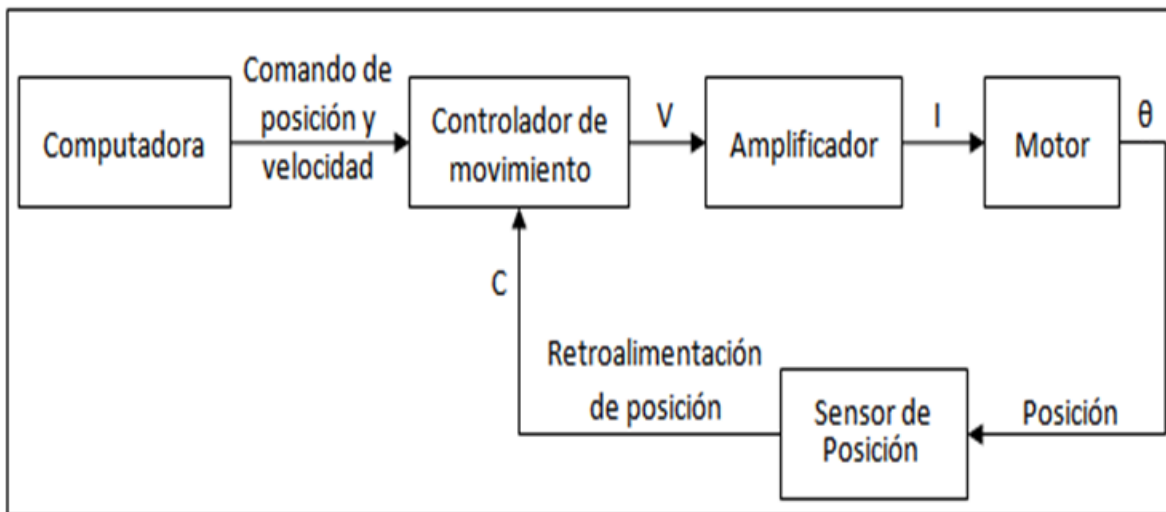


Figura 16. Estructura de un sistema de control de lazo cerrado (Jacob Tal, 1989).

3.3 Sistema de control en diagrama de bloques.

La figura 17 es un sistema de control representado por un diagrama de bloques, que nos muestra como es el control de posición, control de velocidad, control de corriente, Amplificador, Motor y el mecanismo del sistema. Si se divide en los bloques que anteriormente se describen quedaría de la siguiente manera:

- Controlador de movimiento: cuenta con el control de posición y control de velocidad.
- Amplificador: cuenta con el control de corriente y el amplificador de potencia.
- Motor: este último tiene integrada los encoders que retroalimenta el primer bloque y su función es proporcionar el trabajo deseado por el controlador de movimiento.

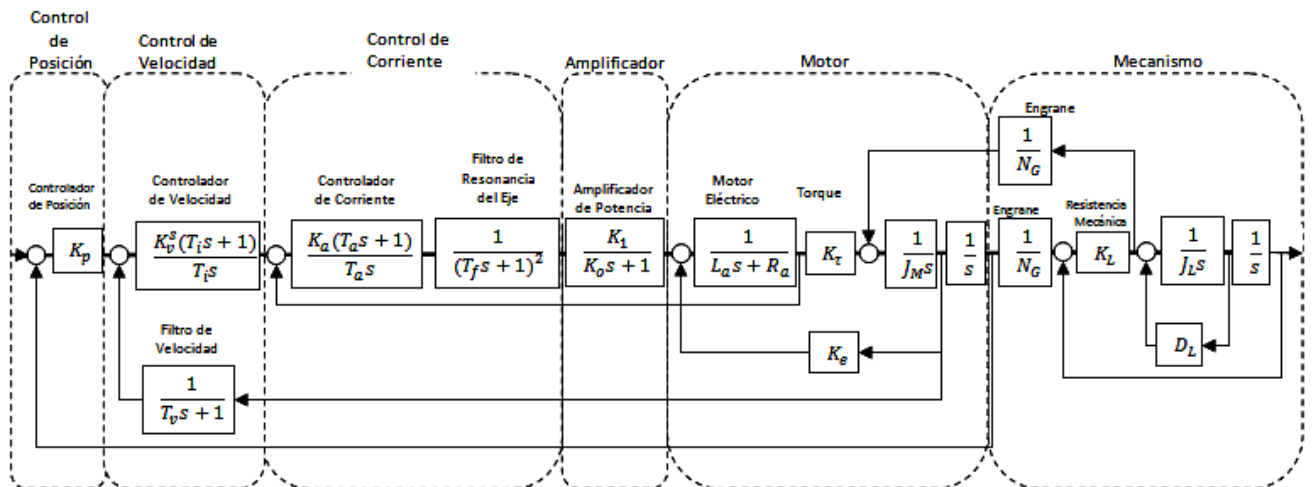


Figura 17. Sistema de control de una articulación (Masatoshi Nakamura, 1998).

3.4 Sistema de control del robot Romat 56.

Como se mencionó anteriormente, el robot Romat 56 cuenta con 6 grados de libertad por lo que su control se muestra en la figura 18, donde se cuenta con 6 ejes.

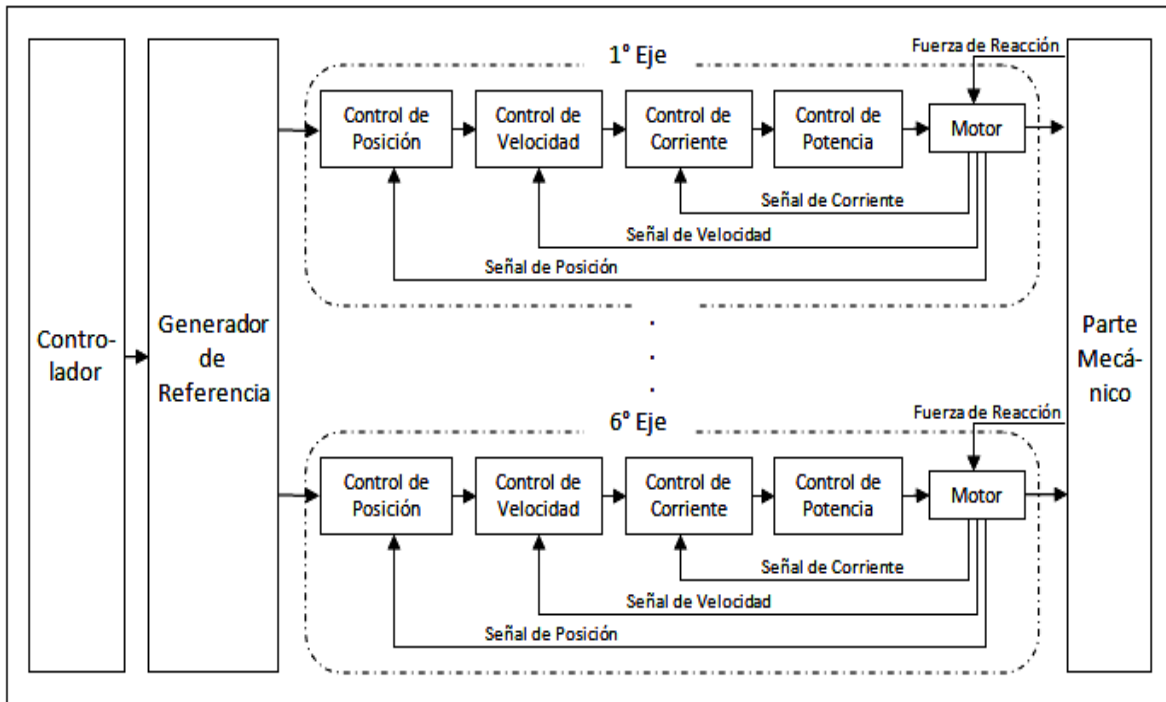


Figura 18. Control del robot Romat 56 (Masatoshi Nakamura, 1998).

En la figura 19 se muestran las conexiones de los dispositivos que componen el control del robot Romat 56.

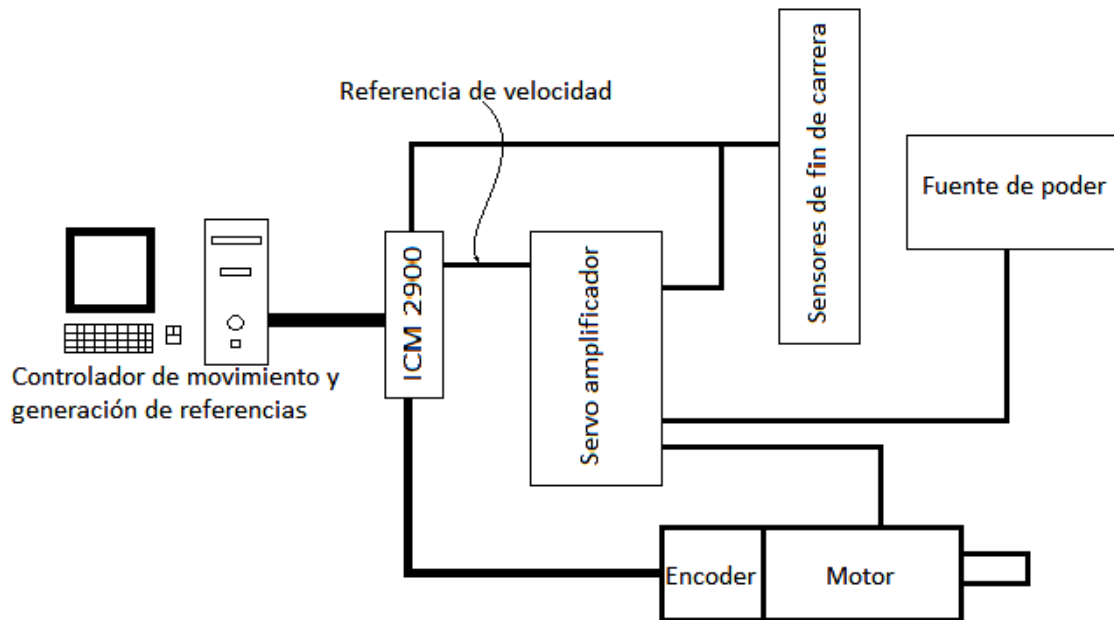


Figura 19. Diagrama de conexiones de dispositivos para el Romat 56.

La computadora

Esto ayuda a que el usuario tenga una interacción con el robot y poder programar una serie de trayectorias que el robot debe seguir con la ayuda de una cinemática ya sea directa ó inversa, esto genera una serie de comandos de referencias que pueden ser de posición, velocidad los parámetros de configuración tales como la aceleración, desaceleración de cada articulación.

Controlador de movimiento

El controlador de movimiento es la parte inteligente del sistema ya que esta se encarga de todo el control de movimiento. El controlador que se usa consiste de un controlador de movimiento DMC-1866 (la tarjeta tiene una capacidad de 6 ejes) son tarjetas de muy alto rendimiento con la que cuenta Galil Motion Control. Este controlador de movimiento cuenta con bus PCI para la

comunicación, es de la última generación con la que cuenta Galil: la serie Accelera, acepta señales de encoder de hasta 22 MHz, ofrece tasas de actualización de servos de hasta 32 kHz, y los procesos de órdenes son tan rápido que llegan a los 40 microsegundos, esto es 10 veces más rápido que las generaciones anteriores (Figura 20).

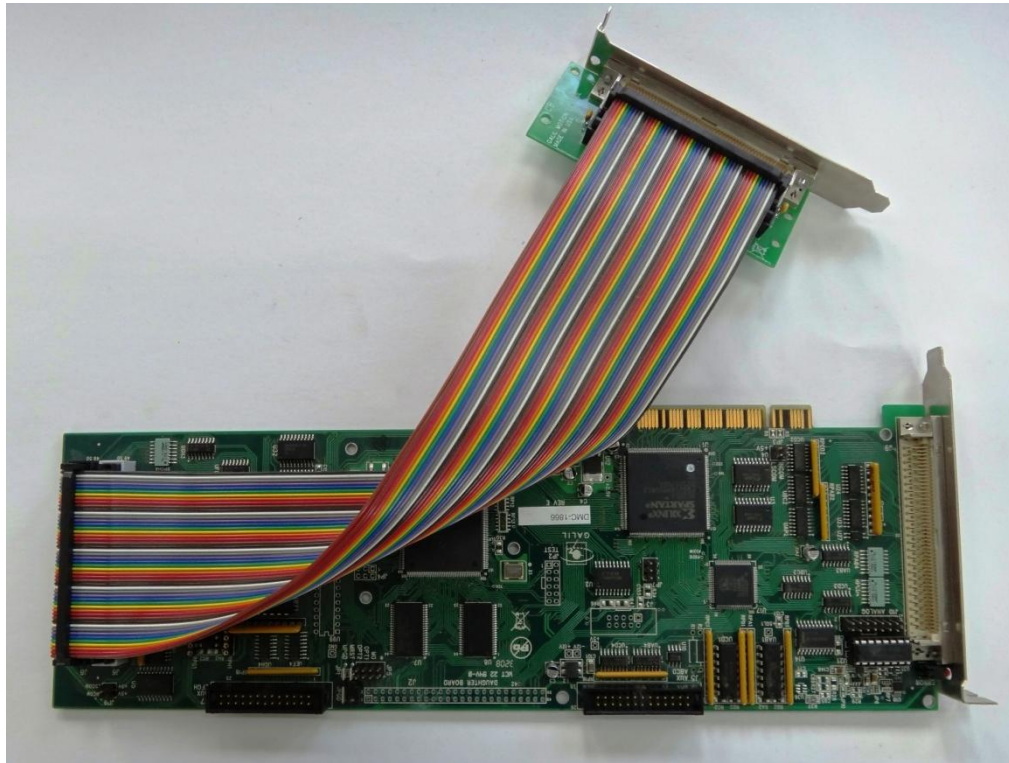


Figura 20. Controlador de movimiento DMC-1866 de Galil Motion Control.

El controlador de movimiento se comunica con la computadora para recibir los comandos de movimiento. Los requerimientos de comando son especificados por la computadora en términos de pocos parámetros, los parámetros pueden ser incluidos como por ejemplo, La posición, la velocidad, la aceleración y la desaceleración, esto se sigue hasta conseguir el perfil de movimiento y se construye una referencia R (figura 21).

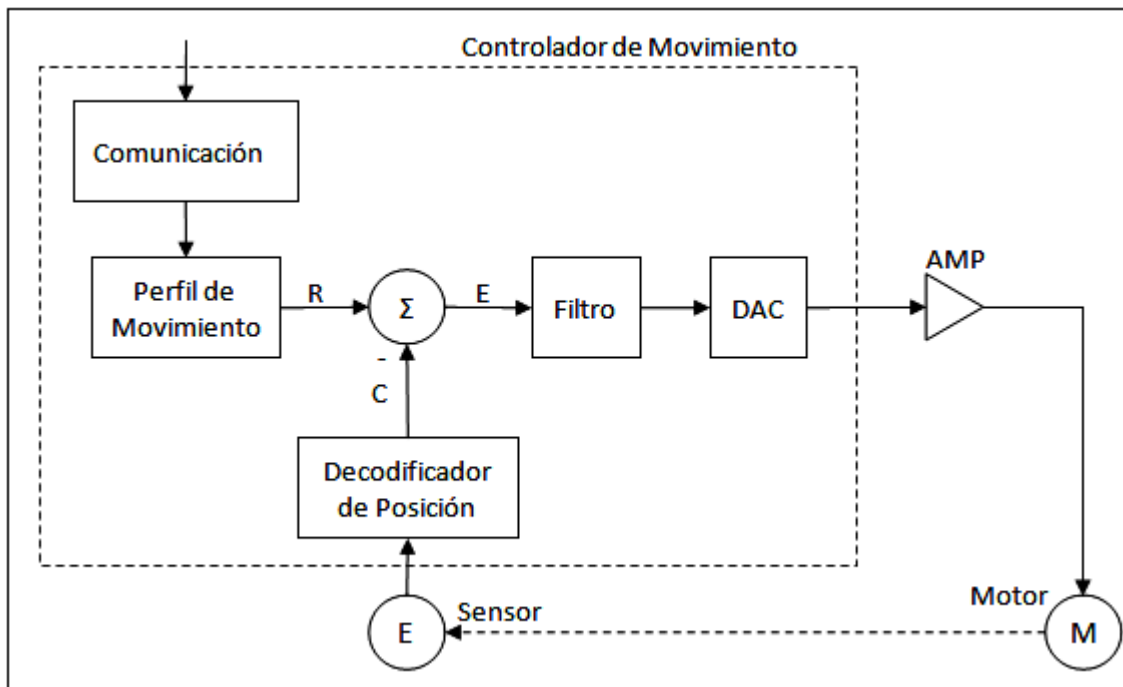


Figura 21. Función elemental de un controlador de movimiento (Jacob Tal, 1989).

El controlador de movimiento se puede dividir en dos partes: la primera en generar una señal de referencia R , y la segunda tarea es seguir la función de la referencia.

El control comienza con la decodificación de la señal de posición, C , y se cierra el lazo de control. La posición del motor C es determinada sobre la señal de retroalimentación (se consigue con un encoder incremental) y se compara con la posición de referencia, R . En donde E es la diferencia entre R y C y es conocida como el error de posición. El objetivo del controlador es reducir error de posición a un valor mínimo sin causar una oscilación en el sistema. El controlador incluye un filtro de estabilidad y se envía a un convertidor digital a analógico (DAC) para después enviar una señal de ± 10 V a un servo amplificador.

Amplificador

Los servo amplificadores que se utiliza reciben un comando de señal que típicamente son una señal analógica de un rango de +/- 10 V donde amplifican el nivel de corriente requerida. Sin embargo los amplificadores se pueden configurar en modo de corriente ó en modo de velocidad. Para el modo de velocidad son preferidos cuando la retroalimentación es de velocidad, por lo tanto, los amplificadores son típicamente configurados en modo de corriente (figura 23).



Figura 23. Servo amplificador.

En el modo de corriente, los amplificadores producen una corriente que son directamente proporcional al voltaje de entrada, estos son obtenidos por un lazo de retroalimentación de corriente cual monitorea la corriente y asegura que la señal de comando si sea proporcional (figura 24).

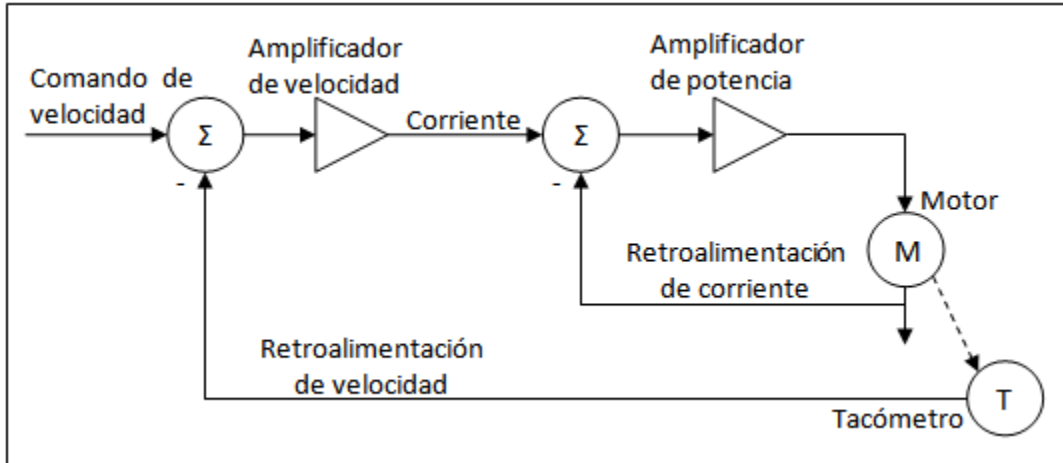


Figura 24. Elementos de un amplificador en modo de velocidad (Jacob Tal, 1989).

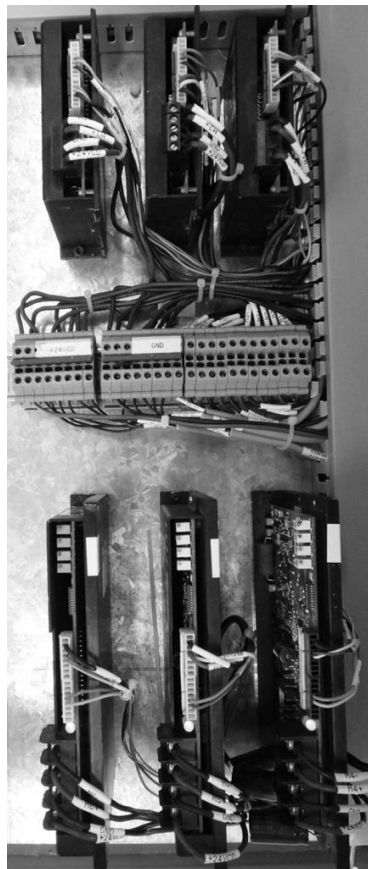


Figura 25. Servo amplificadores instalados para el robot Romat 56.

En la figura anterior se muestra la instalación de los 6 servos amplificadores para los 6 motores del robot.

Motor DC

Los servo motores (figura 26) utilizan un voltaje de DC, estos servo motores convierten la energía eléctrica en energía mecánica, o dicho de otra manera, convierte la corriente en torque rotacional, los parámetros de los motores son la constante de torque, K_t , resistencia de la armadura, R_a , el momento de inercia, J_m , y el nivel máximo de torque. La constante de torque es expresada en unidades de $N \cdot m/A$, esto indica la cantidad de torque que puede generar el motor por una unidad de corriente. Por ejemplo, un motor de DC con un torque constante de $0.1 N \cdot m/A$ al convertir la corriente en 2 Amperes el toque nos dará de $0.2 N \cdot m$. La resistencia de la armadura, es la resistencia total del embobinado de la armadura y las escobillas, estos son expresados en ohms. El momento de inercia, J_m es la suma del momento de inercia del rotor y es expresado en $kg \cdot m^2$.

Un motor también se caracteriza por su nivel de torque que éste puede producir. La capacidad del torque del motor se dividen en dos parámetros: valores de pico y continuos. A torque continuo es el torque que el motor puede producir continuamente, frecuentemente en alguna velocidad y sin calentamiento. El torque pico es el torque máximo que puede generar en un corto periodo de tiempo sin causar daños mecánicos ó desmagnetización.

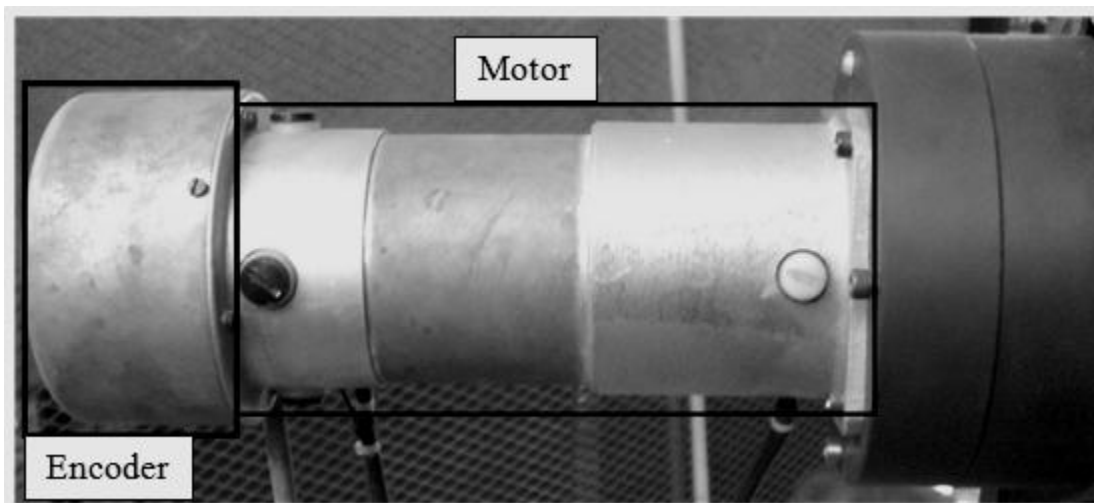


Figura 26. Motor DC.

Encoder

El encoder es esencial ya que con este se logra que el sistema sea de lazo cerrado y pueda tener una retroalimentación, logrando tener el control de la posición y la velocidad en cada motor. Los motores del Romat 56 cuenta con encoders de tipo incremental que generan pulsos, estos encoders generan seis señales que son: canal +A y -A, canal +B y -B y por ultimo +índex y -índex. El canal A y B están desfasados a 90° y con estos se determina la velocidad, posición y la dirección del motor, la resolución incrementa al dividir en cuatro la señal, este proceso es llamado contador de cuadratura. Por último el índice genera un pulso por cada revolución del motor.

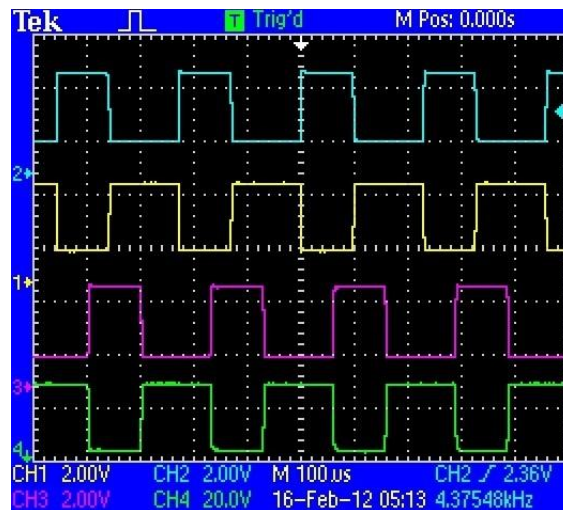


Figura 27. Señal de encoder; señal 1 (canal -A), señal 2 (canal +A), señal 3 (canal +B) y señal 4 (canal -B).

Sensor de fin de carrera

Estos sensores se ubican cerca de cada articulación y se encargaran de producir una señal de cuando el eslabón este llegando al límite del recorrido (figura 28) estos son muy importantes por razones de seguridad. Cada articulación posee con un campo de trabajo que está cerca del límite del desplazamiento lo que produce un cambio de señal de voltaje, que el sensor envía al controlador.

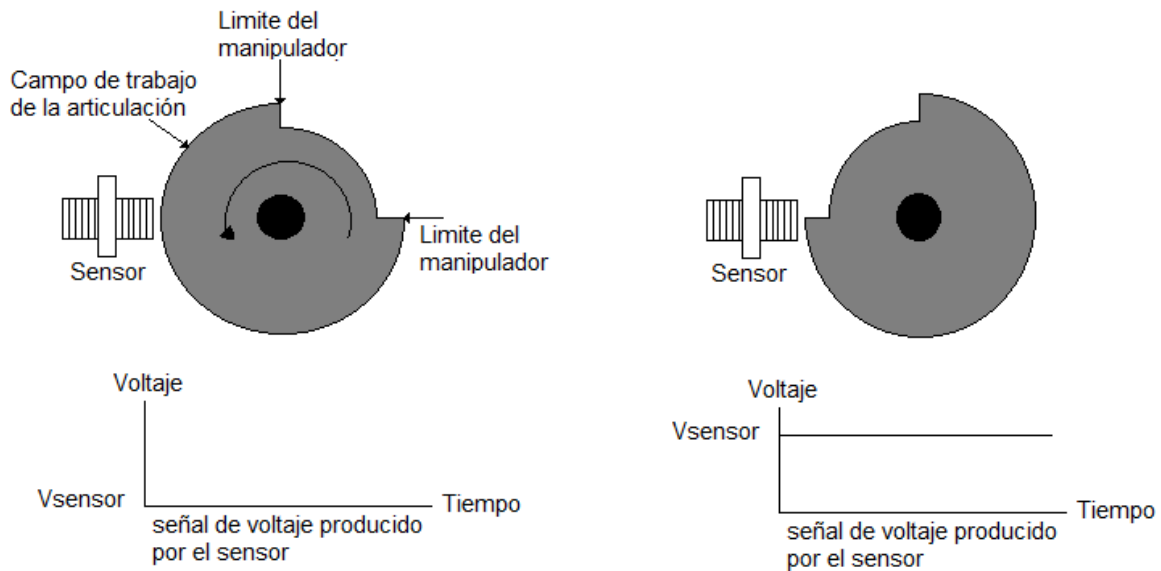


Figura 28. Función de sensor de fin de carrera.

En la figura 29 se muestra las tarjetas que se diseñaron para el control de los sensores de fin de carrera, estas tarjetas suministran energía eléctrica a los sensores capacitivos que se alimentan con 24 VCD y se recibe señales de esta magnitud. También se nota que contamos con dos tarjetas aparentemente similares la diferencia es que su función es inverso, esto ocurre porque se cuenta con 2 tipos diferentes de servo amplificadores. El primer tipo es de la marca Copley Controls Corp este tipo de amplificadores se activan en estado alto, es decir, que para actuar necesitan una señal de +5 VCD. El segundo tipo es de la marca Advanced Motion Controls estos se activan en bajo, es decir, que actúan cuando reciben una señal de 0 VCD.

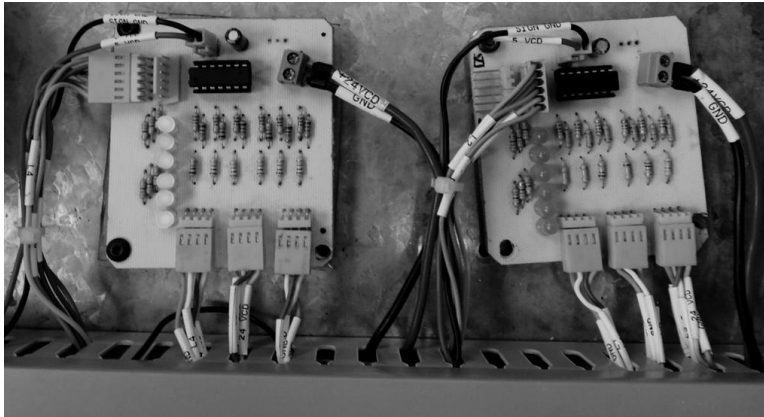


Figura 29. Tarjetas de control de los sensores de fin de carrera.

Fuente de energía.

Este proporciona la energía necesaria para el funcionamiento del control de movimiento del robot, la gran mayoría de los casos este se diseña en base de la cantidad de energía requerida de todo el sistema. En este trabajo se cuenta con una fuente de energía que puede llegar a suministrar voltaje de 0 – 250 VDC y corriente de 0 – 40 Amperes.

3.5 Interfaz gráfica de usuarios

La interfaz gráfica de usuarios o GUI (Graphics User Interface) es la parte más importante de este proyecto porque a través de ella es posible la interacción del usuario con el robot sin la necesidad de conocer el fondo de la programación, de la instalación eléctrica o el tipo de dispositivos que este tiene integrados, simplemente sigue los pasos para la operación del robot, por ello se diseñó una GUI que sea fácil de manejar y que no sea necesario contar con fórmulas matemáticas para el trazo de trayectorias, de lo contrario se piensa que esto llega a desgastar al personal y la operación de estos equipos solo puede ser realizada por personal calificado que cuente con un alto conocimiento en el área de la robótica.

La GUI está diseñada en LabView 10.0 que es un lenguaje gráfico, se decidió construir la GUI en este software por los beneficios que este tiene, como lo es su fácil manejo de programación y también porque este software está siendo muy bien aceptado en las industrias por lo que nuestra intención es dar más herramientas a las empresas que adquieran este trabajo y si en un futuro este requiere de una actualización la empresa tiene la posibilidad de hacerlo.

LabView 10.0 está desarrollado por National Instruments donde una de sus ventajas es poder acoplar otros dispositivos externos a la robótica, ya que esta compañía cuenta con cientos de equipos para una infinidad de aplicaciones que pueden ser interactuados desde LabView. Por lo que nos enfocamos en este programa y en el controlador de movimiento DMC-1866 que es desarrollado por Galil Motion Control donde fue necesario buscar librerías que permitieran abrir comunicación entre estas dos partes. Las librerías DMC fueron adquiridos en la página de internet de la compañía Galil Motion Control, así como algunos manuales para el manejo de estas librerías.

En la figura 30 se muestra la GUI donde daremos una explicación de ellas y los pasos a seguir para el manejo de este. En la GUI se cuenta con pestañas que son el Inicio, Manual, Editor P2P, Archivos y Configurar.

Comenzaremos con la figura 30 que es el Inicio donde se muestra los datos de la tesis, también es muy importante ya que con el botón de encender hace que establezca una comunicación con el control de movimiento o en su caso apagar la comunicación, como también se cuenta con un Reset que reinicia el sistema en caso de algún conflicto, el botón de Stop detiene la ejecución del programa de LabView.

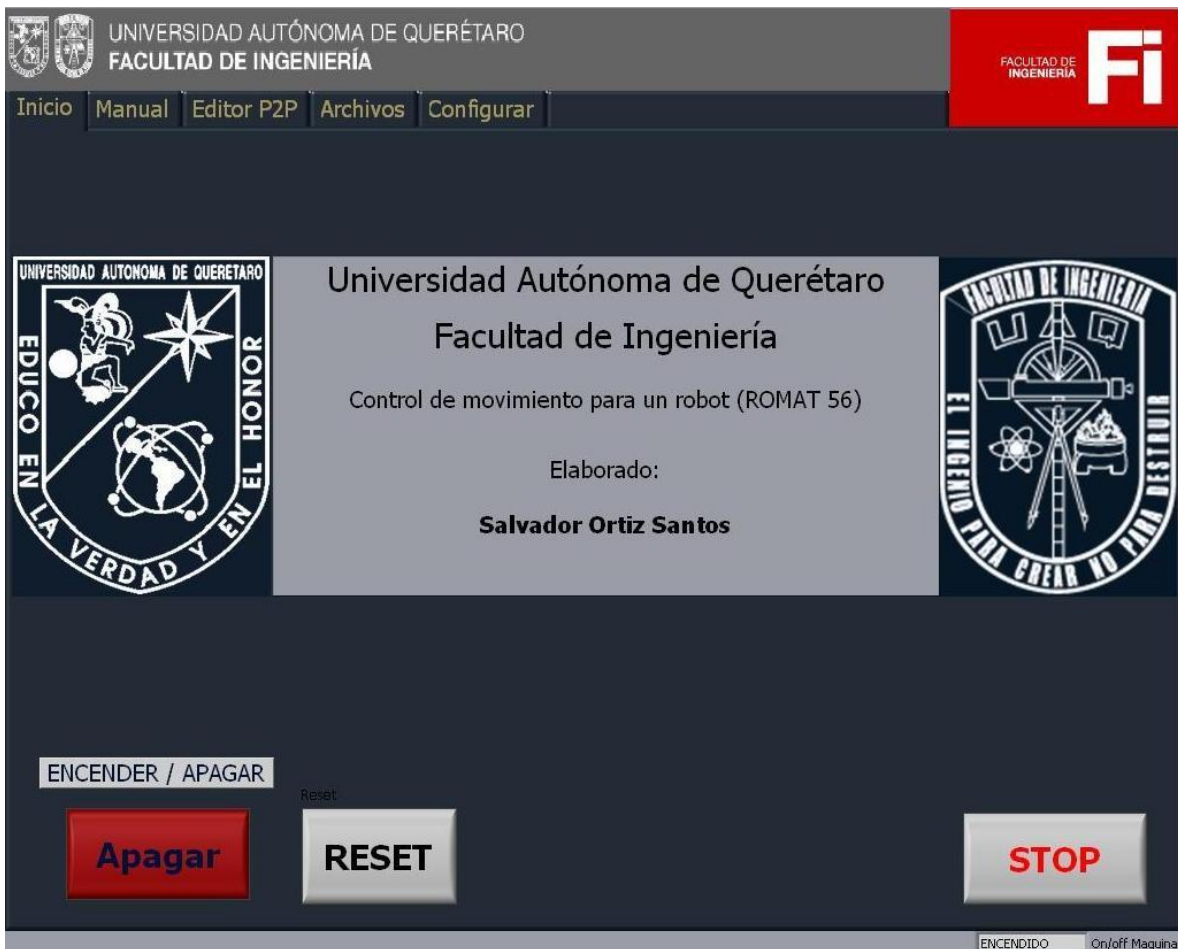


Figura 30. Página de inicio.

La parte manual que se muestra en la figura 31 es una parte donde el usuario tiene la posibilidad de hacer movimientos con cada una de las articulaciones y el cambio de velocidad de cada uno de los motores, es importante contar con esta parte ya que se pueden realizar trabajos sin la necesidad de una programación.

Por otro lado la parte manual sirve para establecer la posición y la orientación del robot en tiempo real. Para el movimiento de cada motor solo se debe de oprimir el boton de signo menos que significa movimiento en contra de las manecillas del reloj o en su caso el de signo más que es un movimiento en el sentido de las manecillas del reloj.



Figura 31. Página de control manual.

Al realizar una investigación de cómo es la programación de estos robot industriales nos encontramos con una variedad de lenguajes, donde para su manejo es necesario conocer los comandos de programación y además conocer muchas variables del robot, tales como su estructura, el campo de trabajo, los parámetros de Denavit-Hartenberg entre muchos otros. Uno de los lenguajes que se trabajó a detalle fue el VAL (Variable Assembly Language) que es producido por Unimation Inc.

Al notar que los lenguajes de programación comerciales requieren de un conocimiento amplio en robótica, se trabajó con una GUI la cual su manejo fue muy gráfico y no de comandos de lenguaje de programación.

La programación de trayectorias del robot industrial Romat 56 está diseñada para dirigirse a un punto y guardar en una cadena de String por lo que la parte de programación se llama Editor P2P (punto a punto), que consiste en grabar puntos comenzando desde un punto de referencia.

La figura 32 es una pestaña de la GUI llamada Editor P2P sus movimientos son muy similares al de la forma manual excepto por la velocidad que está determinado solo de un rango de 0 a 100% que es establecido para todos los motores, el controlador de movimiento nos muestra una posición y orientación absoluta es decir la medición tiene una referencia que es el origen, pero en nuestros cálculos para guardar la posición y orientación es necesario contar con mediciones de posición y orientación relativa es decir que esta medición tenga como referencia el último punto. Los botones que se presentan en el Editor P2P son:

- Stop: Este botón permite que el usuario detenga toda acción que el robot este efectuando.
- Reiniciar valores: Es importante cuando se inicia un nuevo programa donde su trabajo es dar valores iniciales a todas las variables, esto hace que no se cuente con errores.

- Guardar punto: Cuando se haya encontrado el punto deseado en el espacio tridimensional y con la orientación deseada se guardan todas las coordenadas para posteriormente buscar un nuevo punto y orientación.
- Guardar archivo: Una vez terminado de guardar los puntos que se desea que recorra el robot, lo que hace es guardar estos puntos en un archivo que posteriormente se ejecutara.



Figura 32. Página de control de punto a punto.

En la figura 33 se muestra la parte de Archivos, recordemos que en el Editor P2P ha programado trayectorias y que al finalizar se guardó en un archivo, el nombre y la dirección de este documento es generado por el usuario. Esto nos lleva que en Archivos se ejecute el documento, se tienen los siguientes botones:

- Abrir Archivo: nos permite que el usuario busque el archivo en la dirección que anteriormente fue establecido, cuando se haya cargado el archivo nos muestra en Archivos a Ejecutar La dirección y el nombre del documento que se va a ejecutar.
- Ejecutar Archivo: ejecuta el archivo y recorre todos los puntos guardados.
- Cambiar velocidad: cambia la velocidad según la selección que este tenga que va de 0 a 100%.



Figura 33. Página de archivos.

Las empresas siempre están en competencia y muchas otras tienen una visión de avance tecnológico, en consecuencia toda industria busca mejorar la calidad de su producto y la eficiencia de la industria. Lo cual conlleva en aplicar tecnología que cuente con una buena precisión, esta variable puede confundir porque la precisión es una variable en la que se puede lograr de varias maneras pero en este trabajo se busca que esta precisión sea obtenida bajo una respuesta de control de forma trapezoidal.

En la figura 34 se trata de la configuración de cada motor, la configuración se hace a través del control PID, al configurar el PID nos ayuda que el robot tenga una respuesta rápida y sin sobrepaso y una buena precisión, para lograr esto se cambian las ganancias KP (ganancia proporcional), KI (ganancia integral) y KD (ganancia derivativa), para contar con una buena respuesta también contamos con dos variables más que nos ayudan a que nuestro control tenga una respuesta rápida o lenta dependiendo de la aceleración y la desaceleración, estas dos últimas nos pueden modificar los tiempos de respuesta es importante que este cuente con un balance adecuado de lo contrario la respuesta de control puede cambiar y afectar a otras variables tales como la precisión.

Una vez ajustados los parámetros del control PID, aceleración y desaceleración contamos con tres botones:

- Abrir archivo: Nos permite abrir un archivo previamente guardado que nos configura los parámetros de control.
- Guardar archivo: Cuando hayamos terminado de establecer la configuración a cada uno de los motores o solo de los motores que se necesitan se tiene la posibilidad de guardar en un archivo la cual su dirección y el nombre es seleccionado por el usuario que posteriormente solo se ejecute este archivo y aplicar la configuración.
- Aplicar configuración: lo que sucede en este paso es que la configuración del control es enviada al controlador de movimiento para

su ejecución física de los motores, los valores ejecutados pueden venir de la selección de las variables o desde un archivo.

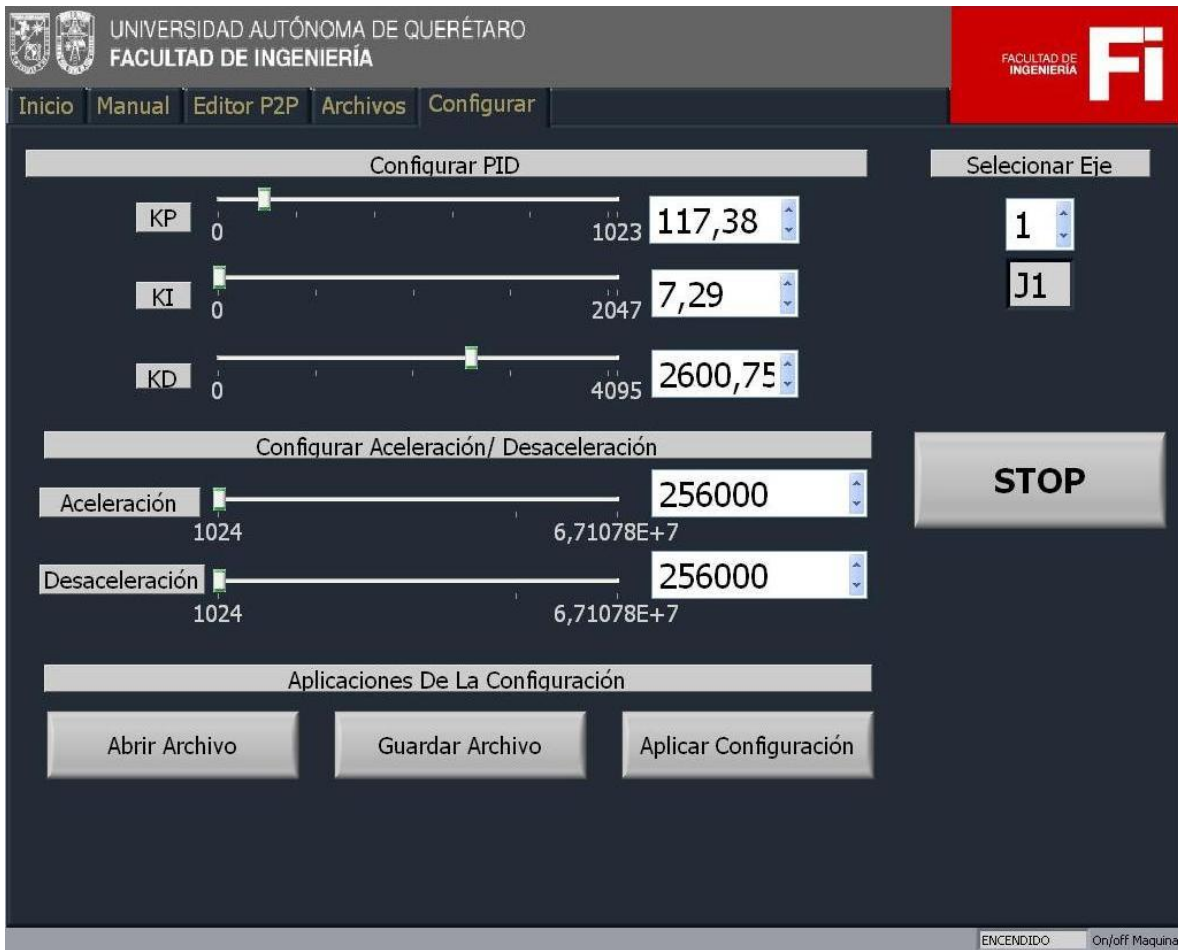


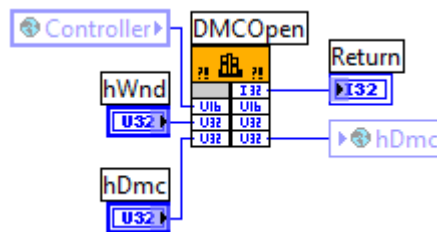
Figura 34. Página de configuración.

3.6 Librerías que permiten establecer la comunicación entre Labview 10.0 y el controlador de movimiento.

En el trabajo se presenta tres librerías que fueron adquiridos por Galil Motion Control que son: OPEN, COMMAND y CLOSE.

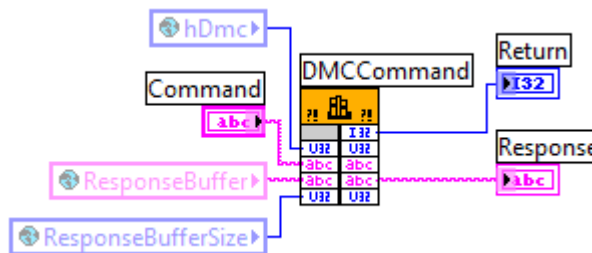
Open

Esta librería abre la comunicación entre LabView y la tarjeta de control de movimiento.



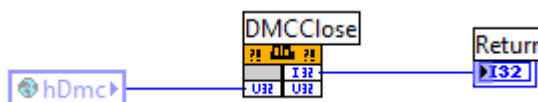
Command

La librería command permite enviar comandos a la tarjeta de control de movimiento, con lenguaje específico de de esta tarjeta.



Close

Esta librería permite cerrar la comunicación entre Labview 10.0 y la tarjeta de control de movimiento.



4. Resultados y conclusiones.

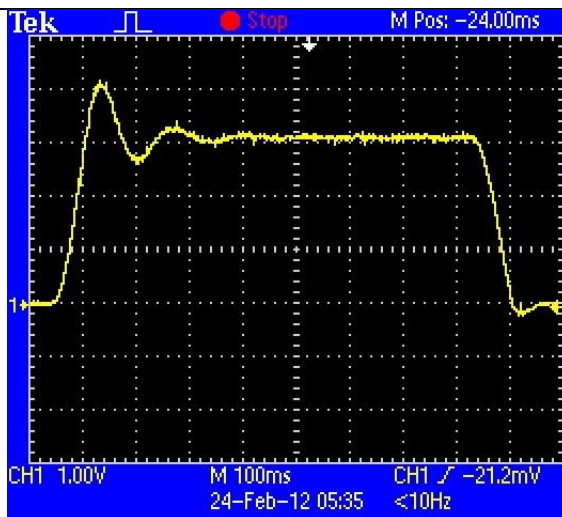
4.1 Resultados

Pruebas para conocer la respuesta del controlador.

Esta prueba se le realizó a uno de los motores que componen el Romat 56 ya que este robot usa motores con las mismas características o muy similares por lo que se supuso que las respuestas deben ser muy similares. Esta prueba nos ayuda a descubrir cuál es la precisión del robot.

En todas las pruebas se usó una velocidad constante de 10000 cuentas/segundo. Por lo cual se realizaron diferentes pruebas con diferentes parámetros de control para observar cuál de estas es la más adecuada para nuestro sistema.

En esta prueba se requería llegar a una posición de 8000 cuentas y con un control PID que esta por defecto por el controlador de movimiento teniendo como valores: $K_P = 6.0$, $K_I = 0.0$ y $K_D = 64.0$.



Parámetros y resultados:

Aceleración = 128,000

Desaceleración = 128,000

Punto inicio: -1

Punto final: 8006

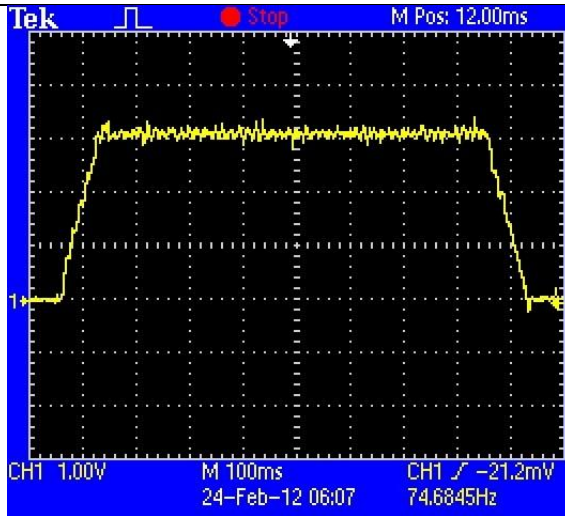
Se muestra un experimento para mostrar el sobrepaso y el tiempo de subida.

	<p>Parámetros y resultados:</p> <p>Aceleración = 256,000 Desaceleración = 256,000 Punto inicio: -1 Punto final: 7996</p> <p>Al incrementar la aceleración el sobrepaso es más grande.</p>
	<p>Parámetros y resultado:</p> <p>Aceleración = 512,000 Desaceleración = 512,000 Punto inicio: -1 Punto final: 7992</p> <p>Al pretender tener una respuesta a corto tiempo hace que se presente un sobrepaso más grande.</p>

Tabla 4.1: Prueba 1 respuesta del sistema.

En la prueba anterior podemos concluir que la respuesta no es la óptima para llegar a la velocidad deseado y que además se cuenta con error de posición.

Esta prueba es similar con la anterior se requiere de una posición de 8000 cuentas, pero la diferencia con la anterior esta prueba maneja un controlador PID adecuada para este motor con los siguientes valores: $K_P = 84.63$, $K_I = 9.09$ y $K_D = 1178.88$.



Parámetros y resultado:

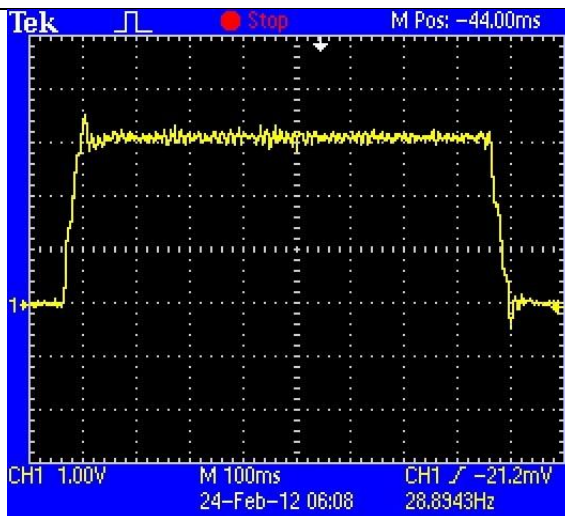
Aceleración = 128,000

Desaceleración = 128,000

Punto inicio: 0

Punto final: 8000

Muestra un trapezoide ideal para el sistema.



Parámetros y resultado:

Aceleración = 256,000

Desaceleración = 256,000

Punto inicio: 0

Punto final: 8000

Muestra un trapezoide pero este cuenta con un sobrepaso muy pequeño.

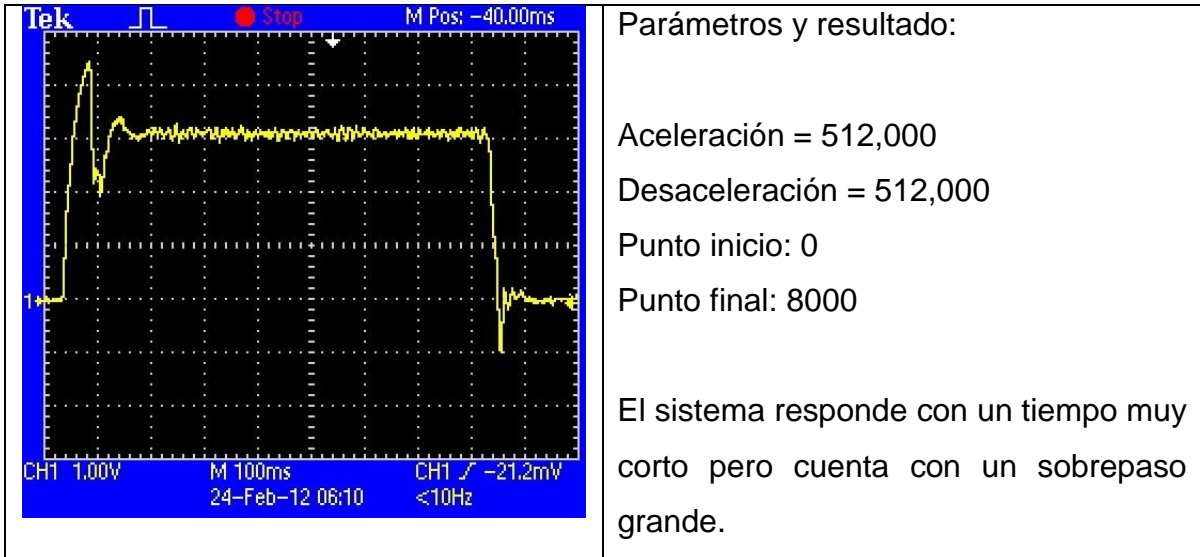
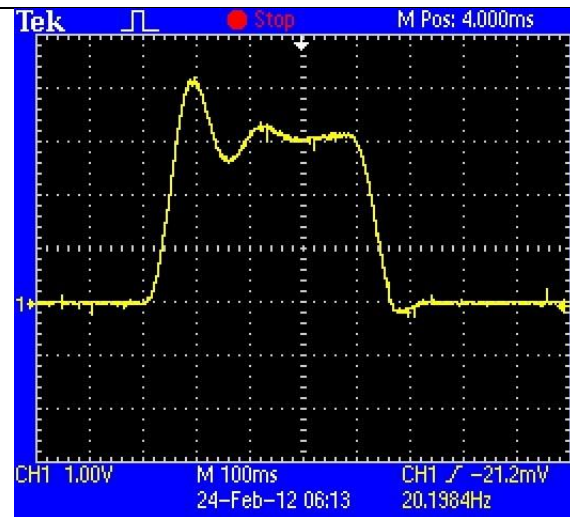


Tabla 4.2: Prueba 2 respuesta del sistema con PID.

En esta prueba podemos concluir que al desear una inmediata respuesta este tiene un sobrepaso más grande, de tal manera que este puede sobre desbatar la pieza a maquinarse o chocar con lo que se esté soldando o pintando.

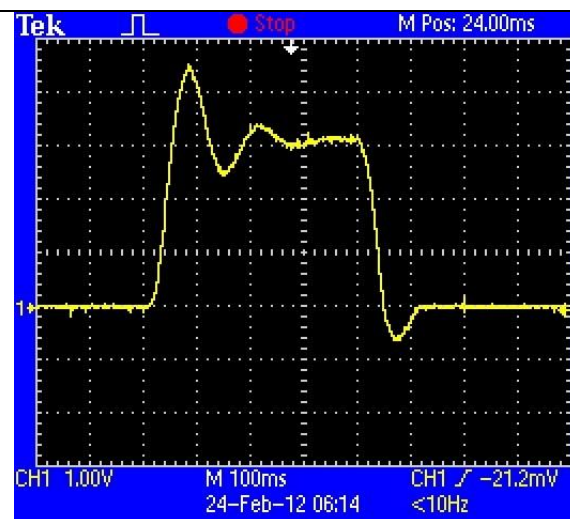
En esta prueba se requería llegar a una posición de 4000 cuentas y con un control PID que esta por defecto por el controlador de movimiento teniendo como valores: $K_P = 6.0$, $K_I = 0.0$ y $K_D = 64.0$.



Parámetros y resultados:

Aceleración = 128,000
Desaceleración = 128,000
Punto inicio: 0
Punto final: 4007

Se muestra un sobrepaso con una oscilación.



Parámetros y resultados:

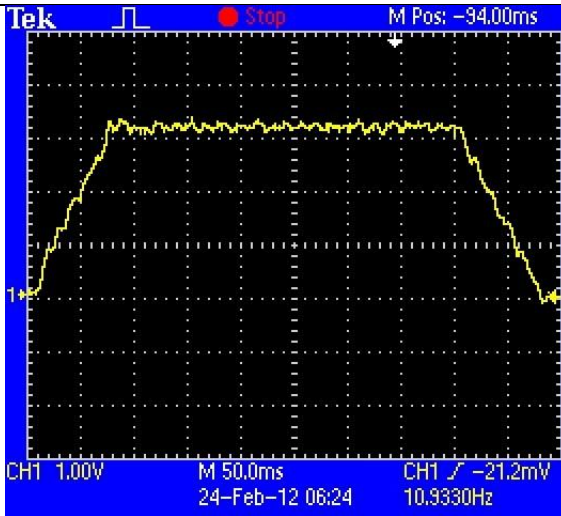
Aceleración = 256,000
Desaceleración = 256,000
Punto inicio: 0
Punto final: 3993

Se muestra un sobrepaso con una oscilación.

Tabla 4.3: Prueba 3 respuesta del sistema.

En esta prueba se puede concluir que al desear una posición corta con los mismos parámetros de las pruebas anteriores este cuenta con un sobrepaso más grande y también cuenta con error de posición.

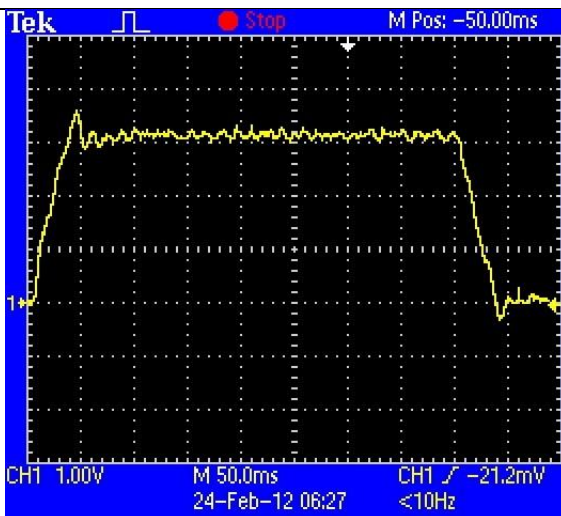
Esta prueba es similar con la anterior se requiere de una posición de 4000 cuentas, pero la diferencia con la anterior esta prueba maneja un controlador PID adecuada para este motor con los siguientes valores: $K_P = 87.75$, $K_I = 9.0$ y $K_D = 1166.38$.



Parámetros y resultado:

Aceleración = 128,000
Desaceleración = 128,000
Punto inicio: 0
Punto final: 4000

Muestra una respuesta trapezoidal ideal pero cuenta con una respuesta lenta.



Parámetros y resultado:

Aceleración = 256,000
Desaceleración = 256,000
Punto inicio: 0
Punto final: 8000

Muestra una respuesta más rápida que la anterior pero con un sobrepaso.

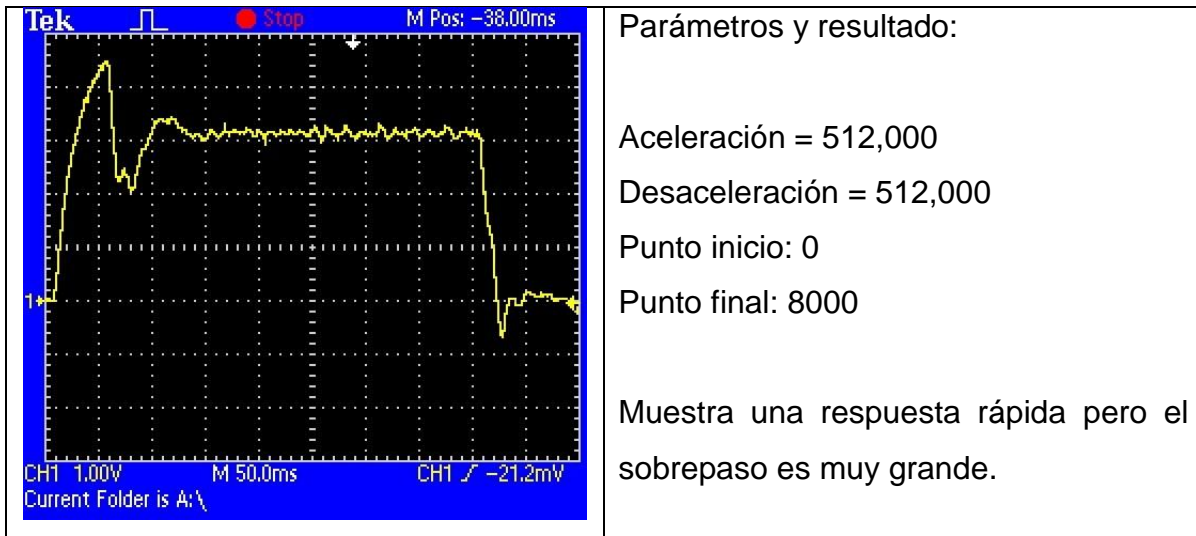


Tabla 4.4: Prueba 4 control del sistema con PID.

En conclusión los parámetros que son adecuados para este sistema son con el controlador PID y con valores de aceleración y desaceleración pequeñas para obtener una respuesta de forma **trapezoidal**. Una vez hecho este experimento se introdujeron los parámetros al programa de interfaz gráfica de usuario.

4.2 Conclusión

Al termino de este trabajo se cuenta con un gabinete que integra dispositivos que permiten el control del robot Romat 56, también se cuenta con la interfaz gráfica de usuario en Labview 10.0 donde se generan trayectorias y que a su vez permite la sintonía de los controladores PID, cabe mencionar que la interfaz gráfica de usuario es una aportación que se hace al robot Romat 56 del laboratorio de Mecatrónica donde dicho sistema al ser de bajo costo, será factible su implementación en los sistemas productivos de la región.

Por otra parte el robot Romat 56 cuenta con 6 grados de libertad de los cuales 5 grados de libertad están operando adecuadamente, el sexto grado de libertad donde se instala la herramienta a trabajar no está en operación debido a que no se cuenta con el encoder y por cuestiones de tiempo debo de obtener mi grado.

En las siguientes dos figuras nos muestran las gráficas de cómo el robot Romat 56 ejecuta la trayectoria, contamos con tres tipos de gráficas posición, velocidad y torque. Con estas figuras podemos concluir que el sistema funciona adecuadamente, donde en la primera columna nos muestra las graficas de posición real y posición deseada podemos percibir que en un comienzo la posición real y la posición deseada no son exactamente la misma pero en un determinado tiempo la posición real llega a ser la misma a la posición deseada como mencionamos en la parte de resultados esto pasa por tener una respuesta lenta en nuestro control y que nos ayuda a que nuestro sistema no cuente con sobrepasos y que el robot no realice movimientos extraños, en segunda columna de las figuras muestra la velocidad y en la tercera columna el torque.

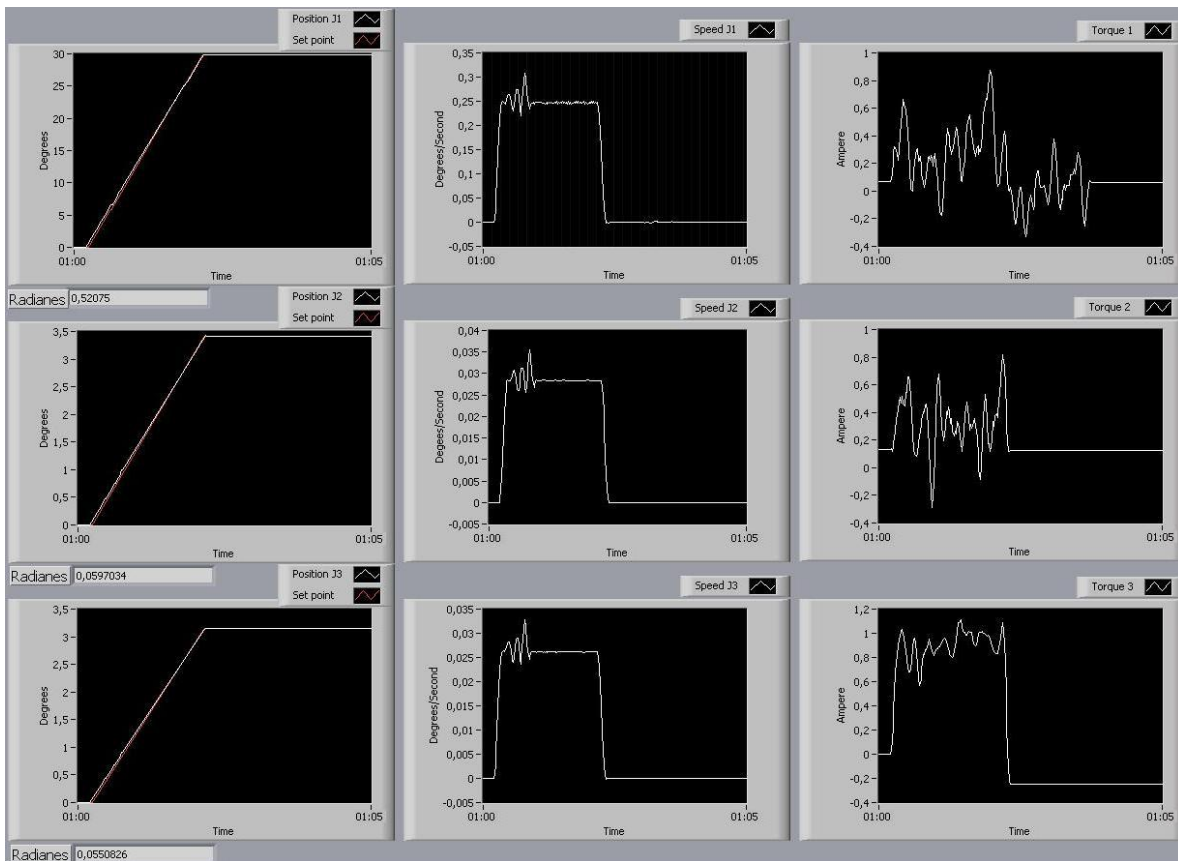


Figura 35. Gráfica de posición, velocidad y torque (prueba 1).

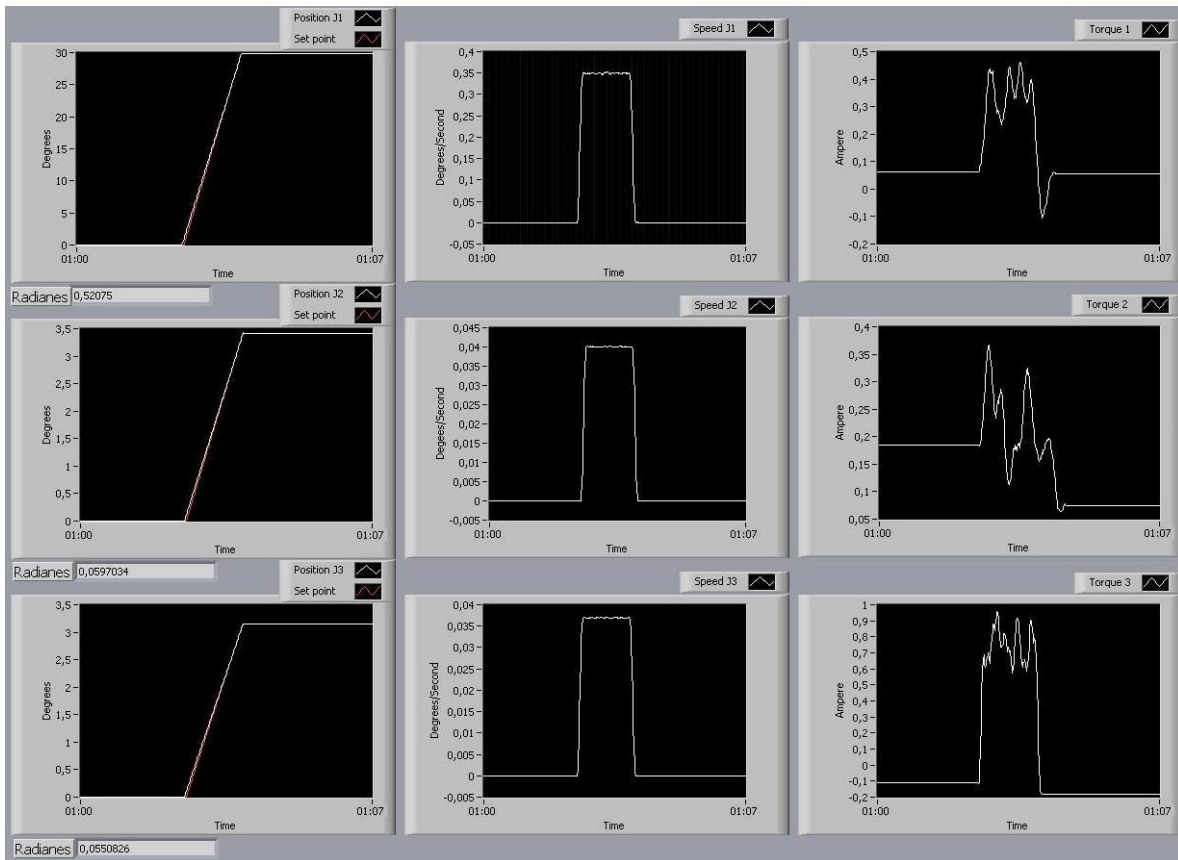


Figura 36. Gráfica de posición, velocidad y torque (prueba 2).

Bibliografía

Angeles Jorge, Fundamentals of Robotic Mechanical Systems. Springer, 1997.

Ángeles Álvarez Félix Jorge. *Análisis y Síntesis Cinemáticos de Sistemas Mecánicos*. Limusa, 1978.

Angulo Usategui y José Avile. Curso de Robótica. Madrid: Paraninfo, 1985.

Asada, H. y J.J.E. Slotine. Robot Analysis and Control. E.E.U.U: John Wiley & Sons, 1986.

Barrientos Antonio y Peñin Luis Felipe. Fundamentos de robótica (2ª edición). McGRAW - HILL, 2007.

B.M. Olds, Moskal y R.L. Miller. Assessment in engineering education: Evolution, approaches, and future collaborations. J. Eng. Educ. vol 94, no. 1. pp.13-25, 2004.

Gomez Loenzo Roberto Augusto. Tesis "Diseño e implementación de una celda robotizada de soldadura". Universidad Autónoma de Querétaro, 2003.

Jacob Tal. Motion Control Application. Galil Motion Control Company. pp.1 - 40, 1989.

Jacob Tal. Step by step desing of motion control systems. Galil Motion Control Company, 1994.

John J. Craig. Robótica (3a edición). México. Pearson educación, 2006.

Krzysztof R. Koslowski. Robot Motion and Control 2009. Polonia: Poznan University of technology. vol 396, 2009.

Martínez Ramirez Erika. Simulador de manipuladores robóticos desacoplados. Master's thesis, Universidad Autónoma de Querétaro, Facultad de Ingeniería, 2001.

Masatoshi Nakamura y Satoru Goto. Mechatronic Servo System Control. Japón: Saga University. vol. 300, 1998.

- N. Arda Erol, Yusuf Altintas, and Mabo Ito. Open system architecture modular tool kit for motion and machining process control. IEEE Trans. on Mechatronics. vol. 5, no. 3, pp. 281-291, 2000.
- Ollero Baturone Ánibal. Robótica: Manipuladores y robots móviles. Barcelona (España). Marcombo, 2001.
- Groover Mikell, Weiss Mitchel, Nagel Roger y Odrey Nicholas. Robotica Industrial: tecnología, programación y aplicación. McGRAW - HILL, 1990.
- Spong, M.W. y M. Vidyasagar. Robot Dynamics and Control. E.E.U.U: John Wiley & Sons, 1989.
- Timothy N. Chang, Biao Cheng and Paiboon Sriwilajaroen. Motion Control Firmware for high-speed robotics system. IEEE Trans. Ind. Elect., vol. 53, no. 5. pp. 1713-1722, 2006.

5. APENDICE

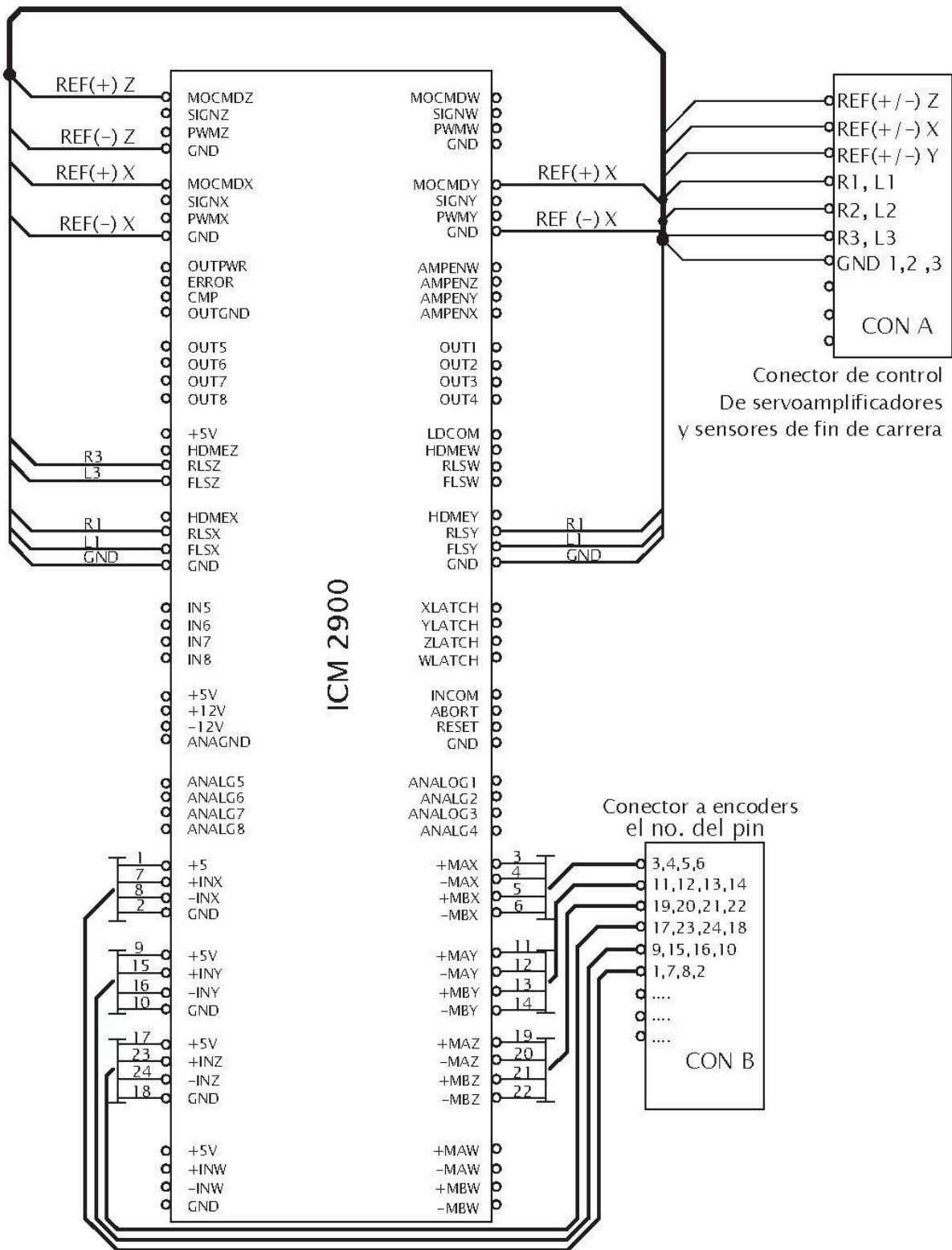
5.1 Diagrama de conexión

Hoja 1

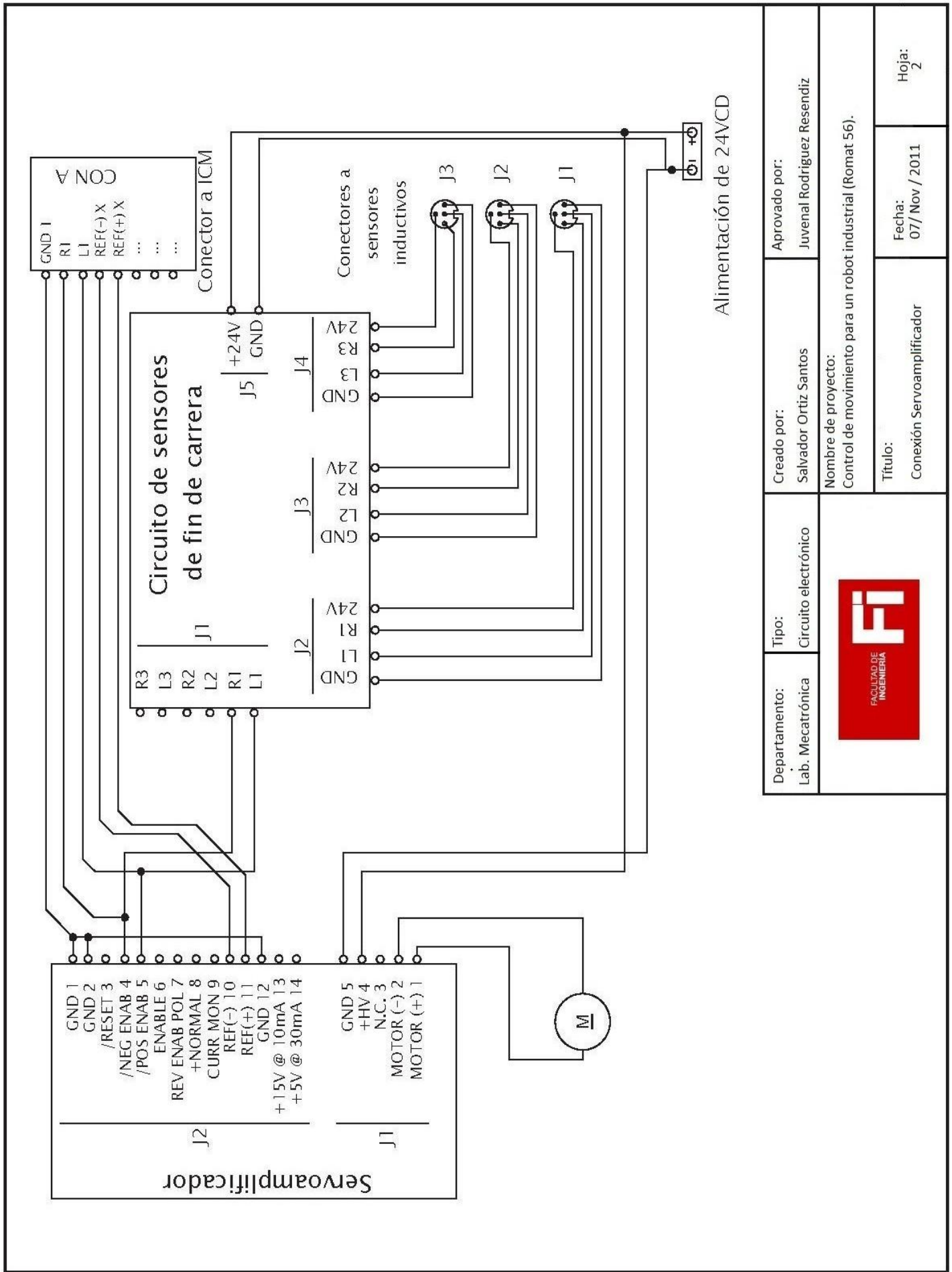
En este diagrama muestra la conexión del ICM 2900, este dispositivo es una terminal de conexiones para la tarjeta de control de movimiento modelo DMC-1866 con los dispositivos tales como los servo amplificadores que envían una señal analógica de +/- 10 Volts, como las entradas de pulsos que son generados por los encoders que son los canales A, B, Índice y la alimentación del encoder y además se tienen entradas de señal digital por parte de los sensores de fin de carrera.

Hoja 2

En este diagrama muestra la conexión del servo amplificador y el motor de corriente continua donde la velocidad o torque (según la configuración del servo amplificador) depende de la entrada de señal digital que es generado por el controlador de movimiento y que es conectado en la referencia del servo amplificador, también se muestra la conexión de los sensores de fin de carrera a la tarjeta de circuito que este manda una señal al servo amplificador y al controlador de movimiento.



Departamento: Lab. Mecatrónica	Tipo: Circuito electrónico	Creado por: Salvador Ortiz Santos	Aprobado por: Juvenal Rodríguez Resendiz
		Nombre de proyecto: Control de movimiento para un robot industrial (Romat 56).	
		Título: Conexión ICM 2900	Fecha: 07/ Nov / 2011



Departamento: Lab. Mecatrónica	Tipo: Circuito electrónico	Creado por: Salvador Ortiz Santos	Aprobado por: Juvenal Rodriguez Resendiz
		Nombre de proyecto: Control de movimiento para un robot industrial (Romat 56).	Fecha: 07/ Nov / 2011
		Título: Conexión Servoamplificador	Hoja: 2

5.2 Conectores

En la figura 39 se muestra conectores de marca Phoenix Contact, estos conectores hacen la conexión entre el gabinete y los motores del robot.



Figura 39. Conectores del gabinete.

En la siguiente tabla se muestra la conexión de los encoders hacia las terminales ICM y la división que este tiene con cada motor en el conector antes mencionada.

Función	Motor 1	Motor 2	Motor 3	Motor 4	Motor 5	Motor 6
- Index	1	9	17	25	33	41
+ Index	2	10	18	26	34	42
Canal -B	3	11	19	27	35	43
Canal +B	4	12	20	28	36	44
Canal -A	5	13	21	29	37	45
Canal +A	6	14	22	30	38	46
GND	7	15	23	31	39	47
+ 5VCD	8	16	24	32	40	48

Tabla 5.a. Conexión en el conector de los señales de encoders de cada motor.

Función	Motor 1	Motor 2	Motor 3	Motor 4	Motor 5	Motor 6
Motor (+)	1	9	17	25	33	41
Motor(-)	2	10	18	26	34	42
Freno(+)	3	11	19	27	35	43
Freno(-)	4	12	20	28	36	44
+24 VCD	5	13	21	29	37	45
GND	6	14	22	30	38	46
Señal Der	7	15	23	31	39	47
Señal Izq	8	16	24	32	40	48

Tabla 5.b. Conexión en el conector de alimentación y señales de control.

Conexión de encoders

La conexión de señales de encoder se hace a través de conectores de tipo DB – 9, donde estas presentan un código de colores que se presentan en la figura

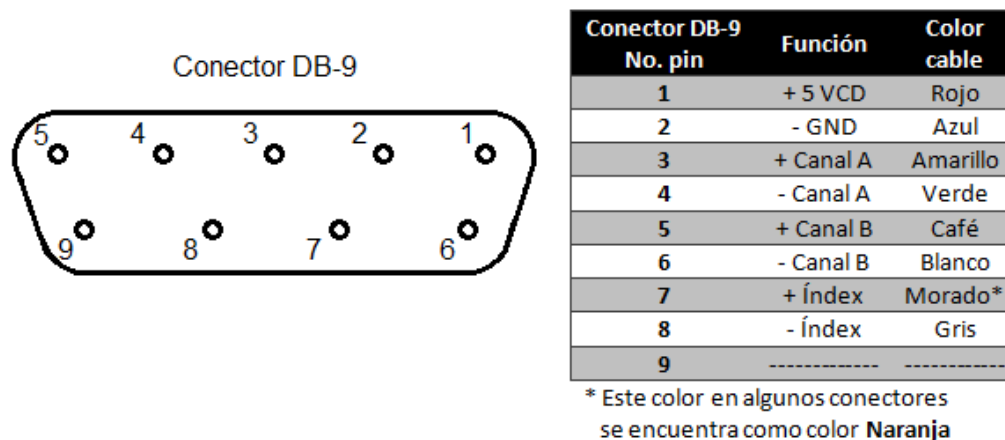


Figura 40. Conector DB-9 para señales de encoder.

Conector de motores y sensores.

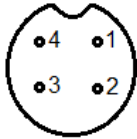
En la figura 41 se muestra el tipo de conector que se usa para conectar el robot con el gabinete. Estos conectores se usan para los motores, frenos, alimentación de sensores y señales de sensores capacitivos.



Figura 41. Conector para motor y sensores.

Conector para sensores de fin de carrera.

En la figura 42 se muestra las características de los conectores y la conexión que se hizo con estos.

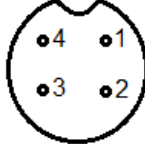


No. pin	Función	Color cable dentro del gabinete	Color cable después del gabinete
1	+ 24 VCD	Amarillo	Rojo
2	GND	verde	Azul
3	Sensor Derecho	café	Naranja
4	Sensor Izquierdo	Blanco	Negro

Figura 42. Conector para sensor de fin de carrera.

Conector para alimentación de motor y sensores.

En la siguiente figura se muestra cual fue la configuración de conexión para este conector y su característica.



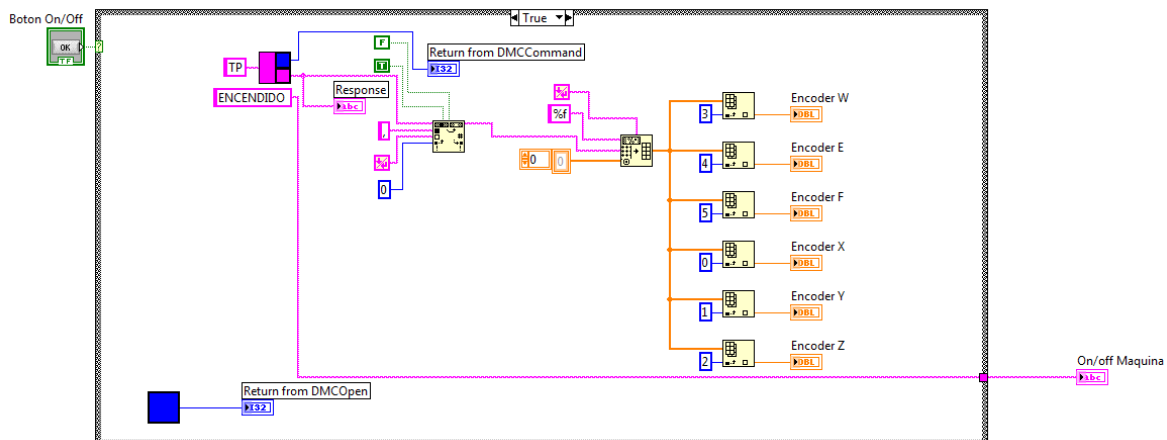
No. pin	Función	Color de cable dentro del gabinete	Color de cable fuera del gabinete
1	Motor (+)	Rojo	Blanco
2	Motor(-)	Negro	Negro
3	Freno(+)	Negro	Café
4	Freno(-)	Negro	Azul

Figura 43. Conector para motor y freno.

5.3 Partes de código para el control de movimiento.

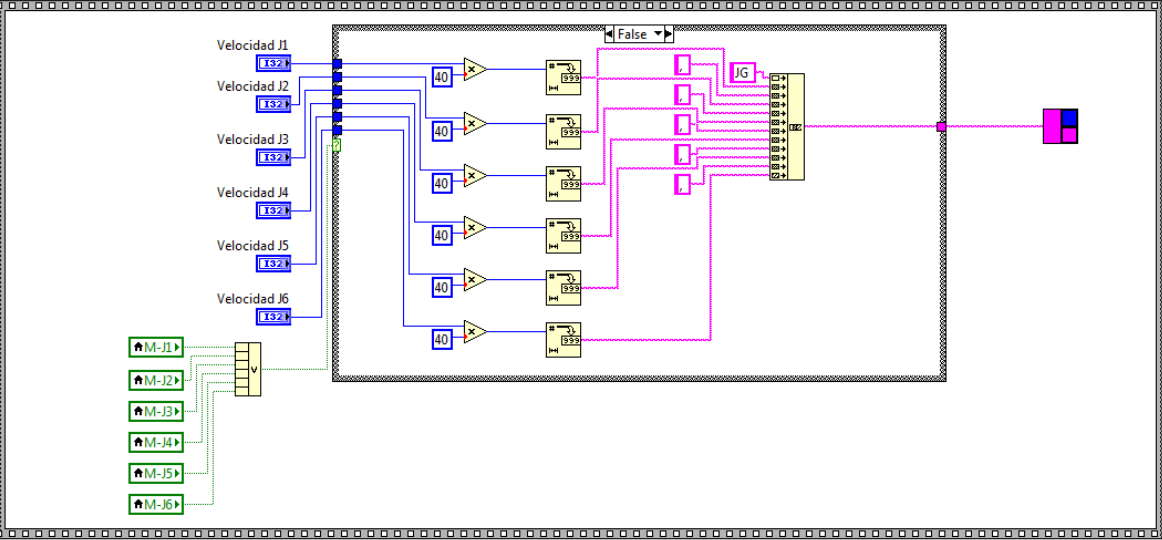
En este trabajo se muestran algunas partes del código fuente de la interfaz gráfica de usuario que es desarrollado en LabView 10.0, se considera necesario mostrar ya que se tuvo que conseguir algunas librerías que ayudaron hacer la comunicación entre la computadora y la tarjeta de control de movimiento, estas librerías son nuevas por lo que hubo algunas dificultades en operarlas, pero al final del proyecto podemos decir que valió todo el esfuerzo.

En código 1 se muestra una parte principal del programa ya que en este se manda a llamar un archivo llamado OPEN que permite abrir la comunicación con el controlador de movimiento esto se hace atreves del cuadro azul. La otra parte es que cuando la comunicación está abierta entonces le pedimos que nos establezca la posición en tiempo real por todo el tiempo que el equipo este en línea, por lo que la repuesta es una cadena de String lo cual se separa y se convierte en números enteros.

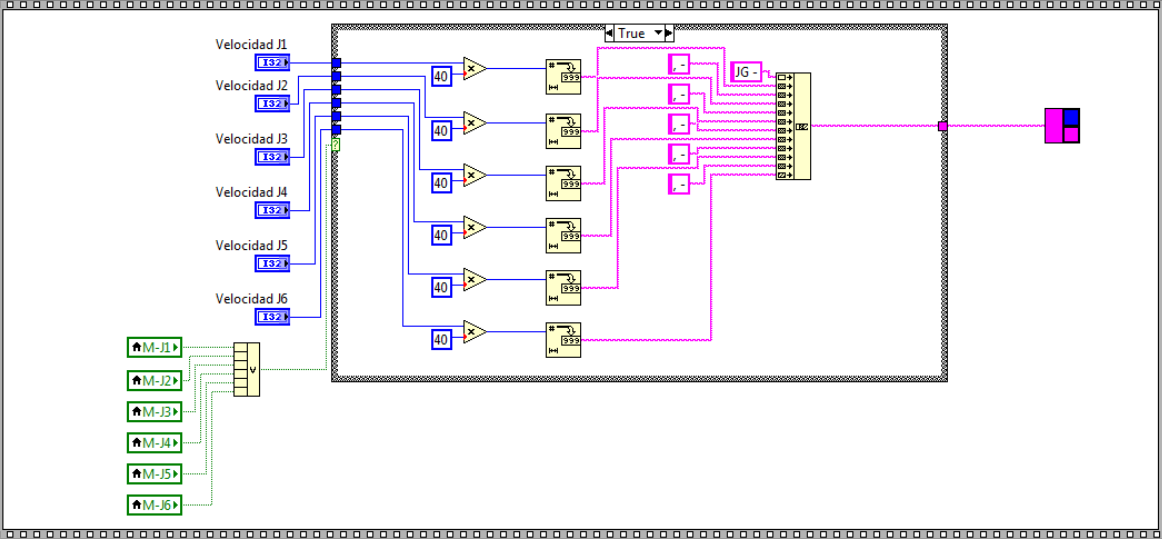


Código 1. Abre la comunicación y establece la posición en tiempo real.

En código 2 y código 3 muestran el comando JG que significa JOG este es un comando que reconoce el controlador de movimiento que implica tener un movimiento dependiendo de la velocidad requerida por el usuario.

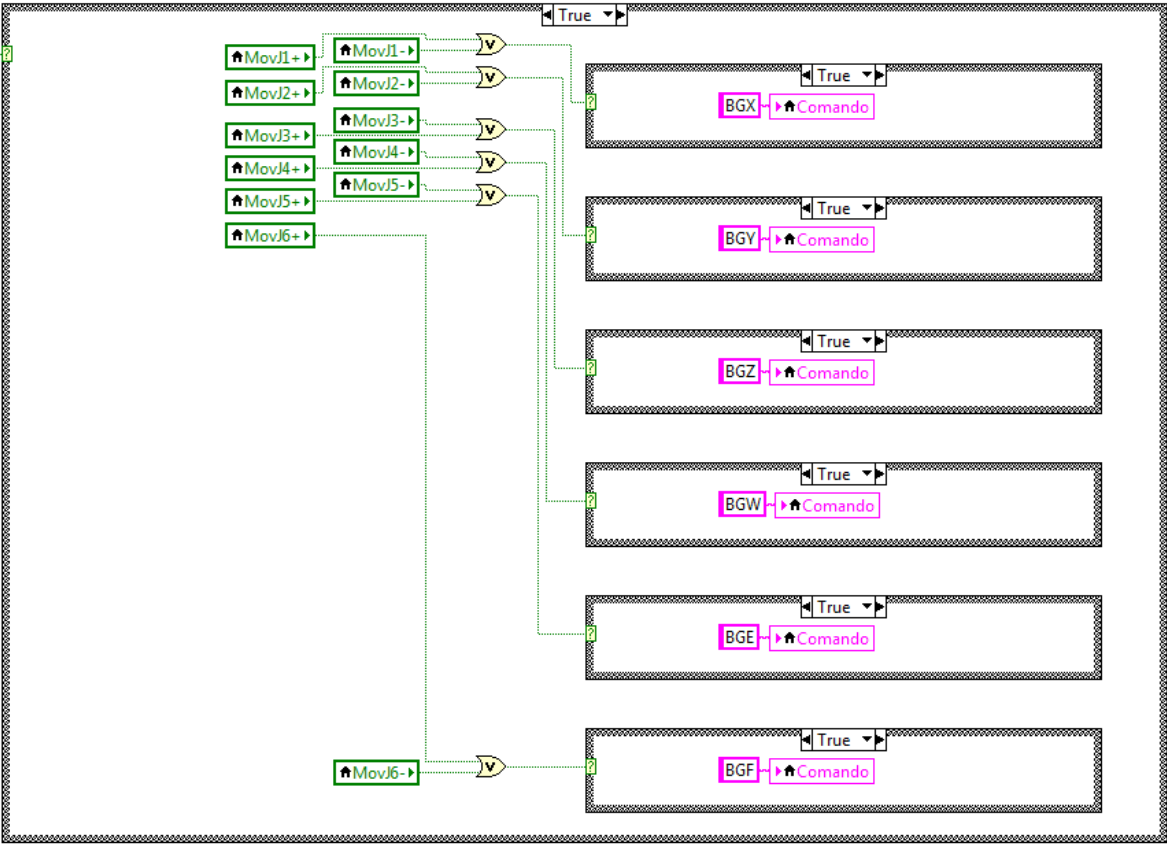


Código 2. Movimiento en sentido de las manecillas del reloj.



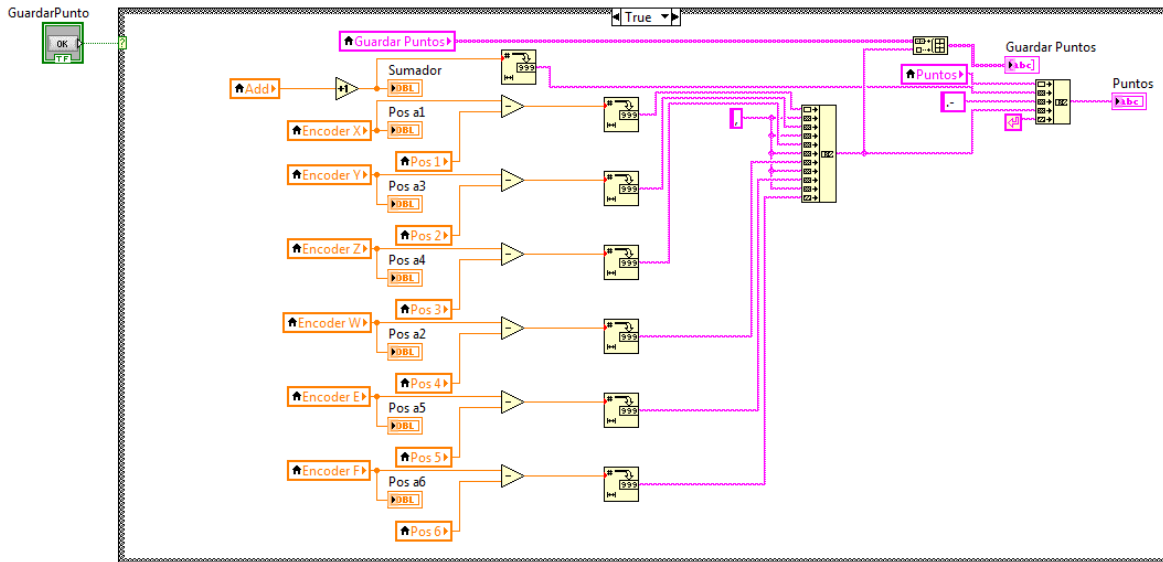
Código 3. Movimiento en sentido contrario de las manecillas del reloj.

Para la parte del editor de punto a punto los movimientos son individuales son similares al manual pero con la diferencia de que este usa un comando BG (Begin o Comienza) que requiere de otra variable para reconocer el movimiento a realizar que recordemos que nuestro robot cuenta con 6 grados de libertad donde las variables son asignados por el controlador de movimiento como X, Y, Z, W, E y F como se muestra en código 4.



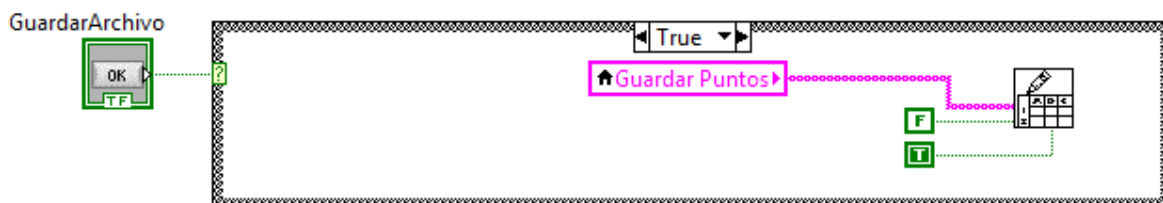
Código 4. El uso del comando BG en el editor punto a punto.

Una vez que se hayan realizado los movimientos que el usuario considere que son las apropiadas para su propósito se guarda la posición y la orientación del robot este código se muestra en la código 5.



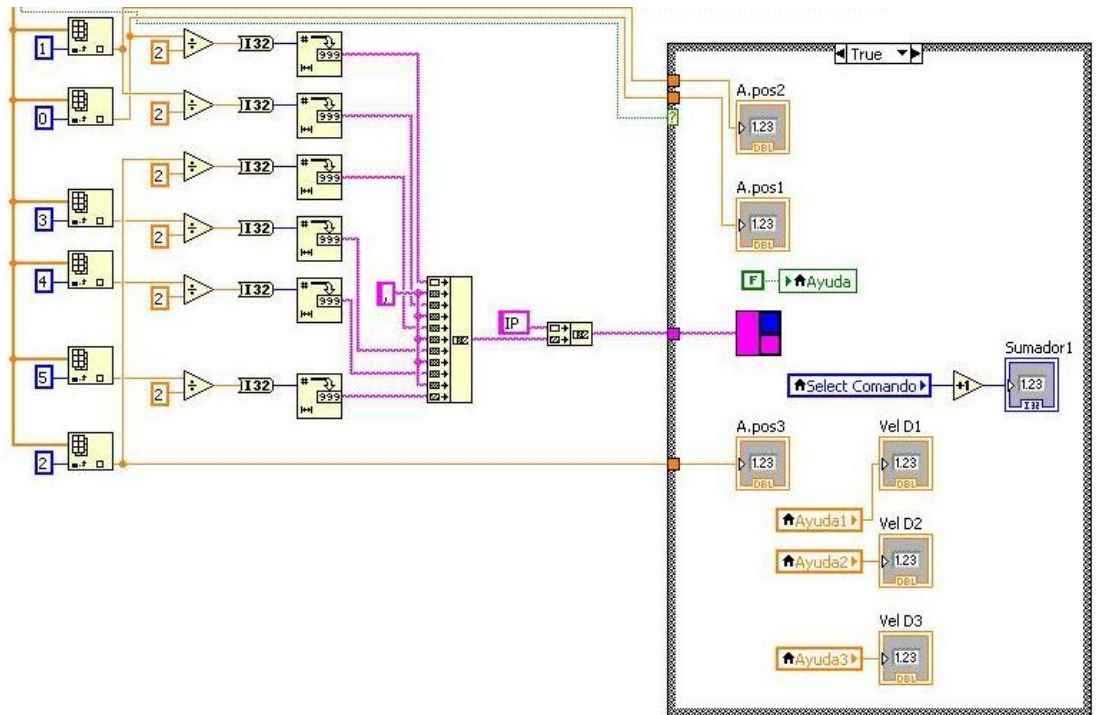
Código 5. Guarda la posición y la orientación del robot.

Cuando el usuario haya recorrido todos los puntos que este requiera es posible guardar en un archivo donde el usuario tiene la libertad de definir la dirección y el nombre del archivo para posteriormente ser abierto y recorrer todos los puntos que se hayan guardado.



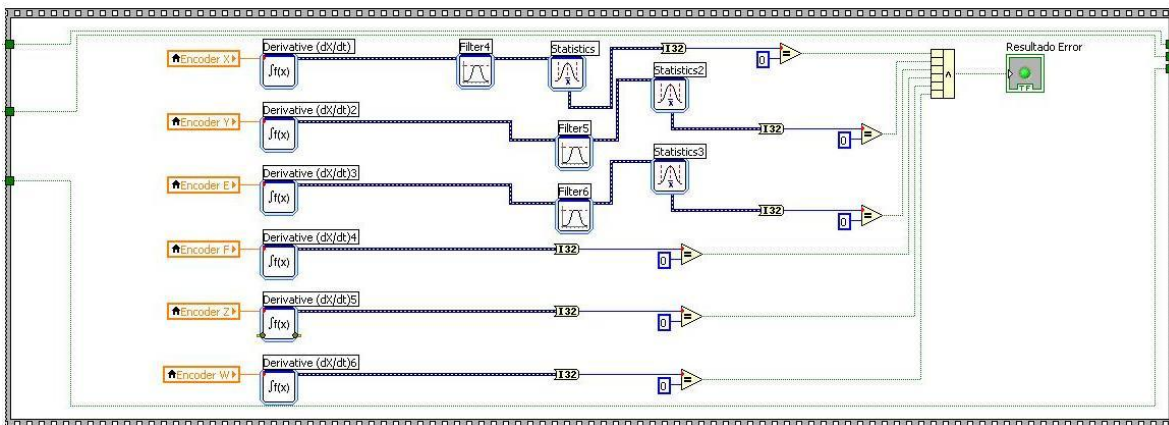
Código 6. Guardar todos los puntos en un archivo.

Código 9 es para enviar la trayectoria al controlador de movimiento para su ejecución.



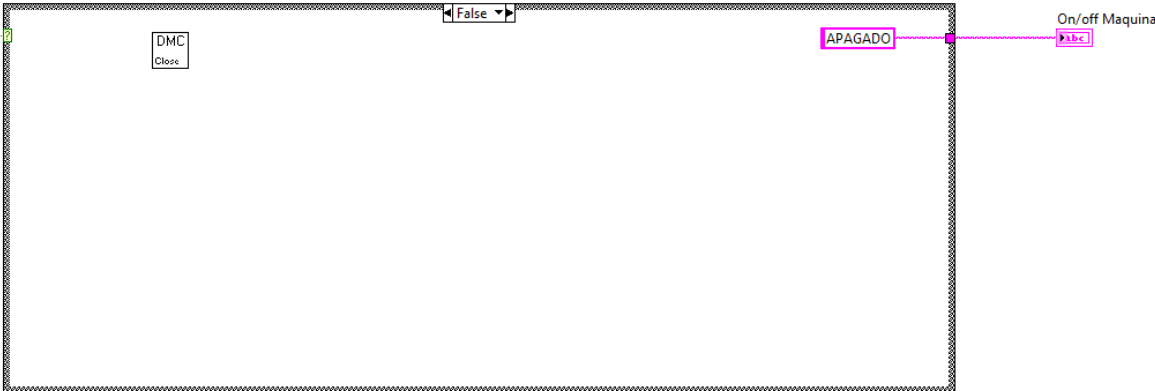
Código 9. Envía comando de movimiento al controlador.

Una vez abierto el archivo se van ejecutando las posiciones y orientaciones anteriormente guardados, para el envío de una trayectoria a otra es necesario que la velocidad de la trayectoria sea de cero, es decir, cuando la velocidad es igual a cero envía la siguiente trayectoria así sucesivamente hasta terminar de leer el contenido del archivo (código 10).



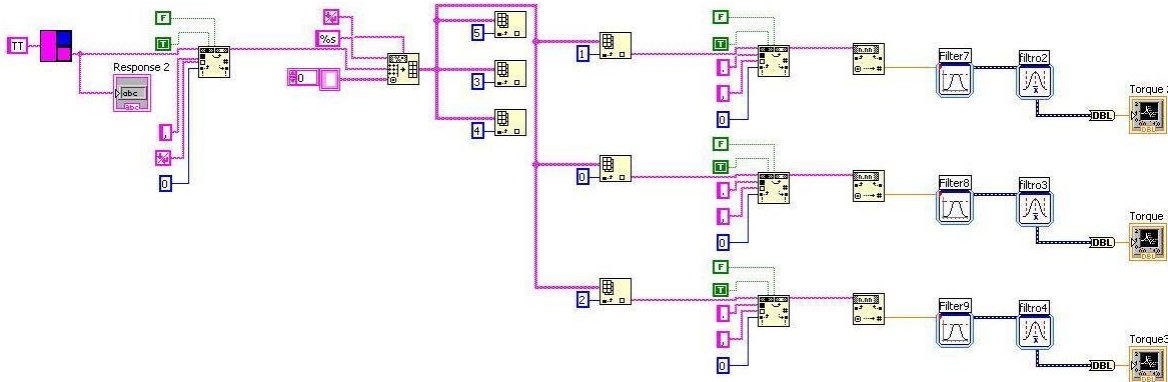
Código 10. Muestreo de velocidad para verificar trayectoria.

Otro de las librerías importantes que contamos es el Close (cerrar) cuando apagamos el sistema es necesario cerrar la comunicación entre la computadora y el controlador de movimiento al mandar a llamar esta librería automáticamente nos cierra la comunicación.



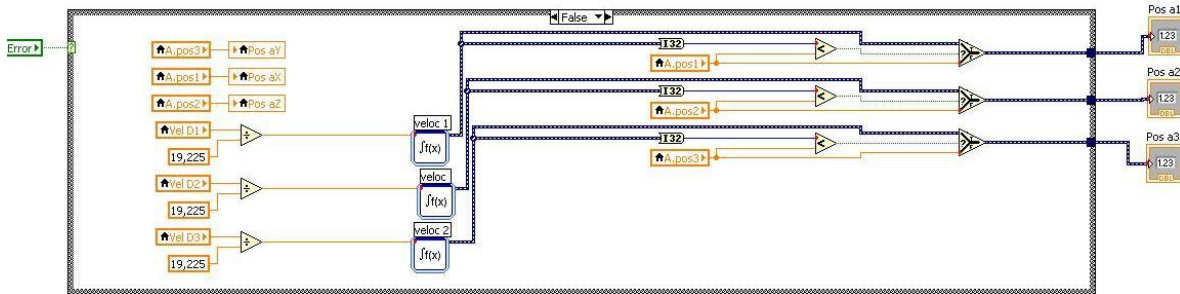
Código 14. Librería Close que cierra la comunicación.

(En código 15) muestra el código de cómo graficar el torque de los motores 1, 2 y 3. Esto se hace a través de la lectura del torque de la tarjeta de control de movimiento.



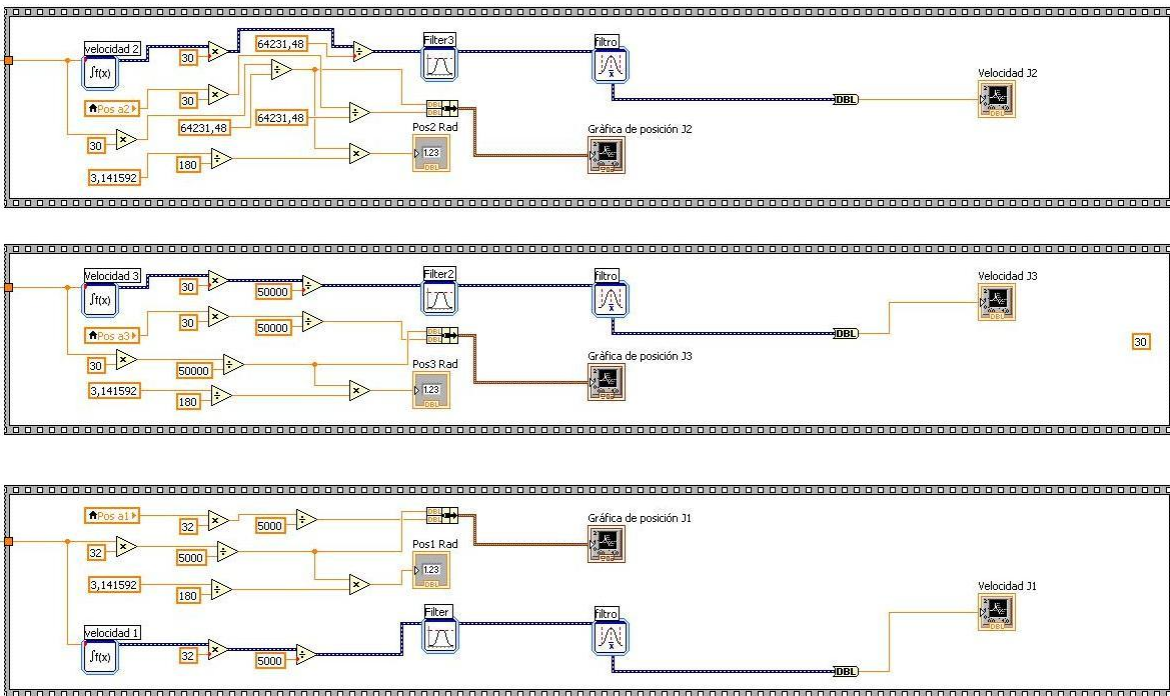
Código 15. Gráfica Torque.

(Ver código 16) muestra el código para graficar la posición deseada, en resumen esta gráfica se consigue al integrar la velocidad deseada a través del tiempo.



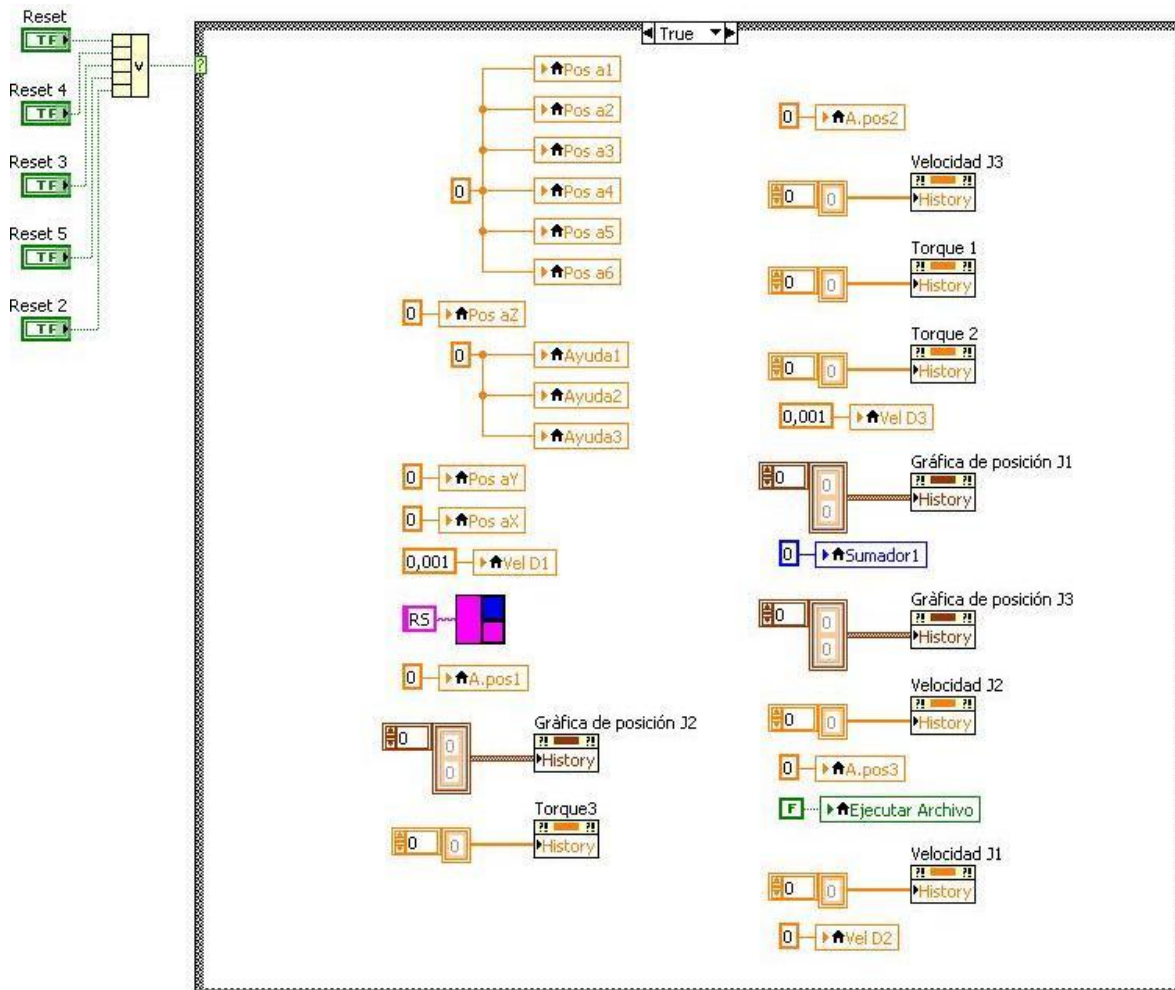
Código 16. Gráfica posición deseada.

(Ver código 17) en este se presenta el código para graficar la posición y la velocidad real. Al adquirir la posición en tiempo real este valor es en cuentas, por lo que se debe de convertir en grados y posteriormente para graficar la velocidad tenemos que derivar la posición a través del tiempo.



Código 17. Gráfica la posición y velocidad real.

En código 18 se reinician algunas variables y graficas a sus valores iniciales, esto sucede cuando se oprime el botón de reset o stop de la interfaz grafica de usuario.



Código 18. Reiniciar valores de las variables.