PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL E INTERFAZ PARA UN BRAZO ROBÓTICO DE 5 GLD

Tesis para optar el título de Ingeniero Electrónico, que presenta el bachiller:

Christian Karl Luyo Gonzales

ASESOR: Ing. Willy Carrera Soria

Lima, Julio del 2015

RESUMEN

En el presente documento de tesis se desarrolla un sistema de control para la manipulación y planificación de trayectorias de un robot de cinco grados de libertad. Este robot que ha sido fabricado por la Maestría de Ingeniería Mecatrónica de la Pontificia Universidad Católica del Perú, forma parte de un proyecto multidisciplinario que busca desarrollar un sistema robótico capaz de ser utilizado para pruebas dinámicas y cinemáticas en este área.

En el mencionado proyecto se realizó el diseño mecánico correspondiente al brazo robótico, que incluyó el dimensionamiento y posterior fabricación de cada uno de los eslabones. En la parte electrónica, se dimensionó y adquirió motores con *encoders* que permitirían obtener señales de posición y dar movimiento a los eslabones. Con toda la información adquirida a través de los dimensionamientos y ensayos mecánico-eléctricos, se logró generar modelos matemáticos de función de transferencia de motores para el diseño del control discreto.

Usando estas funciones de transferencia, se implementó un sistema de control distribuido basado en la implementación de un algoritmo PD que permite el control angular de los motores con 5 microcontroladores comunicados en protocolo I2C.Las referencias para cada control de lazo cerrado, son generadas desde una interfaz que tiene internamente un generador de referencias y enviadas por protocolo serial al microcontrolador maestro. Este generador tiene un funcionamiento basado en modelos cinemáticos que toma como principio los algoritmos de cinemática directa e inversa para la generación de trayectorias.

Finalmente, el sistema es comandado por un usuario que podrá definir el punto final de la trayectoria del brazo en coordenadas X, Y, Z. Además, el usuario podrá realizar control individual de cada uno de los eslabones y observar los valores de posición X, Y, Z en tiempo real mediante gráficas y valores.

FACULTAD DE
CIENCIAS E INGENIERÍA

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

TEMA DE TESIS PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

Título	:	DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL E INTERFAZ PARA UN BRAZO ROBÓTICO DE 5GLD
Área	:	Control y Automatización #1278
Asesor	:	Ing. Willy Carrera
Alumno	:	Christian Luyo Gonzales
Código	:	20100327
Fecha	:	01/04/15

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
Facultad de Ciencias e Ingeniería
22 JUN. 2015
MIGUEL MEJIA PUENTE
DECANO

Descripción y Objetivos

La búsqueda del aumento de la productividad y de la mejora de la calidad de los productos sin dejar de lado la eficiencia, ha impulsado a la industria al uso de maquinaria automatizada que permita reemplazar técnicas manuales que son lentas , imprecisas o peligrosas.

En la implementación y desarrollo de brazos robóticos, se utiliza diversos algoritmos de control para realizar movimientos que sean cada vez más precisos y rápidos. Esto trabaja en conjunto con una interfaz que permita el comando en tiempo real por parte del usuario y que hagan más dinámica la interacción hombre máquina.

El objetivo de esta tesis es el diseño e implementación del control del brazo robótico de 5 grados de libertad que permita el ingreso de la posicion final deseada, determine la trayectoria de movimiento utilizando algoritmos de cinemática directa e inversa, grafique el movimiento en tiempo real y utilice un sistema de control distribuido con todos los motores del brazo. Además, el sistema de control debe funcionar con un amortiguamiento crítico.

El desarrollo del presente trabajo implica la implementación de una interfaz grafica que permita la realización de la cinemática directa, cinemática inversa e interpolación de puntos de trayectoria; implementación de un sistema de control distribuido en microcontroladores maestro y esclavo con el algoritmo PID; obtención del modelo de los motores; simulación de los algoritmos PID y modelos cinemáticos en MATLAB-Simulink y finalmente la validación mediante ensayos de movimiento del brazo robot.

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
SECCIÓN ELECTRICIDAD Y ELECTRÓNICA
Ing. WILLY CARRERA SORIA
PROFESOR ASOCIADO

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA
M. Sc. Ing. MIGUEL ANGEL CATAÑO SÁNCHEZ

MÁXIMO 50 PÁGINAS

A mis padres, por su amor y apoyo incondicional.

A Jorge, José, Rodrigo e Ing. Willy Carrera,

por ser parte de esta experiencia.

A todos ellos… Gracias!

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE ECUACIONES

INTRODUCCIÓN

La búsqueda del aumento de la productividad y de la mejora de la calidad de los productos sin dejar de lado la eficiencia, ha impulsado a la industria al uso de maquinaria que permita la manufactura de manera automática. Dependiendo del contexto, este tipo de estructuras puede tener muchos tipos de aplicaciones, en su mayoría en procesos industriales; ya que, su flexibilidad limitada y su alto costo sólo se justifican en casos de producción de lotes muy grandes, o por la precisión y velocidad que se pueden aplicar en diversos procesos.

Uno de estos tipos de maquinaria automática es el robot industrial tipo manipulador de uso general, controlado generalmente por computadora que consiste principalmente en la unión de manera serial de eslabones rígidos con un extremo final libre y equipado con una herramienta para manipular objetos o realizar tareas. El diseño del sistema de control adecuado para el brazo robótico radica en un estudio previo a nivel dinámico y cinemático.

El estudio de la cinemática involucrada en el brazo robótico se divide en 2 tipos de cálculos. Por un lado, la cinemática directa trabaja directamente con los datos de entrada que especifican cuánto es el movimiento requerido por el usuario para que el brazo llegue a una posición. Por otro lado, la cinemática inversa utiliza solo la posición final enviada por el usuario para realizar los cálculos matriciales respectivos para la creación de referencias de movimiento para cada articulación de manera simultánea.

La dinámica del robot, por otra parte, trata no sólo de la geometría del movimiento sino de las causas que lo originan: fuerzas y momentos. El modelo dinámico real de un brazo de robot se puede obtener de leyes físicas como las de Newton. Esto conduce al desarrollo de las ecuaciones dinámicas de movimiento para las distintas articulaciones del manipulador con base en los parámetros geométricos e inerciales especificados para los distintos elementos.

El objetivo de este proyecto es servir como un equipo didáctico para la comprensión de temas de dinámica, cinemática, control automático y electrónica de potencia para el futuro desarrollo en robótica. Para controlar los movimientos del brazo robot, el usuario será capaz de introducir las coordenadas finales del trazo por medio de una interfaz gráfica. Esta interfaz realizará el cálculo de la trayectoria y enviará estos datos a cada articulación o motor del brazo robótico. Este brazo contará con

un sistema de control que permita llevar el último eslabón del brazo a la posición deseada a partir de las referencias enviadas por la interfaz.

Capítulo 1 SISTEMA DE CONTROL DE BRAZO ROBÓTICO

1.1 Estado del Arte

1.1.1 Presentación del objeto de estudio

En décadas pasadas han habido cambios rápidos en el uso de tecnología a nivel industrial; en este sentido, los manipuladores robóticos tuvieron que ir mejorando y adaptando su nivel de técnicas de control debido a que la manufactura automatizada de la industria y el comercio ha ido escalando a nivel de complejidad y robustez.

En el área de control aplicado a robótica, la cual es incipiente en la tecnología moderna por su creciente desarrollo y difusión, se han ido desarrollando algoritmos que se basan en modelos matemáticos y determinísticos con la finalidad de mejorar la precisión, velocidad y sobre todo la estabilidad de la respuesta del sistema. Muestra de ello es el desarrollo permanente del algoritmo PID, control difuso, redes neuronales y además del desarrollo continuo de interfaces que parten desde el envío de comandos mediante la computadora vía remota hasta el control por recepción de señales cerebrales con el fin de desarrollar futuras prótesis .

Dentro de los algoritmos de control la propuesta del algoritmo PID, es una de las más usadas para el control de sistemas robóticos y de sistemas de control propiamente dichos, debido a su simplicidad, nivel de precisión y capacidad de adaptarse a diferentes contextos físicos.

En el presente estudio, se detallan los modelamientos matemáticos inherentes al sistema del brazo robótico, los cuales se busca controlar ya sea a nivel de movimiento angular y de posición con los algoritmos más modernos aplicados a este tipo de sistema.

1.1.2 Estado de la investigación

Según las investigaciones realizadas en el College Técnico de Palestina en Gaza por Abu Qassem y su equipo, se identificaron 3 modelos matemáticos importantes para la implementación de un control completo del brazo robótico [3]. Uno de los más importantes es la cinemática directa, en la cual se utilizó el análisis de Denavit Hartenberg y las matrices de transformación que permiten a través del cálculo matricial determinar el punto final de movimiento del brazo cuando se plantean puntos de referencia a cada articulación. Este cálculo utiliza los parámetros físicos como longitudes, inercias de movimiento y alturas de cada eslabón del brazo.

Sin embargo, con la implementación de la cinemática directa solo se puede predecir la posición final si es que hay un movimiento del robot y no se podría determinar cuánto es el movimiento necesario de cada eslabón, por ello según las investigaciones realizadas por Zafer Bingul y Serdar Kucuk , la cinemática inversa tiene 3 opciones de trabajo de los cuales se eligió usar un algoritmo que tenga respuesta rápida para tener acciones de control ya que se trabaja con un conjunto de sensores en todas las articulaciones del brazo robótico [1].

El procedimiento a seguir con el algoritmo elegido se basó en la determinación de parámetros espaciales, ya sea posición, orientación y ubicación de ejes de cada articulación del sistema robótico. Con esta información se generan matrices con las cuales se hacen aproximaciones geométricas a partir de la posición deseada del usuario brindado en un sistema cartesiano que determinará los valores de referencia para el sistema de control de lazo cerrado.

Dado que la tendencia de los sistemas de control es utilizar un control robusto y preciso, es necesario incluir un sistema de control que de alguna manera utilice los algoritmos anteriores ya sea de cinemática inversa o directa para conseguir referencias para cada articulación, y que permita llevar estas variables de posición y velocidad del brazo a las de referencia. En este sentido, D. J. Foster y A.J. L. Harrison desarrollaron y presentaron el modo de control de deslizamiento, el cual se basa en el trabajo directo con ecuaciones dinámicas del sistema; además definieron una función de cambio, la cual lleva las ecuaciones iniciales a una forma que define la parte continua y descontinua del sistema [4]. A medida que el algoritmo permite controlar directamente la parte discontinua, este programa permite la disminución de las perturbaciones estocásticas y cambios bruscos como entrada lo cual convierte al algoritmo de control en uno de los más efectivos.

Por otro lado, Alassar y Elaydi tomando en cuenta el robot ya modelado matemáticamente mediante modelamiento matricial, presentan el uso del algoritmo PID, ya que este tiene una robustez y simplicidad de configuración de parámetros, lo cual hace el sistema más rápido a nivel de respuesta de sistema [2]. El algoritmo PID utiliza principalmente los valores de realimentación para generar una señal de error mediante la cual, en tiempo real, calcula o define cambios con parámetros P, I y D para los motores de tal manera que se puede controlar el movimiento cambiando las señales de entradas de los actuadores.

Además, a diferencia del caso del control de deslizamiento el algoritmo PID no tiene grandes problemas en la implementación ya que este requiere un número grande

de variables y gran procesamiento [4]. Este sistema PID debe ser robusto ante variaciones grandes en la entrada ya que finalmente es un sistema expuesto a entradas como torques y perturbaciones; en este sentido, previamente a la etapa de control de lazo cerrado de PID se implementa un bloque de control difuso para mejorar el desempeño del sistema de control en medida que este configura automáticamente los parámetros del PID [2].

Una vez obtenido el sistema de control PID en conjunto con el control difuso acoplado al sistema de brazo robótico, se necesita el desarrollo de una interfaz gráfica mediante la cual el usuario pueda generar las trayectorias deseadas a partir de puntos. Por ello, el primer paso fue modelar el sistema robótico en el programa AutoCAD y luego guardarlo como archivo stl. o slp con la intención de invocarlo en el GUI de Matlab con el comando y generar los movimientos dinámicos de manera gráfica.

1.2 Declaración de la problemática

En la implementación y desarrollo del sistema de control distribuido de un brazo robótico de 5 grados de libertad, se presentan diversos aspectos y casos que se desprecian; ya que, se busca simplificar el modelamiento matemático y sistemático del robot. No obstante, en la realidad, el sistema puede perder estabilidad y precisión al exponerse a condiciones de perturbación como torques o fuerzas externas.

Para el desarrollo del sistema del brazo robótico, se debe realizar un modelo matemático dinámico que permita el cálculo tanto de las posiciones finales del punto final del brazo robótico como la determinación de los ángulos en que cada articulación debe posicionarse para llegar al punto final solicitado por el usuario. Entre los problemas que se puede presentar están el de la acumulación de error; es decir, los factores mecánicos como las inercias, pesos y torques que hacen que los cálculos matemáticos sean más imprecisos a medida que se hagan cambios bruscos en las condiciones físicas.

En el sistema de control distribuido existen diversos modos de comunicación utilizados en varias etapas del sistema. En la primera etapa, el envío de los datos se basa en el bus RS232 el cual es utilizado para la comunicación entre el microcontrolador maestro y la interfaz gráfica que ha sido previamente programada e implementada en computadora. En la etapa siguiente, la comunicación entre el microcontrolador maestro y los esclavos es realizada en el bus I2C que permite la

conexión casi simultánea para el control de los 5 motores. Sin embargo, cuando el sistema está expuesto a condiciones ambientales existen diversos tipos de ruidos y perturbaciones que pueden generar que la comunicación por bus serial o I2C no sea exitosa lo cual provoca errores en el comportamiento del sistema principalmente por la precisión.

Luego de que se realiza la respectiva distribución de datos de referencia a cada esclavo para el control de cada articulación, se implementa un lazo de realimentación basado en la cuenta de pulsos enviado desde el encoder del motor que permite calcular el error entre el ángulo realizado en la articulación y la referencia. En este sentido, se realiza una aproximación entre el número de pulsos contados y el ángulo que corresponde al caso ideal de movimiento.

1.3 Algoritmos matemáticos

Los algoritmos empleados en la interfaz gráfica para el cálculo de las posiciones y generación de referencias en tiempo real para el movimiento de cada motor son los siguientes:

1.3.1 Cinemática directa

Se utiliza fundamentalmente el álgebra vectorial y matricial para representar y describir la localización de un objeto en el espacio cartesiano con respecto a un sistema de referencia fijo. Cabe resaltar que el brazo robótico es una cadena cinemática formada por objetos rígidos o eslabones unidos entre sí mediante articulaciones en la que se puede establecer un sistema de referencia fijo situado en la base del robot y describir la localización de cada uno de los eslabones con respecto a dicho sistema de referencia. De esta forma, el problema cinemático directo se reduce a encontrar una matriz que permita relacionar tanto la rotación como la traslación de cada eslabón en cadena para hallar la posición final incluyendo parámetros físicos como longitud y grados de libertad.

1.3.2 Cinemática Inversa

El objetivo de la solución de la cinemática inversa consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot $q=[q1,q2,\ldots,qn]T$ donde qn es el ángulo de cada eslabón con respecto a una referencia fijada para cada caso. La idea es tener un juego de ángulos como solución que permita al eslabón final llegar a la posición final definida.

Dada la posición y orientación del actuador final del robot, el problema cinemático consiste en calcular todos los posibles conjuntos de ángulos entre las articulaciones

que podrían usarse para obtener la posición y orientación deseada. La obtención de las ecuaciones que rigen la cinemática inversa son fuertemente dependientes de la configuración del robot.

La ausencia de una solución significa que el robot no puede alcanzar la posición y orientación deseadas porque se encuentra fuera de su espacio de trabajo o fuera de los rangos permisibles de cada una de las articulaciones.

Capítulo 2 HARDWARE DEL BRAZO ROBÓTICO DE 5GLD

En este capítulo, se brindará detalles del *hardware* que ya había sido implementado en la estructura y la que se desarrolló para el brazo robótico de la ilustración 1.

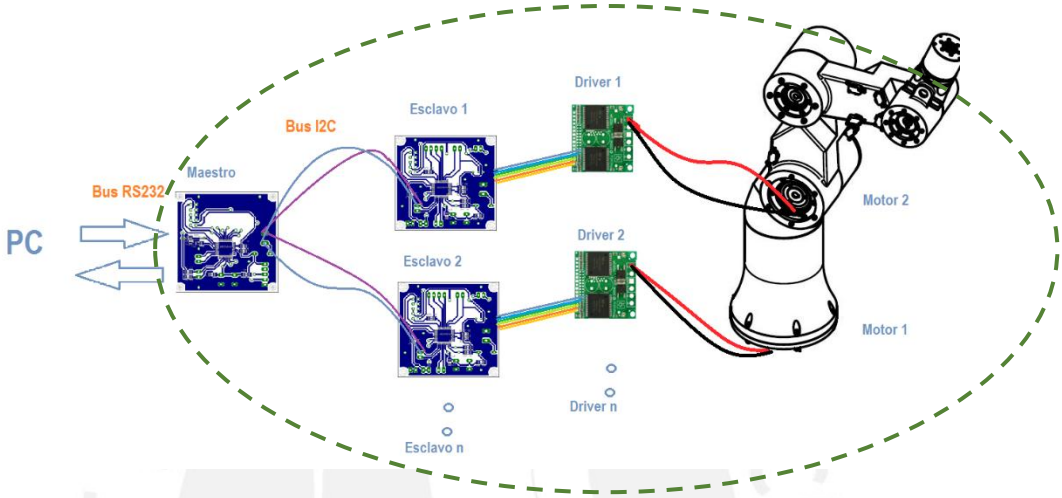


Ilustración 1 Solución a nivel de hardware

2.1 Componentes de hardware

2.1.1 Características físicas

Longitud de cada eslabón

El brazo robótico se conforma de 5 eslabones en serie, los cuales tienen diferentes longitudes y se detalla en la ilustración 2.

L1 = 17.5 cm, con ejes Zo Y Z_1 a 2 cm de distancia d.

L2 = 28 cm, ejes paralelos.

L3 = 28 cm, ejes paralelos.

L4 = 19 cm, del harmonic drive del M4 hasta el harmonic drive del motor 5, ejes en cuadratura, con d = 0cm.

L5 = 09 cm, del harmonic drive del M5 hasta el centro del gripper.

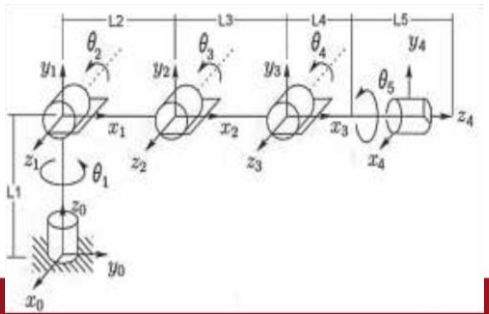


Ilustración 2 Estructura y ejes de referencia del brazo

Verificación de los ángulos de rotación de cada articulación

El brazo robótico está diseñado para tener los siguientes grados de libertad en cada eslabón:

θ1 = ± 360°
-45° ≤ θ2 ≤ 225°
-135°≤ θ3 ≤ 135°
-135°≤ θ4 ≤ 135°
Θ5 = ± 360°

Sin embargo, por defectos mecánicos en el eslabón 2 debido a una falta de encaje entre el *armonic drive* y la armadura metálica que lo contiene, se generan fuerzas que se oponen al movimiento lo cual restringe el movimiento al siguiente rango.

25°≤ θ2 ≤ 120°

Peso de cada eslabón

Tabla 1 Masas de los componentes del brazo. Elaboración propia

		Kg
Peso de la carga:	1	**1**
Peso que soporta el M5:	1.973	**2.973**
Peso del gripper más casco del M5:	1.973	
Peso que soporta el M4:	1.86	**4.833**
Peso del M5	0.13	
alma,brida,harmonic drive del M5	0.394	
acople del eslabón 4 y 5	0.795	
casco del M4	0.541	
Peso que soporta el M3:	4.941	**9.774**
Peso del M4	0.507	
estructura, brida, rodamientos, tornillos	1.204	
eslabón une M3 y M4	2.096	
casco del M3	1.134	
Peso que soporta el M2:	6.278	**16.052**
Peso del M3	1.377	
estructura, rodamientos, distanciador, tapa	1.519	
bridas y harmonic drive	0.513	
tornillos	0.053	
eslabón une M2 y M3	2.816	
Peso que soporta el M1:	8.477	**24.529**
Peso del M2	1.385	
estructura, har.drive, distanciador, rodamientos, disco, tornillos	2.778	
casco del M2	1.07	
acople de eslabón 1 y 2	2.6	
tapa de acople	0.644	
Adicionalmente se tienen los siguientes pesos:		
Motor M1	1.36	
anillo ditanciador, harmonic drive	0.516	
alma con anillo de tapa	2.011	

2.1.2 Circuito Maestro

Funcionamiento:

Esta tarjeta con microcontrolador recibe la información de ángulo y sentido desde la interfaz de la PC, y luego la distribuye a los 5 esclavos para realizar los movimientos necesarios. Tiene la particularidad de comunicarse hasta con 5 esclavos a través de protocolo I²C. La tarjeta impresa y el esquemático con la distribución de componentes se detallan en las figuras 3 y 4.

Detalles técnicos de la tarjeta:

- Voltaje de alimentación: 5 V
- Interfaz para comunicación: I²C

- Dimensiones: 5 x 5 cm
- Material: Fibra de vidrio

Componentes Empleados:

- Atmega8A-AU
- 9 leds
- 12 resistencias (9 de 1K, 1 de 10K, 2 de 4.7K Ohmios)
- 5 condensadores de 100nF
- 1 inductancia de 10uH
- 1 pulsador de reset

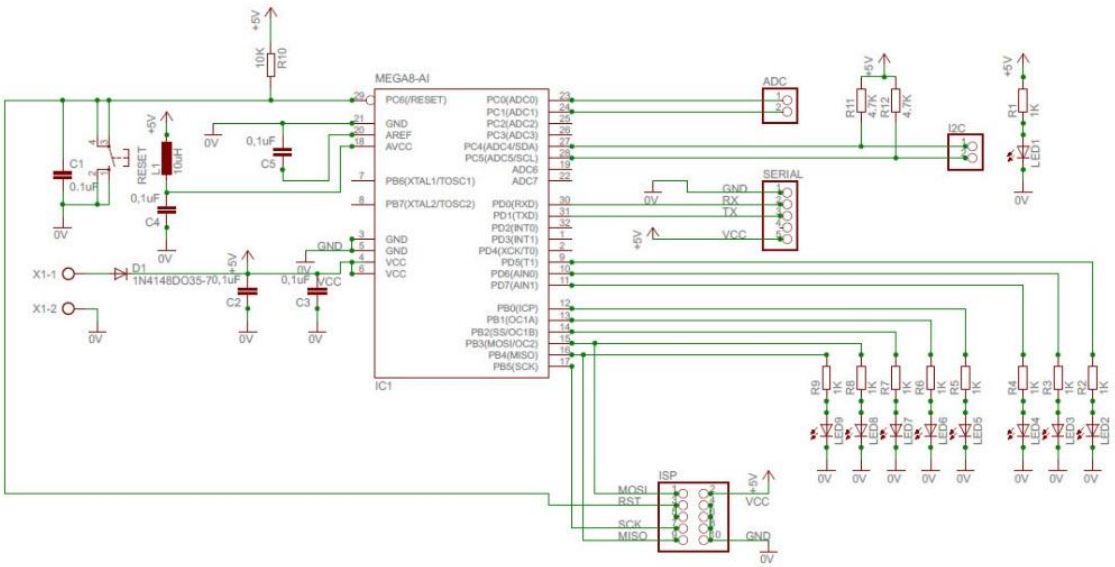


Ilustración 3 Circuito esquemático de maestro.[13]

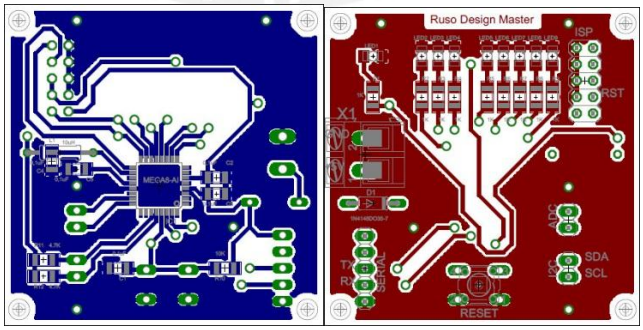


Ilustración 4 Tarjeta impresa circuito maestro. [13]

2.1.3 Circuitos Esclavos

Los esclavos se encargan de la acción de control, emiten señales hacia los motores con el fin de producir movimiento de acuerdo a los requerimientos provenientes del maestro. En otras palabras, de acuerdo con las cuentas provenientes del encoder, ejecuta el movimiento de acuerdo con ángulo estimado de acción. Se cuenta con cinco de esto circuitos, uno por cada motor a controlar. Su distribución en esquemático y *la* tarjeta impresa se detallan en las figuras 5 y 6.

Detalles técnicos: (Por cada tarjeta)

- Voltaje de alimentación:5 V
- Interfaz para comunicación: I²C
- Dimensiones: 5 x 5 cm
- Material: Fibra de vidrio

Componentes Empleados:

- Atmega8A-AU
- 9 leds
- 12 resistencias (3 de 1K, 1 de 10K, 1 de 4.7K, 2 de 2K Ohmios)
- 4 condensadores de 100nF
- 1 pulsador de reset

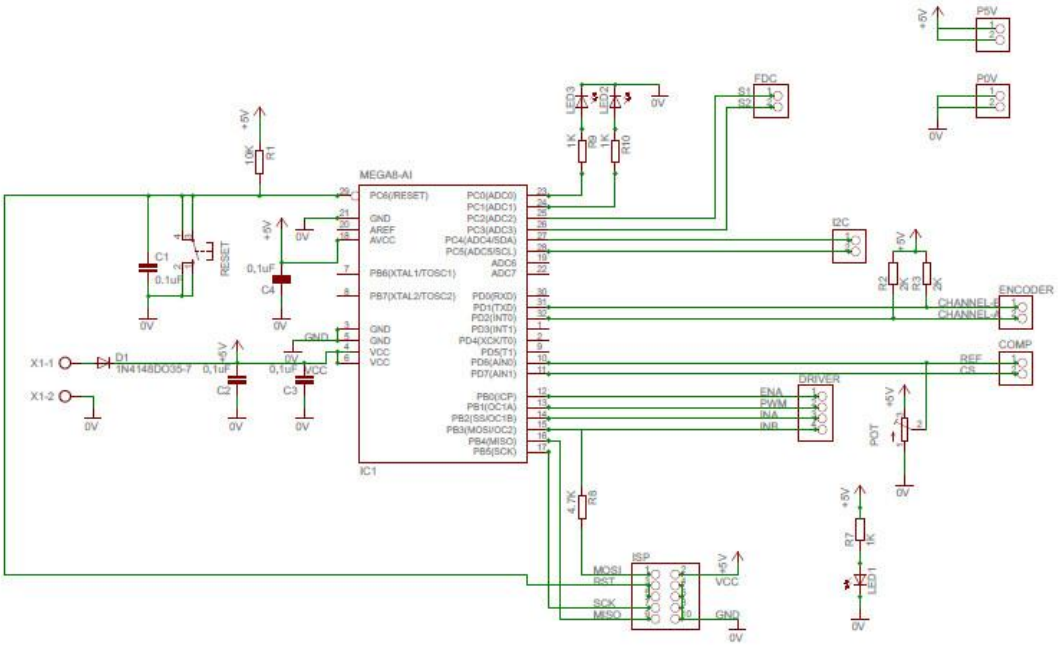


Ilustración 5 Esquemático del esclavo.[13]

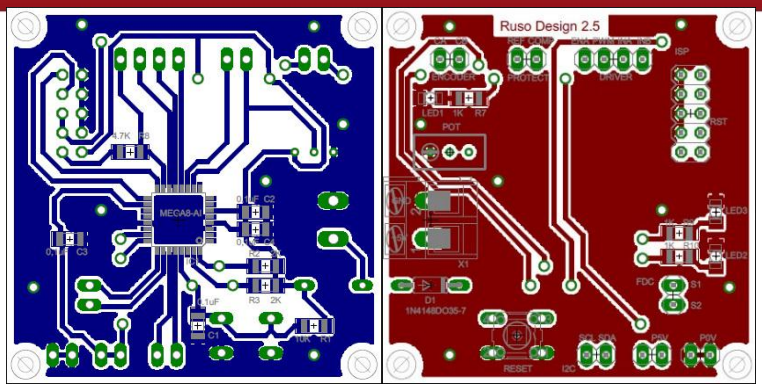


Ilustración 6 Tarjeta impresa del circuito esclavo.[13]

2.2 Hardware de control de motores

2.1.1 Circuito del Driver

Es el circuito encargado del manejo del sentido, velocidad y habilitación de los motores, utiliza el driver VNH2SP30-E, el cual puede manejar hasta dos motores de forma independiente.

2.1.2 Circuito del Encoder

Es el circuito encargado de medir la posición en la que se encuentra el motor. El sensor que se utiliza en esta aplicación es el encoder óptico. El sistema cuenta con 5 encoders y sus características son detalladas en la tabla 2.

Motor	Modelo	PPR	Cover	Electrical	Hub(diametro del eje)	Largo de cable con conector	Fabricante
1	M53	2048	Con salida axial para conector JST	Open Collector	3/8"	8 ft	Dynapar
2	M53	2048	Con salida axial para conector JST	Open Collector	3/8"	8 ft	Dynapar
3	M53	2048	Con salida axial para conector JST	Open Collector	3/8"	8 ft	Dynapar
4	M15	1024	Enclosed, End-of-the-shaft mount	Open Collector	1/4"	8 ft	Dynapar
5	HEDS-9140	500	-	-	Trabaja con discos ópticos de 11 mm de diámetro	-	Avago

Tabla 2 Tabla de características de encoders. Elaboración propia

Los encoders resaltados en amarillo indican que no estuvieron disponibles para la implementación.

2.2.3 Circuito de protección del motor

La protección se determina de manera que cumplan las exigencias de funcionamiento normales:

- Soportar la corriente a plena carga y las sobre intensidades a corta duración.
- No provocar caídas de tensión que pudiera perjudicar el rendimiento del motor y su respectivo driver.
- Garantizar la protección de las personas que manipulan el equipo.
- Proteger el motor en caso de cortocircuito, debido a trabas del motor o aumento de torque.

Se utilizará el driver, Dual VNH2SP30 Motor Driver Carrier MD03A (ver ilustración 7), el cual nos proporciona un pin de censado de corriente del motor CS.

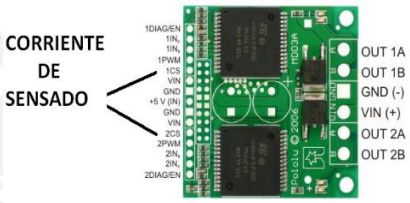


Ilustración 7 Distribución de pines de driver

El funcionamiento del motor, es decir su movimiento respectivo, se controlará mediante modulación de ancho de pulsos (PWM). Por ello, se debe considerar la gráfica 8, ya que, nos muestra el comportamiento de la corriente de sensado (motor) debido a una señal PWM.

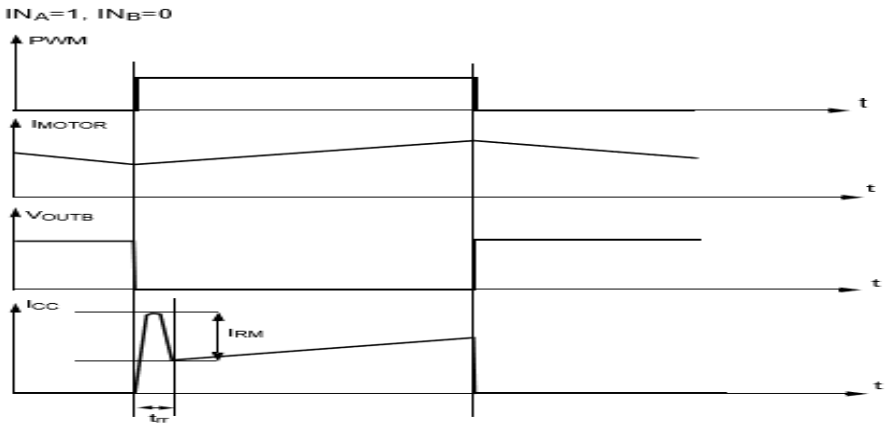


Ilustración 8 Diagrama de tiempos pin CS

Como podemos observar a la salida del pin CS se le acopló una resistencia de censado R7 (1.5KΩ). Además, de un circuito “tanque”, con lo cual en la salida del pin 5 de JP1, se obtiene una señal de la forma mostrada en la señal de la izquierda (ver ilustración 9), por ello, se le acoplará un condensador en paralelo a la salida

para de esta manera reducir el voltaje de rizado y obtener una nueva señal de salida (ver ilustración 9).

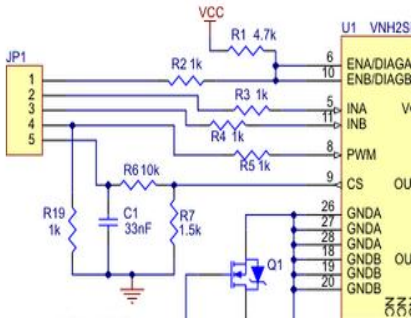


Ilustración 9 Pines de ATMEGA8A

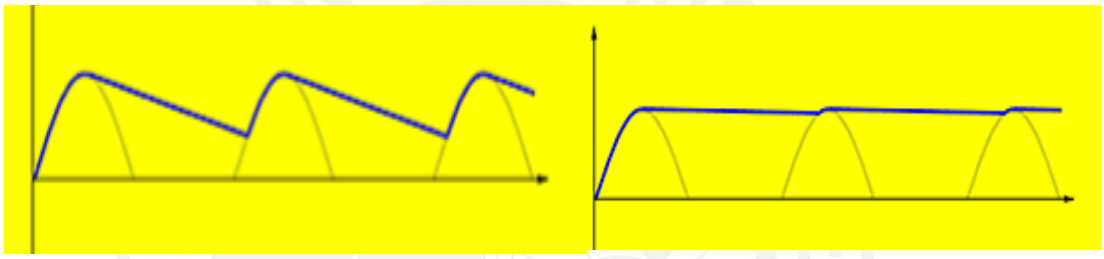


Ilustración 10 Efectos del condensador en CS

Según la hoja de datos del driver a la salida del pin CS obtenemos aproximadamente 0.13 V/A; por ello, es necesario un proceso de amplificación de la señal.

Se utiliza una amplificación no inversora de la señal, el diagrama esquemático de la ilustración 11 lo muestra a continuación:

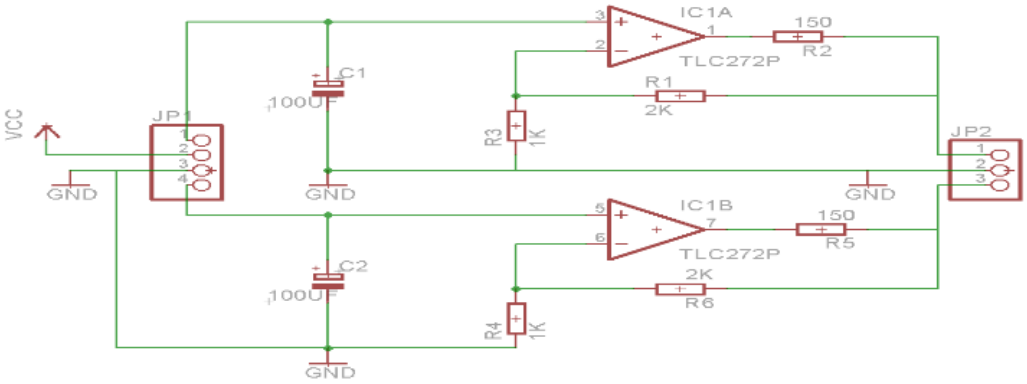


Ilustración 11 Esquemático de amplificador. [12]

Se acaba de detallar el proceso de obtención del voltaje de sensado del motor, una vez obtenida esta señal se utilizará un comparador analógico para su evaluación (ver ilustración 12).

El ATMEGA8A que se utiliza tiene 2 pines (ver figura 12) para uso del comparador analógico.

El pin PD6(AIN0), se utilizará como fuente de referencia.

El pin PD7(AIN1) recibirá la señal del OPAMP y hará la comparación con el pin PD6.

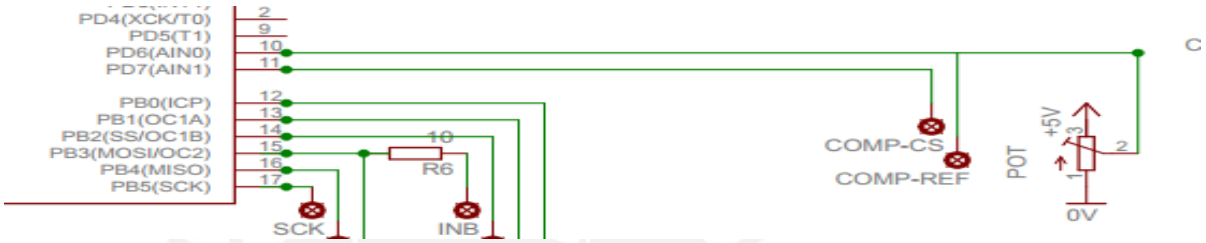


Ilustración 12 Comparador analógico [12]

La rutina de control que se llevará a cabo es la siguiente:

La interrupción del comparador se activará una vez que el ordenador (PC) indique una sobrecorriente. Cuando el voltaje de referencia sea mayor al voltaje del censado, no habrá riesgo de sobre intensidad de corriente, el proceso continuará normalmente. Pero si el voltaje de referencia fuese menor en ese caso se activará un timer que contará un determinado tiempo antes de mandar una señal al maestro (se activa una bandera) y este a su vez al *driver* que deshabilitará los motores en marcha.

2.3 Diseño de tarjetas de encoder para implementación de motor 1 y 2

2.3.1 Diseño disco 1

Se consideraron 51 agujeros concéntricos y equidistantes en un disco que esta acoplado en el eje del motor 1 (ver ilustración 13).

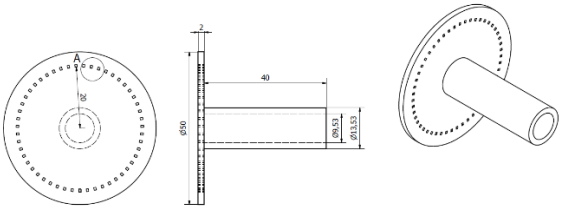


Ilustración 13 Encoder de motor 1 implementado. Elaboración propia

2.3.2 Diseño disco 2

Se consideraron 51 agujeros concéntricos y equidistantes en un disco que esta acoplado en el eje del motor 2. (Ver ilustración 14)

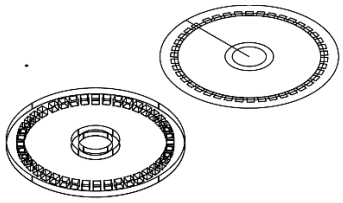


Ilustración 14 Encoder de motor 2. Elaboración propia

2.3.3 Diseño tarjeta de generación de señal del encoder 1 y 2

El disco sería acoplado a los ejes respectivos de los motores 1 y 2 y además se le integraría un optocoplador Código MOC70T3. Luego de ello se procedió a cablear sus señales A y B hasta una tarjeta que contiene un OPAMP con la idea de mejorar la forma de onda de la señal recibida desde los optocopladores en cada motor cuyo esquemático se muestra en la ilustración 15.

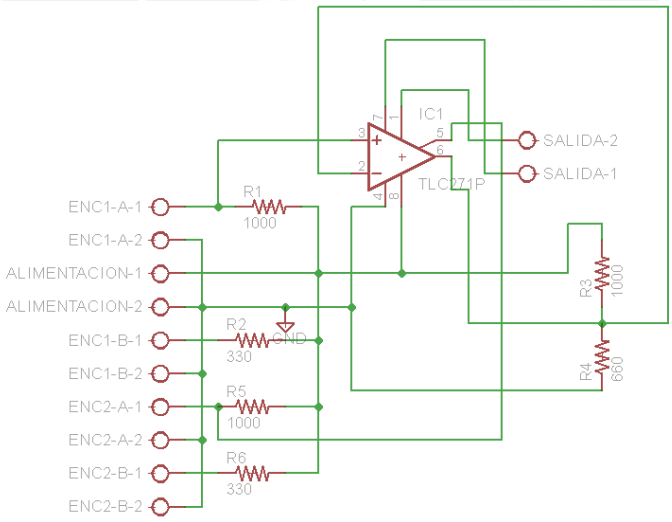


Ilustración 15 Tarjeta Comparadora de señales de optocoplador en motor 1 y 2.Elaboración propia

En la tarjeta se incluyen 2 salidas con las señales mejoradas que irán a las interrupciones INT0 de las tarjetas esclavas 1 y 2 (ver ilustración 16 y 17). Además, se está tomando una referencia de 2.5V ya que las ondas recibidas de los optocopladores varían entre 2V y 3.8V en ambos casos.

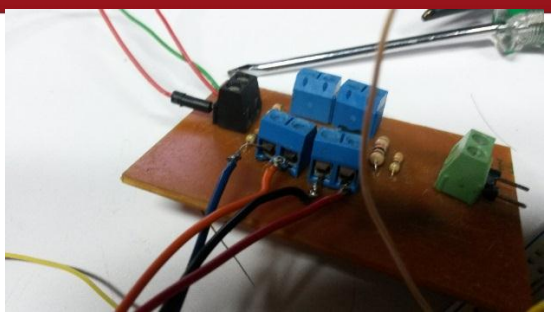


Ilustración 16 Tarjeta de optocopladores implementado .Elaboración propia

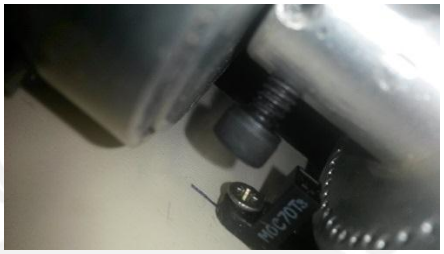


Ilustración 17 Optocopladores instalados. Elaboración propia

Capítulo 3 DISEÑO DEL SISTEMA DE CONTROL

3.1 Modelo teórico

La integración de un sistema de control distribuido y de una interfaz didáctica con el hardware de un brazo robótico de 5 grados de libertad, debe hacer posible la generación de movimientos de las 5 articulaciones de manera precisa y anticipada; es decir, debe enviar comandos a partir de la interfaz que defina la trayectoria que el brazo efectuará.

La estructuración que se presenta se basa inicialmente en el uso de una interfaz gráfica trabajada en el programa Visual Studio del sistema operativo Windows, permite al usuario ingresar las especificaciones espaciales de movimiento. Esta interfaz utiliza algoritmos matemáticos con el fin de operar y definir la ubicación de puntos por medio de cálculos basados en la cinemática inversa y cinemática directa. Por un lado, la cinemática inversa que determina el movimiento de cada articulación para que el extremo final del brazo llegue al punto indicado por el usuario en unidades cartesianas. Por otro lado, la cinemática directa permite saber la ubicación de la última extensión del brazo a partir de coordenadas articulares fijadas.

Cabe resaltar que para aplicar los algoritmos matemáticos, se emplea datos físicos como el peso, dimensiones e inercias de toda la estructura, los cuales son vitales para modelar el brazo con el fin de realizar estimaciones dinámicas del brazo robótico.

Estos datos generados a partir de la interfaz gráfica para la generación de trayectorias, son principalmente los ángulos y sentidos de referencia para cada uno de los motores. Toda esta información es enviada por medio del protocolo serial a

un microcontrolador llamado maestro para su posterior distribución hacia los controladores esclavos (ver ilustración 18). En este sentido, el control distribuido planteado al generar referencias en cada motor de cada articulación produce el movimiento esperado de manera simultánea y precisa.

Para determinar la eficiencia del sistema se requerirá el cálculo de los puntos finales se considera el uso de una interfaz gráfica y de mediciones en coordenadas cartesianas. Aparte de ello, también se usará un programa de Matlab que permite anticipar el movimiento del brazo a nivel angular.

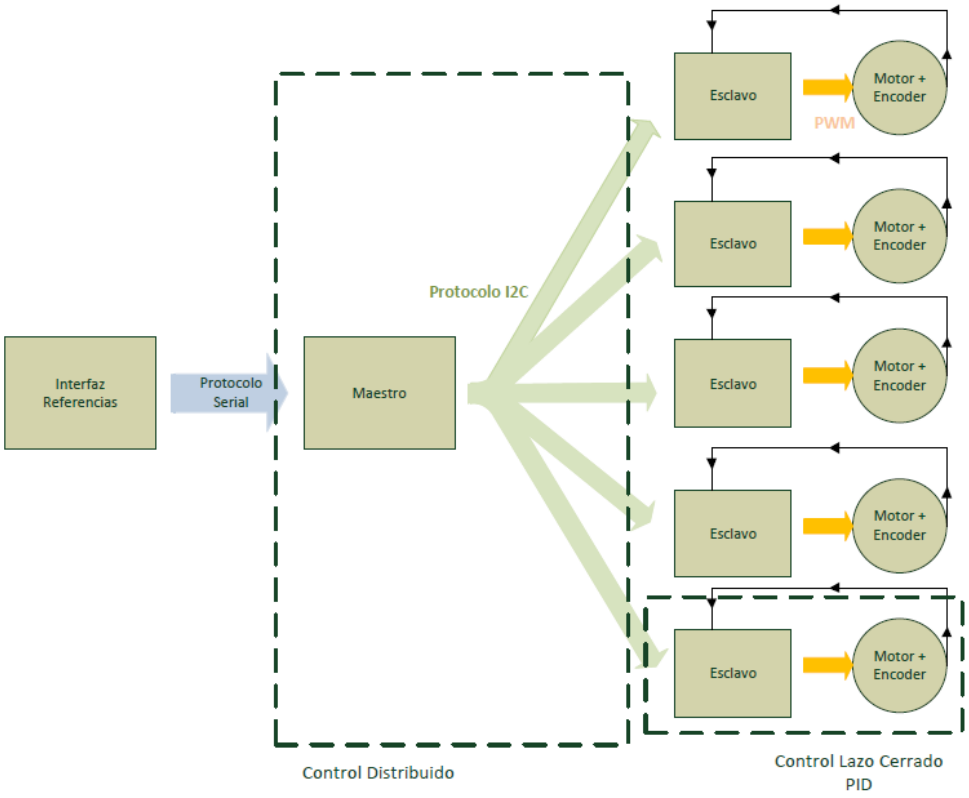


Ilustración 18 Diagrama de solución general. Elaboración propia

3.2 Objetivos Generales

El objetivo del desarrollo de un sistema de control con una interfaz gráfica, es de implementar un brazo robot como herramienta didáctica para la comprensión del comportamiento dinámico y cinemático de un sistema robótico.

El sistema de control comprende el uso de una lógica distribuida; es decir, las tareas de control se subdividen a los controladores que actúan directamente en el comportamiento de cada moto. Esto será posible con el uso de la comunicación I2C entre maestro y esclavos y la comunicación serial entre maestro e interfaz. Con esto, el control de posición será realizado por cada microcontrolador esclavo con un amortiguamiento menor al 5%, un error menor a 8 grados por eslabón y con el

tiempo de establecimiento menor en cada caso. A nivel de trayectorias, el sistema debe trabajar de manera simultánea y el volumen de error debe ser menor a 9cm de radio tomando el punto deseado como centro de la esfera que se forma.

Las referencias necesarias para el sistema de control son enviadas por la interfaz gráfica implementada en la pc. Esta interfaz tiene la capacidad de generar los datos necesarios como la posición y ángulos de referencia para los motores usando algoritmos cinemáticos de cinemática directa e inversa; además tiene la opción de graficar en tiempo real el movimiento del brazo de tal manera que se pueda monitorear las referencias enviadas.

3.3 Modelo cinemático del brazo robótico

3.3.1 Modelo cinemático directo

Para la obtención del modelo cinemático directo del brazo, en primer lugar, se deben hallar los parámetros del algoritmo Denavit-Hartenberg el cual se basa en encontrar las relaciones que permiten conocer la localización espacial del extremo del robot a partir de los valores de sus coordenadas articulares [23].

Para este caso, al tener varios grados de libertad se debe plantear un modelo sistemático basado en la utilización de matrices de transformación homogénea. Es decir, se puede asociar a cada eslabón un sistema de referencia adecuado a cada uno representando las rotaciones y traslaciones relativas entre los distintos eslabones que componen el brazo robótico (ver ilustración 19).

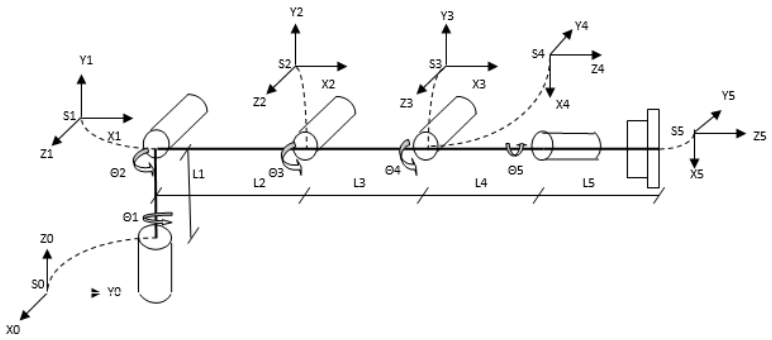


Ilustración 19 Distribución de ejes en brazo robótico 5GLD. Anexo 1

El procedimiento para poder localizar los ejes de cada eslabón (mostrado en la ilustración 19), requiere de la traslación y rotación de los mismos para relacionarlos de manera geométrica y poder generar los parámetros Denavit Hartenberg [23]. Los procedimientos seguidos son mostrados en el anexo 1 y los resultados en la tabla 3.

Tabla 3 Parámetros Denavit Hartenberg

Articulación	Θ	D	A	A
1	Θ_1 + 90°	L1	0	90°
2	Θ_2	0	L2	0
3	Θ_3	0	L3	0
4	Θ_4 - 90°	0	0	-90°
5	Θ_5	L4 + L5	0	0

3.3.2 Modelo cinemático inverso

El objetivo de hallar un modelo cinemático inverso es el de darle referencias articulares a todos los motores q= [q1 q2 q3 q4] a partir de considerar como entrada coordenadas cartesianas. Con ello, sería posible el control de movimiento del extremo final del brazo para generación de futuras trayectorias comandadas desde la interfaz gráfica.

Para la resolución del problema de cinemática inversa se usarán métodos geométricos. Asumiendo tener los datos ya dados X, Y y Z.

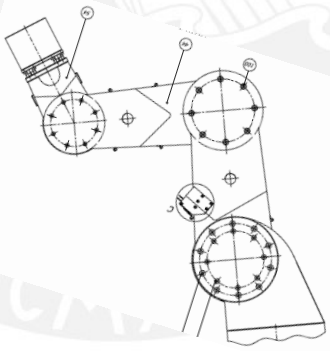


Ilustración 20 Gráfica de brazo robótico. Anexo 14

Ángulo1

El valor de q1 se obtiene de manera inmediata con la fórmula:

Ángulo1 =arctg(y/x)

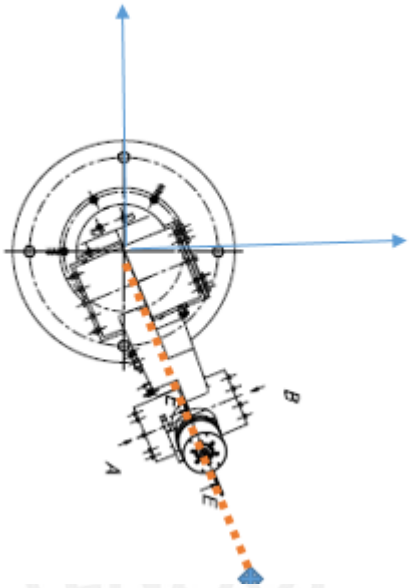


Ilustración 21 Cinemática inversa q1.Elaboración propia

Luego de definir el ángulo 1, el brazo se alinea en coordenadas X y Y con respecto al punto definido (ver ilustración 21).

Ángulo2

Para la simplificación del algoritmo de cinemática inversa de 5GLD se usará un algoritmo que determine el ángulo 2 antes de hallar los ángulos 3 ,4 y 5; con la idea de llevar un problema de inversa 5GLD a 3GLD.

Luego de que se halla el ángulo 2 en el cual el brazo este alineado con el punto en coordenadas X, Y; el problema de cinemática inversa se convierte en un problema de 2D como se muestra en la siguiente figura 22.

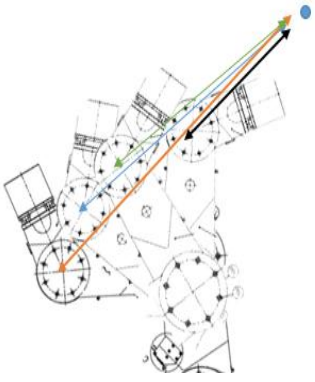


Ilustración 22 Cinemática inversa q2. Elaboración propia

Distancia en el caso que el ángulo 2 esté en 0 grados

Distancia en el caso que el ángulo 2 esté en 30 grados

Distancia en el caso que el ángulo 2 esté en 60 grados

Distancia en el caso que el ángulo 2 esté en 90 grados

Este algoritmo busca las posiciones que se acerquen más al punto deseado, hace un muestreo a partir de 20 grados hasta 120 grados en el eslabón 2 como se muestra en la figura 22. El criterio para la elección del ángulo 2 se basa en que la distancia entre el eje del motor 2 y el punto seleccionado sea menor a 56cm y sea mayor a 20cm.

Ángulos 3,4 y 5

Una vez seleccionado el ángulo 2, el punto final del eslabón 2 es tomado como el punto inicial del algoritmo de cinemática inversa de 3GLD como se muestra en la ilustración 23.

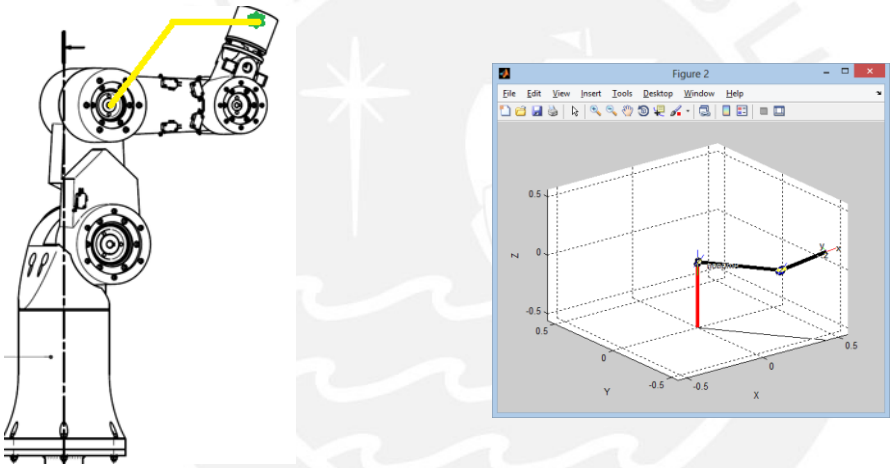


Ilustración 23 Cinemática inversa q3, q4 y q5.Elaboración propia

El algoritmo de 3GLD busca la configuración combinada de los eslabones 3,4 y 5 de tal manera que lleguen al punto final deseado [24], cuyo código se encuentra en el anexo 12.

3.4 Desarrollo del sistema de control

3.4.1 Modelo matemático del brazo robótico

Debido a que el diseño del sistema de control para el brazo robótico de cinco grados de libertad es de forma distribuida, el modelo a utilizar en cada controlador esclavo sería el de un motor DC cuyo modelo eléctrico se muestra en la ilustración 24.

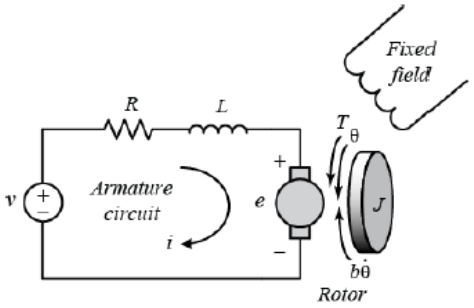


Ilustración 24 Diagrama del modelo de un motor DC. [10]

Para ello, se pretende hacer mediciones previas para llegar al modelo de cada motor DC a controlar. Realizando el análisis eléctrico del sistema se obtiene:

$$L \times \frac{di}{dt} + Ri + Ke \times \dot{\theta} = V(s)$$

Ecuación 1 Análisis eléctrico de motor DC

Realizando el análisis por leyes de newton se obtiene:

$$J_{planta} \times \ddot{\theta} + B \times \dot{\theta} + \tau_p = Kt \times i$$

Ecuación 2 Análisis por leyes de newton

Al realizar combinaciones algebraicas, eliminar la perturbación y aplicando transformada de Laplace en ambas ecuaciones se obtiene lo siguiente:

$$\frac{\theta(s)}{v(s)} = \frac{Kt}{[(La*s+Ra)(Ja*s+B)+Kt*Ke]*s}$$

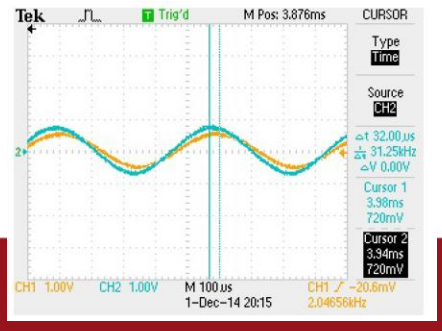
Ecuación 3 Función de transferencia de motor DC

3.4.1.1 Cálculo de resistencias e inductancias

A continuación, se muestran las constantes inherentes a los actuadores que se obtuvieron al realizar la medición de las resistencias de armadura de cada actuador del sistema con un multímetro Fluke 179 en modo ohmímetro. Se tomaron las constantes de torque y voltaje de las hojas de datos; dichos valores son numéricamente iguales. Los resultados en la primera prueba y hallados en hoja de datos se muestran en la tabla 4.

Tabla 4 Tabla de propiedades de motores

Señal medida en Rprueba
Señal medida en fuente



Actuador	Ra (Ohm)	Kt = Ke
M1	7.50	0.141
M2	14.80	0.257
M3	14.80	0.257
M4	3.80	0.031
M5	4.02	0.024

Ilustración 25 Ondas visualizadas en prueba de constantes eléctricas. Elaboración propia y [3]

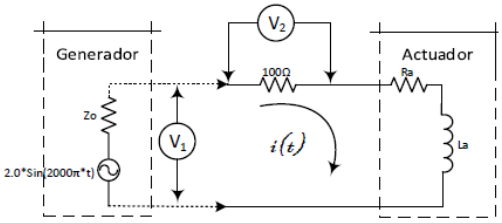


Ilustración 26 Prueba eléctrica. Elaboración propia y [13]

La prueba consistió en suministrar tensión alterna al motor en serie con una resistencia (Rprueba) conocida, la idea sería medir la corriente y el desfase entre voltaje y corriente para determinar la resistencia e inductancia del motor(ver ilustración 26).

$$\frac{v_1}{v_2/R_{prueba}} = \sqrt{(\omega La)^2 + (R_{prueba} + Ra)^2}$$

$$\phi = tan^{-1}\left(\frac{X_L}{R_{prueba} + Ra}\right)$$

Ecuación 4 Ecuaciones de prueba eléctrica

Tabla 5 Valores hallados en prueba eléctrica. Elaboración propia y [13]

Actuador	v_2/R_{prueba}	Ra (Ohm)	La (mH)
M1	$0{,}0116 * \text{sen}(4000\pi t + 21{,}70°)$	24.95	3.956
M2	$0{,}0112 * \text{sen}(4000\pi t + 31{,}49°)$	20.87	5.287
M3	$0{,}0111 * \text{sen}(4000\pi t + 31{,}51°)$	20.90	5.252
M4	$0{,}0130 * \text{sen}(4000\pi t + 43{,}86°)$	21.93	0.174
M5	$0{,}0145 * \text{sen}(4000\pi t + 18{,}08°)$	4.12	2.640

3.4.1.2 Cálculo coeficiente de rozamiento viscoso

Para el cálculo de este coeficiente se energizó el motor de tal manera que llegue a una velocidad constante, luego se mide la corriente consumida por este con un multímetro FLUKE 179 y la velocidad del motor con un tacómetro. Finalmente se despejaría la constante B utilizando como dato el Kt anteriormente encontrado,

tomando en cuenta el τp= 0 y la ecuación 5. Los resultados se muestran en la tabla 6:

Ecuación 5 Ecuación de prueba de rozamiento viscoso

$$J_{planta} \times \ddot{\theta} + B \times \dot{\theta} + \tau_p = Kt \times i$$

Tabla 6 Valores de Rozamiento viscoso hallados. Elaboración propia y [13]

Actuador	Velocidad (rad/s)	Corriente (A)	Rozamiento viscoso $(Nm.s/rad)$
M1	118.66	0.75	8.90×10^{-4}
M2	41.86	1.23	6.54×10^{-3}
M3	41.86	1.23	6.54×10^{-3}
M4	45.10	0.99	6.80×10^{-4}
M5	99.43	0.38	0.091×10^{-3}

3.4.1.3 Cálculo de momento de inercia referido a los actuadores

Para el análisis del momento de inercia de todo el sistema se asumió que todas las distancias son máximas; es decir, cuando los eslabones están todos estirados formando línea recta.

Ecuación 6 Ecuación inicial de inercia

$$Ja = \sum_{i=1}^{n} M_i \times r_i^2$$

El momento de inercia es calculado bajo la ecuación x, donde Ja es el momento de inercia respecto al eje de giro;M_i, las masas manipuladas por la articulación;ri, la distancia del eje de giro a la masa M_i y n la cantidad de masas [22]. Para facilitar el análisis, se asumió que los elementos concentran sus masas en su centro de gravedad para definir el momento de rotación en función de la masa y la distancia que se define entre el centro del eje y los centros de masa.

En la ecuación 7, m_{pload} es la masa manipulada por el actuador p para el caso en el que el sistema está desarticulado; m_{p+1}, la masa del actuador p+1;vm_{load}, la masa de la carga final que manipula el robot que puede ser como máximo 1.0Kg; r_{pcm}, la distancia al centro de la masa manipulada por el actuador p; L_{load}, la distancia a la carga m_{load} que manipula el robot y q, el índice que representa el último actuador manipulado por el actuador p en análisis.

$$J_p = m_{pload} * Cm^2 + m_{p+1} * L_{p+1}^2 + ... + m_{qload} * Cm^2 + m_{q+1} * L_q^2 + m_{load} * L_{load}^2$$

Ecuación 7 Cálculo de inercia de movimiento

Los datos a usar son:

Tabla 7 Datos hallados por prueba y de datasheet de los motores del brazo. Elaboración propia y [13]

Eslabón	PPR	Cte.AD	R (ohm)	L (mH)	Kt=Ke	B(Nm.s/rad)
Grado1	51	120	24.95	3.956	0.141	8.9*10^-4
Grado2	51	160	20.87	5.287	0.257	6.54*10^-3
Grado3	2048	120	20.9	5.252	0.257	6.54*10^-3
Grado4	1024	130	21.93	0.174	0.031	6.8*10^-4

Donde R= resistencia, L = inductancia, B= rozamiento viscoso y Cte. AD= constante de *armonic drive*.

Para M4:

$$J_4 = m_{5load} * (L5/2 + L4)^2 + m_5 * L4^2 + m_{4load} * (L4/2)^2 + m_{load} * (L5 + L4)^2$$

Para M3:

$$\begin{aligned} J_3 = \; & m_{5load} * (L5/2 + L4 + L3)^2 + m_5 * (L4 + L3)^2 + m_{4load} * (L4/2 + L3)^2 + \\ & m_{load} * (L5 + L4 + L3)^2 + m_4 * L3^2 + m_{3load} * (L3/2)^2 \end{aligned}$$

Para M2:

$$\begin{aligned} J_2 = \; & m_{5load} * (L5/2 + L4 + L3 + L2)^2 + m_5 * (L4 + L3 + L2)^2 \\ & + m_{4load} * (L4/2 + L3 + L2)^2 \\ & + m_{load} * (L5 + L4 + L3 + L2)^2 + m_4 * (L3 + L2)^2 \\ & + m_{3load} * (L3/2 + L2)^2 + m_3 * L2^2 + m_{2load} * (L2/2)^2 \end{aligned}$$

Para M1:

$$\begin{aligned} J1_{planta} = \; & m_{5load} * (L5/2 + L4 + L3 + L2)^2 + m_5 * (L4 + L3 + L2)^2 \\ & + m_{4load} * (L4/2 + L3 + L2)^2 + m_4 * (L3 + L2)^2 \\ & + m_{3load} * (L3/2 + L2)^2 + m_3 * L2^2 \\ & + m_{2load} * (L2/2)^2 + m_{load} * (L5 + L4 + L3 + L2)^2 \\ & + \frac{1}{2} m_{1load} * (r_{11}^2 + r_{12}^2) \end{aligned}$$

Para M5:

$$J5_{planta} = \frac{1}{2} m_{5load} * (r_{51}^2 + r_{52}^2) + m_{load} * L5^2$$

Tabla 8 Valores de inercia de movimiento hallados. Elaboración propia

Actuador	Jplanta (Kg - m^2)	Jreductor (Kg - m^2)	Jrotor (Kg - m^2)	Ja (Kg - m^2)
M1	3.953	$0.36x10^{-4}$	$0.423x10^{-4}$	$3.528\ x10^{-4}$
M2	3.913	$1.30\ x10^{-4}$	$0.704x10^{-4}$	$3.532\ x10^{-4}$
M3	1.235	$0.36x10^{-4}$	$0.704x10^{-4}$	$1.921\ x10^{-4}$
M4	0.207	$0.14x10^{-4}$	$0.0190x10^{-4}$	$0.281\ x10^{-4}$
M5	0.0106	$0.033\ x10^{-4}$	$0.0162x10^{-4}$	C

Donde las inercias resultantes Ja son:

$$Ja = \frac{Jplanta}{R\hat{}2} + Jreductor + Jmotor$$

Donde:

Jplanta= Inercia de movimiento de la planta. Calculado previamente.

Jreductor= Inercia de movimiento referenciado al reductor. Tomado de hoja de datos (anexo 18).

Jmotor= Inercia del motor desacoplado. Tomado de hoja de datos (anexos 4 al 9)

R= Constante de reductor

Ja1	Ja2	Ja3	Ja4	Ja5
3.528 x10-4	3.532 x10-4	1.921 x10-4	0.281 x10-4	0.281 x10-4

Al expandir el denominador de la ecuación 3 que corresponde a la función de transferencia; al ser multiplicado el valor de la inductancia por la inercia y el coeficiente de rozamiento viscoso se obtienen valores que son muy bajos en comparación al resto de componentes; por lo tanto, es factible despreciar la inductancia. En consecuencia, se obtiene el modelo simplificado mostrado en la ecuación.

$$\frac{\theta(s)}{V(s)} = \frac{Kt}{[R * Ja * s + R * B + Kt * Ke] * s}$$

Los modelos resultantes son:

M1 $$\frac{\theta(s)}{V(s)} = \frac{0.141}{0.00002463s + 0.00095}$$

M2 $$\frac{\theta(s)}{V(s)} = \frac{0.257}{0.000617341s + 0.01587}$$

M3 $$\frac{\theta(s)}{V(s)} = \frac{0.257}{0.004016225s + 0.2026}$$

M4 $$\frac{\theta(s)}{V(s)} = \frac{0.031}{0.00737236s + 0.2024}$$

Como siguiente paso, en la sintonización de los controladores se tomará en cuenta el modelo PD debido a que las funciones de transferencia de los motores presentan un polo en el origen o también llamado integrador.

3.4.2.1 Sintonización de lazo cerrado

Para la sintonización de los controladores se utilizó la herramienta de tuning de PID del programa de Matlab en el cual se tiene como dato el modelo de la planta del motor respectivo y las propiedades como el tiempo de establecimiento con el criterio de mínimo tiempo por cada motor. Cabe resaltar que al disminuir el tiempo de establecimiento y acelerar el sistema, aumentan las posibilidades de oscilaciones e inestabilidad del sistema. Por ello, se debe buscar además, un sobre impulso que sea menor del 5%.

Sintonización de controlador de M1

$$\frac{\theta(s)}{V(s)} = \frac{0.141}{0.00002463s + 0.00095}$$

Θ=1.57 rad

Ts=4.8s

Se utilizan los siguientes criterios:

$$\%\,\text{Sobre Impulso} = 100 \cdot e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}}$$

4.8s

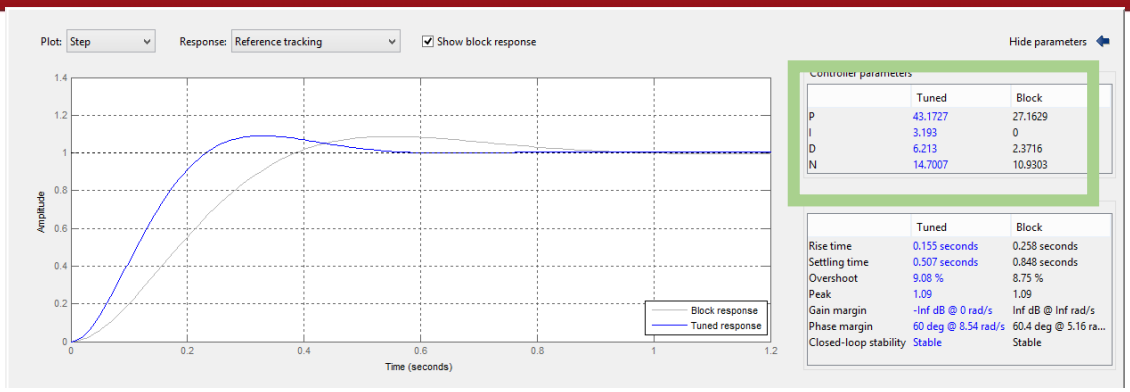


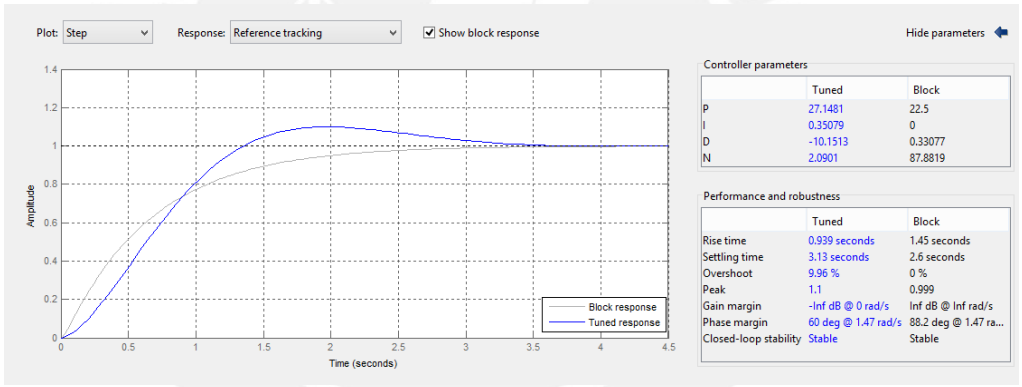
Ilustración 27 Sintonización de motor1. Elaboración propia

Sintonización de controlador de M2

$$\frac{\theta(s)}{V(s)} = \frac{0.257}{0.000617341s + 0.01587}$$

Θ= 0.7 rad

Ts=6.4s



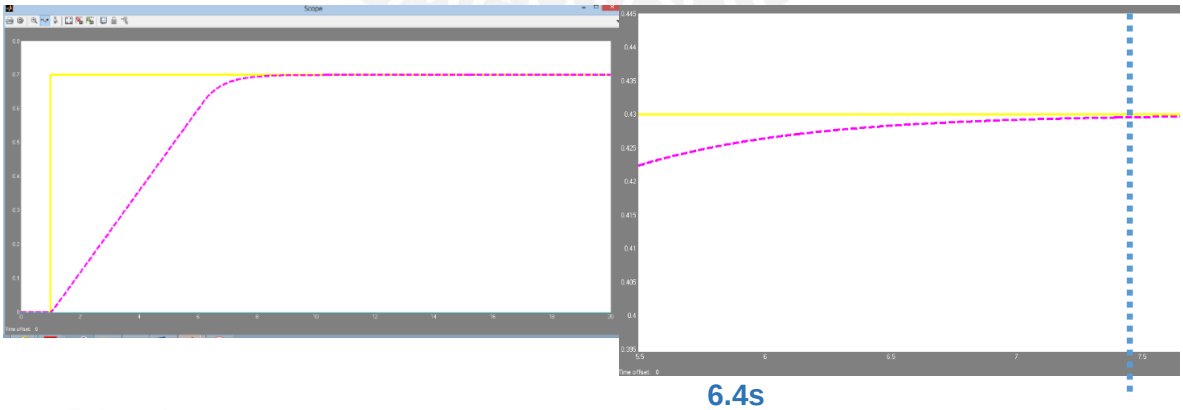


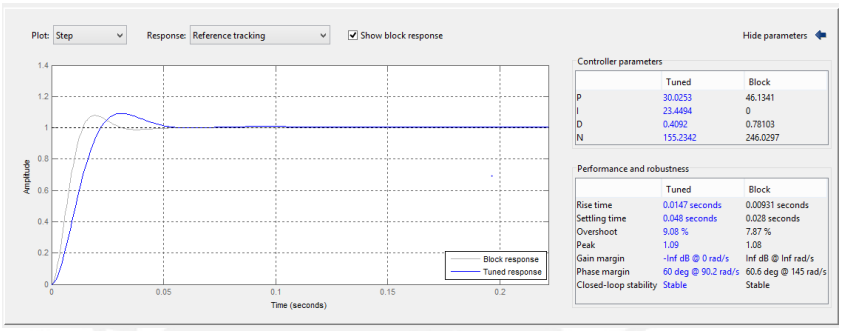
Ilustración 28 Sintonización de motor 2. Elaboración propia

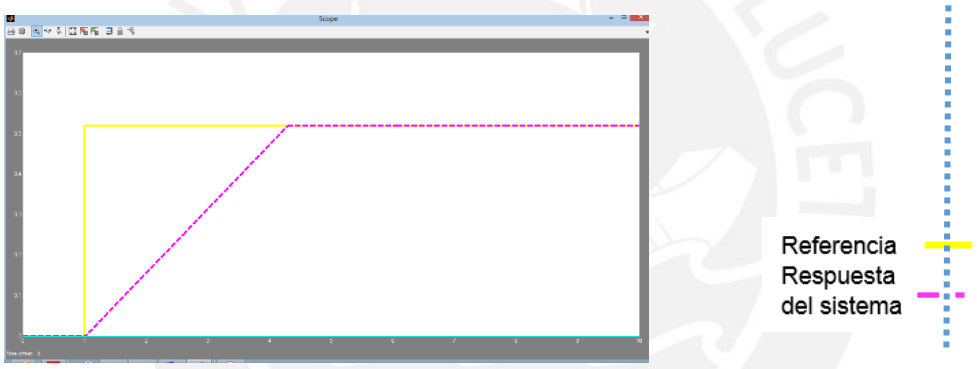
Sintonización de controlador de M3

$$\frac{\theta(s)}{V(s)} = \frac{0.257}{0.004016225s + 0.2026}$$

Θ=0.52 rad

Ts=4.3s





4.3s

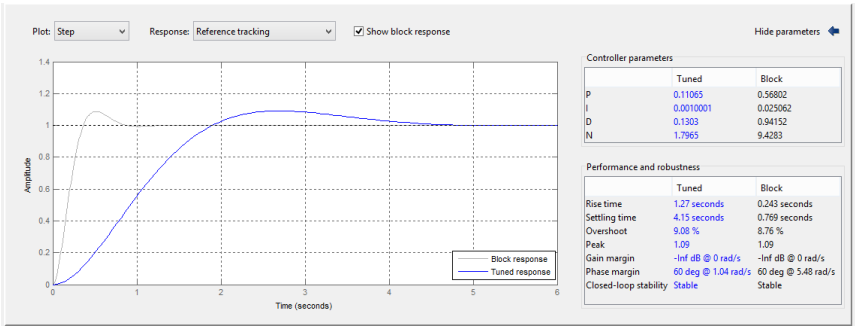
Ilustración 29 Sintonización de motor 3. Elaboración propia

Sintonización de controlador de M4

$$\frac{\theta(s)}{V(s)} = \frac{0.031}{0.00737236s + 0.2024}$$

Θ=1.22 rad

Ts=8s



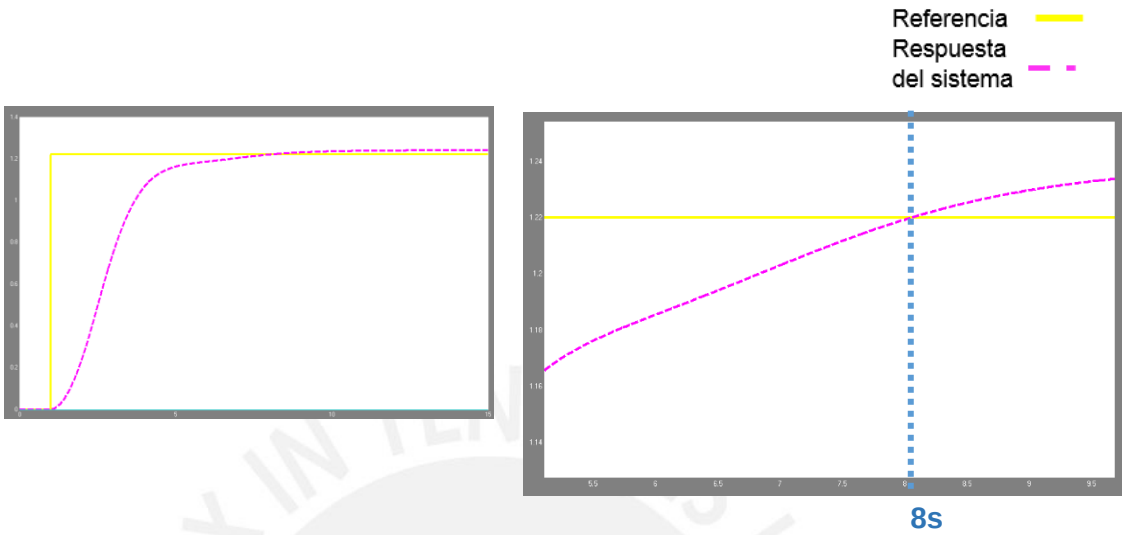


Ilustración 30 Sintonización de motor 4. Elaboración propia

Finalmente, la sintonización tiene los siguientes resultados que se muestran en la tabla 9:

Tabla 9 Valores PID hallados. Elaboración propia

Eslabón	Ángulo (sexagesimal)	Ángulo (rad)	Parámetro PID			
			P	D	PPR	R
Grado1	90	1.57	27	2.37	51	120
Grado2	40	0.7	22.5	0.33	51	160
Grado3	30	0.52	0.127	0.233	2048	120
Grado4	70	1.22	2.19	0.575	1024	130

donde PPR= pulsos por revolución y R constante de reducción.

La forma final del algoritmo PD continua a implementar es:

$$PID(s) = Kp + Kd * s$$

Al transformar el controlador PID(s) a control digital mediante la aproximación de Euler [18], se obtiene:

$$PID(z) = Kp + Kd * \frac{z-1}{Tz}$$

Luego al aplicar la transformada inversa Z, se puede establecer un algoritmo en función del tiempo y los parámetros PID hallados, que puede ser implementado en los 5 esclavos.

$$PID(k) = kp * e[k] + kd * \frac{1}{T} * (e[k] - e[k-1])$$

Ecuación 8 Función discreta PID

3.4.2 Adaptaciones del controlador PID

3.4.2.2 Back calculation Anti Wind up

A pesar de hacer una sintonización que permite tener pequeñas oscilaciones en la salida, nuestro sistema presenta grandes oscilaciones pico en la salida del controlador cuando hay grandes variaciones de ángulos en la entrada de referencia. Ya que el sistema de control está constantemente expuesto a cambios bruscos de referencia por los cambios de ángulos exigidos a los motores, esto pone en riesgo el actuador o el driver del motor ya que este hardware tiene un voltaje límite de exigencia y con esta modificación se buscaría disminuir las ocasiones de saturación.

El esquema planteado es el de la ilustración 31:

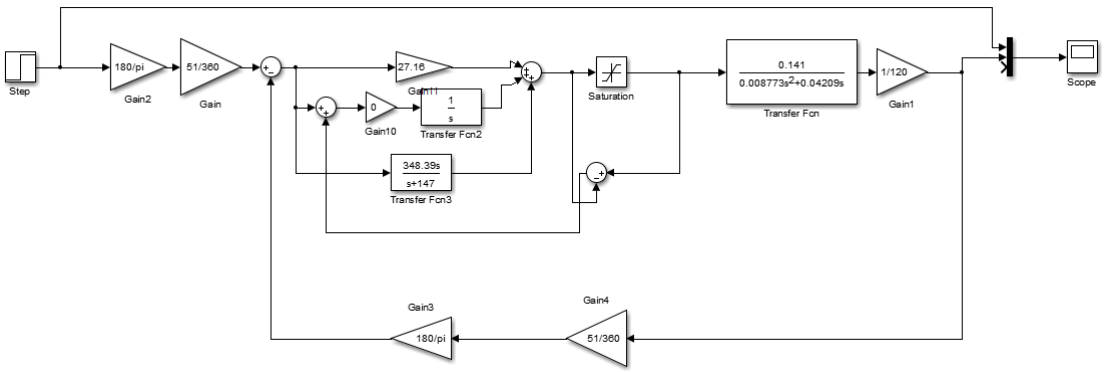


Ilustración 31 Esquema de simulación en simulink completo. Elaboración propia

El esquema *back calculation anti wind up* se puede definir como una realimentación suplementaria en el sistema que afecta directamente al integrador [20], la idea es incluir un integrador y un bloque de saturación para asegurar la estabilización del sistema. En el caso de nuestro modelo el controlador no incluye un integrador (Ki=0); por lo tanto, nuestro esquema se reduce a lo siguiente (ver ilustración 32):

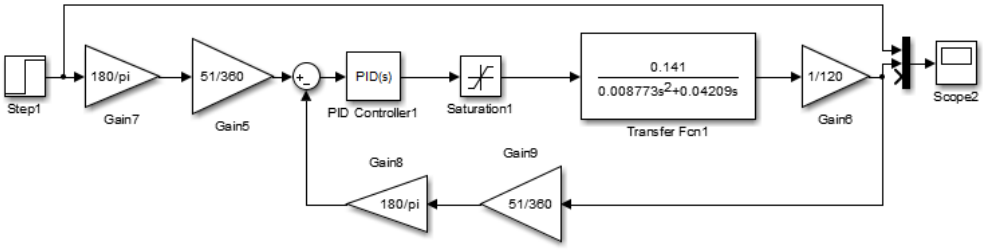


Ilustración 32 Esquema de simulación en simulink simplificado. Elaboración propia

3.4.2.3 Comportamiento del controlador y de la salida

Tomando en cuenta que nuestro controlador solo tiene la capacidad de suministrar 15 voltios por propiedades del driver, se estaría trabajando con un sistema con un comportamiento más lento comparado a lo hallado teóricamente. A partir de esta premisa, el planteamiento de sintonización con el *toolkit* sería el siguiente en la ilustración 33:

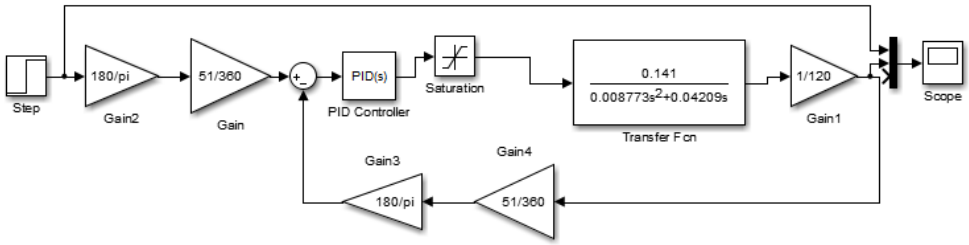


Ilustración 33 Prueba de controlador. Elaboración propia

En la imagen 34, se puede apreciar un bloque de saturación que se encuentra después del controlador, este bloque representaría la limitación física del driver que está implementado en el sistema de control distribuido en los 5 motores.

Luego se planteó el siguiente esquema para estudiar el comportamiento de los controladores cuando hay el bloque de saturación y cuando no.

Para el caso del motor1, aplicándole un escalón de entrada que representa un comando de cambio de ángulo. En la siguiente figura, se puede apreciar 2 sistemas idénticos pero con la variación de que uno tiene el bloque de saturación y otro no, esto con la finalidad de comparar los comportamientos de la salida de los controladores.

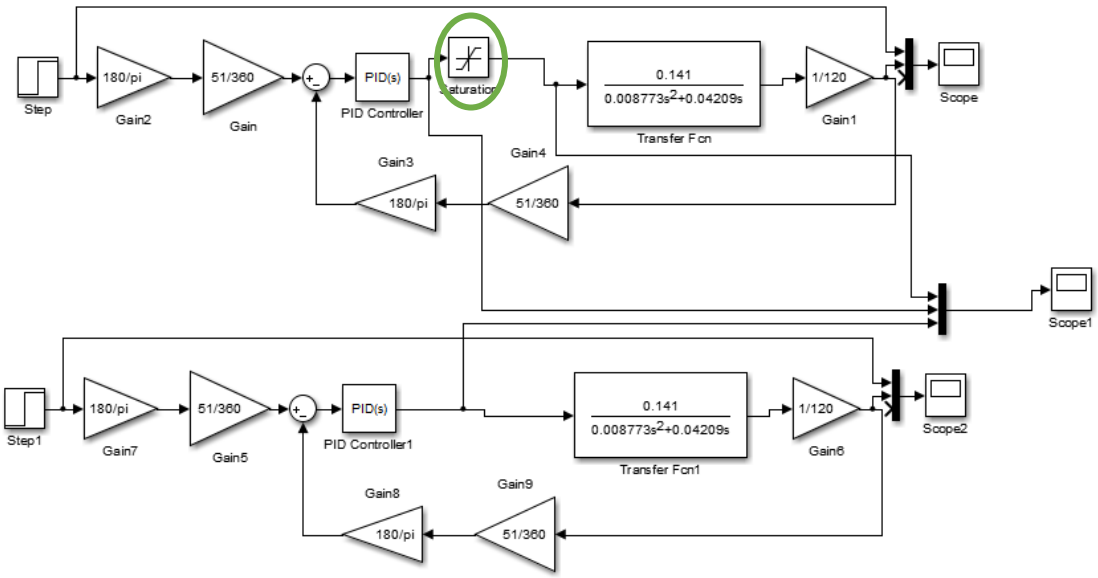


Ilustración 34 Comparación de controladores. Elaboración propia

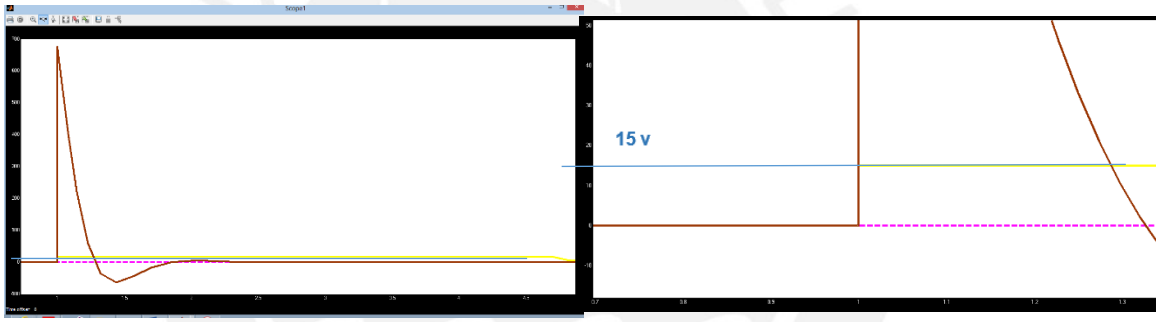


Ilustración 35 Curvas de simulación de controladores. Elaboración propia

En la figura 35, se pude notar las 2 señales representando los diferentes comportamientos de los sistemas con saturación y sin saturación. La señal de color rojo tiene un comportamiento más abrupto y rápido porque no tiene ningún bloque de saturación, lo cual permite que el sistema tenga un tiempo de establecimiento menor; sin embargo, podría dañar el hardware que trabaja con tensión nominal de 15v. Por otro lado, tomando en cuenta el hardware implementado, nuestro sistema tiene un comportamiento con un tiempo de establecimiento mucho mayor a pesar de la sintonización previa para que el tiempo sea el menor posible. La señal amarilla representa la salida del controlador que se estaría presentando en el actuador.

Cabe resaltar que a pesar de que la sintonización del PID no abarca el modelo que representaría la saturación, el hecho de que los parámetros estén ajustados para que el sistema actúe de manera más rápida y robusta hace que nuestro sistema tenga una respuesta rápida a pesar de la saturación de salida del controlador.

En la figura 36 se puede apreciar el comportamiento de la salida de nuestro sistema comparando con la secuencia de ángulos enviados desde la interfaz, con el fin de efectuar trayectorias con un tiempo de muestreo de 2.5 segundos.

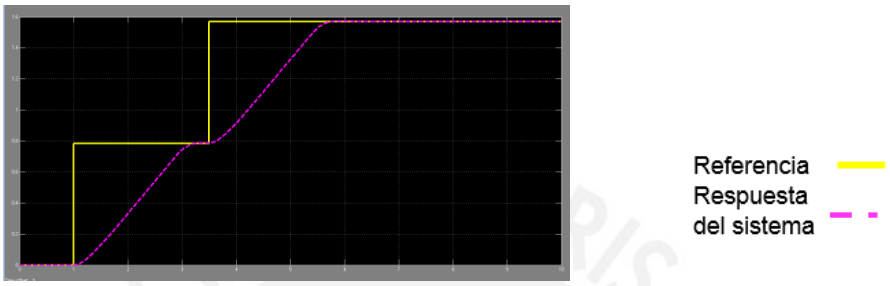


Ilustración 36 Prueba de 2 puntos de trayectoria

En la figura 37, se muestran 2 tipos de comportamiento del controlador a la hora de realizar una trayectoria con 3 puntos de trayectoria; es decir, con 3 barridos de ángulos.

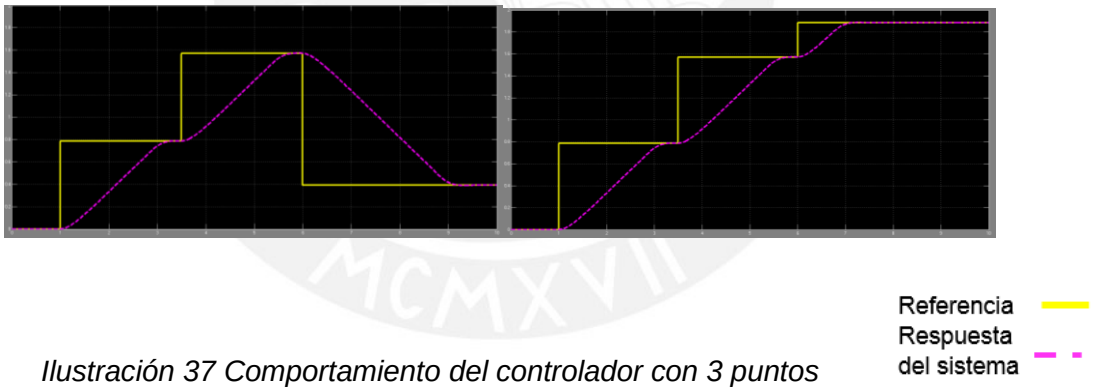


Ilustración 37 Comportamiento del controlador con 3 puntos

Se puede concluir que con el tiempo de muestreo en la interfaz de 2.5, no hay problema en el comportamiento del sistema para barridos de referencias que puede llegar de la interfaz por la generación de trayectorias.

3.4.2.4 Tiempo de muestreo

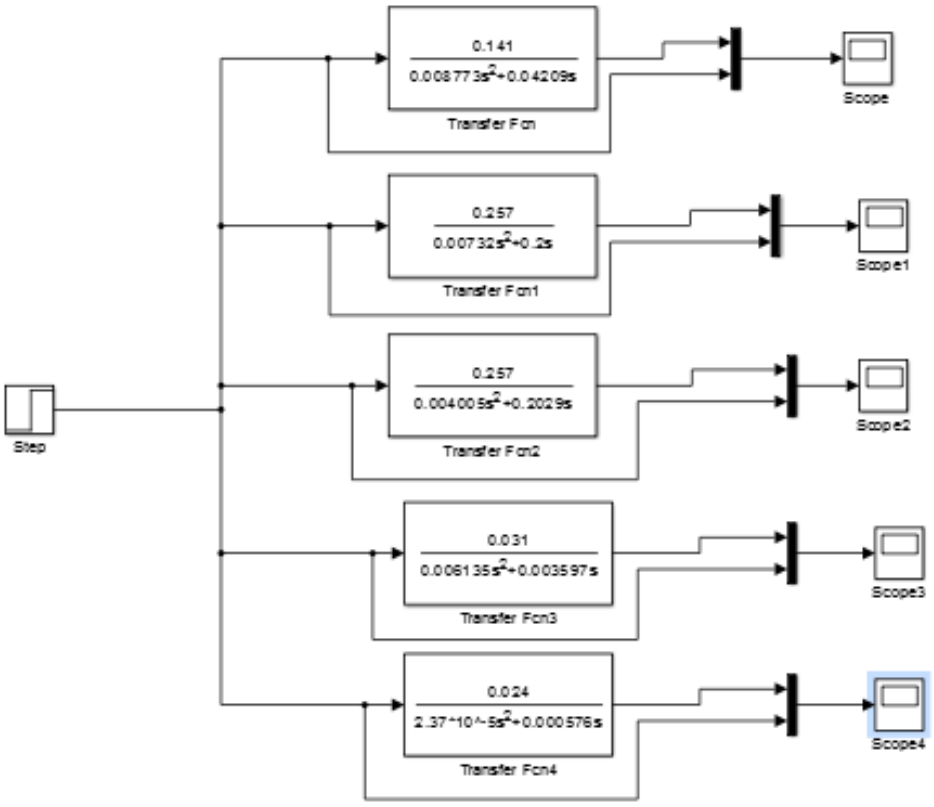


Ilustración 38 Pruebas de planta en lazo abierto. Elaboración propia

Para la estimación del tiempo de muestreo se aplicó a las 5 plantas en lazo abierto un escalón unitario (ver ilustración 38), con el fin de estimar el tiempo de respuesta Tr, en la imagen 39 se puede apreciar la respuesta en lazo abierto del motor 5.

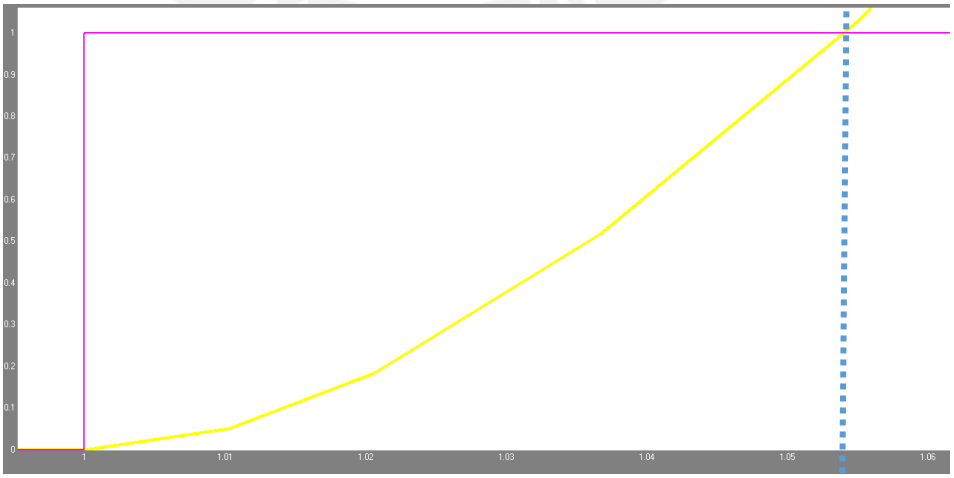


Ilustración 39 Curva de simulación de motor 5 en lazo abierto. Elaboración propia

Luego de hallar el tiempo de respuesta menor entre los 5 motores del sistema (54ms), se hace una aproximación de T=Tr/10 [18], mediante un cálculo se halla que T= 5.4ms y luego para hacer un muestreo más preciso se redondea a 5ms.

3.3.3 Comunicación interfaz-5 esclavos (prueba error comunicación y trama RS232 -I2C)

3.4.3.1 Comunicación RS232

Ilustración 40 Comunicación serial. Elaboración propia

-Baudrate: 9600

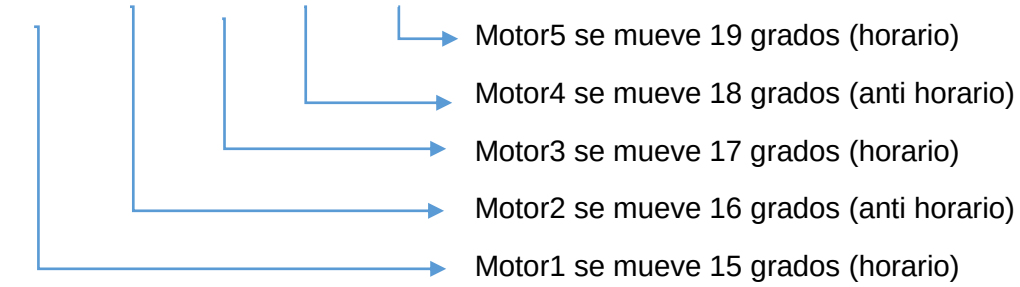
-No bit stop

Para esta prueba, se realizaron comandos desde la interfaz y analizados con un osciloscopio para determinar el error rate que podría haber en la comunicación RS232 entre el circuito maestro y la PC.

La trama empleada depende de diferentes factores como: el ángulo, el sentido y la dirección del esclavo donde se está enviando el comando.

Ejemplo:

015+@016-@017+@018-@019+@,



3.4.3.2 Comunicación I2C

El I2C es un protocolo que se utiliza por excelencia en la comunicación entre microcontroladores. En nuestro caso, permite la comunicación entre los 5 esclavos

con la tarjeta maestra con el fin de enviarle las referencias para los sistemas de lazo cerrado como se muestra en la figura 41.

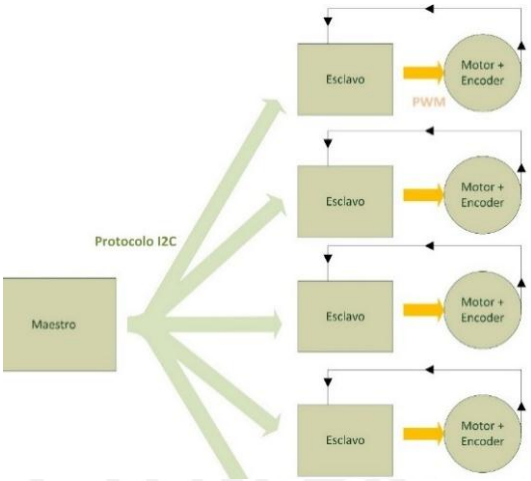


Ilustración 41 Esquema de comunicación maestro esclavo

La trama de comunicación sería la siguiente:

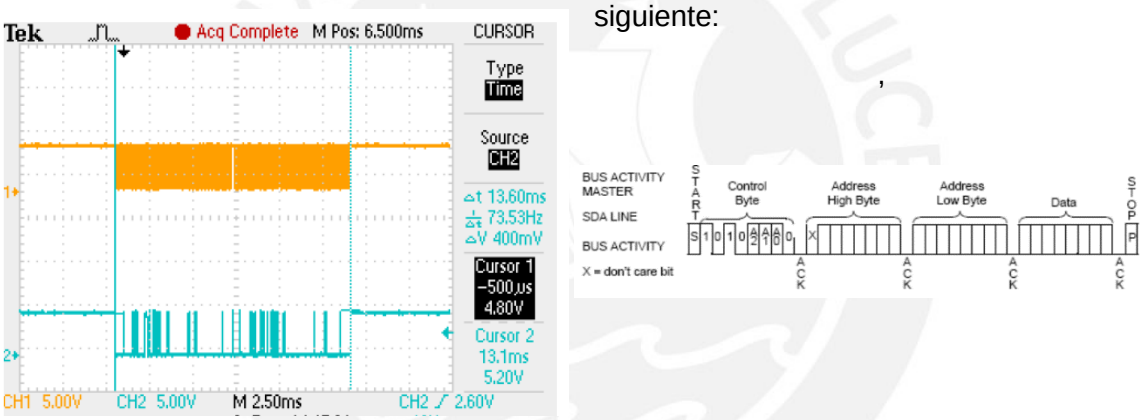


Ilustración 42 Trama de comunicación I2C. Elaboración propia y [13]

3.4.4 Desarrollo del control del motor con los 5 esclavos (diagrama de flujo)

3.4.4.1 Maestro

El maestro se encarga de repartir la información de ángulos y sentido al esclavo respectivo, ya que la interfaz envía la información por tramas de 5 motores. En el anexo 23 se detalla el código del microcontrolador.

- Se inicializan los puertos, I2C, interrupciones y comunicación serial.
- El maestro espera que le lleguen 25 caracteres que representan los ángulos y los sentidos de los 5 motores.
- El maestro enciende un LED indicando que la trama llegó correctamente.

- El maestro direcciona la comunicación I2C para que los datos de ángulo y sentido lleguen al esclavo respectivo.
- Espera que le llegue la siguiente trama.

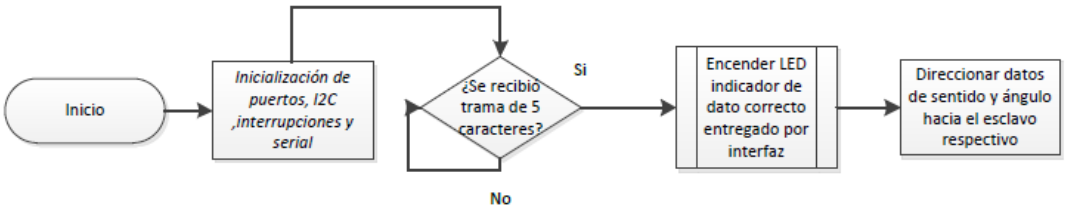


Ilustración 43 Diagrama de flujo de controlador maestro Elaboración propia

3.4.4.2 Esclavos

- Inicialización de puertos, comunicación I2C, interrupciones y PWM.
- El esclavo espera que le lleguen los 5 caracteres desde la comunicación I2C.
- El esclavo separa los datos de sentido y ángulo del esclavo respectivo.
- El esclavo determina la cantidad de pulsos a partir del factor de reducción y los pulsos por revolución que tiene del encoder instalado en el motor.
- Se ejecuta el algoritmo de PID sintonizado previamente con lo cual se determina a cada instante el valor de PWM adecuado para hacer que el motor gire lo necesario. Este algoritmo trabaja a partir de la señal de error que se actualiza con las interrupciones por pulsos del encoder.
- Cuando el número de pulsos asignado es igual al número de pulsos contados, el movimiento se detiene de tal manera que el motor queda en la posición deseada.
- Espera que le llegue el siguiente grupo de datos.

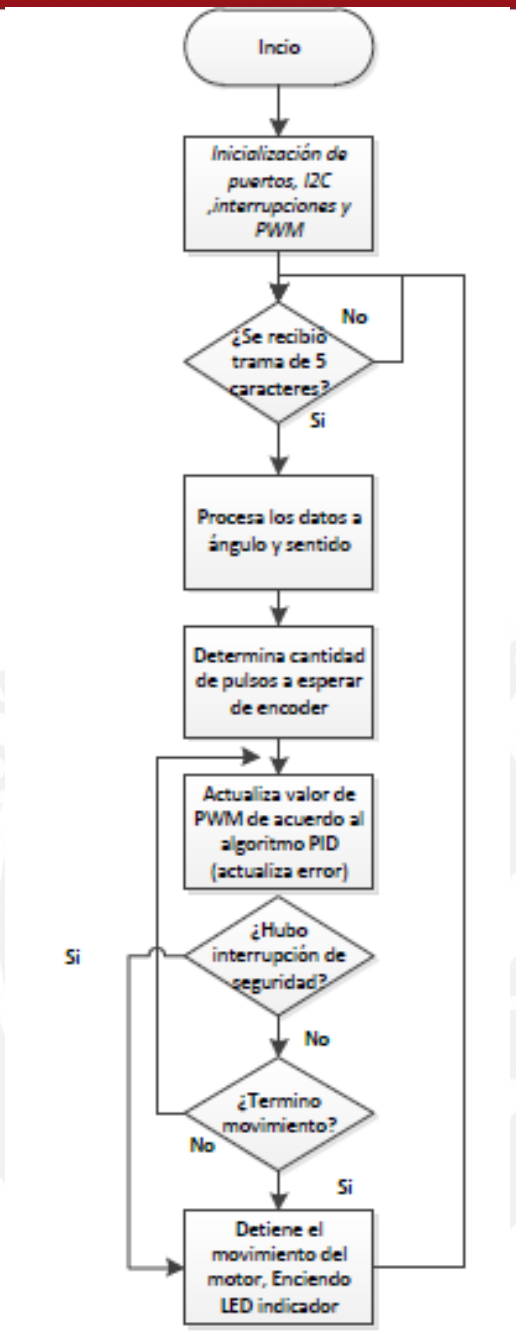


Ilustración 44 Diagrama de flujo de controlador esclavo. Elaboración propia

Caso de seguridad:

En el caso que se produzca un evento de interrupción de seguridad, por ejemplo en caso de sobrecorriente y/o pulso de fin de carrera, el motor debe detenerse inmediatamente.

3.5 Generador de trayectorias (interfaz)

Luego de haber diseñado el sistema de control se necesitó un intermediario entre el usuario en el sistema que incluso genere las referencias para las trayectorias que se desea generar. Por ello, se implementó un software que permite la generación de referencias dependiendo del punto final definido del usuario, esto basado en los modelos cinemáticos antes hallados. Además, permite la observación de valores en tiempo real de las referencias generadas y estas son graficadas para ver la forma de la trayectoria. Los códigos se detallan en los anexos 25,26 y 27.

Pantalla principal

La pantalla principal cuenta con diversas opciones con el objetivo de realizar movimientos del brazo robótico de manera individual o simultánea formando trayectorias.

La operación de la pantalla principal se puede dividir en 3 grupos principales:

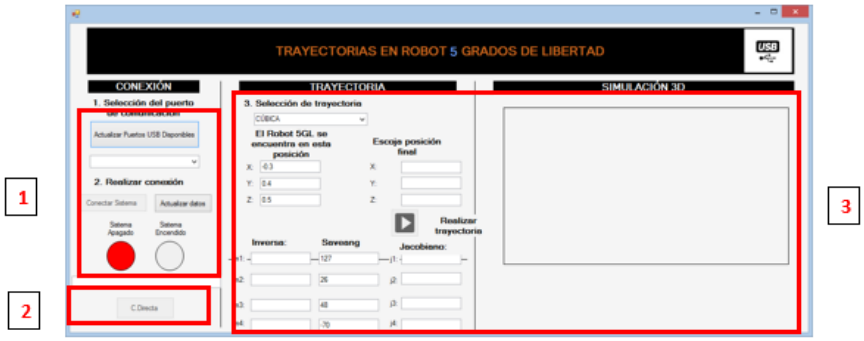


Ilustración 45 Partes pantalla principal. Elaboración propia

3.5.1 Inicialización de sistema

Al comenzar el uso del ejecutable de la aplicación de generador de trayectorias, el siguiente paso a tomar es la de inicializar el puerto serial. Para ello es necesario elegir el puerto de la barra después de apretar el botón “Actualizar puertos USB disponibles” (ver ilustración 46).

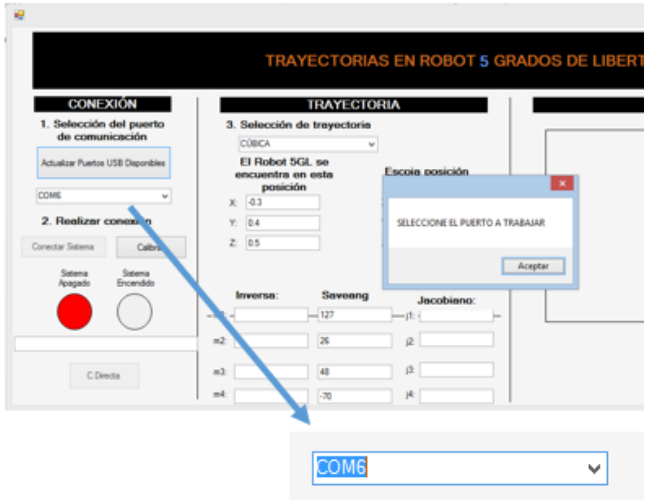


Ilustración 46 Inicialización de puerto. Elaboración propia.

Luego se procede a apretar el botón “Conectar sistema”. Inmediatamente el sistema mostrará la posición y los ángulos finales del brazo en un uso previo como en la ilustración 47.

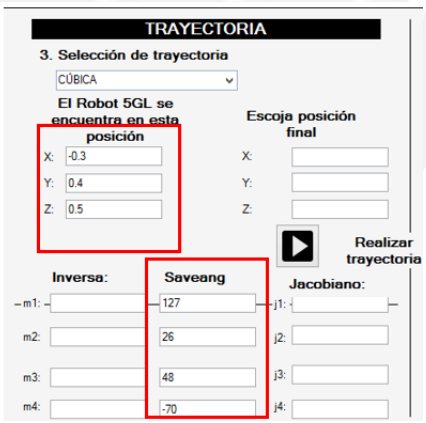


Ilustración 47 Sistema inicializado. Elaboración propia

Donde:

Saveang: ángulos de cada motor en tiempo real

En el caso que el brazo haya tenido algún movimiento inesperado por alguna razón externa o falla de energía. La opción “actualizar datos” permite registrar los datos de ángulos medidos previamente, y actualizarlos al sistema para asegurar que la

interfaz y el brazo están trabajando con las mismas coordenadas angulares o cartesianas.

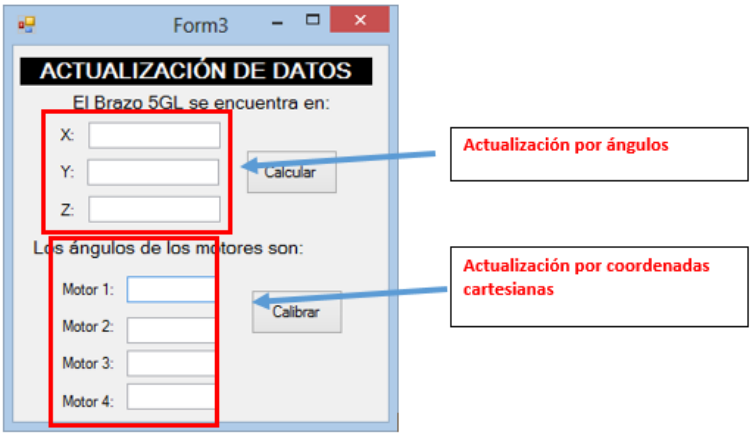


Ilustración 48 Actualización de datos. Elaboración propia

En esta ventana es posible colocar los valores reales de los ángulos de cada motor o agregar los valores X, Y, Z reales para que el sistema los guarde y actualice para correcta operación a futuro.

3.5.2. Cinemática directa

Para la realización de movimientos independientes de cada motor, se puede utilizar el botón “C.Directa” de la pantalla inicial. En esta pantalla se puede ingresar los ángulos finales de cada motor; es decir, el sistema indica al controlador esclavo cuánto le falta moverse al eslabón para llegar al ángulo deseado por el usuario.

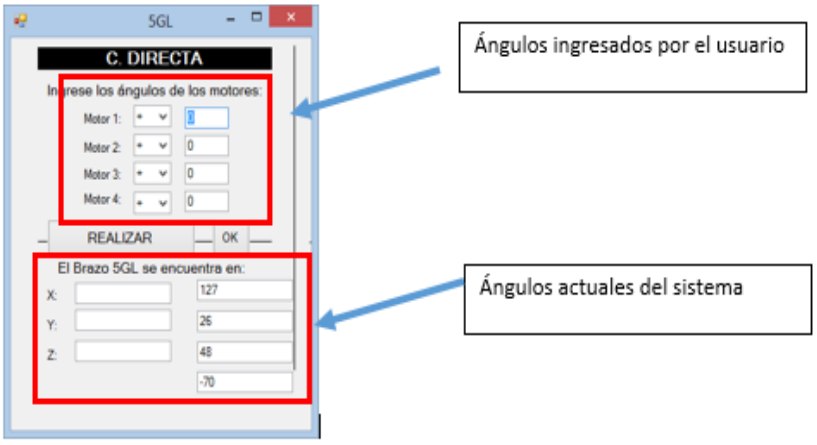


Ilustración 49 Cinemática directa. Elaboración propia

3.5.3 Generación de trayectorias

Esta es la herramienta más potente de la interfaz gráfica, ya que con ella el usuario define la posición final X, Y, Z del eslabón final del brazo. Además de realizar el movimiento, el sistema actualiza durante el movimiento los valores de ángulos, y los grafica en tiempo real como se muestra en la ilustración 50.

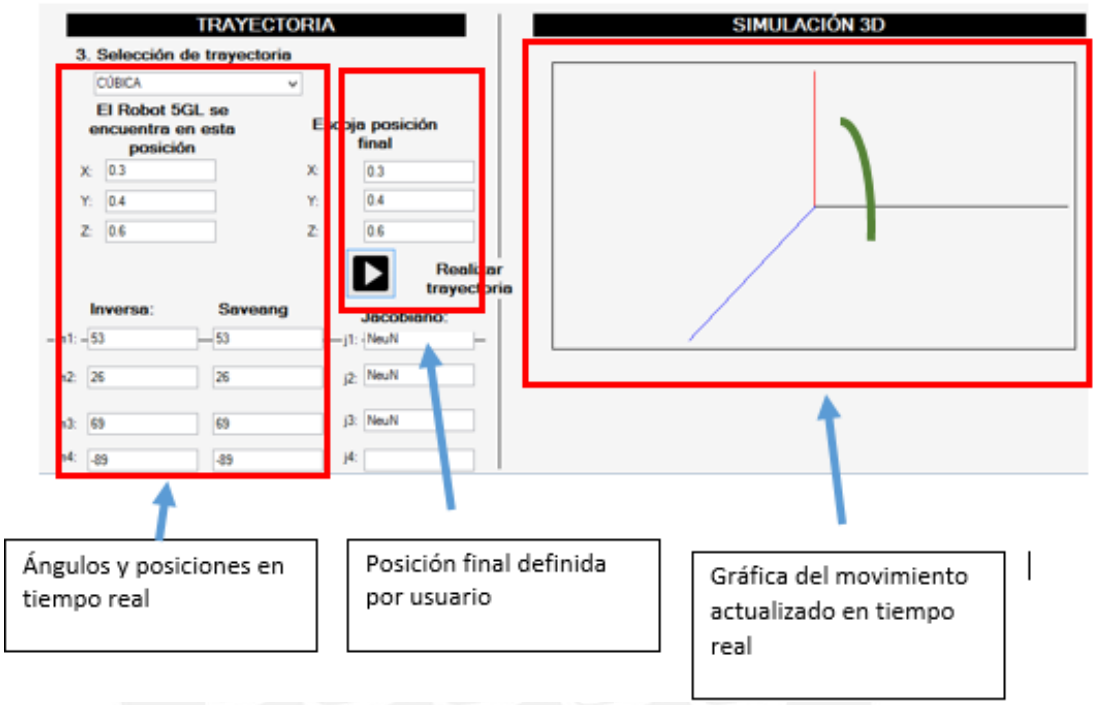


Ilustración 50 Generador de trayectorias. Elaboración propia

CAPÍTULO 4: SIMULACIONES Y PRUEBAS FINALES

En el presente capítulo, se detalla las pruebas realizadas a diferentes partes de nuestro sistema. En primer lugar, se hace simulación a los programas implementados a los microcontroladores maestro y esclavo para comprobar el buen funcionamiento a nivel de comunicación y de lectura de tramas; además, se hizo pruebas de los algoritmos de cinemática directa e inversa implementados en la interfaz para analizar sus errores. En segundo lugar, con el sistema implementado a nivel de hardware y software se hicieron pruebas de envío de trayectorias y su posterior medición de errores en coordenadas cartesianas. Finalmente, se hizo la prueba de tiempos a nivel de todo el sistema desde la interfaz hasta el movimiento de cada eslabón.

4.1 Simulación en software del sistema de control distribuido (Proteus – Matlab algoritmo de cinemática directa e inversa)

Simulación funcionamiento de maestro

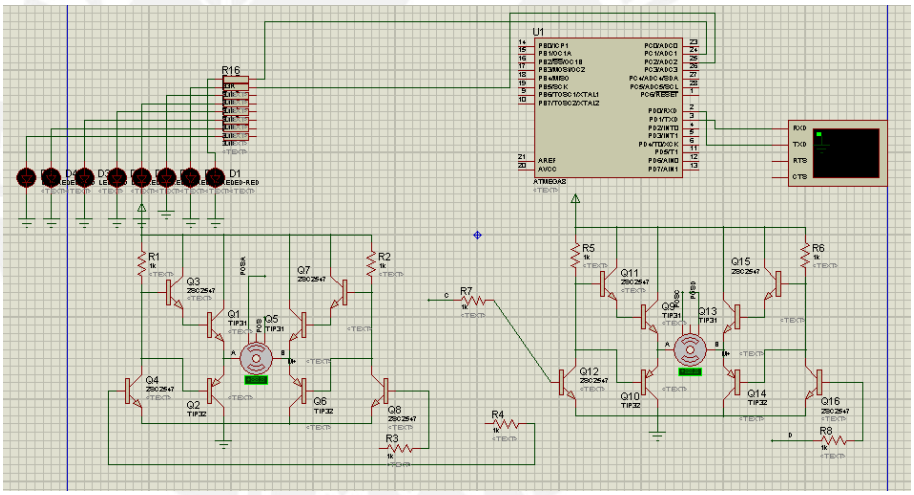


Ilustración 51 Simulación en PROTEUS del maestro. Elaboración propia

En la simulación implementada en Proteus, se probó el funcionamiento de la comunicación serial entre la interfaz y el maestro en el envío de referencias. El maestro debe poder mostrar la información numérica mandada desde la interfaz gráfica del programa Proteus en los LED (ver ilustración 51).

Simulación de funcionamiento de comunicación interfaz-maestro-esclavo

En la segunda simulación en PROTEUS, se implementó 1 microcontrolador maestro y otro esclavo, la idea es enviar por protocolo serial ángulos de prueba al maestro y luego que el microcontrolador maestro envíe la información al esclavo por protocolo I2C. Con ello, se estaría comprobando el buen funcionamiento del código de los microcontroladores y de los protocolos de comunicación.

En la imagen 52, se aprecia el terminal serial y el monitoreo de I2C que se utiliza para poder simular la comunicación esclavo, maestro e interfaz.

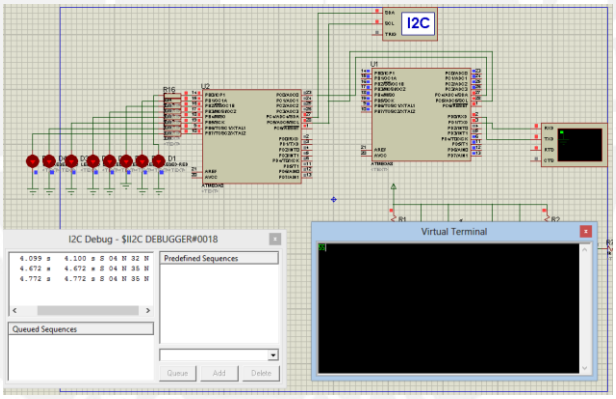


Ilustración 52 Simulación en PROTEUS del esclavo. Elaboración propia

Cinemática directa

Simulación de algoritmo de cinemática directa e inversa en MATLAB

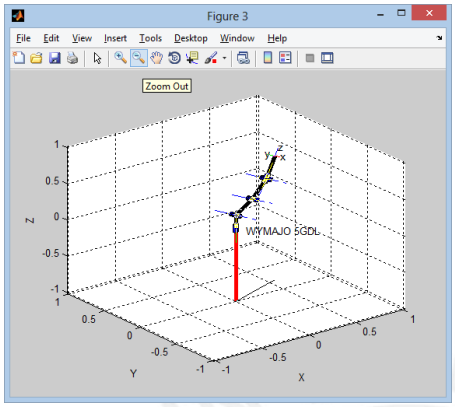
Cinemática directa: En la siguiente simulación se colocan los valores de los ángulos de cada uno de los motores con la idea de comparar el resultado del algoritmo de Denavit-Hartenberg que será implementada en la interfaz y la simulación del robot *wymajo* cuyo código está en el anexo 12 y utilizando el *toolbox* de robótica Matlab.

Ejemplo: Se define los siguientes ángulos para cada motor

M1	M2	M3	M4
20 grados/0.349 rad	20 grados/0.349 rad	20 grados/0.349 rad	20 grados/0.349 rad

Se procede a ejecutar el simulador de cálculos de cinemática inversa y directa (archivo pruebainversa5GLDv2.m).

Luego se procedió a usar el simulador WYMAJO para comparar si la respuesta del algoritmo de cinemática directa es correcta (ver



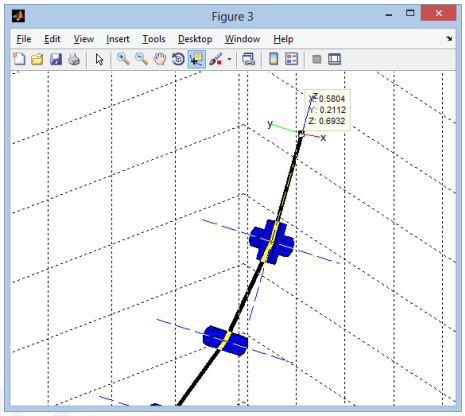
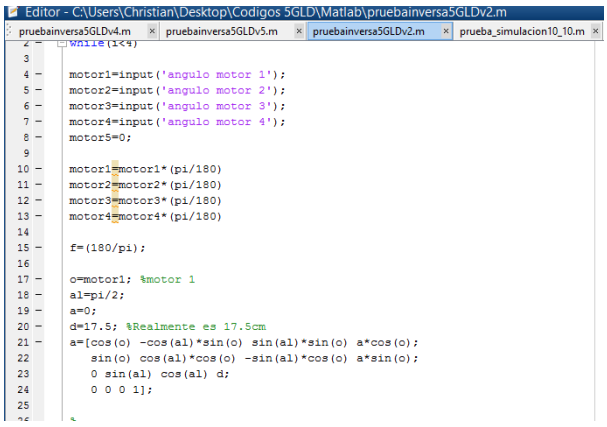


ilustración 53).

Ilustración 53 Prueba de cinemática directa con Toolbox de robótica. Elaboración propia

Las coordenadas finales son: x=0.58, y=0.2112 y z= 0.69 (unidades en metros) y son comparadas como se muestra en la ilustración 54.



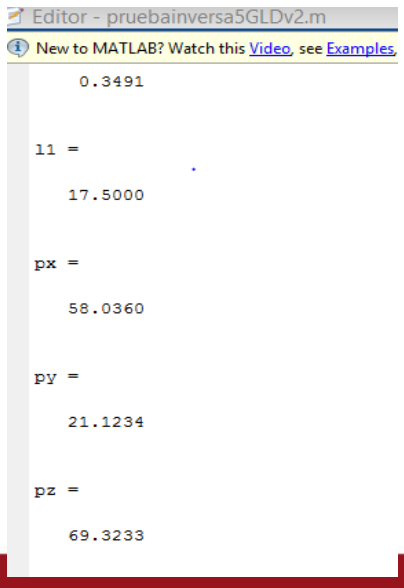


Ilustración 54 Prueba de cinemática directa con algoritmo de interfaz. Elaboración propia

Se puede concluir que el algoritmo está ejecutándose de manera correcta ya que las coordenadas finales corresponden una con la otra.

Cinemática Inversa

Al comprobar el correcto funcionamiento del algoritmo de cinemática directa, este es utilizado para implementar un programa que permita comprobar el grado de error que tiene la cinemática inversa (ver ilustración 55).

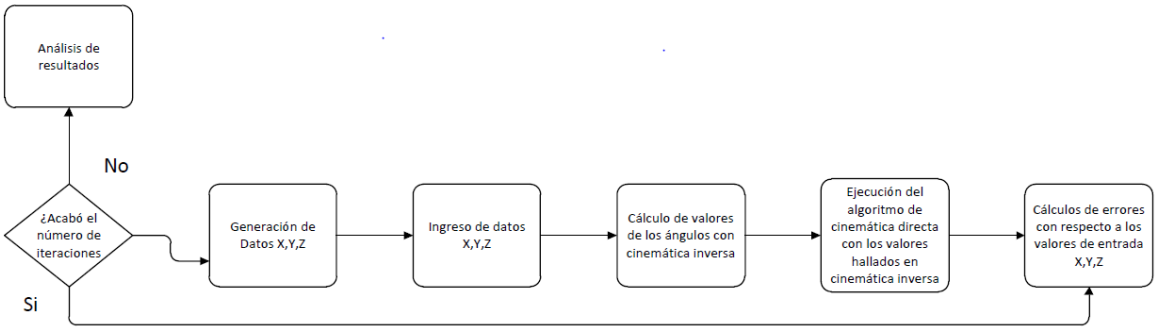


Ilustración 55 Diagrama de flujo y código de cinemática inversa. Elaboración propia

Resultados:

Se programó para que la simulación del algoritmo de cinemática inversa calcule un millón de puntos del volumen de trabajo del brazo robótico; con esta información se determinó la cantidad de errores posibles que podría tener el algoritmo en imprecisión o posible saturación del programa como se muestra en la ilustración 55.1.

Errorcount: determina el número de iteraciones en las cuales el error en los 3 ejes supere 5cm.

j : determina el número de iteraciones en las cuales el algoritmo no pueda encontrar los valores de los ángulos.

Count: número de iteraciones.

```
>> pruebainversa5GLDv5

errorcount =

        1079


j =

     0


count =

     1000000

>>
```

Ilustración 55.1 Resultado programa de prueba de cinemática inversa

Análisis de resultados

Considerando el número de iteraciones o de posibilidades del área de trabajo procesadas con el programa, el error es calculado por 1079/100000*100%= 0.1079% de error. Además, considerando el número elevado de pruebas, se puede concluir que es un algoritmo robusto ya que no presenta problemas de error en el volumen de trabajo que es un radio mayor a 25cm respecto a la base en el plano X, Y(círculos azules).Por otro lado, en los puntos cercanos a la base (menor a 25cm en X,Y) los errores son mayores a 5cm (equis rojas), esto se puede apreciar en la ilustración 56 y se detalla en el Anexo 12.

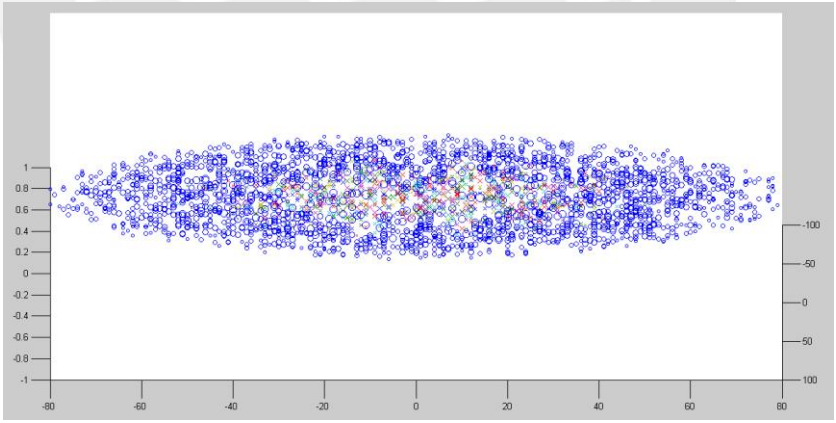


Ilustración 56 Resultados de algoritmo de cinemática inversa. Elaboración propia

4.2 Comunicación interfaz-maestro y control de esclavos

Para la prueba de comunicación y control, se consideró hacer las pruebas individuales de cada motor con la idea de determinar no solo si la comunicación es correcta con cada uno de los esclavos, sino también de determinar el error máximo que puede haber en cada uno de los motores al hacer un movimiento. Los detalles de los ensayos individuales se detallan en el anexo 13.

Prueba de implementación motor 1

Para el ejemplo se considerará un giro de 90 grados respecto al ángulo 0 donde es su posición final.

Posición inicial en 0 grados Posición final en 90 grados

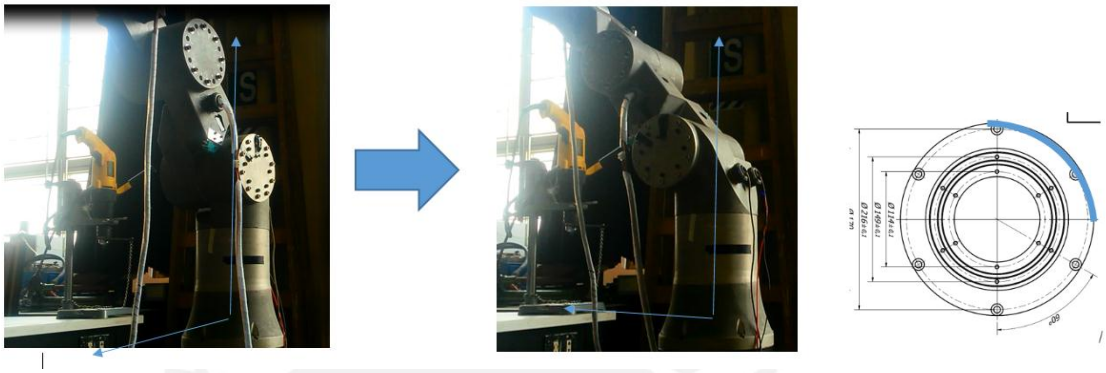


Ilustración 57 Pruebas de implementación de motor 1.Elaboración propia

La medición de grados se hizo midiendo el arco del grado 1 (ver ilustración 57). Por ejemplo, tomando el radio 10.8cm el arco a medir desde la posición inicial debe ser 2*(pi/2)*(10.8)=33.92 cm.

Luego de la realización de 10 repeticiones del movimiento se tomaron los siguientes datos:

Error máximo hallado: 4 grados.

Tiempo de establecimiento: 4 segundos.

Prueba teórica motor 1

$$\frac{0.141}{[0.008773s + 0.04209]s}$$

Datos extraídos con 90 grados:

Tiempo de establecimiento: 4.5 segundos

Overshoot= 0%

Prueba de implementación motor 2

Para el ejemplo se considerará un giro a 50 grados respecto al ángulo 90 (ver ilustración 58).

El segundo caso corresponde a un arco de 40 grados de desplazamiento; es decir, (40/180*pi)*2*(6.8)= 9.48cm.

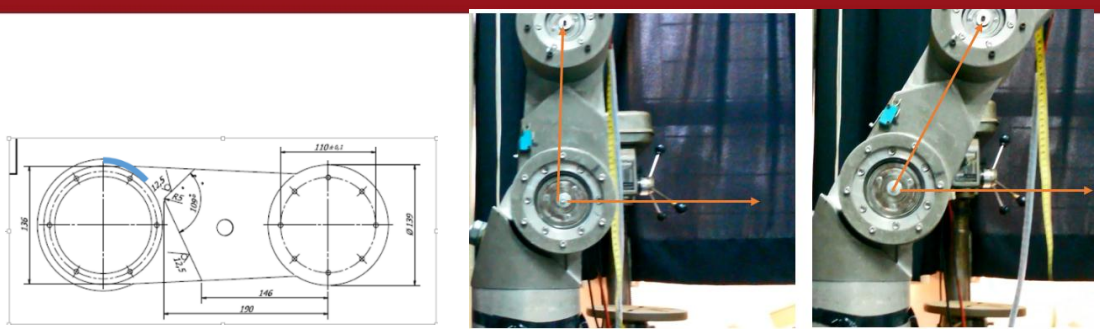


Ilustración 58 Pruebas de implementación de motor 2.Elaboración propia

Luego de la realización de 10 repeticiones del movimiento se tomaron los siguientes datos:

Error máximo hallado: 7 grados

Tiempo de establecimiento: 4.5 segundos

Prueba teórica motor 2

$$\frac{0.257}{[0.007361s + 0.2027]s}$$

Datos extraídos con 50 grados con 90 grados en inicio:

Tiempo de establecimiento: 6.8 segundos

Overshoot= 0%

Prueba de implementación motor 3

Para el ejemplo se considerará un giro a 40 grados respecto al ángulo 10 (Ver ilustración 59).

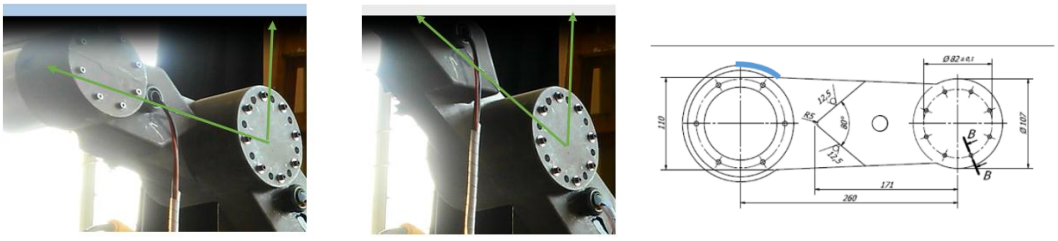


Ilustración 59 Pruebas de implementación de motor 3.Elaboración propia

El tercer caso corresponde a un arco de 30 grados de desplazamiento; es decir, (30/180*pi)*2*(5.5)= 5.28cm

Luego de la realización de 10 repeticiones del movimiento se tomaron los siguientes datos:

Error máximo hallado: 3 grados

Tiempo de establecimiento: 3 segundos

Prueba teórica motor 3:

$$\frac{0.257}{[0.004005s + 0.2029]s}$$

Datos extraídos con 40 grados en una posición inicial de 10 grados:

Tiempo de establecimiento: 4.2 segundos

Overshoot= 0%

Prueba de implementación motor 4

Para el ejemplo se considerará un giro a 70 grados respecto al ángulo 0 o inicial (ver ilustración 60).

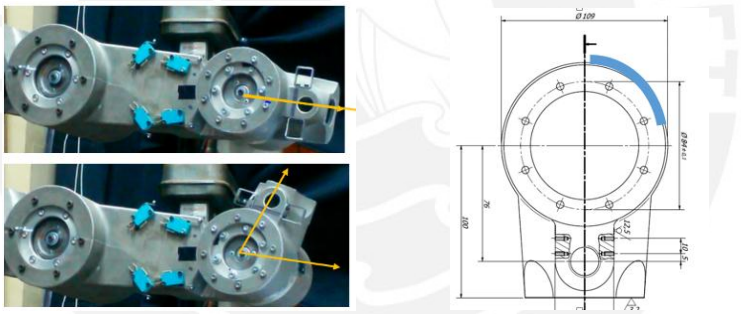


Ilustración 60 Pruebas de implementación de motor 4.Elaboración propia

El cuarto caso corresponde a un arco de 70 grados de desplazamiento; es decir, (70/180*pi)*2*(4.2)= 10.26cm

Luego de la realización de 10 repeticiones del movimiento se tomaron los siguientes datos:

Error máximo hallado: 4 grados

Tiempo de establecimiento: 2 segundos

Prueba teórica motor 4:

$$\frac{0.031}{[0.0006135s + 0.003597]s}$$

Datos extraídos con 70 grados:

Tiempo de establecimiento: 6.18 segundos

Overshoot= 4.2%

4.3 Implementación integrada (trayectoria)

Como pruebas finales, el sistema integrado desde la interfaz hasta los esclavos es probado para la realización de trayectorias. Para que la función del sistema sea correcto, deben funcionar las etapas de generación de trayectorias (interfaz), distribución de datos por parte del maestro y finalmente el funcionamiento correcto del sistema de lazo cerrado de cada esclavo de manera simultánea en los 4 motores implementados. Como procedimiento de cálculo de error, se mide en X, Y, Z el punto final del eslabón terminado la trayectoria; con ello, se determinaría el grado de exactitud que tiene el sistema de control distribuido en conjunto con la interfaz.

Interpolación de polos

Como primer paso, la interpolación forma parte importante del programa de la interfaz ya que este asegura que las referencias sean correctas para los algoritmos de cinemática inversa. El código utilizado se detalla en el anexo 11, en este programa se establecen puntos intermedios entre el punto final e inicial del eslabón final del brazo formando una trayectoria cúbica. En la prueba se tomaron 40 puntos pero en la implementación se usaron 3, con la posibilidad de aumentarlos de acuerdo al criterio del usuario.

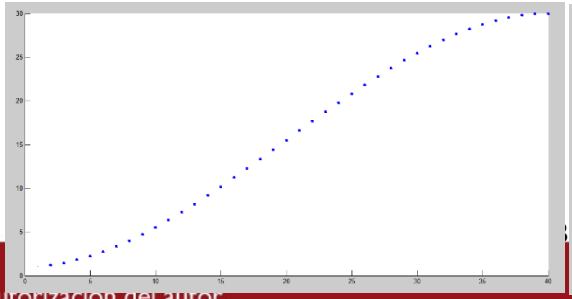
Donde: Zo, Posición inicial en el eje Z

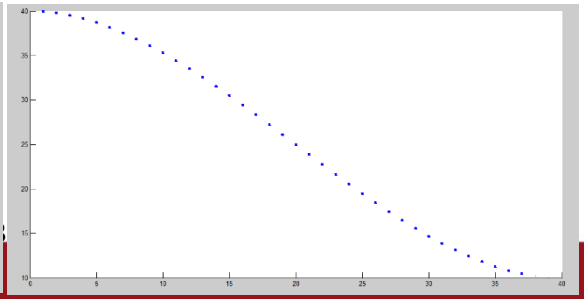
Zf, Posición final en el eje Z

Tf, número de puntos intermedios

Resultados:

Para Zf= 30cm y Zo= 0cm con Tf=40





pasos (en ascenso y descenso).

Ilustración 61 Prueba de interpolación de polos1 Elaboración propia

Prueba de trayectoria cubica

En los siguientes ensayos, se enviará coordenadas X, Y, Z deseados con la idea de que el brazo realice el movimiento necesario para llegar al punto, y con ello, realizar las mediciones de error respectivas.

Es importante recalcar 2 definiciones de volumen de error que servirán para determinar si el error que se midió de manera **práctica** está dentro de los errores previstos a **nivel teórico** usando los datos de la precisión de los eslabones individualmente en las pruebas anteriores.

Primera prueba:

X	-0.3
Y	0.4
Z	0.5

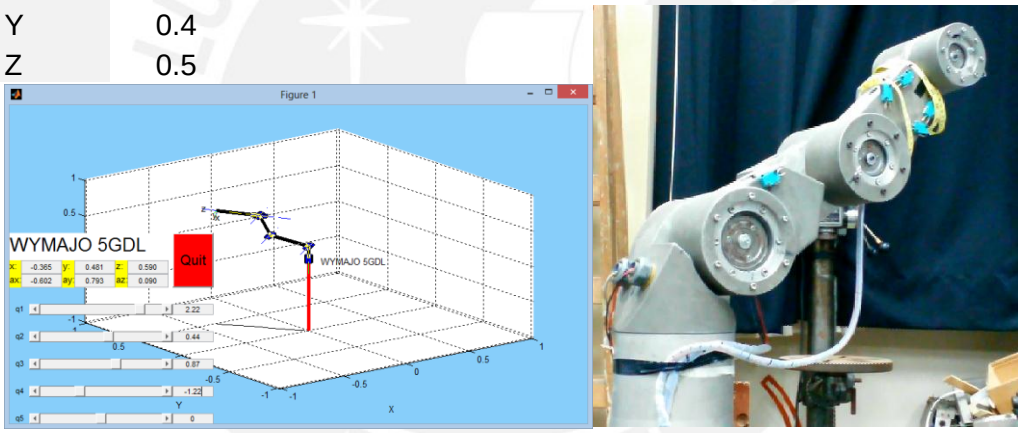


Ilustración 62 Prueba de trayectoria 1.Elaboración propia

Cinemática inversa: Los ángulos de cada motor para llegar al punto final deseado se muestra en la tabla 10.

Análisis de error teórico

De acuerdo al análisis anterior correspondiente a las pruebas individuales, los rangos de los posibles ángulos que se pueden dar se muestran en la tabla 11.

q1	123
q2	18
q3	47
q4	-74

	Min	Nom	Max
q1	123	127	131
q2	18	25	32
q3	47	50	53
q4	-74	-70	-64

Tabla 10 Ángulos ideales **Tabla 11 Análisis angular**

Luego, para poder cuantificar el error que se puede presentar a medida que se envía posiciones finales, se define un volumen de error. Este volumen conforma todos los posibles valores donde el eslabón final puede terminar teniendo como origen el punto deseado.

Por ello, para generar un volumen de error se toma en cuenta los valores angulares máximos y mínimos de la tabla anterior distribuyéndolo en 8 casos. Cada caso toma en cuenta el máximo o mínimo error de uno de los eslabones generando así un caso extremo (ver tabla 12).

Tabla 12 Ángulos teóricos trayectoria1 (Unidades en grados sexagesimales).Elaboración propia

Pruebamax1	Pruebamax2	Pruebamax3	Pruebamax4	Pruebamin1	Pruebamin2	Pruebamin3	Pruebamin4
131	127	127	127	123	127	127	127
25	32	25	25	25	18	25	25
50.04	50.04	53.04	50.04	50.04	50.04	47.04	50.04
-70	-70	-70	-66	-70	-70	-70	-74

Ángulo máximo

Ángulo mínimo

Cada caso tomado genera una coordenada del eslabón final del brazo, el cual es tomado en cuenta para formar el volumen de error teórico (ver tabla 13).

Tabla 13 Valores teóricos trayectoria 1 (posiciones y coordenadas en centímetros).Elaboración propia

Pruebamax1	Pruebamax2	Pruebamax3	Pruebamax4	Pruebamin1	Pruebamin2	Pruebamin3	Pruebamin4	
-21.4	-16.63	-18.76	-19.6333	-17.76	-22.33	-20.468	-19.633	x
24.62	22.07	24.9	26.0543	27.36	29.64	27.162	26.05	y
56.3792	60	56.7	56.3792	56.37	52.11	55.9689	56.3792	z

	Pos Real pro	Pos Teorico	minXYZ	maxXYZ	Errorpos	Errorneg	Errorpruebaprom	Errormax
x	-19.6	-19	-22.33	-16.63	2.37	-3.33	-0.6	3.33
y	26	26.5	22.07	29.64	3.14	-4.43	-0.5	4.43
z	56.3	57	52.11	60	3	-4.89	-0.7	4.89

Errmin / minXYZ: Es el mínimo valor en cada eje con respecto a las pruebas teóricas.

Ejemplo en x:
$\min(-21.4, -16.63, -18.76, -19.633, -17.76\ , -22.33, -20.468, -19.633) = -22.33$

Errmax/ maxXYZ: Es el máximo valor en cada eje con respecto a las pruebas teóricas.

Ejemplo en
x: $\max(-21.4, -16.63, -18.76, -19.633, -17.76\ , -22.33, -20.468, -19.633) = -16.63$

Errorpos y Errorneg: Son los valores de los errores de cada eje, tomando en cuenta los valores máximos y mínimos teóricos y el valor teórico.

Ejemplo en x: $$Errorpos = Errormax_x - PosTeórico$$

$$Errorpos = Errormin_x - PosTeórico$$

Errorpruebaprom: Es el valor de error entre el promedio de las pruebas reales de cada eje y el valor teórico.

$\boldsymbol{PosRealPromedio_x} = (PruebaReal1 + PruebaReal2 + PruebaReal3 + PruebaReal4)/4$, fundamentado en tabla 15.

$$\mathbf{Errorpruebaprom} = PosRealPromedio_x - PosTeórico$$

Errormax: Es el valor mayor entre Errorpos y Errorneg.

Finalmente, para hallar el radio del volumen de **error teórico** se toma el eje con el mayor error. En este caso, el valor sería de 4.89 cm en el eje Z; por lo tanto, el radio es de 4.89 con centro en el punto teórico de la tabla 10.

Volumen error teórico: $(x - x_{teórico})^2 + \ (y - y_{teórico})^2 + (z - z_{teórico})^2 \ = \max(errormax)^2$

Ecuación 9 Ecuación de volumen teórico

Análisis de error real (Unidades en centímetros):

Los valores teóricos son:

	Pos Teorico
x	-19
y	26.5
z	57

Tabla 14 Valores medidos en X1, Y1, Z1 trayectoria 1(Unidades en centímetros).Elaboración propia

Pos Teóricc	Prueba Real1	Prueba Real2	Prueba Real3	Prueba Real4	Promedio	emin	emax
-19	-19.45	-19.55	-19.7	-19.7	-19.6	0.45	0.7
26.5	25.7	26	26.4	25.9	26	0.1	0.8
57	57.3	56.5	56.8	54.6	56.3	0.2	2.4
Error p1		0.9656604	Max error	2.57099203			
Error p2		0.89582364					
Error p3		0.73484692					
Error p4		2.57099203					

Se aprecia que el eje que contribuye con mayor error absoluto es el eje z con 2.4 cm.

Dado que la ubicación del extremo del robot es en el espacio de coordenadas cartesiana, entonces calcularemos el volumen de error práctico para cada prueba según la fórmula 10. Donde n es el número de prueba, ya que se realizaron 4 pruebas repetitivas.

$$(xprueban - x_{teórico})^2 + \quad (yprueban - y_{teórico})^2 \; + (zprueban - z_{teórico})^2 = Errorpn^2$$

Ecuación 10 Cálculo de volúmenes de error práctico

Donde x, y, z son las coordenadas de la posición teórica y x1, y1, z1 son las coordenadas medidas para cada una de las 4 repeticiones.

Para la formar el volumen de error posible se toma en cuenta los valores de error pn, en este caso sería el valor 2.57cm por ser mayor a los demás.

Volumen de error práctico: $(x - x_{teórico})^2 + \quad (y - y_{teórico})^2 \; + (z - z_{teórico})^2 \quad = \max(Errorpn)^2$

Ecuación 11 Volumen de error práctico

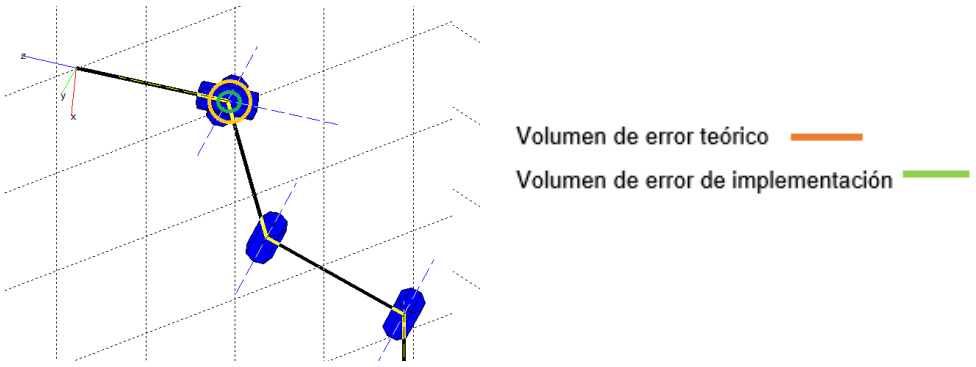


Ilustración 63 Volumen de error teórico y hallado en implementación de trayectoria 1.Elaboración propia

Finalmente se puede apreciar en la ilustración 63, que los valores prácticos están dentro de la esfera de errores teórico; por ello, se puede concluir que los errores hallados individualmente corresponden a los errores hallados en conjunto al hacer la implementación de todo el sistema. De manera análoga se hicieron 6 pruebas más en las cuales hay 2 y 4 intentos repetitivos, los detalles de los resultados se muestran en el anexo 24 y 28. Tomando en cuenta todas las pruebas hechas, el radio de volumen práctico hallado es de 9cm; cabe resaltar, que en todas las pruebas se hallaron puntos dentro del volumen de error teórico.

4.4 Análisis de resultados

4.4.1 Análisis de tiempo de respuesta

Inicialización de interfaz

En inicialización de la interfaz, el código demora en ejecutarse 0.352s.

En el proceso de conexión inicial con el puerto serial activado por el botón "Determinar conexión" y posterior activación toma 0.059s

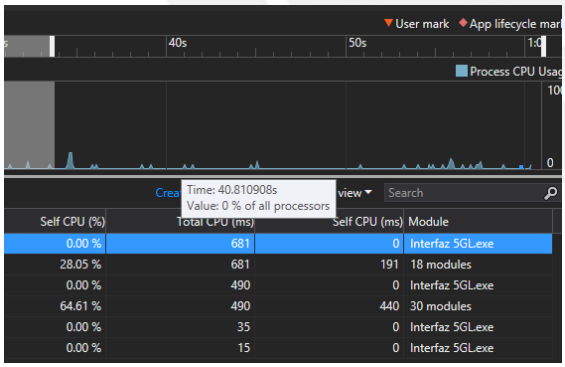


Ilustración 64 Medición tiempos interfaz. Elaboración propia

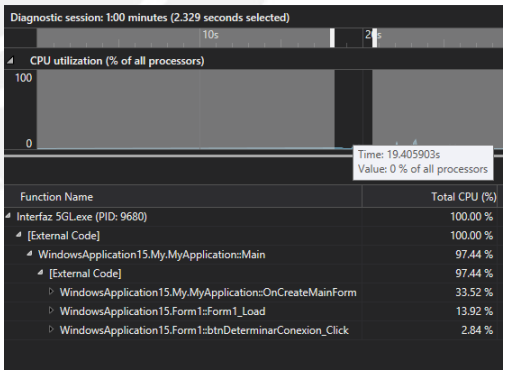


Ilustración 65 Medición de tiempo de inicialización. Elaboración propia

Tiempo de ejecución de interfaz

En la interfaz

Los resultados en segundos son:

0.681s Obtención de ángulos de movimiento para los 5 motores a partir de algoritmos de cinemática inversa y trayectoria

0.490s Realización de gráfica de los movimientos del brazo robótico

0.034s Actualización de datos en interfaz gráfica

0.015s Actualización de timer

Total: **1.22 s**

Tiempo de transmisión de datos

Transmisión interfaz maestro

Se considera el envío de (8bits x 5 caracteres x 4 esclavos = 160bits)

Con la velocidad de 9600 baudrate (bits por segundo), la transmisión demoraría 160bits/ 9600 bauds/s= **0.0167 s.**

Transmisión maestro esclavos

Se configuró el I2C a la velocidad 100Kbits/s, tomando en cuenta la repartición de la misma cantidad de bits y colocándolo en la trama I2C correspondiente.

| start | A7 A6 A5 A4 A3 A2 A1 R/W | ACK | ... DATA ... | ACK | stop | idle |

1 bit + 8 bits + 1 bit cada 8 bits enviados por la Interfaz; por lo tanto, cada 8 bits enviados por el serial se enviaría 10 bits de I2C.

En total serían 160 bits * (10/8) + 1 bit de idle= 201 bits enviados

Tiempo entre maestro y los 4 esclavos sería 201 bits / (100 000 bits/s) = 0.00201s

Por lo tanto el tiempo de transmisión es en total: 0.0167 + 0.00201= **0.01871s**

T**iempo de ejecución en microcontrolador maestro**

8 instrucciones de inicialización

23 Instrucciones de transmisión de ángulos

31 instrucciones x 0.125us/instrucción= **3.875 us**

Tiempo de ejecución en microcontrolador esclavo

20 instrucciones de inicialización

110 instrucciones de procesamiento de ángulo

130 instrucciones x 0.125us/instrucción= **16.25 us**

Ilustración 66 Diagrama de tiempos. Elaboración propia

Cabe resaltar que el tiempo de ejecución del microcontrolador esclavo se define más por la cuenta final de pulsos definidos por el maestro. Por lo tanto, se puede decir que el tiempo se define por la respuesta del sistema de lazo cerrado ante el set point.

CONCLUSIONES

Tomando en cuenta todo lo diseñado e implementado del sistema de brazo robótico de cinco grados de libertad se puede concluir lo siguiente:

El sistema tuvo éxito al ser implementado con los 4 microcontroladores esclavos siendo guiados por un microcontrolador maestro que a su vez es guiado por una interfaz gráfica que funciona con un generador de trayectorias basado en cinemática inversa o también en movimientos individuales. A nivel de comunicación entre maestro y esclavos hubo una comunicación I2C que permitió que el sistema pueda hacer trabajar a los 4 eslabones del brazo de manera simultánea y sin retardos para llegar a las posiciones finales. Además, el maestro pudo obtener los datos necesarios para realimentar a los esclavos debido a que la comunicación serial implementada entre maestro y generador de trayectorias fue exitosa al tomar una velocidad de 9600baud/s.

El sistema de control desarrollado e implementado cubrió las expectativas a nivel de control angular en los 4 primeros motores montados en el brazo. Se hicieron pruebas individuales en las cuales se encontró un error máximo de 7 grados detectado en la motor 2 ya que este presenta problemas mecánicos y tiene acoplado un encoder hecho en impresora 3D lo cual disminuye la precisión a la hora de emitir pulsos lo cual para las condiciones dadas es aceptable. Con respecto al tiempo de establecimiento, hay una diferencia comparando las respuestas teóricas con la de implementación de máximo 1.8 segundos con excepción al motor 4; ya que, a nivel teórico se consideró la inercia del motor 5 y gripper cuando todavía no han sido implementados. A nivel de pruebas en conjunto, el brazo robótico presenta un volumen de error práctico máximo

de 9 cm de radio debido a la acumulación de errores angulares de toda la cadena de eslabones funcionando en conjunto. Se comprobó a nivel de implementación y pruebas repetitivas que el brazo no tiene errores que sean mayor al radio de los volúmenes de error hallados.

El sistema del brazo robótico es controlado completamente por una interfaz gráfica, la cual permitió no solo movimientos aislados de los 4 eslabones disponibles en la estructura sino también, realizar trayectorias, al definir el punto final donde debe llegar. Esta interfaz realizó una secuencia de interpolación de polos entre el punto inicial y final en combinación de cinemática inversa modificada debido a las limitaciones presentadas en el motor 2; sin embargo, el error del algoritmo combinado es casi 0 en todo su volumen de trabajo permitido. Cabe mencionar, que la interfaz permitió también observar las referencias enviadas a los esclavos y ver en forma gráfica la trayectoria en tiempo real.

RECOMENDACIONES

Tomando en cuenta el diseño e implementación realizados durante la tesis y el deseo de seguir mejorando esta interfaz y brazo robótico para el desarrollo de un futuro módulo didáctico, se hacen las siguientes recomendaciones. En primer lugar, se debe depurar las deficiencias mecánicas que pueden limitar el rango de movimientos de cada eslabón.

En segundo lugar, a nivel de comunicación es posible mejorar la comunicación I2C implementando los microcontroladores en una sola tarjeta, o también es posible implementar el sistema con un ATMEGA328 que tiene 6 canales de PWM que le permitirían controlar 5 motores sin problema evitando la comunicación I2C y definiendo las instrucciones de tal manera que trabaje con procesamiento paralelo. Por otro lado, la comunicación serial debe mejorar adaptando las tarjetas a conectores DB9 y crear un bus serial para que haya una comunicación bidireccional que permita comunicar validaciones de movimiento hasta la interfaz gráfica.

Por último, sería apropiado adaptar al sistema un acelerómetro con la idea de realimentar el error de las posiciones finales del brazo; además, se podría validad velocidad y aceleración. Esto iría de la mano con la adquisición de drivers de motor que soporten valores mayores a 15 voltios con la idea de llevar a los motores cada vez más cerca a condiciones nominales y potenciar los movimientos a nivel de trayectorias. Con ello, se podrían analizar e implementar opciones de velocidad máxima y algoritmos matemáticos que incluyan jacobiano directo e inverso.

BIBLIOGRAFÍA

[1] KUCUK, S. and BINGUL, Z.

2004 "The inverse kinematics solutions of industrial robot manipulators,"[En linea] Mechatronics, ICM '04. Proceedings of the IEEE International Conference on , vol., no., pp.274,279, [Consulta 2014/08/23] <http://ieeexplore.ieee.org/Xplore>

[2] ALASSAR, A.Z.; ABUHADROUS, I.M.; ELAYDI, H.A.,

2010 "Modeling and control of 5 DOF robot arm using supervisory control,"[En línea] Computer and Automation Engineering (ICCAE), The 2nd International Conference on , vol.3, no., pp.351,355, [Consulta 2004/08/20] <http://ieeexplore.ieee.org/Xplore>

[3] QASSEM, M.A.; ABUHADROUS, I.; ELAYDI, H.,

2010 "Modeling and Simulation of 5 DOF educational robot arm,"[En línea] Advanced Computer Control (ICACC), 2010 2nd International Conference on , vol.5, no., pp.569,574, [Consulta 2014/09/04] <http://ieeexplore.ieee.org/Xplore>

[4] FOSTER, D.J.; HARRISON, A.J.L.,

2011 "Experimental investigation of a 5-DOF robot arm with sliding mode control,"[En línea] Robotics, Automation and Mechatronics (RAM), IEEE Conference on , vol., no., pp.125,130, [Consulta 2004/09/02] <http://ieeexplore.ieee.org/Xplore>

[5] BARRIENTOS, ANTONIO.

2007 Fundamentos de Robótica. Madrid: McGraw-Hill

[6] OGATA, KATSUHIKO.

1996 Sistemas de control en tiempo discreto. México, D.F. : Prentice Hall

[7] CRAIG,JOHAN

2005 Introduction to Robotics Mechanics and Control, 3rd Edition, pp 109-114, Prentice Hall.

[8] MORALES HERNÁNDEZ ,O.

2003 "Interface gráfica para el medidor del nivel". Puebla :Universidad de las Américas: Escuela de ingeniería. [Consulta 2014/09/18]

http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/morales_h_oe/indice.html

[9] NATIONAL INSTRUMENTS

2013 "Adquisición de datos" [Consulta 2014/09/20]

http://www.ni.com/sensors/esa/

[10] ZEPEDA, HECTOR E. ET. AL.

2010 "Desarrollo del modelo físico, diseño y análisis matemático de un brazo robot-ESIME" México DF: INSTITUTO POLITECNICO NACIONAL. Pp 55-56 [Consulta 2014/10/10]

http://tesis.ipn.mx:8080/xmlui/bitstream/handle/123456789/9947/43.pdf?sequence=1

[11] YIDONG WANG ET. AL.

2002 "Serial Communication in DNC Information Systems", China [Consulta 2014/10/15]

http://www.moxa.com/newsletter/connection/c_images/5-003/Serial%20_Comm_in_DNC_Information_System.pdf

[12] CARELLI, RICARDO

2001 "Dinámica y control de manipuladores robóticos".Lima: PUCP

[13] CHAVEZ, JOSE

2014 "Programación del sistema electrónico de un robot manipulador de cinco grados de libertad". PUCP.

[14] COCOLETZI, RICARDO

2007 "Interfaz gráfica para el control de un brazo robótico educativo de 5 grados de libertad". México: Tlaxcala. Universidad Autónoma de Tlaxcala.

[15] MARTINEZ,GLORIA ET. AL.

2008 "Diseño propio y Construcción de un Brazo Robótico de 5 GDL".México: Revista de ingeniería eléctrica, electrónica y computación. http://www.itson.mx/publicaciones/rieeyc/Documents/v4/art2junio08.pdf

[16] KUMAR,RAVI

2014 "Comparison between FSC and PID Controller for 5DOF Robot Arm". International journal of emerging trends in electric and electronics.

http://www.tuat.ac.jp/~venture/ronbun/ReadingGroupControl/Control_Of_5DOF_Robot_Arm_using_Fuzzy.pdf

[17] JURAIZA, ASNOR ET. AL.

2013 "Position control of arm mechanism using pid controller".Malasia: Universiti Putra Malaysia. http://www.jatit.org/volumes/Vol47No2/53Vol47No2.pdf

[18] IBRAHIM, D.

2006 "Microcontroller based applied digital control". John Wiley &Sons

[19] K. J.ASTROM AND B. WITTENMARK

1997 "Computer-Controlled Systems". Department of Automatic.Control - Lund Institute of Technology

[20] K.J. Åström

2002 "Control system design". Caltech. Åström. Department of Mechanical and Environmental Engineering University of California.

[21] Hayk Markaroglu

"Tracking time adjustment in back calculation anti-windup scheme". Istanbul Technical University.

[22] Bruyninckx, Herman

2010 "Robot Kinematics and Dynamics". Katholieke Universiteit Leuven Department of Mechanical Engineering.Leuven, Belgium

[23] Corke, Peter

2014 "Denavit-Hartenberg notation for common robots". http://petercorke.com/doc/rtb_dh.pdf

[24] Zi, Bin

2012 "Inverse Kinematics and Singularity Analysis for a 3-DOF Hybrid-driven Cable-suspended Parallel Robot". School of Mechanical and Electrical Engineering, China University of Mining and Technology