ESPE

UNIVERSIDAD DE LAS FUERZAS ARMADAS

INNOVACIÓN PARA LA EXCELENCIA

DEPARTAMENTO DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA EN MECATRÓNICA

TRABAJO DE TITULACIÓN, PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN MECATRÓNICA

TEMA: DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR MEDIANTE SOFTWARE LIBRE PARA EL MANIPULADOR SCORBOT-ER III DE 5 GRADOS DE LIBERTAD

AUTOR: NELSON HERNÁN ALDAZ SACA

DIRECTOR: ING. FAUSTO ACUÑA

LATACUNGA

2016

DEPARTAMENTO DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA EN MECATRÓNICA

CERTIFICACIÓN

Certifico que el trabajo de titulación, "**DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR MEDIANTE SOFTWARE LIBRE PARA EL MANIPULADOR SCORBOT-ER III DE 5 GRADOS DE LIBERTAD**", ha sido revisado en su totalidad y analizado por el software anti-plagio, el mismo cumple con los requisitos teóricos, científicos, técnicos, metodológicos y legales establecidos por la Universidad de Fuerzas Armadas ESPE, por lo tanto permito acreditarlo y autorizar al señor **Nelson Hernán Aldaz Saca** para que lo sustente públicamente.

Latacunga, 19 de Febrero de 2016

Ing. Fausto Acuña

DIRECTOR

DEPARTAMENTO DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA EN MECATRÓNICA

AUTORÍA DE RESPONSABILIDAD

Yo, **Nelson Hernán Aldaz Saca**, con cédula de identidad N° 180461150-5, declaro que este trabajo de titulación "**DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR MEDIANTE SOFTWARE LIBRE PARA EL MANIPULADOR SCORBOT-ER III DE 5 GRADOS DE LIBERTAD"** ha sido desarrollado considerando los métodos de investigación existentes, así como también se ha respetado los derechos intelectuales de terceros considerándose en las citas bibliográficas.

Consecuentemente declaro que este trabajo es de mi autoría, en virtud de ello me declaro responsable del contenido, veracidad y alcance de la investigación mencionada.

Latacunga, 19 de Febrero de 2016

Nelson Hernán Aldaz Saca

C.C.: 180461150-5

DEPARTAMENTO DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA EN MECATRÓNICA

AUTORIZACIÓN

Yo, **Nelson Hernán Aldaz Saca**, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar en la biblioteca Virtual de la institución el presente trabajo de titulación **"DISEÑO E IMPLEMENTACIÓN DE UN CONTROLADOR MEDIANTE SOFTWARE LIBRE PARA EL MANIPULADOR SCORBOT-ER III DE 5 GRADOS DE LIBERTAD"** cuyo contenido, ideas y criterios son de mi autoría y responsabilidad.

Latacunga, 19 de Febrero de 2016

Nelson Hernán Aldaz Saca

C.C.: 180461150-5

DEDICATORIA

El Presente Proyecto va dedicado a:

Mis Padres Nelson y Marilú, y mi hermano Cristian por su apoyo incondicional y constante, que en incontables madrugadas me dijeron ¨Tu puedes¨, siendo ellos un pilar fundamental para hacer esto posible.

A mis Abuelitos que de una y otra manera han estado siempre pendientes de mi bienestar.

A mis Amigos Gaby, Luis, Cris, que me demostraron una amistad sincera entre buenas y malas calificaciones.

Y cada una de las personas que a lo largo de mi carrera compartieron momentos felices y difíciles que se me presentaron, y que con sus palabras me elevaban el ánimo para alcanzar mis objetivos.

AGRADECIMIENTO

A Dios por darme la oportunidad de vivir junto a unos padres extraordinarios, que gracias a ellos he podido culminar una etapa muy importante de mi vida, con los sacrificios sobrenaturales que han hecho día tras día, a mi hermano por saber apoyarme y por el cariño que me brinda, a mis familiares y amigos que estuvieron siempre predispuestos a ayudarme, a Paul por haberme compartido sus conocimientos y experiencias.

Y como no agradecerle al Ing. Fausto Acuña quien más que un tutor fue un amigo que me regalo lo más valioso que es el conocimiento, y que además ha despejado todas mis dudas durante toda esta investigación.

Sería una lista muy extensa si agradezco a cada una de las personas que conforman la Universidad de la Fuerzas Armadas ESPE Extensión Latacunga, por eso en general les doy las gracias a todos los que conforman esta Institución Educativa que más que otorgarme un título, me otorgaron buenos valores y grandes conocimientos con alta conciencia social.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

En este proyecto se desarrolló un controlador genérico de cinco grados de libertad operado mediante software libre para el manipulador SCORBOT-ER. El desarrollo del controlador se resume en cuatro etapas: diseño e implementación del controlador, desarrollo de la HMI y pruebas de funcionamiento. Durante la fase del diseño del controlador se estudiaron los actuadores y sensores del SCORBOT para determinar los parámetros de control y seleccionar los componentes del controlador: controladores para motores DC L298N, tarjeta de control Arduino Mega, puertos de comunicación DB25, encoders a medida, circuito distribuidor de voltaje y fuente ATX de 700 W; cumplida esta tarea, la implementación del controlador se realizó de acuerdo a las hojas técnicas y especificaciones del robot y de cada uno de los componentes del controlador. Como punto de partida en el desarrollo de la HMI, fue necesario realizar la selección del sistema operativo de trabajo y el programa específico sobre el que se escribió la aplicación; el desarrollo de la HMI en VPython satisfizo los requerimientos del robot y el controlador ensamblado en las fases anteriores; además, el lenguaje de programación Python presenta la ventaja de ser multiplataforma por lo que la HMI del controlador puede ejecutarse sobre los principales sistemas operativos; en la HMI, las siguientes funciones fueron implementadas: cinemática directa, cinemática inversa, rutina de secuencias, interpolación lineal e interpolación circular. Finalmente tanto el controlador como el HMI fueron sometidos a pruebas de posicionamiento para verificar su desempeño dando como resultado que las posiciones alcanzadas tuvieron un error máximo de 0,0001 mm con respecto a las deseadas.

PALABRAS CLAVE:

- **BRAZO ROBÓTICO**
- **SOFTWARE LIBRE**
- **SCORBOT**
- **ARDUINO**

ABSTRACT

The aim of this project was to develop a 5 DOF generic controller for robotic arm SCORBOT ER III. The development of the controller was divided in 4 stages: circuit design, circuit implementation, HMI design and HMI implementation. In circuit design stage, sensors and actuators were studied in order to determinate the requirements of the controller and select each one of its components: 2 A L298N motor drivers, Arduino Mega board, DB25 ports, encoders, voltage distributor circuit and 700W ATX power supply. In the next stage, circuit implementation, all electronic components were selected and integrated in a fully operative controller following all indication presents in datasheets and specification of components. As a previous step of HMI´s development it was necessary to choose an Open Source IDE due to project requirements; Vpython was chosen for the task and the HMI was developed over Python due to it is a multiplatform language allowing to execute the HMI in most used operative systems; the following functions were implemented: forward kinematics, inverse kinematics, sequence execution, linear interpolation and circular interpolation. Finally, all system were integrated and tested in order to determinate the robot´s accuracy and precision working with the generic controller; results shows that position reached by the robot had a 0,0001 mm error when compared with the positions given.

KEY WORDS:

- **ROBOTIC ARM**
- **OPEN SOURCE SOFTWARE**
- **SCORBOT**
- **ARDUINO**

CAPÍTULO I
MARCO TEÓRICO

1.1 Generalidades

En la actualidad las operaciones que son repetitivas, que necesitan gran exactitud o que son peligrosas obligan a que se emplee manipuladores robóticos en la producción industrial. En procesos repetitivos un operador humano tiende a cansarse y genera menor rendimiento que un robot industrial.

Es por eso que los manipuladores robóticos SCORBOT son una herramienta fundamental en centros educativos donde se forman profesionales que se desempeñen en industrias de automatización.

Los procedimientos automatizados mediante manipuladores robóticos se han convertido en una herramienta de gran importancia para la industria ya que son versátiles, fiables y de gran ayuda para el hombre porque permiten optimizar, agilizar y cuantificar de modo eficiente trabajos que antes se realizaban manualmente con mayor confiabilidad.

1.2 Definición del problema

La industria de los manipuladores robóticos presenta hoy en día un avanzado estado de desarrollo; esto ha permitido que los manipuladores dejen de ser parte de planes pilotos en fábricas y universidades y constituyan pilares fundamentales de las industrias de consumo masivo. La gran demanda que los manipuladores robóticos tienen en la actualidad ha llamado la atención de muchas organizaciones que han financiado acelerados programas de innovación y mejoramiento que permite obtener máquinas cada vez más sofisticadas; sin embargo, debido a este rápido crecimiento, muchos manipuladores quedan obsoletos año tras año debido principalmente a que sus productores descontinúan el modelo y por lo tanto sus piezas de repuesto y el software que utilizan. Un ejemplo claro es la línea de robots SCORBOT; las estructuras de sus modelos más antiguos se encuentran aún en buen estado pero sus controladores son obsoletos para la tecnología actual.

En este proyecto se pretende rescatar un SCORBOT ER III del desuso a través del desarrollo de un controlador genérico que le permita adaptarse a la tecnología de los ordenadores modernos.

1.3 Objetivo general

Diseñar e implementar un controlador genérico mediante software libre para el manipulador SCORBOT-ER de 5 grados de libertad.

1.4 Alcance

Este estudio pretende desarrollar un controlador genérico para el manipulador SCORBOT ER III que pueda ser empleado en tareas de aprendizaje.

El estudio de este manipulador permitirá a sus usuarios acrecentar sus conocimientos respecto al funcionamiento de cada uno de los componentes de los manipuladores industriales: mecánicos, actuadores, sensores, controlador y software de control.

Un controlador genérico para un manipulador industrial cuyo diseño y funciones se encuentre completamente documentados presentan, entre otras, las siguientes características:

- Mayor control del usuario.
- Mantenimiento y reparación sencillos, sin la necesidad de técnicos especializados y con componentes de fácil adquisición.
- Fácil de reproducir en comparación a controladores comerciales.
- Actualización e implementación de mejoras más fáciles en el controlador.
- Actualización e implementación de mejoras más fáciles en el software.
- Debido a que se controla mediante software libre, se evita la adquisición de licencias.

1.5 Descripción del SCORBOT ER III

El manipulador SCORBOT ER III es un brazo robótico articulado diseñado para emular un brazo robótico industrial real. Al día de hoy la línea SCORBOT cuenta con el modelo ER IX por lo que el modelo ER III así como su controlador se encuentran descontinuados; sin embargo, debido a la gran robustez de su estructura mecánica y sus actuadores, muchos modelos ER III siguen operando con controladores e interfaces adaptados a la tecnología actual.

1.5.1 Morfología

El Scorbot puede considerarse un robot de tipo antropomórfico ya que posee las mismas junturas que un brazo humano: hombro, codo y muñeca. Una

característica importante del manipulador es que cada uno de sus eslabones puede girar solamente sobre uno de los ejes cartesianos y los ejes del hombro, codo y muñeca se encuentran alineados. El Scorbot además posee una pinza anclada a la muñeca que le permite asir pequeños objetos.

La Figura 1 muestra los diferentes ejes que posee el Scorbot. El primer eje corresponde a la base y se gira alrededor del eje cartesiano Y gracias a la acción de un motor DC marca Pittman de 24 Watts. Los ejes correspondientes al hombro y el codo del Scorbot se encuentran alineados al eje X y son accionados gracias a sendos motores DC marca Pittman de 24 Watts. La muñeca del Scorbot tiene la capacidad de girar en los ejes X y Z gracias al mecanismo diferencial mostrado en la Figura 2 y dos motores DC marca Pitmman de 24W; el movimiento de la muñeca se logra de la siguiente manera: cuando los motores giran en el mismo sentido la muñeca gira alrededor del eje X mientras que cuando giran en sentidos opuestos la muñeca girará alrededor del eje Z. El robot tiene la capacidad de asir objetos gracias a una pinza accionada por un motor DC de 24W; la apertura máxima de la pinza es 65 mm por lo que las dimensiones de los objetos que se pretenda manipular no deberán exceder este límite (Intelitek, 1995, pág. 7). Las dimensiones del robot se muestran en la Figura 3.

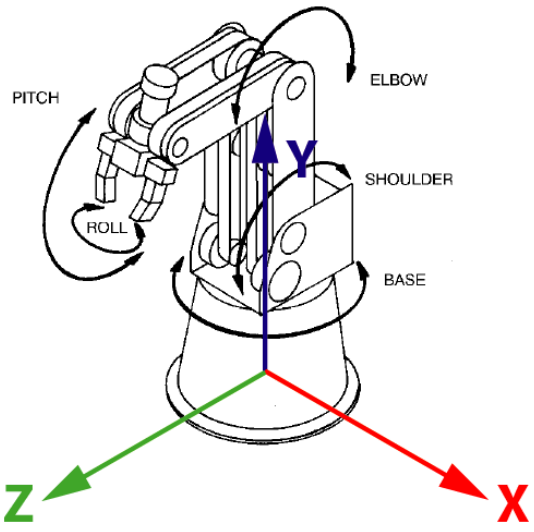


Figura 1. Ejes del Scorbot.

Fuente (Intelitek, 1995, pág. 8)

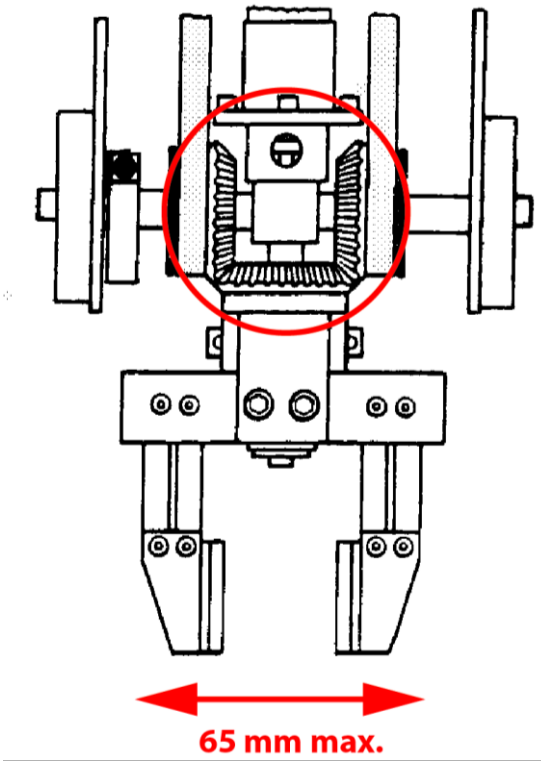


Figura 2. Mecanismo diferencial de la muñeca.

Fuente (Intelitek, 1995, pág. 45)

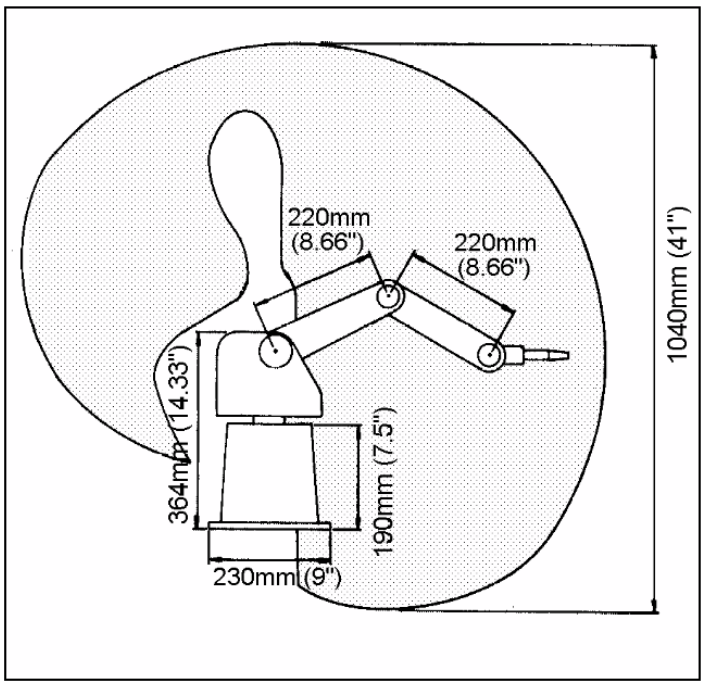


Figura 3. Dimensiones del SCORBOT ER III.

Fuente (Intelitek, 1995, pág. 6)

1.5.2 Actuadores

El movimiento de cada eje del manipulador así como la apertura y cierre de la pinza se logran mediante motores DC de 24 W. El voltaje nominal de estos motores DC es 12 V por lo que la corriente máxima que podrán consumir es de 2 A (Intelitek, 1995, pág. 150).

Todos los motores DC que permiten el movimiento de los ejes se encuentran soportados en la base a fin de reducir el peso de los eslabones; el movimiento es transmitido entonces por medio de poleas y bandas dentadas. El motor DC destinado al accionamiento de la pinza se encuentra montado en el propio mecanismo de la pinza pero es de tamaño relativamente pequeño comparado con el resto de motores DC y no representa una carga excesiva para la estructura.

1.5.3 Sensores

El SCORBOT posee dos tipos de sensores: encoders de cuadratura e interruptores; los interruptores permiten al robot encontrar la posición HOME cuando este empieza a operar mientras que los encoders de cuadratura permiten conocer el número de grados que un eje ha girado y la dirección que ha seguido (Intelitek, 1995, pág. 21).

A. Sensores de cuadratura

Un encoder es un sensor capaz de convertir el movimiento de un dispositivo en trenes de pulso a fin de medir las siguientes variables: desplazamiento, dirección y posición de acuerdo con las capacidades de su tecnología.

Los encoders de cuadratura utilizados en el SCORBOT ER III constan principalmente de un disco ranurado y dos fotointerruptores (Intelitek, 1995, pág. 22); los fotointerruptores se encuentran desfasados noventa grados tal como se muestra en la Figura 4 para que sus señales permitan medir la cantidad de movimiento y el sentido de giro del actuador.

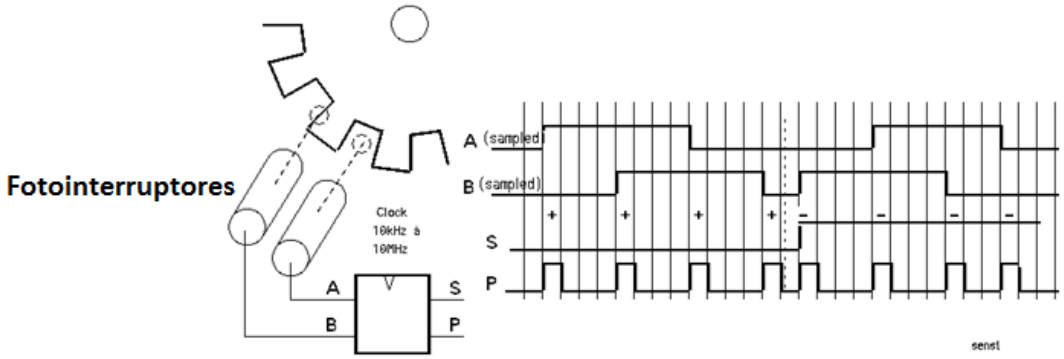


Figura 4. Encoder de cuadratura.

Fuente (DIDEL, s.f.)

B. Interruptores mecánicos.

Un interruptor mecánico es un dispositivo que permite o no el paso de la corriente eléctrica a través de una línea de conducción. La activación del interruptor puede realizarse por medios mecánicos, magnéticos, ópticos etc.; en el caso del SCORBOT ER III los interruptores son mecánicos tal como se aprecia en la Figura 5.

Figura 5. Interruptores mecánicos del SCORBOT ER III.

Fuente: (OBrien, s.f.)

El SCORBOT ER III posee cinco interruptores, uno por cada articulación, que permiten ubicar la posición HOME al inicializar el robot (Intelitek, 1995, pág. 22)

1.5.4 Controlador

El controlador original del SCORBOT es un contenedor en el que se aloja toda la circuitería que permitirá la operación del robot a través de la PC.

Entre los componentes principales del Scorbot podemos nombrar los siguientes:

- Tarjeta madre: gestiona todas las operaciones del robot
- Circuito de control de los motores DC: permite el control de velocidad de los motores DC de los ejes y la pinza
- Circuito para lectura de los sensores: los encoders de cuadratura requieren de un circuito auxiliar para la lectura de sus señales
- Circuito de comunicación: permite la comunicación entre la tarjeta madre y la PC
- Fuente de poder: suministra la energía necesaria para la operación del robot

Entre las funciones que cumple el controlador pueden mencionarse las siguientes:

- Control y alimentación de los cinco motores DC del brazo y el motor DC de la pinza.
- Interpretación de las instrucciones editadas en el software proporcionado por la empresa; el software del controlador corre en una versión de Windows actualmente en desuso.
- Lectura de las señales de los encoders e interruptores del robot.
- Administración de entradas y salidas adicionales.

1.5.5 Software del Scorbot

SCORBASE es el software original para manejar el Scorbot ER III. Este programa venia cargado en diskette por lo que ya no puede ejecutarse en computadoras modernas. Este programa se ejecutaba sobre el sistema DOS actualmente en desuso en máquinas que operan con Windows (Intelitek, 1995, pág. 35).

1.6 Fundamentos de los manipuladores robóticos

El desarrollo del controlador de un robot manipulador requiere del estudio de cuatro campos principales: actuadores, sensores, controlador y software.

1.6.1 Actuadores

Los actuadores para robots manipuladores son principalmente eléctricos, hidráulicos y neumáticos; otros tipos de fuentes de energía también pueden ser consideradas. Otra clasificación, de acuerdo al tipo de movimiento, los agrupa en lineales y rotativos.

En esta sección se estudiarán únicamente los motores DC, que son los utilizados por el SCORBOT ER III.

El funcionamiento de un motor DC es bastante simple: al aplicar una tensión a sus terminales el motor comenzará a girar; si se invierte la conexión, el motor gira en sentido opuesto. Es necesario que la fuente sea capaz de entregar suficiente corriente para alimentar el motor a plena carga.

El funcionamiento descrito en el párrafo anterior permite varios tipos de control para un motor DC; cada uno de estos controladores se describe a continuación:

A. Control unidireccional

Si se requiere que el motor impulse en un solo sentido, el control del motor se logrará con un solo interruptor tal como se puede observar en la Figura 6.

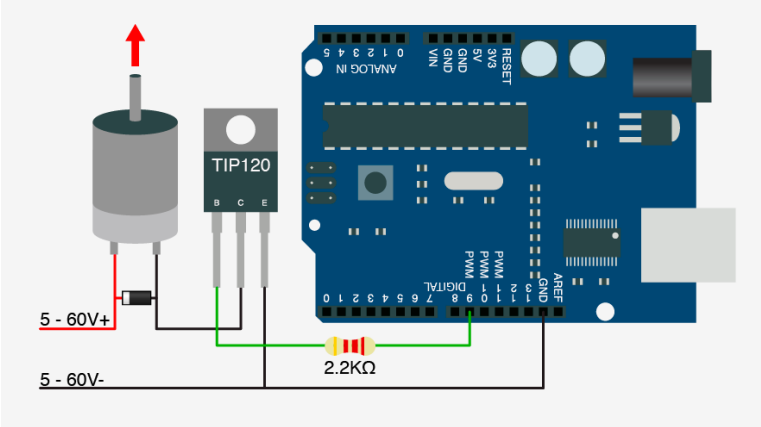


Figura 6. Controlador unidireccional de un motor DC de escobillas.

Fuente: (Magnusglad, s.f.)

B. Control bidireccional

Para lograr el control bidireccional de un motor DC se necesitan cuatro interruptores dispuestos tal como se muestra en la Figura 7, este arreglo se conoce como puente H; un puente H activa el motor cerrando los interruptores opuestos de cada una de sus ramas (Bar, s.f.).

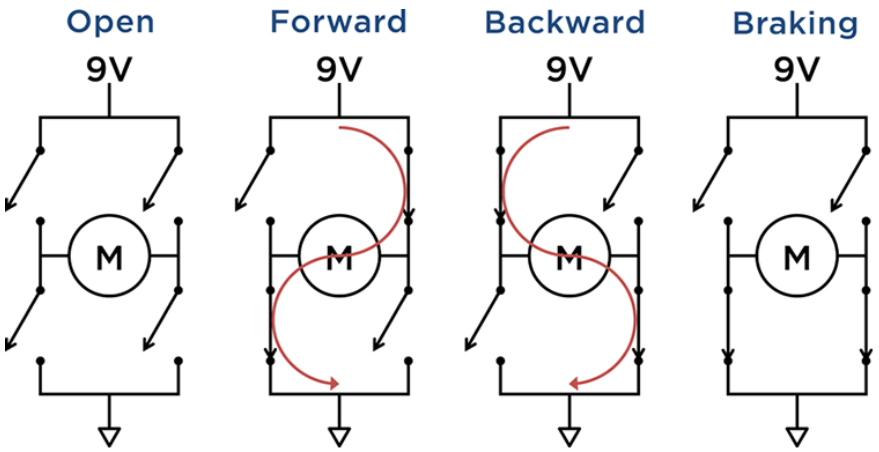


Figura 7. Controlador bidireccional de un motor DC con escobillas

Fuente: (Bar, s.f.)

C. Control de velocidad

La gran mayoría de controladores de velocidad para motores DC basan su funcionamiento en la modulación de ancho de pulso o PWM por sus siglas en inglés. (Sparkfun, s.f.)

Un circuito PWM entrega un tren de pulsos; en este circuito es posible variar la frecuencia con que ocurre cada ciclo y el tiempo que la señal permanece encendida dentro del ciclo. La forma de utilización del circuito más difundida es la de dejar la frecuencia fija y variar el tiempo en ON de la señal tal como se puede apreciar en la Figura 8.

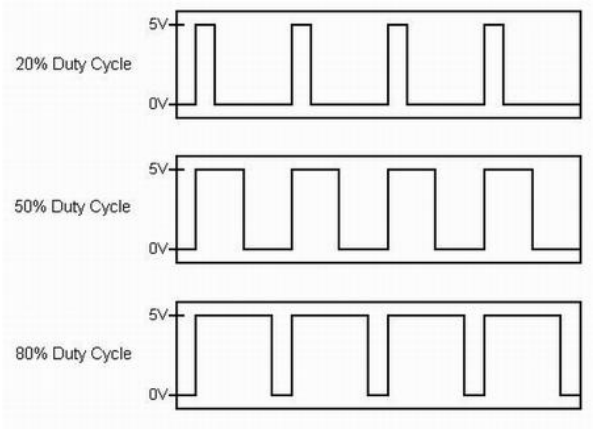


Figura 8. Modulación de ancho de pulso

Fuente: (National Instruments, s.f.)

Una aplicación muy difundida de los circuitos PWM la constituye el convertidor de puente completo ilustrado en la Figura 9; estos convertidores permiten

variar la potencia total entregada a una carga a través de un arreglo tipo puente H y un circuito PWM.

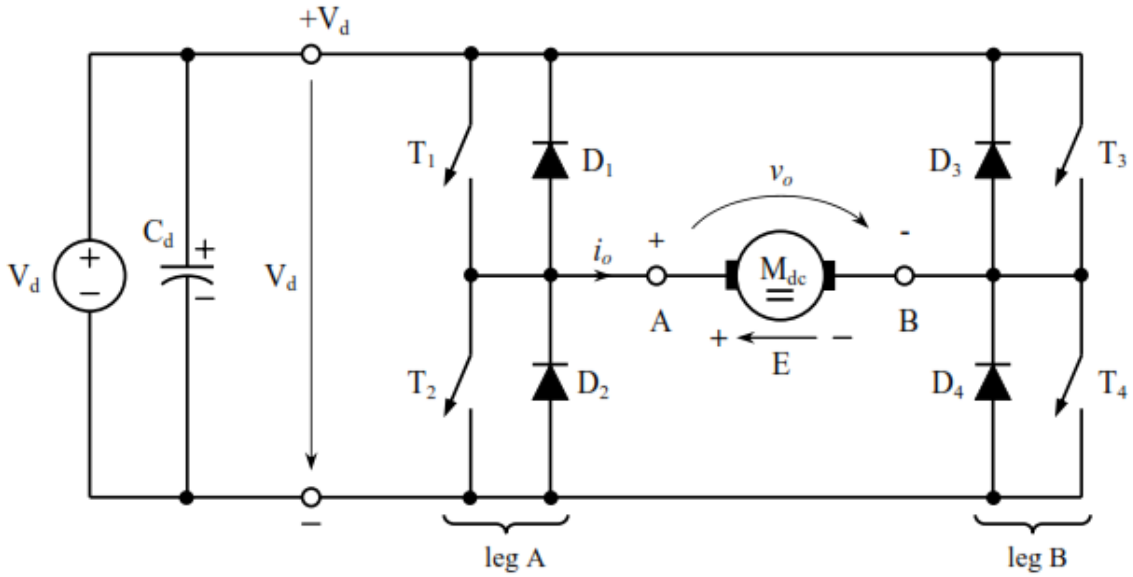


Figura 9. Controlador de velocidad de un motor DC

Fuente: (Albu, s.f., pág. 4)

La señal de control se utiliza en los interruptores T_1 y T_4, la señal invertida se utiliza en T_2 y T_3; este arreglo permite controlar la velocidad y sentido de giro del motor sin la necesidad de circuitos auxiliares; las formas de onda de este controlador se ilustran en la Figura 10.

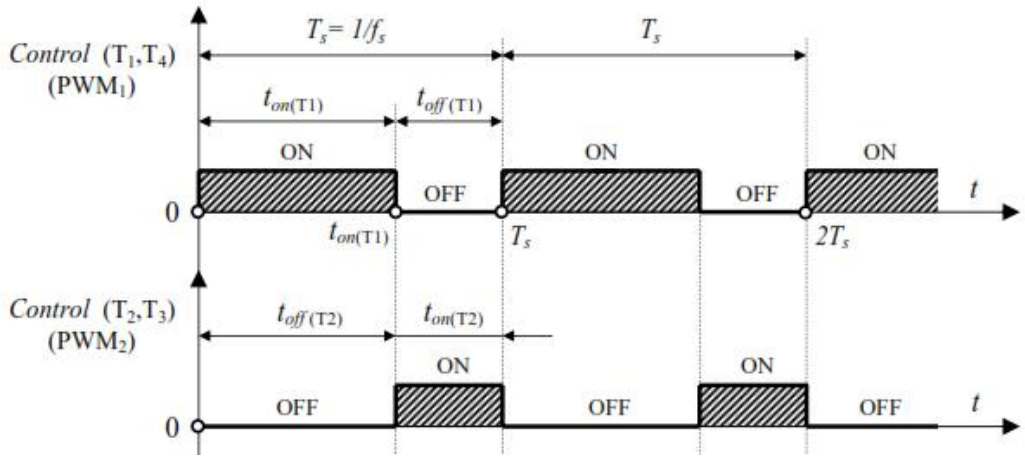


Figura 10. Formas de onda de un convertidor de puente completo

Fuente: (Albu, s.f., pág. 4)

1.6.2 Sensores

Los sensores más importantes para un robot manipulador son los que le permitirán determinar la posición y orientación de cada uno de sus eslabones; para determinar la posición existen varias alternativas entre las que destacan las siguientes:

A. Potenciómetros

Estos dispositivos entregaran una señal analógica que depende del ángulo que esté girada su perilla de control. Este tipo de dispositivos pueden encontrarse de muy bajo costo en el mercado; sin embargo no siempre es fácil adaptarlos al mecanismo que se desee controlar y requiere etapas adicionales de acondicionamiento. La Figura 11 muestra un potenciómetro típico.

Figura 11. Potenciómetro típico

Fuente: (Vex robotics, s.f.)

B. Encoder

Los encoders absolutos son ampliamente utilizados debido a que permiten conocer la posición de los eslabones en cualquier momento; presentan la desventaja de que aún los más simples tienen un costo elevado. Los encoders incrementales tienen precios accesibles para aplicaciones de mediana envergadura pero necesitan sistemas adicionales para referenciarse cuando el robot inicia su funcionamiento. La Figura 12 muestra el funcionamiento de un encoder absoluto; las diferentes ranuras representan una codificación binaría en la que la ranura más interna representa el bit más significativo y la más externa el menos significativo.

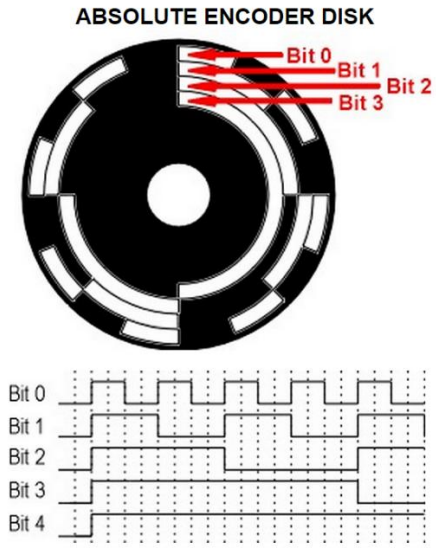


Figura 12. Encoder absoluto.

Fuente: (Automated motion systems, s.f.)

C. Acelerómetro

El acelerómetro es un dispositivo electromecánico capaz de medir aceleraciones; un acelerómetro de 3 ejes informará de la inclinación de su superficie respecto a la de la superficie terrestre; este dispositivo es muy recomendado para medir valores estáticos. Cuando mediciones dinámicas son requeridas se emplea, conjuntamente con el acelerómetro, un giroscopio que es capaz de medir la velocidad de rotación de una superficie en un determinado eje. El cálculo de la inclinación en tiempo real, resultará de la integración de las variables velocidad y aceleración. La Figura 13 muestra el comportamiento de un acelerómetro típico.

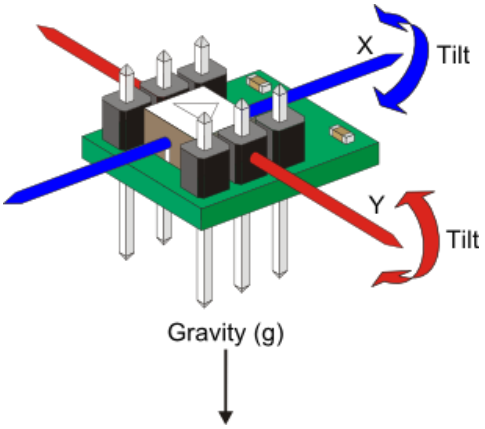


Figura 13. Funcionamiento de un acelerómetro típico.

Fuente: (Parallax, s.f.)

1.6.3 Controlador

El controlador de un manipulador consta de dos partes: hardware y software

A. Hardware del controlador

Un controlador para un robot manipulador puede ser cualquier dispositivo basado en microcontroladores con suficientes entradas y salidas digitales y analógicas tanto para operar actuadores como sensores y permitir comunicación con otros dispositivos tales como computadoras personales.

Los dispositivos más utilizados para controlar robots manipuladores industriales son los controladores lógico programables; para el caso de robots más pequeños pueden utilizarse microcontroladores y tarjetas programables. La Figura 14 muestra ejemplos de estos dispositivos.

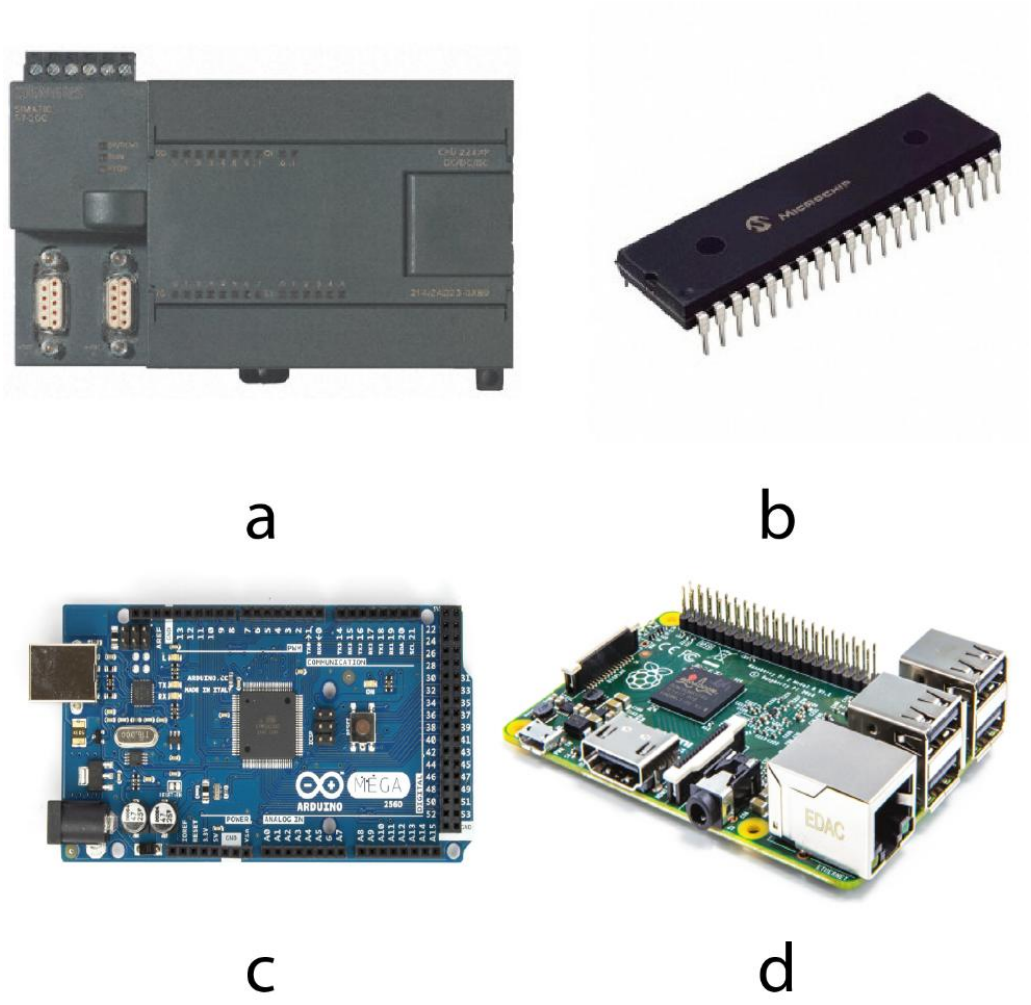


Figura 14. Diferentes controladores para robots manipuladores. (a) Controlador lógico programable SIEMENS (b) Microcontrolador PIC (c) Arduino MEGA (d) Raspberry PI

B. Software del controlador

Independientemente del lenguaje de programación que se utilice para el controlador, cualquier programa escrito para controlar un robot manipulador deberá contener las ecuaciones cinemáticas que describen su movimiento; los modelos que pueden utilizarse para describir el movimiento del robot manipulador son principalmente dos: modelo cinemático directo y modelo cinemático inverso; ambos modelos se relacionan de acuerdo a la Figura 15. En el modelo directo se calculan las coordenadas del extremo del robot partiendo de los valores de los ángulos de las articulaciones mientras que en el modelo inverso se calculan los ángulos de las articulaciones partiendo de las coordenadas del extremo del robot.

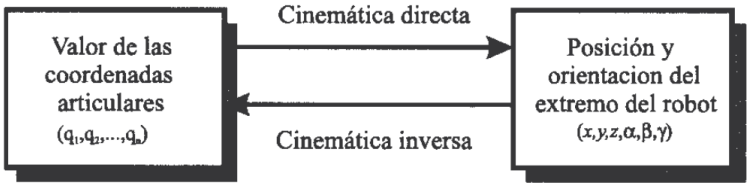


Figura 15. Modelo cinemático directo y modelo cinemático inverso de un robot manipulador.

Fuente: (Barrientos, 2007, pág. 94)

1. Modelo cinemático directo

En este modelo se busca determinar las coordenadas del extremo del robot a partir del conocimiento de las longitudes de los eslabones y los ángulos que forman los eslabones del robot (Barrientos, 2007, pág. 93)

A fin de resolver el modelo cinemático directo se han desarrollado varios procedimientos: geométrico, analítico, matrices de transformación, etc.; en una configuración serial la solución del modelo cinemático directo es única

2. Modelo cinemático inverso

En este modelo se busca determinar los ángulos que forman los eslabones a partir del conocimiento de las coordenadas del extremo del robot y la longitud de los eslabones (Barrientos, 2007, pág. 93)

Los procedimientos para resolver el modelo cinemático inverso son los mismos que para el modelo directo; sin embargo, a diferencia del modelo directo,

existen varias soluciones para este modelo; la Figura 16 ilustra que en un manipulador de dos eslabones existen dos configuraciones que llegan a la misma posición

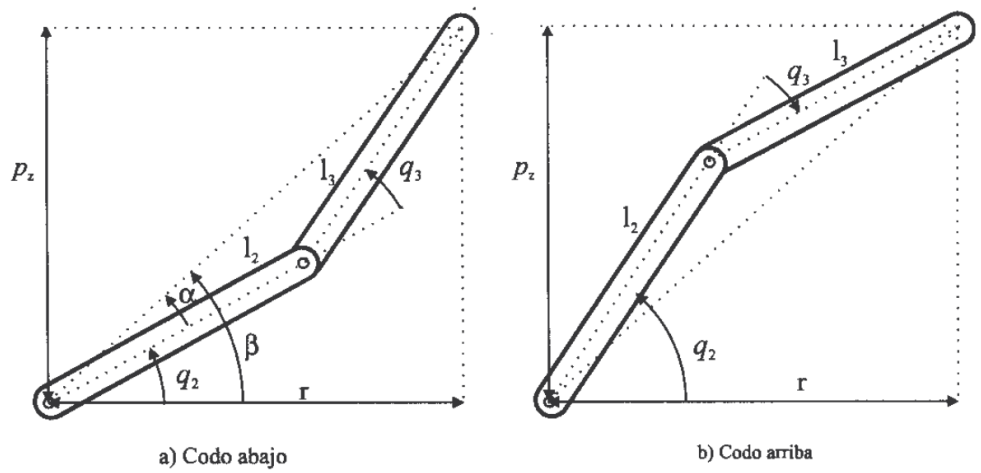


Figura 16. Dos soluciones para un mismo modelo cinemático inverso.

Fuente: (Barrientos, 2007, pág. 111)

1.7 Estado del arte

Existe una gran cantidad de modelos SCORBOT ER III operables; sin embargo, sus controladores se encuentran obsoletos ya que para utilizarlos se requiere una versión de Windows que ya no tiene soporte.

Para aprovechar al máximo las características mecánicas del SCORBOT ER III es necesario acondicionarlo a las nuevas tecnologías de control disponibles tal como se ha logrado en los trabajos estudiados en los siguientes apartados.

1.7.1 Estudio e implementación de un controlador (hardware y software) para tres grados de libertad del manipulador robótico CRS-A255

El propósito de este proyecto fue desarrollar un controlador nuevo para uno de los robots CRS-A255 del laboratorio de robótica del Departamento de Eléctrica y Electrónica de la ESPE cuyo controlador presentaba fallas que no permitían su correcto funcionamiento.

El robot CRS-A255 comparte muchas características con el Scorbot ER III tanto en tamaño como en composición. Posee las siguientes articulaciones: base, hombro, codo, muñeca y herramienta; todos los ejes se mueven de con la ayuda de motores DC que requieren 35 V y 2 A para su funcionamiento.

De forma similar al Scorbot, el CRS-A255 utiliza encoders para su posicionamiento, en este proyecto se afirma que:

> "Cada motor cuenta con un encoder tipo incremental, es decir, que solo puede realizar cuentas ascendentes o descendentes a partir de la posición actual de la articulación, por lo que siempre será necesario un proceso de inicialización manual".

El párrafo anterior expresa la dificultad que tienen los manipuladores con encoders incrementales para posicionarse cuando el robot se inicia, como información adicional hay que mencionar que el proceso de re calibración es también complejo.

La implementación del hardware del controlador se realizó en tres etapas: alimentación, control y potencia. Para la etapa de alimentación se realizó un circuito rectificador/regulador de corriente. En la etapa de potencia se diseñaron circuitos de modulación de ancho de pulso basados en el MOSFET IRF530; la etapa de control consta de varios micro controladores PIC en un arreglo maestro esclavo.

El software controlador del manipulador fue desarrollado en el ambiente de desarrollo integrado NETBEANS del lenguaje JAVA ya que en palabras de los autores del proyecto "Java es el lenguaje más utilizado en la carrera de Electrónica en Automatización y Control". El programa desarrollado es una interfaz gráfica permite verificar la conexión del controlador a la PC, operar tres de los cinco eslabones que posee el robot y la pinza además de regular la velocidad de movimiento de los eslabones (Maldonado, 2013)**.**

1.7.2 Control de movimiento del manipulador de cinco grados de libertad CRS-A255 por esqueletización

El objetivo de este proyecto fue desarrollar un sistema capaz de reconocer la posición de un brazo real con una cámara web y a través de un algoritmo de esqueletización posicionar un manipulador CRS-A255 en la misma posición que el brazo real.

El hardware utilizado en este proyecto consta de dos partes. La primera corresponde al controlador del CRS-A255 que permitirá el movimiento de los eslabones del robot mediante cinemática directa. La otra parte del hardware consta de dos cámaras web que capturan la posición del brazo real que posteriormente será procesada por el software de control.

El software de control fue desarrollado en el lenguaje de programación C++ para aprovechar la librería OpenCV para visión computarizada en tiempo real y la disponibilidad de librerías de comunicación serial. La interfaz gráfica se realizó utilizando el programa Qt Designer y los cálculos necesarios para implementar el algoritmo de esqueletización y la cinemática directa se realizan con funciones estándar de C++ (Rea, 2012).

1.7.3 Control computarizado de brazo robótico

Este proyecto fue desarrollado en la Universidad de Victoria, Columbia británica, Canadá. El objetivo del proyecto fue desarrollar un controlador de bajo costo para un brazo robótico de cinco grados de libertad y un actuador final.
El manipulador seleccionado para este proyecto fue el Armatron robot que originalmente se opera mediante un controlador analógico.

El primer paso fue reemplazar los motores y la circuitería del controlador original por motores a pasos y drivers para estos motores; cada uno de los controladores incluye un registro de cambios bidireccional 74LS194 y un arreglo de transistores Darlington ULN2003A.

La comunicación entre el circuito de control y la PC se logra mediante el puerto paralelo y un decodificador/demultiplexor que interactúa directamente con los componentes antes nombrados; no existe tarjeta madre como tal.

Para la realización de la interfaz computacional se utilizó Visual Basic; en esta aplicación consta de una interfaz gráfica para llevar a cabo las siguientes operaciones (Rasouliha, 2004):

- Movimiento independiente de cada motor
- Información del estado de cada motor
- Control mediante cinemática directa del robot
- Configuración de los registros

1.7.4 Desarrollo de software para controlar el Scorbot ER VII con una PC

Este proyecto fue desarrollado en el Superior School of Technology of the Polytechnic Institute of Viseu, Portugal. El objetivo del proyecto fue desarrollar un software de control para un Scorbot ER VII

En este caso se contaba con el robot y el manipulador por lo que las actividades realizadas se orientaron al desarrollo del programa controlador

La interfaz gráfica fue desarrollada en el Borland C++ Builder; el programa permite la selección del puerto de comunicación, conocer los ángulos en los que se encuentra cada eslabón, conocer las coordenadas del punto final del robot, control de la velocidad de los eslabones y además se ha colocado la opción de ubicar el robot en una posición HOME. Las funciones utilizadas para el control del robot son las siguientes:

- Speed(): cambia la velocidad de los eslabones del robot
- Home(): posiciona el robot en la posición HOME
- MotorsOff(): apaga los motores del robot
- MotorsOn(): activa los motores del robot
- Close(): cierra el gripper
- Open(): abre el gripper

La aplicación contiene además ventanas para visualización del estado de las entradas y salidas y la configuración manual de cada eje del robot (Ferrolho, 2005).

1.7.5 Control de un brazo robótico con un sistema operativo en tiempo real

Este proyecto fue desarrollado en la Universidad de Cantabria, España. El objetivo general fue diseñar y construir un controlador para un Scorbot ER V.

El Scorbot ER V comparte las mismas características que su antecesor el Scorbort ER III; tiene el mismo número de grados de libertad, los mismos actuadores y sensores.

Se hace mención a que en condiciones normales los motores del manipulador no consumen más de 0,65 amperios de corriente por lo que para el control de los motores se seleccionó el driver L6206 de STMicroelectronic; este driver maneja voltajes entre 8 y 52 V una corriente máxima cercana a los 3 A además de ser capaz de operar con señales PWM de hasta 100kHz

La tarjeta madre fue implementada con un CPLD (dispositivo lógico programable complejo); para este proyecto se seleccionó el modelo XC95288-XL-10-TQ144 ya que este dispositivo ofrece todas las características de almacenamiento, entradas y salidas necesarias para la escritura de las rutinas del robot.

El software utilizado consta de dos partes: un módulo de núcleo o kernel informático que realiza el control del robot y la interfaz gráfica desarrollada con la librería GTK. El software está diseñado para correr sobre plataformas basadas en Linux.

El software desarrollado permite las siguientes funciones:

- Control mediante cinemática inversa basada en el algoritmo Denavit-Hartenberg.
- Control mediante cinemática inversa: este programa permite calcular un máximo de dos soluciones posibles y siempre se seleccionara la que deje el codo hacia arriba.
- Control de trayectorias: en este caso se desarrollaron algoritmos para el movimiento punto-punto y lineal.
- Control PID: el programa controla un lazo de control cerrado para la operación de los motores (Martínez, 2008).

1.7.6 Control de un brazo robótico de cinco grados de libertad con Arduino

Este proyecto fue desarrollado en la Universidad de Valladolid, España. Su objetivo general fue diseñar y construir un controlador para operar el robot Velleman KSR10 de cinco grados de libertad desde una PC.

El robot Velleman KSR10 es un robot de pequeña grama controlado originalmente por una serie de contactores que activan o desactivan los motores de cada eje; entre todos los motores no sobrepasan 1 A de consumo.

Para la construcción del controlador se utilizó como tarjeta madre un Arduino Mega 2560 ya que constituye una tarjeta ideal para la interfaz entre dispositivos electrónicos y la PC además de ser de código abierto. Adicional a la utilización de la tarjeta Arduino se recurrió a los drivers L293D que permiten controlad la inversión de giro y la velocidad de los motores.

Para la programación del Arduino se utilizaron librerías para el puerto serial, para realizar funciones matemáticas y PID para el posicionamiento de los eslabones

Los sensores utilizados son resistencias variables adaptadas a cada uno de los ejes a través de varillas metálicas.

La HMI del proyecto se desarrolló en el software .NET y permite controlar manualmente el robot así como guardar pequeñas secuencias para la operación automática (Yague, 2013).

1.7.7 Control de un brazo robótico teleoperado mediante acelerómetro

Este proyecto fue desarrollado en la Universidad Autónoma de México; su objetivo principal fue realizar el control de un Scorbot ER V en una configuración maestro-esclavo mediante un acelerómetro y dos placas Arduino.

Como ya se ha comentado, el Scorbot ER V es similar al Scorbot ER III. Posee 5 motores DC de 12 V; nuevamente se hace mención a que los motores no consumen más allá de 1 A cada uno por lo que se empleó el driver L293D la etapa de potencia.

El controlador del robot tiene dos partes: el maestro y el esclavo. La parte correspondiente al maestro consta de un Arduino Duemilanove y un acelerómetro BMA180 que permitirá controlar el robot en modo manual. El esclavo está formado por un Arduino Mega, la circuitería de potencia para los motores y la circuitería necesaria para la lectura de los encoders del brazo.

La interfaz gráfica y las instrucciones de control del robot fueron desarrolladas con ayuda de C#; el software permite realizar las siguientes actividades:

- Cinemática directa
- Cinemática inversa
- Trayectorias punto-punto
- Trayectorias lineales
- Trayectorias circulares

Una característica importante de este proyecto es que se prescindió del cable de comunicación entre la PC y el robot gracias a la ayuda de antenas Xbee que proveen de comunicación inalámbrica al Scorbot (Holguín, 2011).

1.7.8 Control de brazo robótico a través del movimiento humano utilizando acelerómetros

Este proyecto se desarrolló en el National Institute of Technology of Rourkela, India. Su objetivo principal fue desarrollar un brazo robótico que emule los movimientos de un brazo humano real mediante la utilización de motores DC

para las juntas del brazo y acelerómetros para monitorear la posición de los diferentes “eslabones” del brazo humano.

Para la construcción del brazo se utilizaron seis motores DC HITEC modelo HS-5065MG: dos para el hombro, dos para el codo y dos para la muñeca; para el control del brazo robótico se utilizó el micro controlador ATmega640; este dispositivo es capaz de suministrar el voltaje y corrientes necesarios para el funcionamiento de todos los servos por lo que no se requiere el empleo de drivers adicionales.

Debido a la utilización de motores DC, el brazo robótico no requiere sensores adicionales ya que cada servo se encuentra en lazo cerrado mediante un potenciómetro de precisión.

Para la lectura de la posición del brazo humano fueron utilizados dos acelerómetros MMA7361L y un micro controlador ATmega32 que realiza la adquisición y acondicionamiento de la señal de los sensores.

Las funciones de control así como la interfaz gráfica que permitirán operar todo el sistema fueron desarrolladas en Matlab; este software es una poderosa herramienta computacional que además de la gestión de los micro controladores permite la visualización de datos en tiempo real y la realización de cálculos matemáticos complejos con instrucciones simples; sin embargo Matlab es software comercial (Pattnaik, 2010).

1.7.9 Control de un brazo robótico industrial basado en acelerómetros

Este proyecto fue desarrollado por miembros de la IEEE. El objetivo de este proyecto fue presentar una alternativa al controlador de movimientos en la programación de brazos robóticos industriales.

El brazo en cuestión es el MOTOMAN HP6; este robot posee características y capacidades industriales y su control es realmente complejo por lo que se optó por seleccionar un controlador comercial para su operación. El controlador seleccionado es el NX100 puede conectarse directamente a la computadora como a una gran cantidad de dispositivos.

Para la lectura de la posición del brazo humano se utilizó una tarjeta tipo ANNs (artificial neural networks) que fue programada y entrenada para leer y acondicionar los datos de las salidas de los acelerómetros.

Los acelerómetros utilizados son los ADXL330 que poseen una buena relación costo-precisión.

El programa desarrollado para la administración de todo el sistema fue escrito para la plataforma Windows en forma de DLL y permite el posicionamiento del robot de acuerdo con las lecturas de los acelerómetros (Neto, 2010).

1.7.10 Diseño, Construcción y Control de movimiento de un manipulador de 6 grados de libertad para aplicaciones industriales

Este proyecto fue desarrollado por la Rajamangala University of Technology Thanyaburi. Su principal objetivo fue desarrollar un robot de capacidades industriales para el desarrollo de la materia de robótica dentro de la universidad.

La estructura del robot incluye 6 motores DC que trabajan a 24 V y 6 A por lo que los drivers de bajo nivel quedaron descartados. Debido a la alta precisión que se requiere en los movimientos del robot, se seleccionó un controlador comercial que corresponde al SPiiPlus Motion Controller. Para la parte de instrumentación fueron seleccionados sendos encoders incrementales que se conectan directamente al controlador. La interfaz gráfica está desarrollada en Visual C++, la gran ventaja que se presento es que existe una librería escrita en C para el controlador SpiiPlus que permite controlar el robot sin necesidad de tener conocimientos profundos de programación (Tuntivivat, 2011).

1.7.11 Análisis de resultados

La Tabla 1 muestra el cuadro comparativo de las diferentes tecnologías utilizadas en los proyectos relacionados. Para la ponderación de cada uno de los proyectos se ha decidido evaluar el número de funciones desarrolladas en el robot dándole un punto por cada función.

Tabla 1

Cuadro comparativo de la tecnología utilizada en los proyectos relacionados

Proyecto	Brazo robótico	Actuadores	Sensores	Controlador		Funciones	Ponderación
				Hardware	Software de desarrollo		
1	CRS-A255	Motores DC 35 V, 2 A	Encoders incrementales	Controlador original	NETBEANS, JAVA	Cinemática directa	1
2	CRS-A255	Motores DC 35 V, 2 A	Encoders incrementales	Controlador original	Qt Creator, C++	Visión artificial, cinemática directa	2
3	Armatron	Motores a pasos 12 V, 1 A	-	74LS194, ULN2003A	Visual Basic, Basic	Cinemática directa	1
4	Scorbot ER VII	Motores DC 12 V, 2 A	Encoders incrementales	Controlador original	Borland C++ Builder, C++	Cinemática directa, cinemática inversa, HOME	3
5	Scorbot ER V	Motores DC 12 V, 2 A	Encoders incrementales	CPLD XC95288-XL-10-TQ144 , L6206	GTK, multilenguaje	Cinemática directa, cinemática inversa, control de trayectorias, PID	4
6	Velleman KSR10	Motores DC 12 V, 1 A	Potenciómetros	Arduino Mega, L293D	Visual .NET, Basic	Cinemática directa	1
7	Scorbot ER V	Motores DC 12 V, 2 A	Encoders incrementales, acelerómetro	Arduino Mega, L293D	C#, C	Cinemática directa, cinemática inversa, control de trayectorias	3
8	Diseño y construcción	Motores DC 5V	Acelerómetros	ATmega640	Matlab, M	Cinemática directa	1
9	MOTOMAN HP6	Motores DC 200 W	Encoders incrementales, Acelerómetros	NX100	-	Cinemática directa	1
10	Diseño y construcción	Motores DC 24 V, 6 A	Encoders incrementales	SPiiPlus	Visual C++, C++	Cinemática directa, cinemática inversa, control de trayectorias	3

Como puede apreciarse en la Tabla 1, existe una amplia variedad de controladores y sensores que pueden ser utilizados para controlar el Scorbot ER III. El proyecto con mayor ponderación (4) posee un controlador CPLD (Complex Programmable Logic Device). Los dispositivos CLDP son difíciles de adquirir en el país por lo que se procede a descartar este proyecto y se tendrá en cuenta los siguientes con más alta puntuación. El siguiente proyecto con más alta puntuación (3) utiliza el controlador y sensores originales del robot por lo que también se descarta ya que se busca implementar un controlador genérico. El siguiente proyecto, con una puntuación de 3, está basado en un controlador SPiiPlus que es específico para robótica pero se maneja con software propietario por lo que se descarta esta opción. El siguiente proyecto analizado, con una puntuación de 3, utiliza como controlador una tarjeta Arduino Mega y los sensores originales del robot y se presenta como la mejor opción para implementar un controlador genérico ya que las tarjetas Arduino son de fácil adquisición en Ecuador y es un dispositivo capaz de interactuar con una amplia variedad de software tanto propietario como libre.

CAPÍTULO II
DISEÑO E IMPLEMENTACIÓN DEL HARDWARE DE CONTROL

2.1 Requerimientos del controlador

Los requerimientos del controlador corresponden a características adicionales al control del robot que son necesarias para la utilización del sistema. Estos requerimientos se indican a continuación:

- **Voltaje de alimentación**

  El voltaje de alimentación más utilizado para componentes electrónicos en Ecuador es 110 VAC a 60 Hz.

- **Tipo de comunicación**

  El tipo de comunicación más extendido en ordenadores en la actualidad es USB por lo que el controlador deberá soportar este tipo de comunicación.

- **Precisión**

  La precisión es un factor que depende del estado de los componentes mecánicos, de la tecnología de los componentes del controlador y del tipo de controlador implementado. La pérdida de precisión debida al deterioro de los componentes mecánicos puede aplacarse mejorando el sistema de control. Por lo tanto el controlador deberá ofrecer la máxima precisión que permitan los componentes que lo integran.

- **Paro de emergencia**

  Es necesario que el controlador cuente con este dispositivo para interrumpir la operación del robot ante posibles choques o pérdida de control.

- **Temperatura de operación**

  La temperatura de operación del controlador será la correspondiente al laboratorio de CAD/CAM de la ESPE-L que oscila entre 10 y 25 grados centígrados.

- **Entradas adicionales**

  El controlador original del SCORBOT-ER III dispone de entradas para dos juegos completos de motores. El nuevo controlador debe contar con las siguientes entradas adicionales:

  - Entradas de alimentación para actuadores.
  - Entradas de alimentación para tarjetas controladoras de motores

- Entradas de control para actuadores
- Entradas para lectura de encoders de cuadratura

2.2 Selección de los componentes del controlador

En el controlador genérico del SCORBOT-ER III está conformado principalmente por los siguientes componentes:

- Tarjeta Madre
- Tarjetas controladoras de actuadores
- Fuente de alimentación

Cada uno de estos componentes se selecciona a continuación.

2.2.1 Tarjetas controladoras de los motores

Todos los actuadores del manipulador SCORBOT ER III son motores DC, estos dispositivos deben impulsar la carga a la que se encuentren conectados tanto en sentido horario como anti-horario. Si solamente fuese requerido el movimiento bidireccional el control podría realizarse con puentes H; sin embargo, en manipuladores cuyos eslabones tienen un peso que no puede ser despreciado es necesario utilizar circuitos PWM para regular la potencia entregada cuando los eslabones se mueven a favor o en contra del campo gravitacional. A continuación se estudian algunas tarjetas disponibles para el control de los actuadores del SCORBOT-ER III.

A. Roboclaw 2x5A

Este dispositivo es capaz de controlar dos motores DC mediante comunicación serial, radiofrecuencia o señales analógicas. Puede entregar 5 A continuos por canal y 10 A como máximo (SERVOCITY, s.f.).

Características:

- Alimentación: 6-30V
- Corriente: 2x5A (10A máx.)
- Modos de funcionamiento: Serial TTL, RC, analógico
- Entradas para retroalimentación con PID
- 5V BEC (con jumper)

Ventajas

- Limitación automática de corriente
- Monitorización de batería para evitar sobre descarga
- Freno regenerativo
- Tolera cambios de dirección a alta velocidad

Desventajas

- Costo elevado
- Requiere importación
- Incorpora un micro controlador por lo que su programación es compleja.

B. Sabertooth 2x5A

Con este dispositivo pueden controlarse dos motores de corriente continua a través de comunicación serial, radiofrecuencia o señales analógicas (Dimension Engineering, s.f.).

Características:

- Alimentación: 6-18 V
- Corriente: 2x5 A (10 A máx.)
- Modos de funcionamiento: Serial TTL, RC, analógico
- Entradas para retroalimentación con PID
- 5V BEC (con jumper)
- Soporte de encoders de cuadratura

Ventajas

- Protección sobre corriente
- Protección sobre voltaje
- Freno regenerativo
- Tolera cambios de dirección a alta velocidad

Desventajas

- Costo elevado
- Requiere importación
- Incorpora un micro controlador por lo que su programación es compleja.
- No puede ser utilizado con fuente de alimentación

C. MotorShield V3

Esta es una placa diseñada por Arduino para controlar motores de baja potencia. Está basada en el driver L298 que permite controlar la velocidad y sentido de giro de dos motores DC o de un motor a pasos (ARDUINO, s.f.).

Características:

- Alimentación: 5-12 V
- Corriente: 2x2 A (4 A máx.)
- Compatible con tarjetas Arduino
- Integra sensor de corriente

Ventajas

- Costo moderado
- Función de paro de emergencia
- Fácil programación

Desventajas

- Requiere importación
- No integra retroalimentación

D. L298N dual drive

Este módulo basado en el chip L298N permite controlar dos motores de corriente continua o un motor paso a paso bipolar de hasta 2 amperios. Cuenta con todos los componentes necesarios para funcionar sin necesidad de elementos adicionales, entre ellos diodos de protección y un regulador LM7805 que suministra 5V a la parte lógica del integrado L298N (ELECTRONILAB, s.f.).

Características:

- Alimentación: 5-12 V
- Corriente: 2x1 A (2 A máx.)
- Compatible con la mayoría de micro controladores

Ventajas

- Costo moderado
- Fácil programación

Desventajas

- Requiere importación
- No integra retroalimentación

E. Evaluación de las tarjetas controladoras de los motores

Para evaluar las diferentes alternativas de tarjetas controladoras se ponderará el precio y la disponibilidad de estos dispositivos de acuerdo a los siguientes criterios:

Precio:

- Si la tarjeta tiene un precio bajo se le otorgará un puntaje de 10
- Si la tarjeta tiene un precio medio se le otorgará un puntaje de 5
- Si la tarjeta tiene un precio alto se le otorgará un puntaje de 0

Dificultad de adquisición:

- Si la dificultad de adquisición de la tarjeta es baja se le otorgará un puntaje de 10
- Si la dificultad de adquisición de la tarjeta es media se le otorgará un puntaje de 5
- Si la dificultad de adquisición de la tarjeta es alta se le otorgará un puntaje de 0

La Tabla 2 muestra el análisis comparativo de las tarjetas estudiadas en este apartado.

Tabla 2

Diferentes tarjetas que cumplen los requerimientos de los actuadores del SCORBOT ER III

Modelo	**Marca**	**Vmáx (V)**	**Imáx (A)**	**Precio (USD)**	**Ponderación**	Dificultad **de adquisición**	**Ponderación**	**Ponderación total**
Roboclaw 2x5A	Orion Robotics	34	5	Alto	0	Media	5	5
Sabertooth 2X5	DimensionEngineering	18	5	Alto	0	Media	5	5
Motor Shield V3	Arduino	12	2	Alto	0	Media	5	5
L298N dual driver	STMicroelectronics	12	2	Bajo	10	Baja	10	20
a14010200ux0040	UX Motor	24	2	Bajo	10	Alta	0	10
L6206	STMicroelectronics	52	2.8	Medio	5	Media	5	10
L293D	STMicroelectronics	36	1	Bajo	10	Media	5	15

Del análisis de la Tabla 2 se desprende que el dispositivo con mejor puntuación es el L298N dual debido a su facilidad en la adquisición tanto en disponibilidad como en precio por lo que se selecciona este dispositivo para controlar los motores del brazo robótico.

2.2.2 Tarjeta madre

Para leer las señales de los interruptores de referencia del SCORBOT son necesarios 6 pines digitales. Para leer las señales de los encoders son necesarios 12 pines digitales. La tarjeta madre debe poseer como mínimo 6 pines analógicos y más de 30 pines digitales, debe soportar comunicación USB y debe ser capaz de interactuar con software libre. Algunas alternativas para la selección de la tarjeta madre se estudian a continuación.

A. PIC 18F4550

El pic18F4550 pertenece a la gama alta de microprocesadores PIC y posee el mayor número de entradas y salidas (MICROCHIP, s.f.).

Características:

- 30 pines disponibles para entrada o salida
- Voltaje de alimentación: 5 V
- Soporta comunicación serial
- Capaz de interactuar con software libre

B. RASPBERRY PI

RASPBERRY PI es una tarjeta electrónica basada en un microprocesador Broadcom (ADAFRUIT, s.f.).

Características:

- Voltaje de alimentación: 5 V
- 17 pines disponibles para entrada y salida con capacidad de expansión
- Sistema operativo nativo
- Soporta comunicación serial
- Ideal para interactuar con software libre

C. ARDUINO MEGA 2560

Una alternativa que ofrece gran cantidad de pines digitales y analógicos y gran compatibilidad con software libre son las tarjetas ARDUINO (ARDUINO, s.f.); El modelo MEGA de estas tarjetas satisface todos los requerimientos que el SCORBOT ER III requiere para su completo funcionamiento; este dispositivo incluye 10 pines de comunicación, 11 pines de salida analógica, 16 pines de entrada analógica y 32 pines de entrada y salida digital. La Figura 17 muestra la distribución de pines del ARDUINO MEGA.

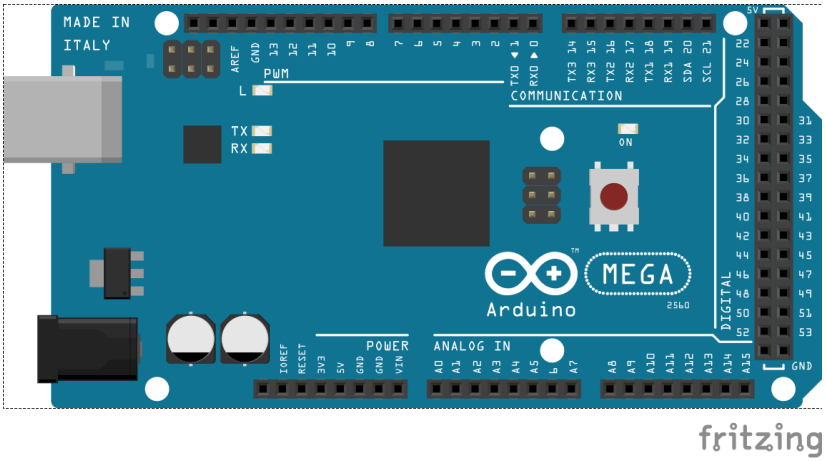


Figura 17. Distribución de pines del ARDUINO MEGA.

Fuente (FRITZING, s.f.)

D. Evaluación de las tarjetas madre

A fin de seleccionar la tarjeta madre se ponderan el número de pines disponibles y la complejidad para programar de acuerdo con los siguientes criterios:

Complejidad de programación:

- Si la complejidad de programación de la tarjeta es baja se otorgará un puntaje de 10.
- Si la complejidad de programación de la tarjeta es media se otorgará un puntaje de 5.
- Si la complejidad de programación de la tarjeta es alta se otorgará un puntaje de 0.

Número de pines:

- Si el número de pines es mayor a 40 se concederá un puntaje de 10.
- Si el número de pines esta entre 30 y 40 se concederá un puntaje de 5.
- Si el número de pines es menor a 30 se concederá un puntaje de 0.

La Tabla 3 muestra el análisis comparativo de las tarjetas estudiadas en este apartado.

Tabla 3

Diferentes tarjetas de control disponibles para este proyecto

Componente	Precio (USD)	Comunicación	Número de pines	Ponderación de número de pines	Complejidad de programación	Ponderación de complejidad de programación	Ponderación total
PIC 18F4550	12	Serial	30	5	Multilenguaje	5	10
Arduino MEGA	22	USB	59	10	Multilenguaje, IDE dedicada y libre	10	20
Raspberry Pi	75	USB	17	0	Requiere sistema operativo	0	0

El análisis de la Tabla 3 dio como resultado que la tarjeta madre más adecuada para el desarrollo de este proyecto es la Arduino Mega ya que posee un gran número de pines, posee una complejidad de programación baja y puede enlazarse con software libre.

2.2.3 Fuente de poder

Para la selección de la fuente de poder ha de tomarse en cuenta que los motores del SCORBOT funcionan a 12 voltios y tienen un consumo de corriente máximo de 2 amperios; en el caso improbable de que todos los motores funcionen a máxima capacidad se requerirá una fuente de 12 voltios y 12 amperios. La potencia de la fuente debe ser entonces de 144W. Los encoders de cuadratura requieren 5 voltios para su funcionamiento y el consumo de corriente puede considerarse despreciable en comparación con los 12 amperios que teóricamente consumirían los

motores. A continuación se analizan algunas alternativas de selección de la fuente del proyecto.

A. Fuente lineal

Este tipo de fuentes son fáciles de construir y tienen buen desempeño en baja potencia. Los valores típicos que ofrece este tipo de fuentes son 5, -5, 12 y -12 V. También es posible implementar salidas con voltaje regulable. La fuente puede fabricarse a medida por un costo aproximado de 25 dólares.

B. Fuente conmutada

Las fuentes conmutadas son más ligeras y eficientes que las lineales pero su diseño es más complejo. Tienen su principal utilización en aplicaciones de alta potencia. Los valores típicos de este tipo de fuentes son 5 y 12 V. Una fuente conmutable con un tamaño apropiado para este proyecto puede adquirirse por 60 dólares.

C. Fuente ATX

Las fuentes diseñadas para equipos de computación son una solución muy cómoda para aplicaciones de baja potencia. Estas fuentes tienen un acondicionamiento de voltaje muy superior al que se puede conseguir en una fuente lineal. Los valores que pueden obtenerse de esta fuente son 3, 5, -5, 12 y -12 V. Una fuente ATX apropiada para este proyecto puede adquirirse por 15 dólares.

D. Evaluación de las alternativas de fuente de poder

A fin de seleccionar la fuente de alimentación más adecuada para este proyecto se ponderan el número de valores disponibles y el costo de los dispositivos de acuerdo con los siguientes criterios:

Valores de voltaje disponibles:

- Si la fuente provee más de dos valores de voltaje se asignará un puntaje de 10.
- Si la fuente provee dos valores de voltaje se asignará un puntaje de 5.
- Si la fuente provee un valor de voltaje se asignará un puntaje de 0.

Costo de la fuente:

- Si el costo de la fuente es bajo se asignará un puntaje de 10.
- Si el costo de la fuente es medio se asignará un puntaje de 5.
- Si el costo de la fuente es alto se asignará un puntaje de 0.

La Tabla 4 muestra el análisis comparativo de las fuentes estudiadas en este apartado.

Tabla 4

Diferentes fuentes de alimentación disponibles para este proyecto

Fuente	Beneficios	Voltajes disponibles (V)	Ponderación de voltajes disponibles	Costo (USD)	Ponderación de costo	Ponderación total
Lineal	Diseño Simple	5, -5, 12, -12, voltaje regulable	10	Medio	5	15
Conmutada	Mayor eficiencia	5 o 12	0	Alto	0	0
ATX	Diseño confiable	3, 5, -5, 12 y -12	10	Bajo	10	20

La selección de la fuente recae sobre una ATX standard de 750 W que ofrece una corriente máxima de 38 amperios para 12 voltios y 30 amperios para 5 voltios.

2.3 Implementación del controlador

La implementación del controlador del SCORBOT ER III puede realizarse teniendo como elemento base la tarjeta Arduino MEGA y comprende las siguientes tareas:

- Conexión de las tarjetas controladoras de los motores
- Conexión de los sensores
- Conexión de la fuente de alimentación

2.3.1 Conexión de las tarjetas de los motores

Los pines de control del driver L298N se muestran en la Figura 18. IN1 e IN2 dan las señales de control para un puente H de acuerdo al voltaje que se les aplique: 5 o 0 voltios. El pin ENA recibe una señal PWM que regula la velocidad del motor

conectado a las terminales OUT1 y OUT2. IN3, IN4, ENB, OUT3 y OUT4 son análogos a los pines IN1, IN2, ENA, OUT1 y OUT2 respectivamente. Para controlar los seis motores del SCORBOT ER III con drivers L298N serán necesarios 18 pines: 12 pines digitales y 6 pines analógicos.

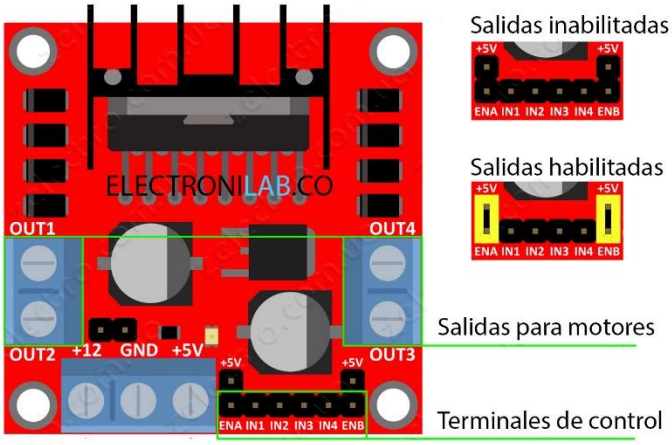


Figura 18. Pines de control del driver L298N.

Fuente (ELECTRONILAB, s.f.)

La distribución de pines en la conexión de la tarjeta ARDUINO MEGA y los drivers de los motores se muestra en la Tabla 5.

Tabla 5

Distribución de pines para la conexión de la tarjeta ARDUINO MEGA y los drivers de los motores

Motor	Pin ARDUINO	Tipo	Tarjeta	Conexión
Base	22	Digital	1	IN1
	23	Digital	1	IN2
	2	Analógico	1	ENA
Brazo	24	Digital	1	IN3
	25	Digital	1	IN4
	3	Analógico	1	ENB
Antebrazo	26	Digital	2	IN1
	27	Digital	2	IN2
	4	Analógico	2	ENA
Muñeca	28	Digital	3	IN1
	29	Digital	3	IN2
	30	Digital	3	ENA

CONTINÚA

	31	Digital	3	IN3
	5	Analógico	3	IN4
	6	Analógico	3	ENB
Pinza	32	Digital	2	IN3
	33	Digital	2	IN4
	7	Analógico	2	ENB

2.3.2 Conexión de los sensores

La lectura de los encoders del SCORBOT ER III requiere un pin digital por cada uno de ellos. La distribución de pines en la conexión de la tarjeta ARDUINO MEGA para la utilización de los sensores se muestra en la Tabla 6 y el diagrama de conexión se muestra en el Anexo 1.

Tabla 6

Distribución de pines en la conexión de la tarjeta ARDUINO MEGA para la utilización de los sensores.

Sensor		Pin ARDUINO	Tipo
Interruptor de la base		34	Digital
Interruptor del hombro		35	Digital
Interruptor del codo		36	Digital
Interruptor de la muñeca		37	Digital
Interruptor de la pinza		38	Digital
Encoder de la base	Canal A	39	Digital
	Canal B	40	Digital
Encoder del hombro	Canal A	41	Digital
	Canal B	42	Digital
Encoder del codo	Canal A	43	Digital
	Canal B	44	Digital
Encoder de la muñeca	Canal A	45	Digital
	Canal B	46	Digital
Encoder de la pinza	Canal A	47	Digital
	Canal B	48	Digital

2.3.3 Conexión de los puertos DB25

El controlador original del Scorbot utiliza un puerto DB50 para operar el brazo. Como este tipo de conector es poco comercial se decidió repartir las conexiones en dos puertos DB25. La Figura 19 muestra la implementación de los puertos en el controlador genérico de este proyecto.

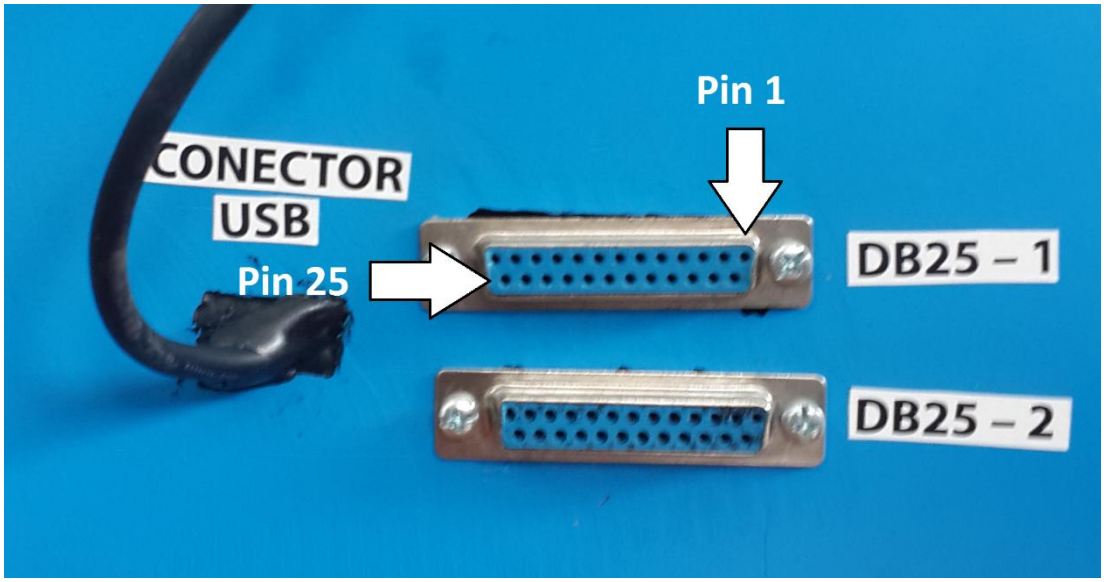


Figura 19. Implementación de los conectores DB25.

La Tabla 7 indica la utilización de los dos puertos DB25.

Tabla 7

Distribución de pines en la conexión de los puertos DB9

Puerto	Pin	Función
DB1	1	Alimentación del motor de la base (-)
	2	Alimentación del motor del brazo (-)
	3	Alimentación del motor del antebrazo (-)
	4	Alimentación del primer motor de la muñeca (-)
	5	Alimentación del segundo motor de la muñeca (-)
	6	Alimentación del motor de la muñeca (-)
	7	Alimentación del interruptor de la base (-)
	8	Alimentación del interruptor del brazo (-)
	9	Alimentación del interruptor del antebrazo (-)
	10	Alimentación del primer interruptor de la muñeca (-)

CONTINÚA

	11	Alimentación del segundo interruptor de la muñeca (-)
	12	n/c
	13	n/c
	14	Alimentación del motor de la base (+)
	15	Alimentación del motor del brazo (+)
	16	Alimentación del motor del antebrazo (+)
	17	Alimentación del primer motor de la muñeca (+)
	18	Alimentación del segundo motor de la muñeca (+)
	19	Alimentación del motor de la muñeca (+)
	20	Alimentación del interruptor de la base (+)
	21	Alimentación del interruptor del brazo (+)
	22	Alimentación del interruptor del antebrazo (+)
	23	Alimentación del primer interruptor de la muñeca (+)
	24	Alimentación del segundo interruptor de la muñeca (+)
	25	n/c
DB2	1	Alimentación del encoder de la base (-)
	2	Alimentación del encoder del brazo (-)
	3	Alimentación del encoder del antebrazo (-)
	4	Alimentación del primer encoder de la muñeca (-)
	5	Alimentación del segundo encoder de la muñeca (-)
	6	Alimentación del encoder del gripper (-)
	7	n/c
	8	Señal del encoder de la base
	9	Señal del encoder del brazo
	10	Señal del encoder del antebrazo
	11	Señal del encoder de la muñeca
	12	Señal del encoder de la pinza
	13	n/c
	14	Alimentación del encoder de la base (+)
	15	Alimentación del encoder del brazo (+)
	16	Alimentación del encoder del antebrazo (+)
	17	Alimentación del primer encoder de la muñeca (+)
	18	Alimentación del segundo encoder de la muñeca (+)

CONTINÚA ⇨

19	Alimentación del encoder del gripper (+)
20	n/c
21	n/c
22	n/c
23	n/c
24	n/c
25	n/c

2.3.4 Conexión de entradas y salidas adicionales

Para conectar hardware y dispositivos adicionales, se instalaron las siguientes salidas adicionales:

- 8 entradas analógicas
- 8 pines digitales que pueden ser utilizados como entradas o salidas
- 4 salidas PWM
- Una conexión de 12 V
- Una conexión de 5V
- Una conexión a GND

La Figura 20 muestra la implementación de las salidas adicionales en el controlador genérico.

Figura 20. Implementación de las entradas y salidas adicionales.

2.3.5 Conexión de la fuente

Las fuentes de poder ATX son utilizadas en equipos de computación y para ser utilizadas en proyectos electrónicos es necesario que se conecte a tierra el cable verde correspondiente a PS-ON tal como se muestra en la Figura 21.

Figura 21. Habilitación de la fuente ATX.

Los voltajes requeridos por el controlador del SCORBOT son +12 y +5 V. Para la adecuada regulación de estos voltajes entre los componentes electrónicos del robot se confecciono la tarjeta de distribución mostrada en la Figura 22.

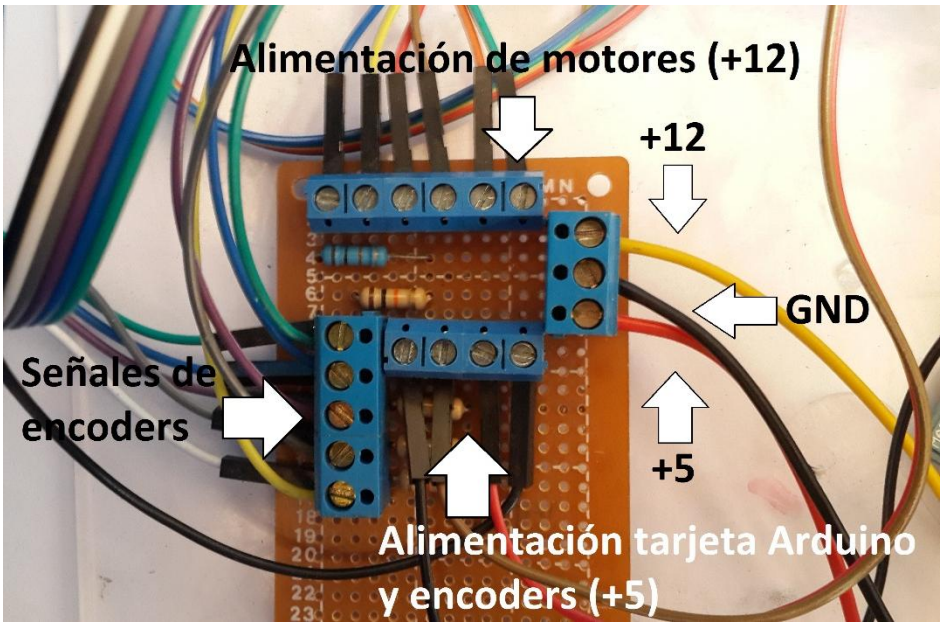


Figura 22. Placa de distribución de voltaje.

Como puede apreciarse en la Figura 22, se han utilizado los cables negro, rojo y amarillo correspondientes a tierra, +5 y +12 V respectivamente. Además esta

tarjeta cuenta con bornes para la alimentación de los motores, componentes de control y lectura de las señales de los encoders. Es muy importante que se respete el orden de los cables ya que de otro modo los componentes del robot sufrirán daños irreversibles. El Anexo 1 detalla la conexión de este dispositivo con el resto del controlador.

2.3.6 Integración de los componentes del controlador

Como tarea final en la implementación del controlador se añadió un paro de emergencia que corta la alimentación de las tarjetas de los motores y una luz led que indica que el controlador está en funcionamiento. La Figura 23 muestra el controlador genérico implementado.

a

b

Figura 23. Integración de los componentes del controlador (a) exterior, (b) interior.

CAPÍTULO III
DISEÑO E IMPLEMENTACIÓN DEL SOFTWARE DE CONTROL

3.1 Requerimientos funcionales

El controlador genérico para el SCORBOT-ER III debe cumplir las siguientes funciones:

- Cinemática directa
- Cinemática inversa
- Interpolación lineal
- Interpolación circular
- Posición HOME
- Programación de secuencias
- Movimiento paso a paso
- Movimiento continuo
- Control de apertura de la pinza

Todas las funciones indicadas deben desarrollarse en software libre.

3.2 Selección del software de programación

Para la selección del software de programación el principal criterio de selección es que sea libre, esto quiere decir que todas sus librerías estén disponibles. Entre los programas más populares dentro de esta categoría destacan los siguientes:

- NetBeans
- Visual Python
- Blender

3.2.1 NetBeans

NetBeans es una IDE libre y sin restricciones de uso. Los programas desarrollados en esta IDE están conformados por conjuntos de componentes de software llamados módulos.

Ventajas:

- Multiplataforma
- Alta funcionalidad
- Entorno amigable

Desventajas:

- Solo acepta programación orientada a objetos
- Requiere cambios significativos para migrar de una plataforma a otra

3.2.2 Visual Python

Visual Python es una interfaz de desarrollo que permite confeccionar aplicaciones dedicadas a ingeniería con base en el lenguaje Python. Visual Python no posee restricciones en su utilización y es libre. Provee importantes librerías matemáticas y de modelamiento 3D.

Ventajas:

- Multiplataforma, no requiere cambios significativos para migrar
- Multiparadigma
- Alta funcionalidad

Desventajas:

- Requiere más tiempo que otras aplicaciones para ejecutar códigos complejos.

3.2.3 Blender

Blender es una suit de diseño, simulación, creación y optimización de proyectos en 3D. Este software posee una interfaz de programación que le permite crear código para controlar sistemas reales que hayan sido virtualizados.

Ventajas:

- Multiplataforma, no requiere cambios significativos para migrar
- Multiparadigma
- Alta funcionalidad
- Interfaz amigable

Desventajas:

- Alta complejidad de programación
- Requiere gran espacio de memoria

3.2.4 Evaluación de las alternativas de software de control

Una vez que se han analizado las alternativas de software libre se procede a evaluar sus características en la Tabla 8 de acuerdo con los siguientes criterios:

Complejidad:

- Si la complejidad de programación es baja se otorgará un puntaje de 10.
- Si la complejidad de programación es media se otorgará un puntaje de 5.
- Si la complejidad de programación es alta se otorgará un puntaje de 0.

Velocidad de trabajo:

- Si la velocidad de trabajo es alta se otorgará un puntaje de 10.
- Si la velocidad de trabajo es media se otorgará un puntaje de 5.
- Si la velocidad de trabajo es baja se otorgará un puntaje de 0.

Funcionalidad (que mide el número de funciones disponibles para programar):

- Si la funcionalidad es alta se otorgará un puntaje de 10.
- Si la funcionalidad es media se otorgará un puntaje de 5.
- Si la funcionalidad es baja se otorgará un puntaje de 0.

Tabla 8

Programas disponibles para el desarrollo de la HMI del controlador

Programa	Complejidad	Ponderación	Velocidad de trabajo	Ponderación	Funcionalidad	Ponderación	Ponderación total
Blender	Alta	0	Alta	10	Media	5	15
NetBeans	Media	5	Media	5	Media	5	15
VPython	Baja	10	Baja	0	Alta	10	20

En la Tabla se aprecia que VPython es el software con mayor puntaje (20) debido a que posee una complejidad de programación baja y porque tiene una alta funcionalidad por lo que se decide utilizar este programa para escribir el código de la HMI.

3.3 Cinemática directa del robot

3.3.1 Algoritmo

La cinemática directa de un manipulador hace referencia a "determinar cuál es la posición y orientación del extremo final del robot, con respecto a un sistema de

coordenadas que se toma como referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot" (Barrientos, 2007, pág. 94). Resolver el problema cinemático directo del Scorbot ER-III consiste en encontrar la posición del extremo de la pinza dados los ángulos de la base (base), hombro (shoulder), codo (elbow) e inclinación de la muñeca (pitch) indicados en la Figura 1. La Figura 24 muestra el análisis geométrico del SCORBOT ER-III para resolver la cinemática directa.

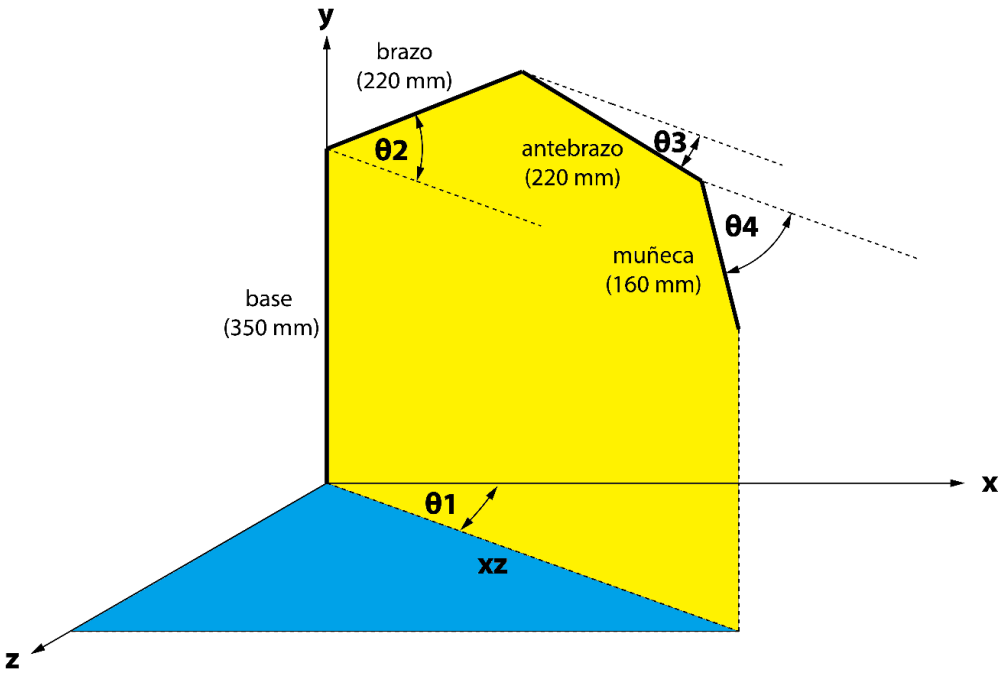


Figura 24. Análisis geométrico del SCORBOT ER-III para cinemática directa.

En la Figura 24 puede apreciase que los eslabones correspondientes al brazo, antebrazo y muñeca rotan alrededor del eje y sobre el plano amarillo un ángulo θ_1. La coordenada y del extremo del manipulador corresponde a la sumatoria algebraica de la todas las coordenadas y de los eslabones tal como se muestra en la Ecuación 3.1.

$$y_f = l_{base} + l_{brazo} * \sin\theta_2 + l_{antebrazo} * \sin\theta_3 + l_{muñeca} * \sin\theta_4$$

Ecuación 3.1

El valor de xz se calcula con la Ecuación 3.2

$$xz = l_{brazo} * \cos\theta_2 + l_{antebrazo} * \cos\theta_3 + l_{muñeca} * \cos\theta_4$$

Ecuación 3.2

La coordenada x del extremo del manipulador se calcula con la Ecuación 3.3 que atiende a la solución del triángulo rectángulo celeste

$$x_f = xz * \cos\theta_1$$

Ecuación 3.3

La coordenada z del extremo del manipulador se calcula con la Ecuación 3.4

$$z_f = xz * \sin\theta_1$$

Ecuación 3.4

El algoritmo para realizar la cinemática directa se describe a continuación:

- El robot debe alcanzar la posicion HOME.
- La lectura de los encoder se realiza de manera simultanea al movimiento del robot.
- La secuencia para calcular la posicion final del robot está enlazada con la rutina que detiene los motores de manera que este cálculo se realiza automaticamente cada vez que el usuario manipule los botones de movimiento.
- Cuando se ha terminado de mover algun eslabón, el programa calcula la posición final del robot.
- El programa imprime la posicion final del robot.

3.3.2 Software

La Figura 25 muestra la interfaz utilizada para la cinemática directa.

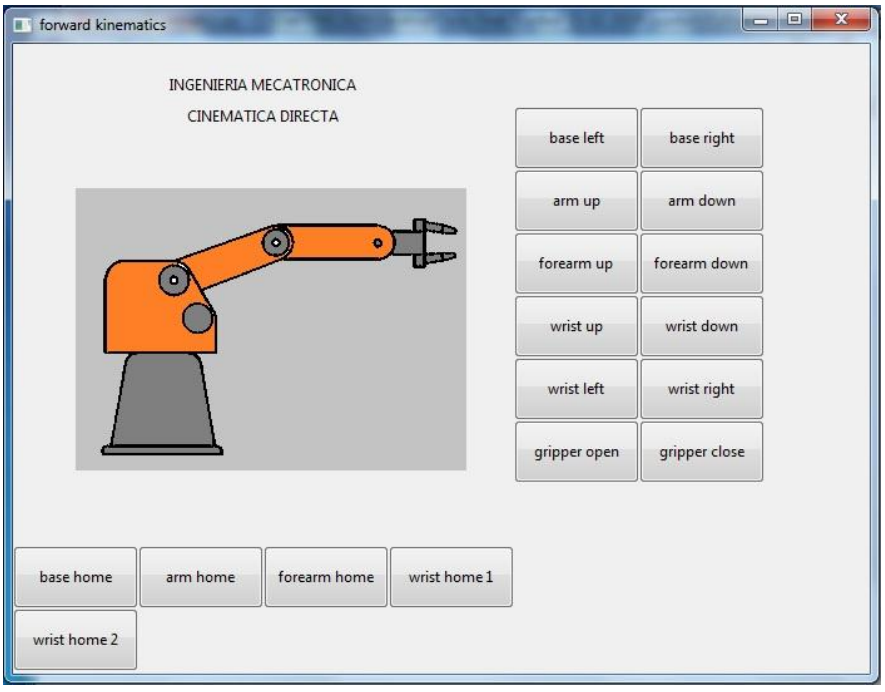


Figura 25. Pantalla HMI para cinemática directa.

La pantalla HMI de la cinemática directa contiene un modelo virtual del SCORBOT que permitirá visualizar los movimientos del robot desde la PC. Los botones que se encuentran debajo de la simulación permiten enviar el robot a su posición HOME. En caso que solo se requiera referenciar una sola articulación, la función HOME está dividida de acuerdo al número de articulaciones. Los botones a la derecha de la simulación permiten mover el robot de acuerdo a una velocidad que previamente debe fijarse.

3.4 Cinemática inversa del robot

3.4.1 Algoritmo

La cinemática inversa de un manipulador hace referencia a "determinar cuál es la configuración que debe adoptar el robot para una posición y orientación del extremo conocidas" (Barrientos, 2007, pág. 94). Resolver el problema cinemático inverso del Scorbot ER-III consiste en encontrar el valor de los ángulos de la base (base), hombro (shoulder), codo (elbow) e inclinación de la muñeca (pitch) dadas las coordenadas (x,y,z) del extremo del manipulador.

Para simplificar los cálculos realizados por el software puede darse como dato de entrada, además de las coordenadas (x,y,z), el ángulo de inclinación de la pinza. Esto no afecta la generalidad del diseño ya que para la mayoría de aplicaciones la inclinación de la pinza debe estar de acuerdo con la geometría del objetivo a asir.

La Figura 26 muestra el análisis geométrico del SCORBOT ER-III para resolver la cinemática inversa.

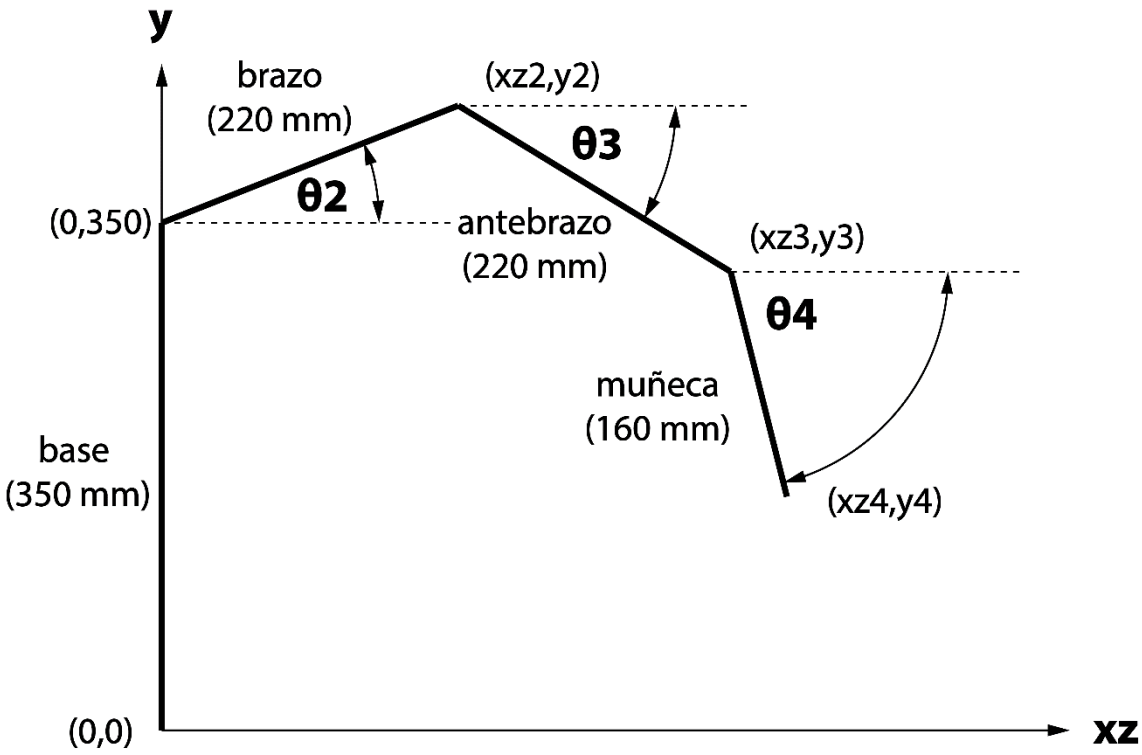


Figura 26. Análisis geométrico del SCORBOT ER-III para cinemática inversa.

Los datos de entrada para este problema son:

- $(x_0, y_0, z_0) = (0,0,0)$
- $(x_1, y_1, z_1) = (0,350,0)$
- (x_4, y_4, z_4) deben especificarse
- θ_4 debe especificarse

El primer valor que debe determinarse es xz_4. La Ecuación 3.5 permite calcular este valor.

$$xz_4 = \sqrt{{x_4}^2 + {z_4}^2}$$

Ecuación 3.5

Es posible determinar el ángulo de la base con la Ecuación 3.6.

$$\theta_1 = \tan^{-1}\left(\frac{z_4}{x_4}\right)$$

Ecuación 3.6

Conocidos los valores de xz_4, y_4 y θ_4 es posible conocer xz_3 y y_3 con la Ecuación 3.7 y la Ecuación 3.8.

$$xz_3 = xz_4 - l_{\text{muñeca}} * \cos\theta_4$$

Ecuación 3.7

$$y_3 = y_4 - l_{muñeca} * \sin\theta_4$$

Ecuación 3.8

Para determinar los valores de xz_2 y y_2 es necesario resolver el sistema conformado por la Ecuación 3.9 y la Ecuación 3.10.

$$(x_2 - xz_3)^2 + (y_2 - y_3)^2 = 220^2$$

Ecuación 3.9

$$x_2{}^2 + (y_2 - 350)^2 = 220^2$$

Ecuación 3.10

La resolucion del sistema formado por las ecuaciones 3.9 y 3.10 arroja los siguientes resultados:

$$xz_2 = \frac{-b \pm \sqrt{b^2 - 4 * a * c}}{2 * a}$$

$$y_2 = \frac{2 * xz_3 * xz_2 + y_1{}^2 - xz_3{}^2 - y_3{}^2}{2 * y_1 - 2 * y_3}$$

Donde:

$$a = 4 * y_1{}^2 - 8 * y_1 * y3 + 4 * xz_3{}^2 + 4 * y_3{}^2$$

$$b = -4 * y_1{}^2 * xz_3 + 8 * y_1 * xz_3 * y_3 - 4 * xz_3{}^3 - 4 * xz_3 * y_3{}^2$$

$$c = y_1{}^4 - 4 * y_1{}^3 * y_3 + 2 * y_1{}^2 * xz_3{}^2 + 6 * y_1{}^2 * y_3{}^2 - 4 * y_1{}^2 * r_1{}^2 - 4 * y_1 * xz_3{}^2 * y_3 - 4 * y_1 * y_3{}^3 + 8 * y_1 * y_3 * r_1{}^2 + xz_3{}^4 + 2 * xz_3{}^2 * y_3{}^2 + y_3{}^4 - 4 * y_3{}^2 * r_1{}^2$$

Para que los resultados sean fisicamente correctos debe verificarse que:

$$b^2 - 4 * a * c \geq 0$$

Despues los ángulos pueden calcularse como

$$A_1 = \tan^{-1}\left(\frac{y_2 - y_1}{xz_2 - xz_1}\right)$$

$$A_2 = \tan^{-1}\left(\frac{y_3 - y_2}{xz_3 - xz_2}\right)$$

En la solución de la cinemática inversa es muy importante verificar que los ángulos calculados se encuentren dentro de los valores permitidos por la geometría del robot. La Tabla 9 muestra los valores límite recomendados para los ángulos de las articulaciones del robot.

Tabla 9

Valores límite recomendados para los ángulos de las articulaciones del robot.

Ángulo	Valor mínimo	Valor máximo
Base	-80	80
Brazo	10	120
Antebrazo	-70	70
Muñeca	-90	0

3.4.2 Software

La Figura 27 muestra la interfaz utilizada para la cinemática inversa.

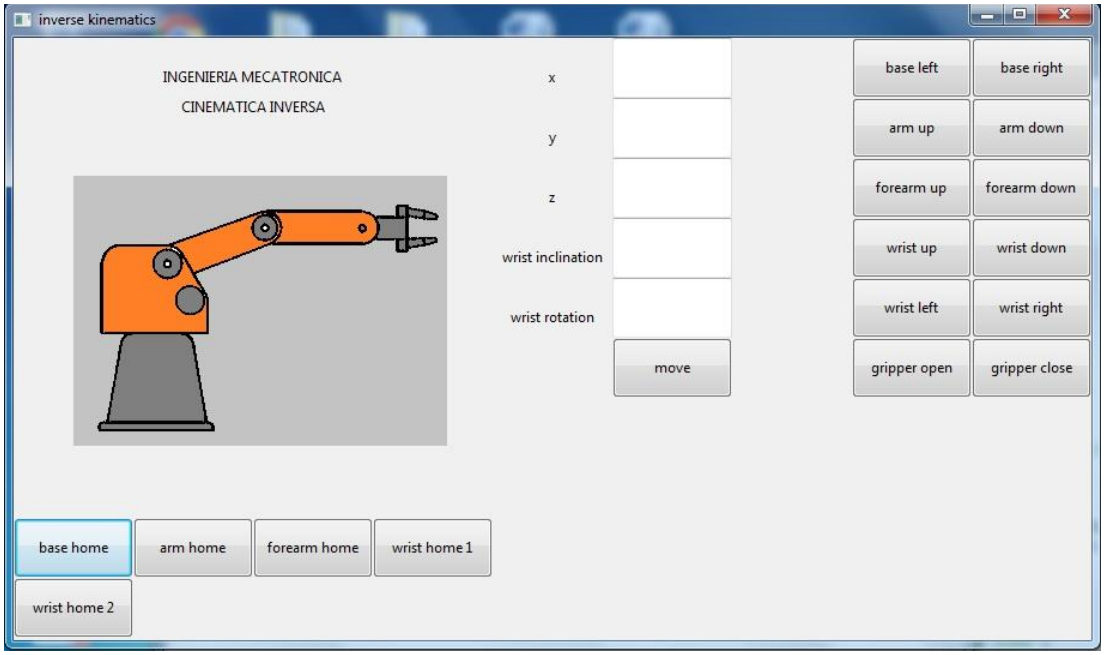


Figura 27. Pantalla HMI para cinemática inversa.

La HMI de la cinemática inversa requiere que se ingresen los valores de la posición del extremo del robot $P = (x, y, z)$ y los ángulos de inclinación y rotación de la pinza. Al iniciar este programa es necesario posicionar todas las articulaciones en HOME. Existe la posibilidad de que el robot se encuentre en una posición incómoda o que no le permita seguir moviéndose; para estos casos se dispone de los botones de la izquierda que permiten al usuario mover libremente el robot.

3.5 Posición HOME

La posición HOME de un manipulador debe seleccionarse de acuerdo con dos criterios:

- Los eslabones estén sometidos a poco esfuerzo. Esto quiere decir que no se encuentren en sus posiciones extremas.
- No existan singularidades en el robot. Esto quiere decir que no existan eslabones alineados.

Teniendo estos criterios en cuenta, ESHED ROBOTEC (1998) propone la posición mostrada en la Figura 28.

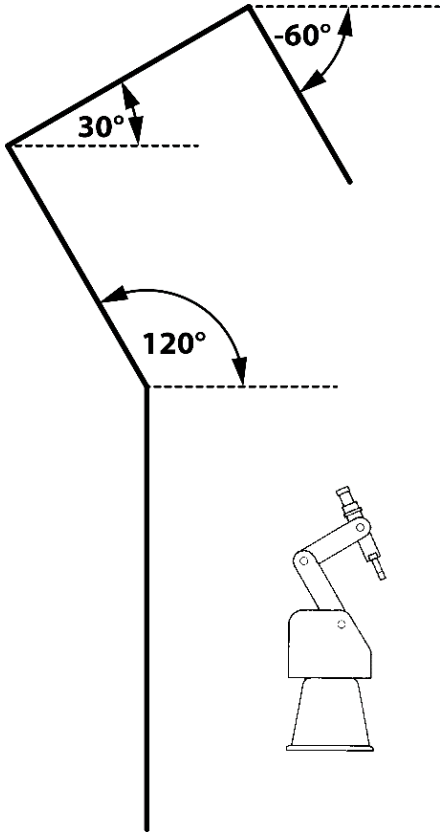


Figura 28. Posición HOME del Scorbot ER III

El Scorbot cuenta con interruptores que le ayudan a alcanzar la posición HOME. El interruptor de la base se encuentra colocado en la posición extrema derecha. Los interruptores del brazo, antebrazo e inclinación de la muñeca se sitúan en la posición extrema superior. El interruptor de rotación de la muñeca se coloca para que la muñeca quede en posición HOME. La pinza no posee interruptor. El algoritmo para alcanzar la posición HOME se describe a continuación:

BASE

- El motor de la base se mueve hacia la derecha
- El interruptor de la base se activa.
- El motor se detiene.
- El contador de la base se fija en -80°.
- El motor se mueve hacia la izquierda.
- El contador alcanza el valor 0°
- El motor se detiene

BRAZO

- El motor del brazo se mueve hacia arriba
- El interruptor del brazo se activa.
- El motor se detiene.
- El contador de la brazo se fija en 120°.

ANTEBRAZO

- El motor del antebrazo se mueve hacia arriba
- El interruptor del antebrazo se activa.
- El motor se detiene.
- El contador del antebrazo se fija en 120°.
- El motor se mueve hacia abajo.
- El contador alcanza el valor 30°
- El motor se detiene

MUÑECA

- El motor de la muñeca se mueve hacia arriba
- El interruptor 1 de la muñeca se activa.
- El motor se detiene.
- El contador 1 de la muñeca se fija en 30°.
- El motor se mueve hacia abajo.
- El contador alcanza el valor -60°
- El motor se detiene

La Figura 29 muestra el diagrama de flujo de la operación del robot para que la base alcance la posición HOME

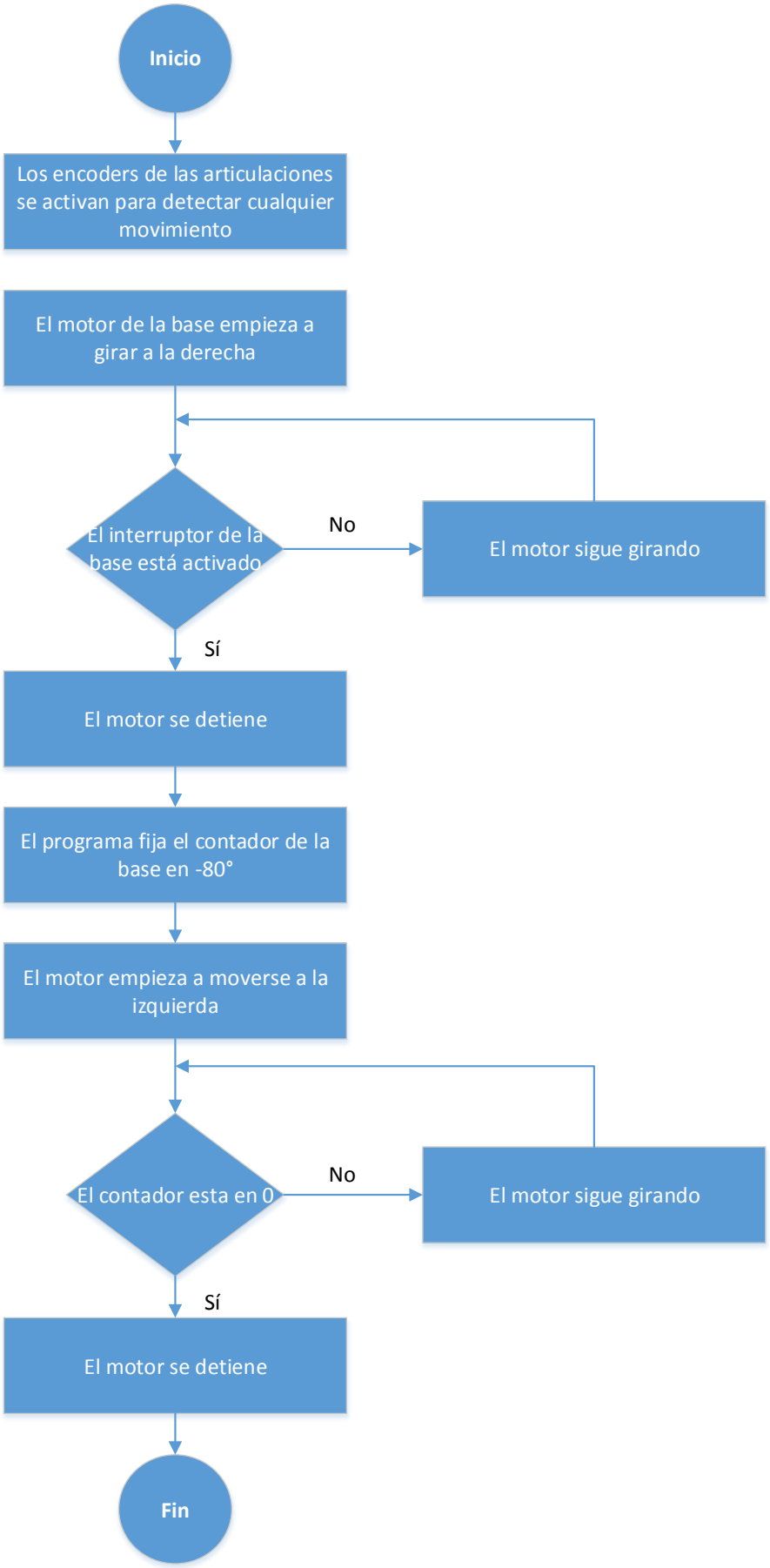


Figura 29. Diagrama de flujo para alcanzar la posición HOME

3.6 Movimiento secuencial

Como parte de las tareas que ejecuta un robot manipulador, debe ser capaz de reproducir una serie de posiciones grabadas por el usuario. El proceso se desarrolla como sigue:

- Al arrancar el programa inicia la lectura de los encoders, los contadores acoplados a estos se mantienen en cero mientras no exista movimiento.
- El usuario mueve el robot hasta la posición deseada.
- Se indica al programa que la posición debe guardarse.
- El programa almacena la lectura de los cinco encoders con funciones correspondientes denominadas readEncoder(); esta función además permite determinar si el motor de cada articulación se está moviendo en sentido horario o anti horario.
- Mientras se continúe guardando posiciones, el programa las apilará en forma de matriz de la siguiente manera

$$P_1 = (A_{10}, A_{11}, A_{12}, A_{13}, A_{14})$$
$$P_2 = (A_{20}, A_{21}, A_{22}, A_{23}, A_{24})$$
$$P_i = (A_{i0}, A_{i1}, A_{i2}, A_{i3}, A_{i4})$$

- El programa permite ejecutar la secuencia en modo paso o en modo continuo.
- Al activar el botón PASO, el programa tomará la primera y segunda posición, calculará el movimiento que debe realizar y moverá el robot. Cuando el robot termina la rutina, un contador incrementa en uno el índice de las posiciones guardadas y las deja listas para el próximo movimiento.
- Si se activa el botón CONTINUO, se ejecutarán tantas instrucciones PASO como posiciones se hayan guardado dentro de un lazo FOR para que las posiciones se actualicen continuamente.
- Al terminar de ejecutar todas las instrucciones, el programa borra de su memoria las posiciones almacenadas.

La Figura 30 muestra el diagrama de flujo de la operación del robot para el movimiento secuencial.

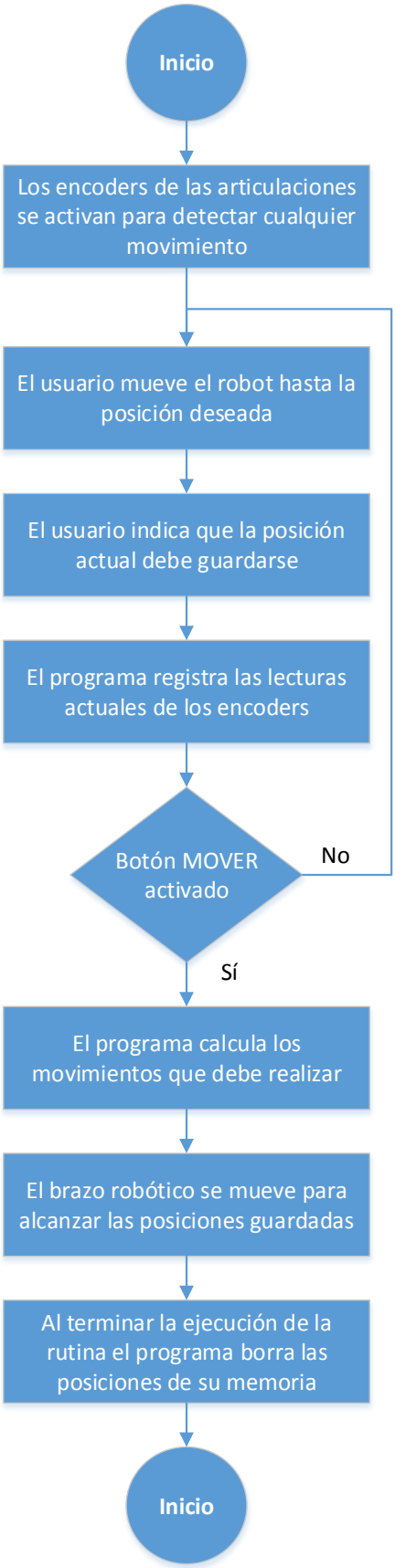


Figura 30. Diagrama de flujo para el movimiento secuencial del SCORBOT ER-III.

3.7 Interpolación lineal

Una línea es una sucesión continua de puntos. La representación de esta idea se muestra en la Figura 31.

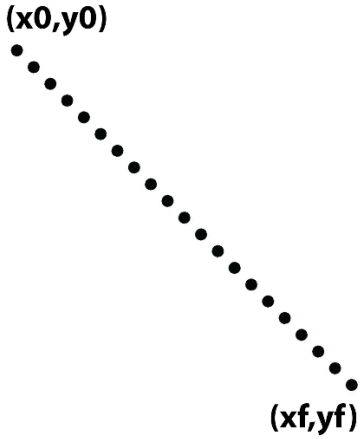


Figura 31. Representación de una recta como una sucesión de puntos.

3.7.1 Matriz de velocidades

Para hallar la velocidad del extremo del robot en cualquier momento debe partirse las ecuaciones que resuelven su posición. Para la coordenada x se tiene:

$$x = l_{brazo} * \cos\theta_1 * \cos\theta_2 + l_{antebrazo} * \cos\theta_1 * \cos\theta_3 + l_{muñeca} * \cos\theta_1 * \cos\theta_4$$

Ecuación 3.11

Derivando la Ecuación 3.11 se obtiene la velocidad del extremo del robot en la direccion x. El resultado se muestra a continuacion:

$$\dot{x} = -l_{brazo} * \cos\theta_1 * \sin\theta_2 * \dot{\theta_2} - l_{antebrazo} * \cos\theta_1 * \sin\theta_3 * \dot{\theta_3} - l_{muñeca} * \cos\theta_1 * \sin\theta_4 * \dot{\theta_4}$$

Ecuación 3.12

Por otro lado la velocidad en la direccion y nace de derivar la Ecuación 3.1

$$\dot{y} = l_{brazo} * \cos\theta_2 * \dot{\theta_2} + l_{antebrazo} * \cos\theta_3 * \dot{\theta_3} + l_{muñeca} * \cos\theta_4 * \dot{\theta_4}$$

Ecuación 3.13

Finalmente, la ecuacion de la posicion en z es:

$$z = l_{brazo} * \sin\theta_1 * \cos\theta_2 + l_{antebrazo} * \sin\theta_1 * \cos\theta_3 + l_{muñeca} * \sin\theta_1 * \cos\theta_4$$

Ecuación 3.14

La velocidad del extremo del robot en la dirección z se obtiene al derivar la Ecuación 3.14. El resultado se muestra a continuacion:

$$\dot{z} = l_{brazo} * \sin\theta_1 * \sin\theta_2 * \dot{\theta_2} + l_{antebrazo} * \sin\theta_1 * \sin\theta_3 * \dot{\theta_3} + l_{muñeca} * \sin\theta_1 * \sin\theta_4 * \dot{\theta_4}$$

Ecuación 3.15

La matriz de velocidades se arma a partir de las ecuaciones 3.12, 3.13 y 3.15. El resultado es el siguiente:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} -l_{brazo} * \cos\theta_1 * \sin\theta_2 & -l_{antebrazo} * \cos\theta_1 * \sin\theta_3 & l_{muñeca} * \cos\theta_1 * \sin\theta_4 \\ l_{brazo} * \cos\theta_2 & l_{antebrazo} * \cos\theta_3 & l_{muñeca} * \cos\theta_4 \\ l_{brazo} * \sin\theta_1 * \sin\theta_2 & l_{antebrazo} * \sin\theta_1 * \sin\theta_3 & l_{muñeca} * \sin\theta_1 * \sin\theta_4 \end{bmatrix} * \begin{bmatrix} \dot{\theta_1} \\ \dot{\theta_2} \\ \dot{\theta_3} \end{bmatrix}$$

$\dot{\theta_1}$, $\dot{\theta_2}$ y $\dot{\theta_3}$ son las velocidades angulares del brazo, antebrazo y muñeca respectivamente.

3.7.2 Matriz de aceleraciones

Las ecuaciones que rigen la aceleración del extremo del robot en las direcciones x, y y z son:

$$\ddot{x} = -l_{brazo} * \cos\theta_1 * \cos\theta_2 * \ddot{\theta_2} - l_{antebrazo} * \cos\theta_1 * \cos\theta_3 * \ddot{\theta_3} - l_{muñeca} * \cos\theta_1 * \sin\theta_4 * \ddot{\theta_4}$$

Ecuación 3.16

$$\ddot{y} = -l_{brazo} * \sin\theta_2 * \ddot{\theta_2} - l_{antebrazo} * \sin\theta_3 * \ddot{\theta_3} - l_{muñeca} * \sin\theta_4 * \ddot{\theta_4}$$

Ecuación 3.17

$$\ddot{z} = l_{brazo} * \sin\theta_1 * \cos\theta_2 * \ddot{\theta_2} + l_{antebrazo} * \sin\theta_1 * \cos\theta_3 * \ddot{\theta_3} + l_{muñeca} * \sin\theta_1 * \sin\theta_4 * \ddot{\theta_4}$$

Ecuación 3.18

La matriz de aceleraciones se escribe como sigue:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}$$

$$= \begin{bmatrix} -l_{brazo} * \cos\theta_1 * \cos\theta_2 & l_{antebrazo} * \cos\theta_1 * \cos\theta_3 & l_{muñeca} * \cos\theta_1 * \sin\theta_4 \\ -l_{brazo} * \sin\theta_2 & -l_{antebrazo} * \sin\theta_3 & -l_{muñeca} * \sin\theta_4 \\ l_{brazo} * \sin\theta_1 * \sin\theta_2 & l_{antebrazo} * \sin\theta_1 * \sin\theta_3 & l_{muñeca} * \sin\theta_1 * \sin\theta_4 \end{bmatrix}$$

$$* \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \\ \ddot{\theta}_3 \end{bmatrix}$$

$\ddot{\theta}_1$, $\ddot{\theta}_2$ y $\ddot{\theta}_3$ son las aceleraciones angulares del brazo, antebrazo y muñeca respectivamente.

3.7.3 Algoritmo de interpolación

El proceso de interpolación lineal se realiza como sigue:

- El usuario ingresa las coordenadas inicial y final de la recta que desea generarse.
- El programa genera las coordenadas de los puntos intermedios. Python posee la instrucción linspace para generar puntos intermedios entre dos valores. La utilización de la instrucción se ve a continuación:

  $x = \text{linspace}(x_0, x_f, 100)$

  $y = \text{linspace}(y_0, y_f, 100)$
- Cada pareja de coordenadas $(x[i], y[i])$ debe ser verificada por el programa para que el robot no realice movimientos ilegales
- Una vez todos los puntos se han verificado, se recurre a la rutina de cinemática inversa para mover el robot a cada uno de los puntos.
- El robot mueve sus articulaciones para alcanzar cada uno de los puntos desviando la trayectoria de la pinza en cada ejecución pero la separación entre puntos es tan pequeña que apenas se percibe.

La Figura 32 muestra el diagrama de flujo correspondiente a la interpolación lineal del Scorbot ER III

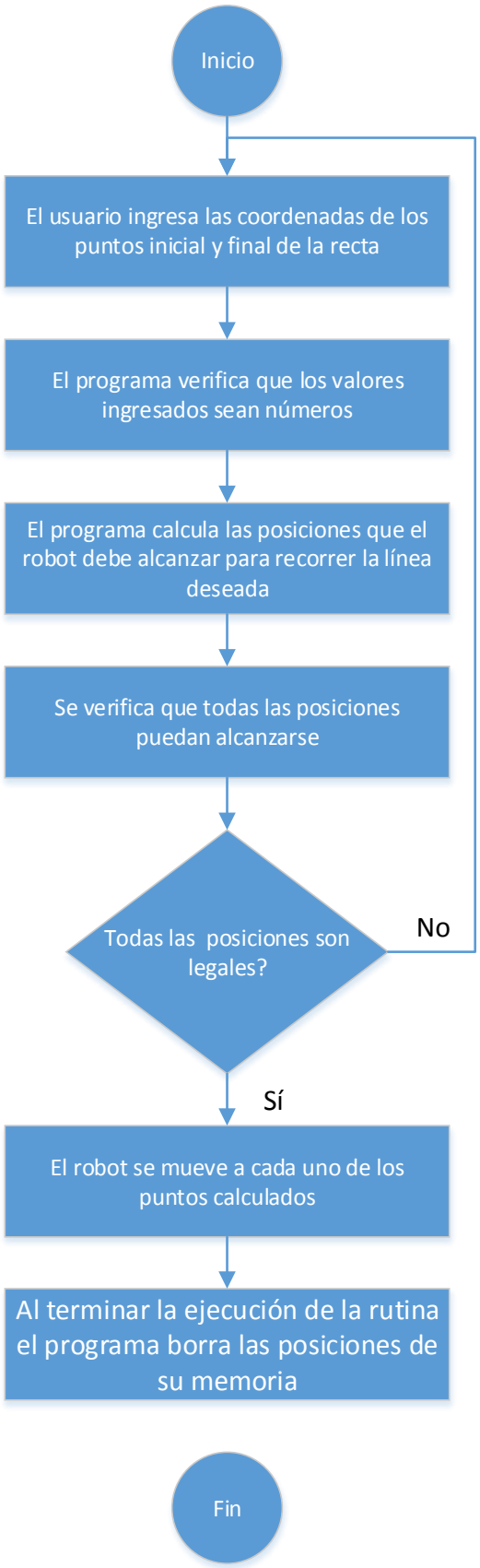


Figura 32. Diagrama de flujo para interpolación lineal del SCORBOT ER-III.

Para verificar el comportamiento del algoritmo de interpolación lineal, se corrió una simulación en el programa MATLAB. Para esta simulación se utilizaron los siguientes datos: $x_1 = 450,\ z_1 = 0,\ x_2 = 100,\ z_2 = 0$. Los principales comandos utilizados en esta simulación fueron:

- Linspace, que permite crear vectores con elementos igualmente espaciados
- Plot3, que permite graficar ternas de valores que tengan la forma (x, y, z)
- Algoritmo de cinemática inversa.

Los resultados de la simulación se aprecian en la Figura 33.

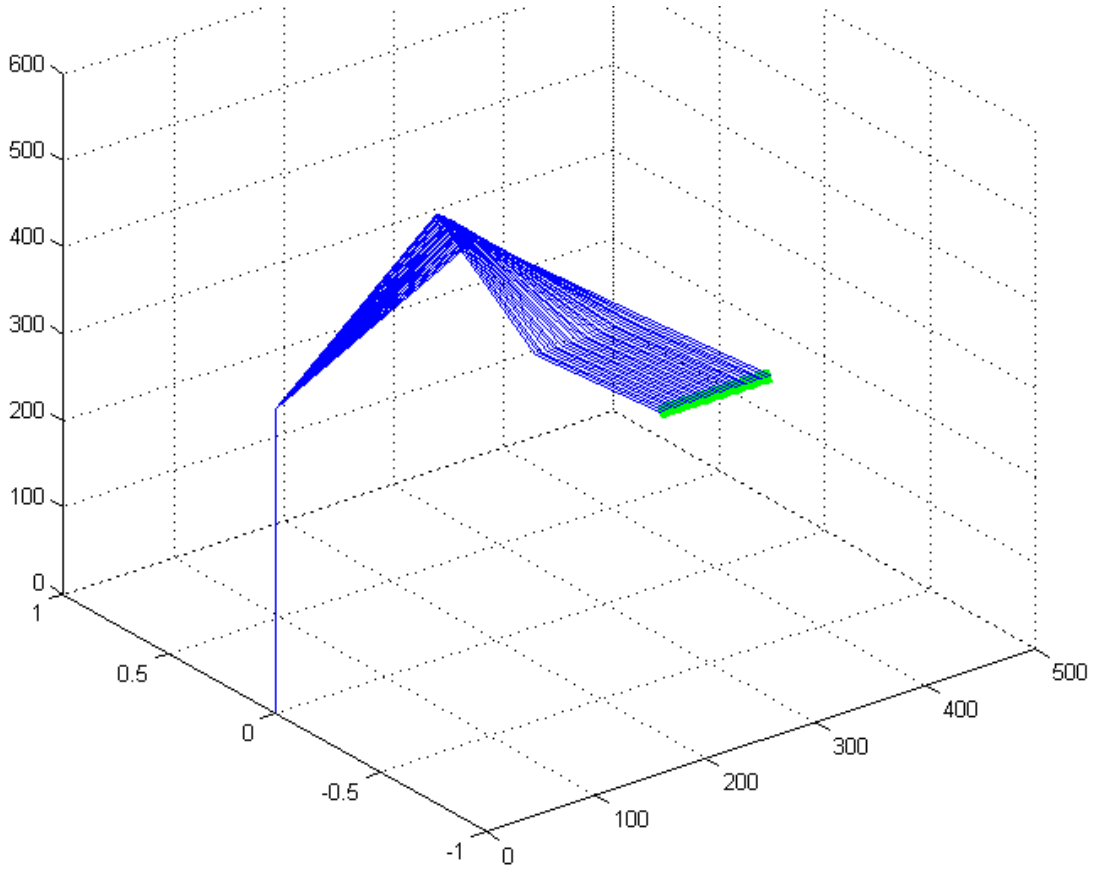


Figura 33. Verificación del comportamiento del algoritmo de interpolación lineal.

En la Figura 33 puede verse que se utilizaron segmentos de color azul para representar los eslabones del robot en cada una de las posiciones que fueron calculadas; la posición del extremo del robot en cada fase fue representada con puntos verdes resultando en una línea prácticamente continua.

3.8 Interpolación circular

Un círculo puede concebirse como una sucesión de puntos igualmente separados de uno llamado centro una distancia llamada radio. Esta idea se muestra en la Figura 34.

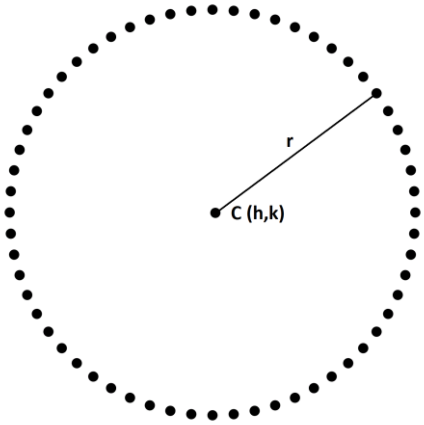


Figura 34. Representación de un círculo como una línea de puntos

La interpolación circular se realiza como sigue:

- El usuario ingresa las coordenadas del centro y el radio del círculo que desea reproducir.
    - Centro: $C(x = h, y = k)$
    - Radio r
- El programa genera las coordenadas de todos los puntos pertenecientes al perímetro de la figura de la siguiente forma: $x_i = r * \cos\theta_i + h,\ y_i = r * \sin\theta_i + h$, con $0 < \theta_i < 2 * \pi$
- El programa calcula todas las posiciones que el robot debe alcanzar con rutinas de cinemática inversa.
- El programa verifica que todas las posiciones calculadas sean legales.
- Si existe alguna posición ilegal se muestra un mensaje de error.
- Si todas las posiciones son legales el programa mueve el robot a cada una de las posiciones.

La Figura 35 muestra el diagrama de flujo correspondiente a la interpolación circular del Scorbot ER III

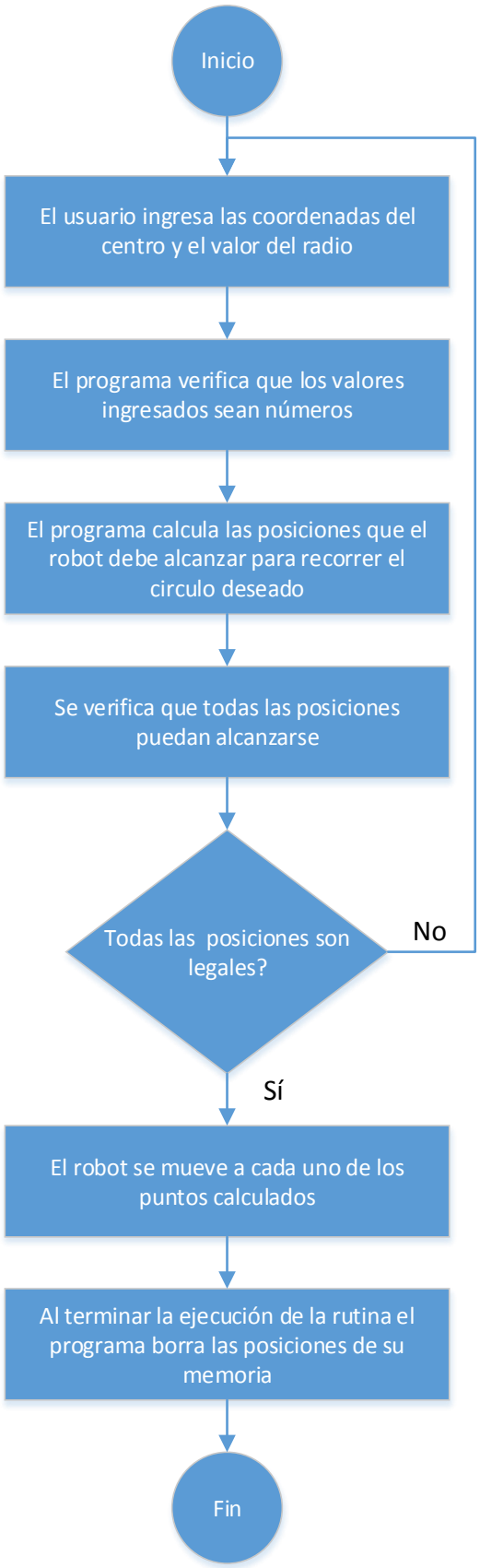


Figura 35. Diagrama de flujo para interpolación lineal del SCORBOT ER-III.

Para verificar el comportamiento del algoritmo de interpolación circular, se corrió una simulación en el programa MATLAB. Para esta simulación se utilizaron los siguientes datos: $x = h = 450$, $z = k = 0$, $r = 100$, $y = 200$. Los resultados de la simulación se aprecian en la Figura 36.

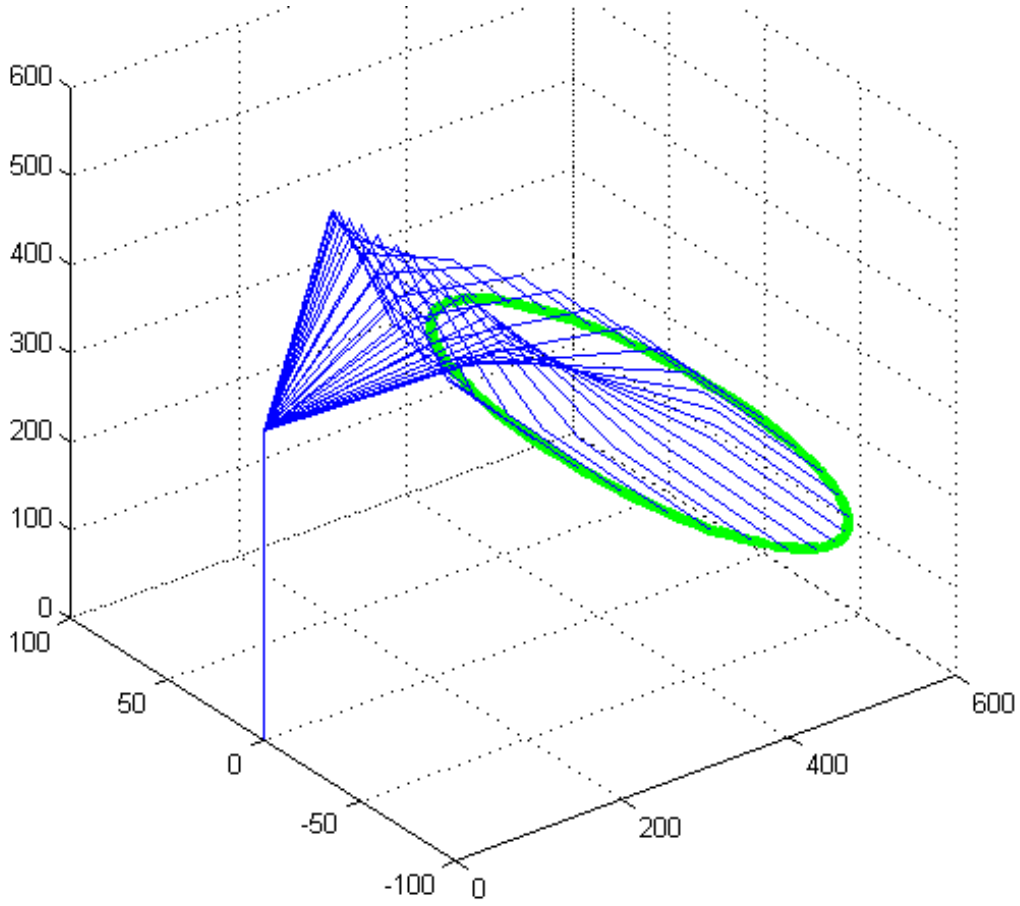


Figura 36. Verificación del comportamiento del algoritmo de interpolación circular.

CAPÍTULO IV
PRUEBAS Y RESULTADOS

4.1 Cinemática directa

El problema de la cinemática directa consiste en encontrar la posición del extremo del robot a partir del conocimiento de la posición de los eslabones. Para estudiar el funcionamiento de la cinemática directa del robot se utilizó el programa de cinemática directa.

4.1.1 Datos recolectados

La prueba inicia con el robot en posición HOME para desplazarse hacia el punto $P = (400,200,50)$. Los resultados de la prueba de exactitud del programa de cinemática directa se muestran en la Tabla 10 y la Figura 37.

4.1.2 Análisis de resultados

Para Analizar la cinemática directa se realizaron 30 pruebas, para lo cual se tuvo en cuenta el valor medido con y sin errores mecánicos, los cuales consisten en recolectar datos sin ayudar al manipulador y ayudando en la sujeción de elementos mecánicos que tengan desgaste, respectivamente; para de esta forma obtener el error absoluto generado por el software del nuevo controlador y determinar las causas que lo generan.

Tabla 10

Valores obtenidos en la prueba de exactitud de la cinemática directa

Prueba	Valor esperado (mm)			Valor con Error Mecánico (mm)			Valor sin Error Mecánico (mm)			Error Mecánico Absoluto (mm)			Error por software (mm)		
	X esperada	Y esperada	Z esperada	X medida	Y medida	Z medida	X medida	Y medida	Z medida	X	Y	Z	X	Y	Z
1	400	200	50	409	190	54	400	200	49,9999	7	13	0	0	0	0,0001
2	400	200	50	408	192	55	400	200	49,9999	7	14	0	0	0	0,0001
3	400	200	50	410	189	53	400	200	49,9999	6	14	0	0	0	0,0001
4	400	200	50	411	185	54	400	200	49,9999	7	13	0	0	0	0,0001
5	400	200	50	410	188	56	400	200	49,9999	7	14	0	0	0	0,0001
6	400	200	50	408	193	54	400	200	49,9999	6	13	0	0	0	0,0001
7	400	200	50	411	187	55	400	200	49,9999	7	14	0	0	0	0,0001
8	400	200	50	406	193	56	400	200	49,9999	7	13	0	0	0	0,0001
9	400	200	50	409	190	53	400	200	49,9999	6	14	0	0	0	0,0001
10	400	200	50	411	188	55	400	200	49,9999	6	14	0	0	0	0,0001
11	400	200	50	412	185	56	400	200	49,9999	7	13	0	0	0	0,0001
12	400	200	50	410	189	54	400	200	49,9999	7	14	0	0	0	0,0001
13	400	200	50	408	193	54	400	200	49,9999	6	13	0	0	0	0,0001
14	400	200	50	407	193	55	400	200	49,9999	6	13	0	0	0	0,0001
15	400	200	50	411	186	53	400	200	49,9999	7	13	0	0	0	0,0001
16	400	200	50	409	190	54	400	200	49,9999	6	13	0	0	0	0,0001
17	400	200	50	412	184	56	400	200	49,9999	7	14	0	0	0	0,0001
18	400	200	50	411	185	55	400	200	49,9999	7	14	0	0	0	0,0001
19	400	200	50	409	191	54	400	200	49,9999	7	14	0	0	0	0,0001
20	400	200	50	413	183	56	400	200	49,9999	7	14	0	0	0	0,0001

CONTINÚA ⇨

21	400	200	50	411	187	55	400	200	49,9999	7	14	0	0	0	0,0001
22	400	200	50	412	186	54	400	200	49,9999	7	13	0	0	0	0,0001
23	400	200	50	410	190	54	400	200	49,9999	6	13	0	0	0	0,0001
24	400	200	50	409	192	53	400	200	49,9999	6	13	0	0	0	0,0001
25	400	200	50	410	191	55	400	200	49,9999	7	14	0	0	0	0,0001
26	400	200	50	408	193	54	400	200	49,9999	6	13	0	0	0	0,0001
27	400	200	50	411	188	55	400	200	49,9999	7	14	0	0	0	0,0001
28	400	200	50	412	185	56	400	200	49,9999	7	13	0	0	0	0,0001
29	400	200	50	410	191	54	400	200	49,9999	7	14	0	0	0	0,0001
30	400	200	50	409	191	55	400	200	49,9999	7	14	0	0	0	0,0001

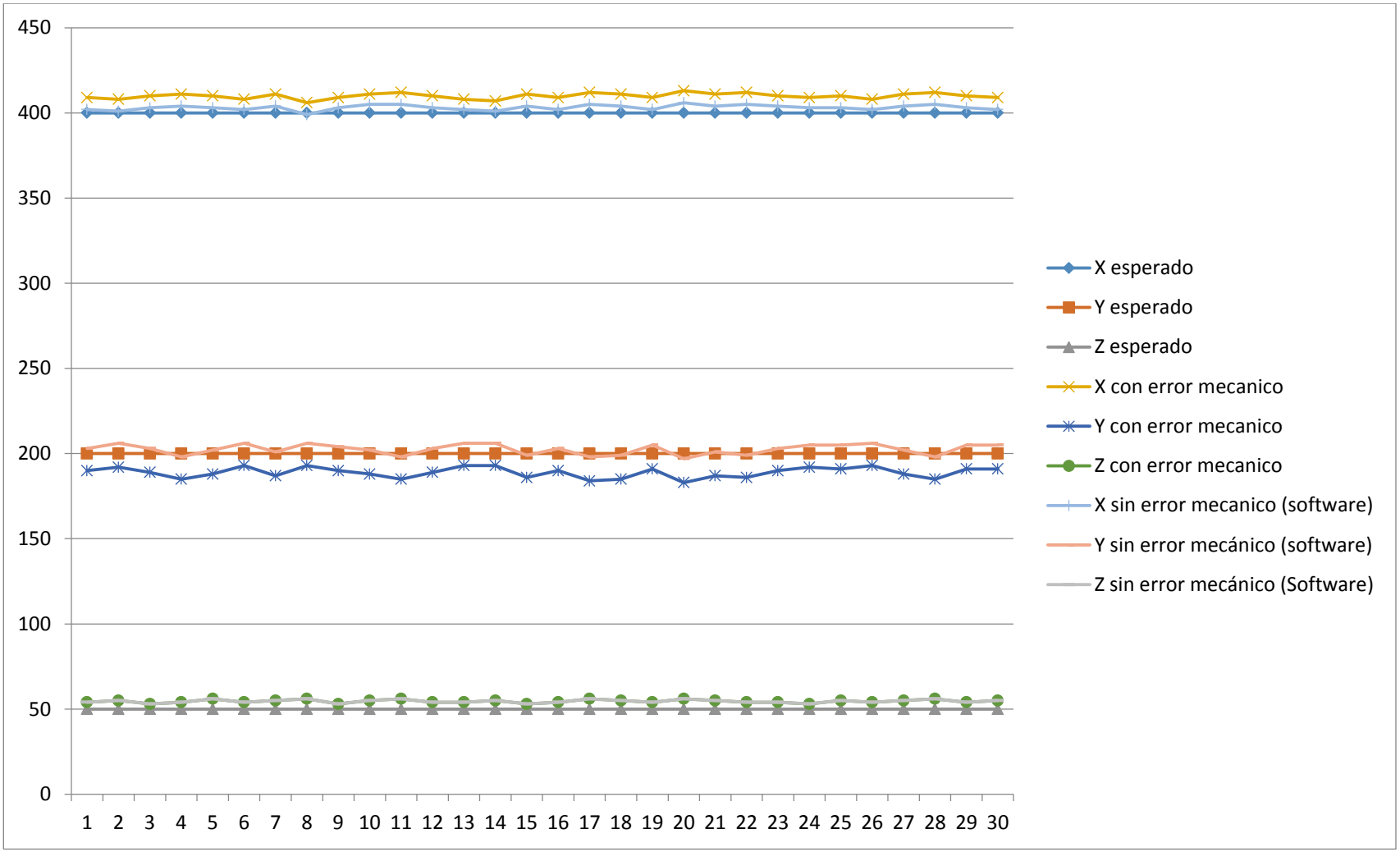


Figura 37. Mediciones de la posición alcanzada por el robot con cinemática directa.

De la Tabla 10 puede rescatarse que el Error Mecánico máximo cometido por el robot en el eje Y es de 14 mm, que representa el 7 % en cuanto a la distancia estudiada, en el eje X es de 7 mm, en Z no existen y en cuanto al error generado por el software del nuevo controlador es de 0,0001 mm como máximo en cada eje del manipulador. Esto se puede verificar gráficamente en la Figura 37.

El cálculo teórico de los pulsos entregados durante el movimiento de la base se ilustra en la Figura 38. En la práctica este valor se verifica, esto quiere decir que en una vuelta completa de la base se contaron 7662 pulsos; Este método de cálculo es válido para todos los eslabones excepto para el brazo en el que teóricamente deberían contarse 6129 pulsos pero en la práctica el valor obtenido oscila entre 4900 y 5200 pulsos; debido al desgaste en elementos mecánicos del manipulador, siendo esta la circunstancia más importante de error.

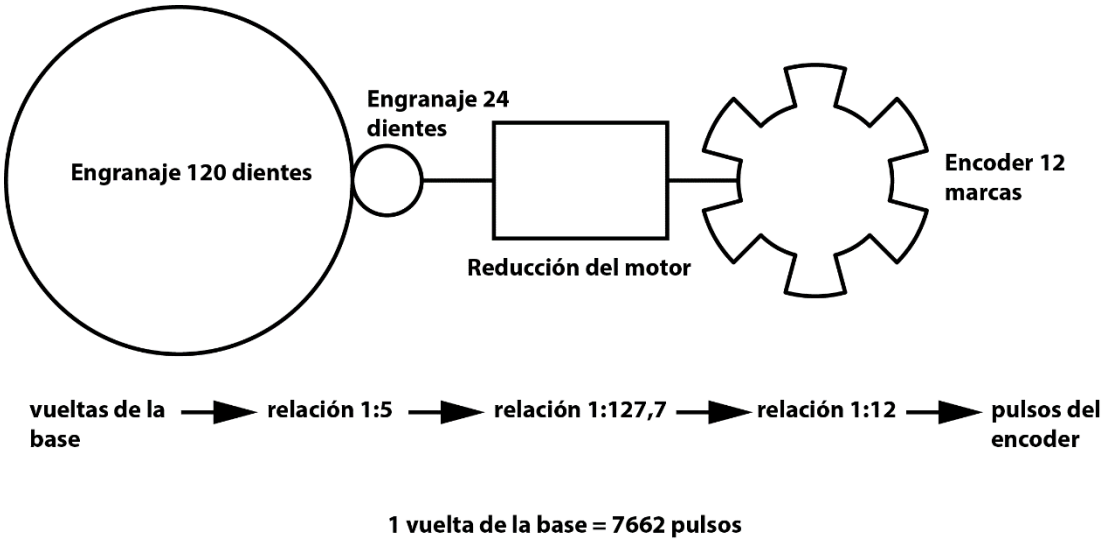


Figura 38. Verificación del comportamiento del algoritmo de interpolación circula.

4.2 Cinemática inversa

El problema de la cinemática inversa consiste en encontrar las posiciones de los eslabones del manipulador a partir de la posición del extremo. Para estudiar el funcionamiento de la cinemática inversa se utilizó el programa desarrollado en Python para este efecto.

4.2.1 Datos recolectados

En esta prueba el manipulador robótico Scorbot debe alcanzar la posición $P = (450,200,100)$. La Tabla 11 y la Figura 39 resumen los resultados obtenidos durante la prueba.

4.2.2 Análisis de resultados

Para Analizar la cinemática inversa se realizó 30 pruebas, para lo cual se tuvo en cuenta el valor medido con y sin errores mecánicos, los cuales consisten en recolectar datos sin y con ayudar al manipulador en la sujeción de elementos mecánicos que tengan desgaste, respectivamente; y así obtener el error absoluto generado por el software del nuevo controlador y determinar las causas que lo generan.

Tabla 11

Valores obtenidos en la prueba de exactitud de la cinemática inversa

Prueba	Valor esperado (mm)			Valor con Error Mecánico (mm)			Valor sin Error Mecánico (mm)			Error Mecánico Absoluto (mm)			Error por software (mm)		
	X esperada	**Y esperada**	**Z esperada**	**X medida**	**Y medida**	**Z medida**	**X medida**	**Y medida**	**Z medida**	**X**	**Y**	**Z**	**X**	**Y**	**Z**
1	450	200	100	460	186	104	450	200	99,9999	7	13	0	0	0	0,0001
2	450	200	100	458	188	103	450	200	99,9999	7	14	0	0	0	0,0001
3	450	200	100	462	184	105	450	200	99,9999	6	14	0	0	0	0,0001
4	450	200	100	461	185	104	450	200	99,9999	7	14	0	0	0	0,0001
5	450	200	100	459	188	105	450	200	99,9999	7	14	0	0	0	0,0001
6	450	200	100	462	184	106	450	200	99,9999	7	13	0	0	0	0,0001
7	450	200	100	461	186	104	450	200	99,9999	6	13	0	0	0	0,0001
8	450	200	100	459	188	105	450	200	99,9999	6	14	0	0	0	0,0001
9	450	200	100	458	189	103	450	200	99,9999	6	14	0	0	0	0,0001
10	450	200	100	457	191	103	450	200	99,9999	6	14	0	0	0	0,0001
11	450	200	100	455	192	105	450	200	99,9999	7	13	0	0	0	0,0001
12	450	200	100	456	190	104	450	200	99,9999	7	14	0	0	0	0,0001
13	450	200	100	459	187	105	450	200	99,9999	6	14	0	0	0	0,0001
14	450	200	100	461	186	103	450	200	99,9999	7	13	0	0	0	0,0001
15	450	200	100	460	186	104	450	200	99,9999	7	13	0	0	0	0,0001
16	450	200	100	462	183	105	450	200	99,9999	7	14	0	0	0	0,0001
17	450	200	100	462	184	106	450	200	99,9999	6	14	0	0	0	0,0001
18	450	200	100	459	188	104	450	200	99,9999	6	14	0	0	0	0,0001
19	450	200	100	460	188	105	450	200	99,9999	7	14	0	0	0	0,0001
20	450	200	100	458	189	105	450	200	99,9999	6	13	0	0	0	0,0001

CONTINÚA ⇨

21	450	200	100	461	188	103	450	200	99,9999	7	13	0	0	0	0,0001
22	450	200	100	459	187	106	450	200	99,9999	7	14	0	0	0	0,0001
23	450	200	100	457	190	104	450	200	99,9999	6	13	0	0	0	0,0001
24	450	200	100	455	192	105	450	200	99,9999	7	13	0	0	0	0,0001
25	450	200	100	456	191	105	450	200	99,9999	7	13	0	0	0	0,0001
26	450	200	100	460	189	104	450	200	99,9999	6	14	0	0	0	0,0001
27	450	200	100	461	188	106	450	200	99,9999	7	14	0	0	0	0,0001
28	450	200	100	462	184	105	450	200	99,9999	6	14	0	0	0	0,0001
29	450	200	100	461	185	104	450	200	99,9999	7	14	0	0	0	0,0001
30	450	200	100	458	189	103	450	200	99,9999	6	14	0	0	0	0,0001

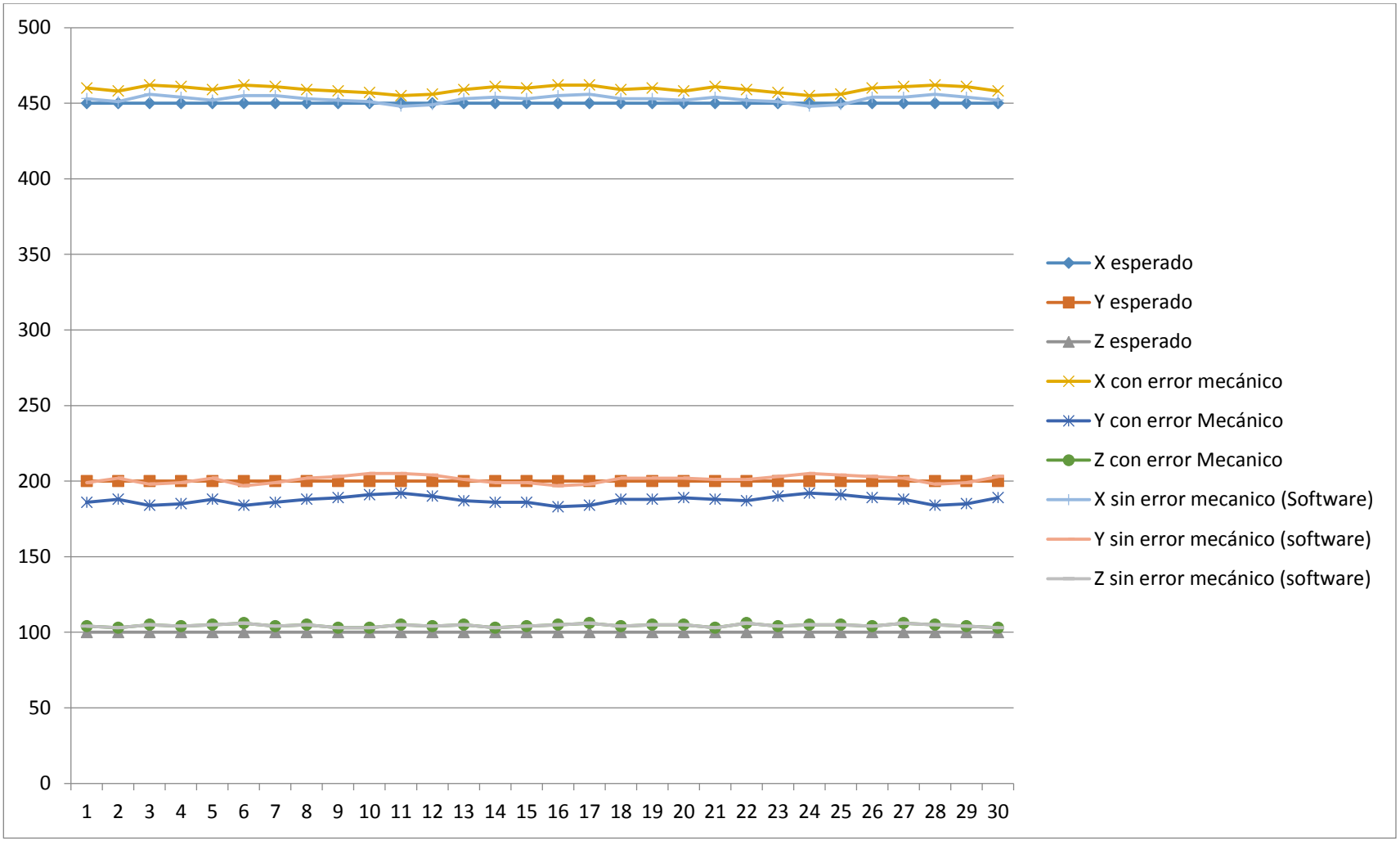


Figura 39. Mediciones de la posición alcanzada por el robot con cinemática inversa.

De la información presentada en la Tabla 11 se obtuvo que el Error Mecánico máximo cometido por el robot en el eje Y es de 14 mm, que representa el 7 % en cuanto a la distancia estudiada, en el eje X es de 7 mm que representa el 1,56 % en tanto que en Z no existen y en cuanto al error generado por el software del nuevo controlador es de 0 mm en X, en Y 0 mm y en Z hay una variación de 0,0001 mm como máximo. Esto se puede verificar gráficamente en la Figura 39. Dichos errores son generados por el estado mecánico que presenta el robot, ya que existen elementos que han sufrido desgaste a lo largo de su existencia.

4.3 Posición HOME

Para alcanzar la posición HOME del robot intervienen principalmente los interruptores de las articulaciones. Excepto para el caso del brazo, todas las articulaciones requieren moverse para alcanzar su posición final después de haber activado su respectivo interruptor.

4.3.1 Datos recolectados

En la posición HOME el extremo del robot alcanza el punto $P = (160,512,0)$. La Tabla 12 muestra los resultados de la prueba de posición HOME.

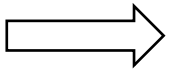
4.3.2 Análisis de resultados

Se realizaron 30 pruebas para analizar la posición HOME, para lo cual se tuvo en cuenta el valor medido con y sin errores mecánicos, los cuales consisten en recolectar datos sin ayudar al manipulador y ayudando en la sujeción de elementos mecánicos que tengan desgaste, respectivamente; y así obtener el error absoluto generado tanto por el software como por la parte mecánica.

Tabla 12

Valores obtenidos en la prueba de exactitud de la posición HOME

Prueba	Valor esperado (mm)			Valor con Error Mecánico (mm)			Valor sin Error Mecánico (mm)			Error Mecánico Absoluto (mm)			Error por software (mm)		
	X esperada	**Y esperada**	**Z esperada**	**X medida**	**Y medida**	**Z medida**	**X medida**	**Y medida**	**Z medida**	**X**	**Y**	**Z**	**X**	**Y**	**Z**
1	160	512	0	166	498	2	160	512	0	6	14	2	0	0	0,0001
2	160	512	0	167	499	1	160	512	0	7	13	1	0	0	0,0001
3	160	512	0	166	498	2	160	512	0	6	14	2	0	0	0,0001
4	160	512	0	167	498	2	160	512	0	7	14	2	0	0	0,0001
5	160	512	0	166	498	1	160	512	0	6	14	1	0	0	0,0001
6	160	512	0	167	499	2	160	512	0	7	13	2	0	0	0,0001
7	160	512	0	166	499	1	160	512	0	6	13	1	0	0	0,0001
8	160	512	0	166	498	2	160	512	0	6	14	2	0	0	0,0001
9	160	512	0	167	499	2	160	512	0	7	13	2	0	0	0,0001
10	160	512	0	166	498	2	160	512	0	6	14	2	0	0	0,0001
11	160	512	0	167	499	1	160	512	0	7	13	1	0	0	0,0001
12	160	512	0	167	498	1	160	512	0	7	14	1	0	0	0,0001
13	160	512	0	166	498	1	160	512	0	6	14	1	0	0	0,0001
14	160	512	0	167	499	2	160	512	0	7	13	2	0	0	0,0001
15	160	512	0	167	499	1	160	512	0	7	13	1	0	0	0,0001
16	160	512	0	167	498	2	160	512	0	7	14	2	0	0	0,0001
17	160	512	0	166	498	1	160	512	0	6	14	1	0	0	0,0001
18	160	512	0	166	498	2	160	512	0	6	14	2	0	0	0,0001
19	160	512	0	167	499	1	160	512	0	7	13	1	0	0	0,0001
20	160	512	0	166	498	1	160	512	0	6	14	1	0	0	0,0001

CONTINÚA 

21	160	512	0	167	499	2	160	512	0	7	13	2	0	0	0,0001
22	160	512	0	166	498	2	160	512	0	6	14	2	0	0	0,0001
23	160	512	0	167	499	2	160	512	0	7	13	2	0	0	0,0001
24	160	512	0	166	498	1	160	512	0	6	14	1	0	0	0,0001
25	160	512	0	167	499	1	160	512	0	7	13	1	0	0	0,0001
26	160	512	0	166	498	2	160	512	0	6	14	2	0	0	0,0001
27	160	512	0	167	499	1	160	512	0	7	13	1	0	0	0,0001
28	160	512	0	166	498	1	160	512	0	6	14	1	0	0	0,0001
29	160	512	0	167	499	2	160	512	0	7	13	2	0	0	0,0001
30	160	512	0	166	498	2	160	512	0	6	14	2	0	0	0,0001

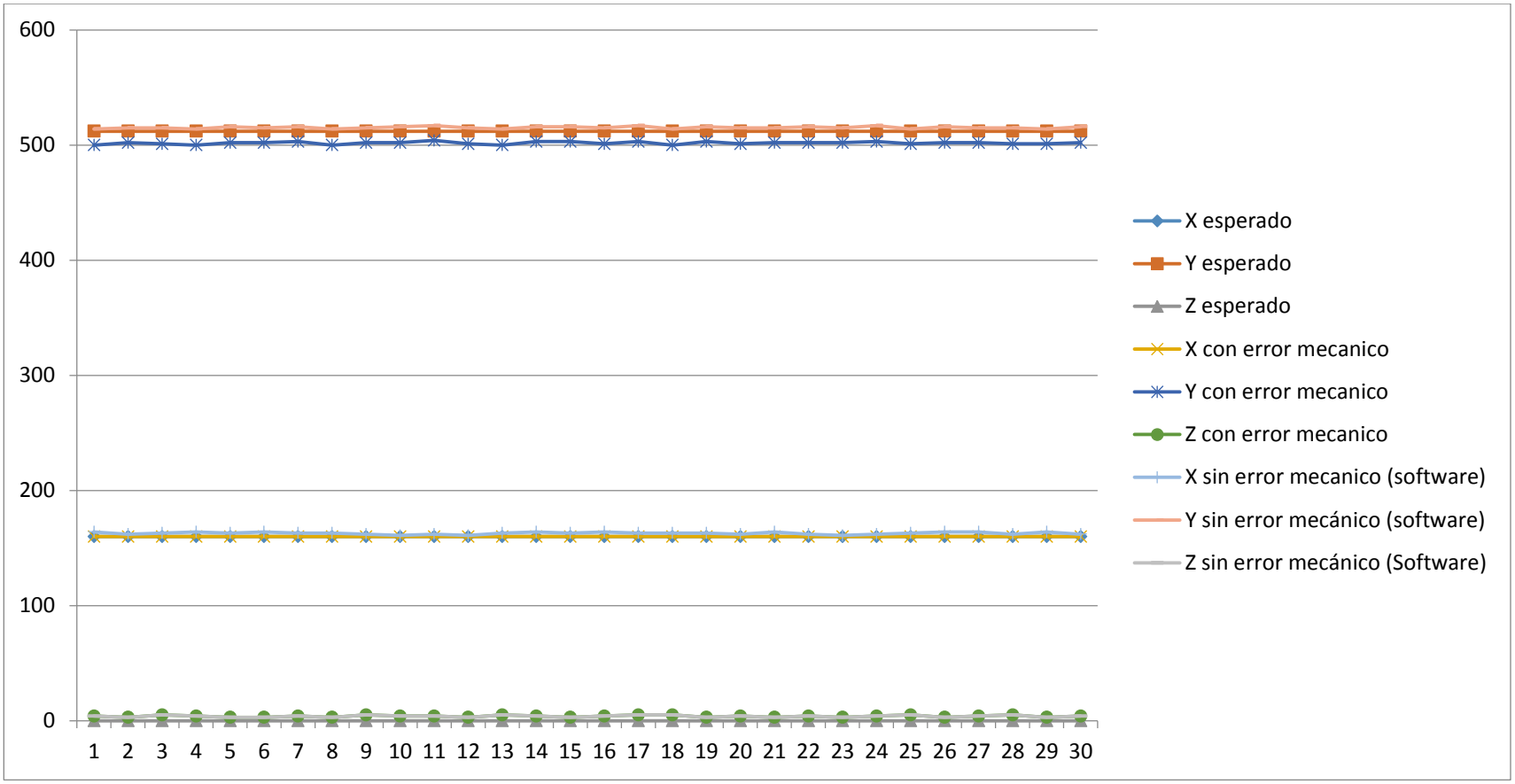


Figura 40. Mediciones de la posición HOME alcanzada por el robot.

De las pruebas realizadas y detalladas en la Tabla 12 se obtuvo un Error Mecánico máximo en el eje Y de 14 mm, que representa el 2,73 % en cuanto a la distancia estudiada, en el eje X es de 7 mm que representa el 4,38 % en tanto que en Z fue 2 mm y en cuanto al error generado por el software del nuevo controlador es de 0 mm en X, en Y 0 mm y en Z hay una variación de 0,0001 mm como máximo. La Figura 40 muestra gráficamente las curvas de cada eje, ratificando los resultados obtenidos en la Tabla 12.

Los errores presentes son generados por el desgaste de los elementos mecánicos, además otra fuente de error en el robot cuando alcanza la posición HOME se debe a la apreciación del instrumento con el que fueron medidos los ángulos iniciales. La medición de estos ángulos fue realizada con un acelerómetro estándar montado en un teléfono inteligente. Durante la medición, se debía adaptar la posición del teléfono a la geometría de los eslabones que no son completamente homogéneos.

4.4 Movimiento secuencial

El movimiento secuencial utiliza rutinas de cinemática inversa para su operación. Los datos utilizados por el programa de movimiento secuencial se apilan en forma de matriz en la que necesariamente el primer conjunto de datos debe ser los de la posición HOME.

4.4.1 Datos recolectados

Para la prueba se realizó movimientos secuenciales entre la posición HOME y la posición $P = (350,350,150)$.

4.4.2 Análisis de resultados

Se realizaron 30 pruebas para analizar el movimiento secuencial. La Tabla 13 y la Figura 41 muestran los resultados de la prueba de la función secuencial, con lo cual se conoció los valores máximos de error cometidos por cada eje ya sea por la parte mecánica o la del software.

Tabla 13

Valores obtenidos en la prueba de exactitud de la función secuencial

Prueba	Valor esperado (mm)			Valor medido (mm)			Error absoluto (mm)		
	x	**y**	**z**	**x**	**y**	**z**	**x**	**Y**	**z**
1	350	350	150	355	340	158	5	10	8
2	350	350	150	358	344	157	8	6	7
3	350	350	150	359	341	161	9	9	11
4	350	350	150	357	339	161	7	11	11
5	350	350	150	360	342	160	10	8	10
6	350	350	150	357	344	157	7	6	7
7	350	350	150	360	340	162	10	10	12
8	350	350	150	359	336	162	9	10	12
9	350	350	150	358	341	161	8	9	11
10	350	350	150	362	339	157	12	11	7
11	350	350	150	358	338	158	8	12	8
12	350	350	150	361	340	160	11	10	10
13	350	350	150	356	339	158	6	11	8
14	350	350	150	361	343	161	11	7	11
15	350	350	150	362	338	160	12	12	10
16	350	350	150	359	339	159	9	11	9
17	350	350	150	361	338	160	11	12	10
18	350	350	150	356	340	161	6	10	11
19	350	350	150	359	340	160	9	10	10
20	350	350	150	361	338	159	11	12	9
21	350	350	150	358	340	160	8	10	10
22	350	350	150	356	338	161	6	12	11
23	350	350	150	361	339	161	11	11	11
24	350	350	150	362	339	162	12	11	12
25	350	350	150	359	340	161	9	10	11
26	350	350	150	356	339	162	6	11	12
27	350	350	150	360	338	158	10	12	8
28	350	350	150	361	344	157	11	6	7
29	350	350	150	358	345	162	8	5	12
30	350	350	150	359	343	158	9	7	8

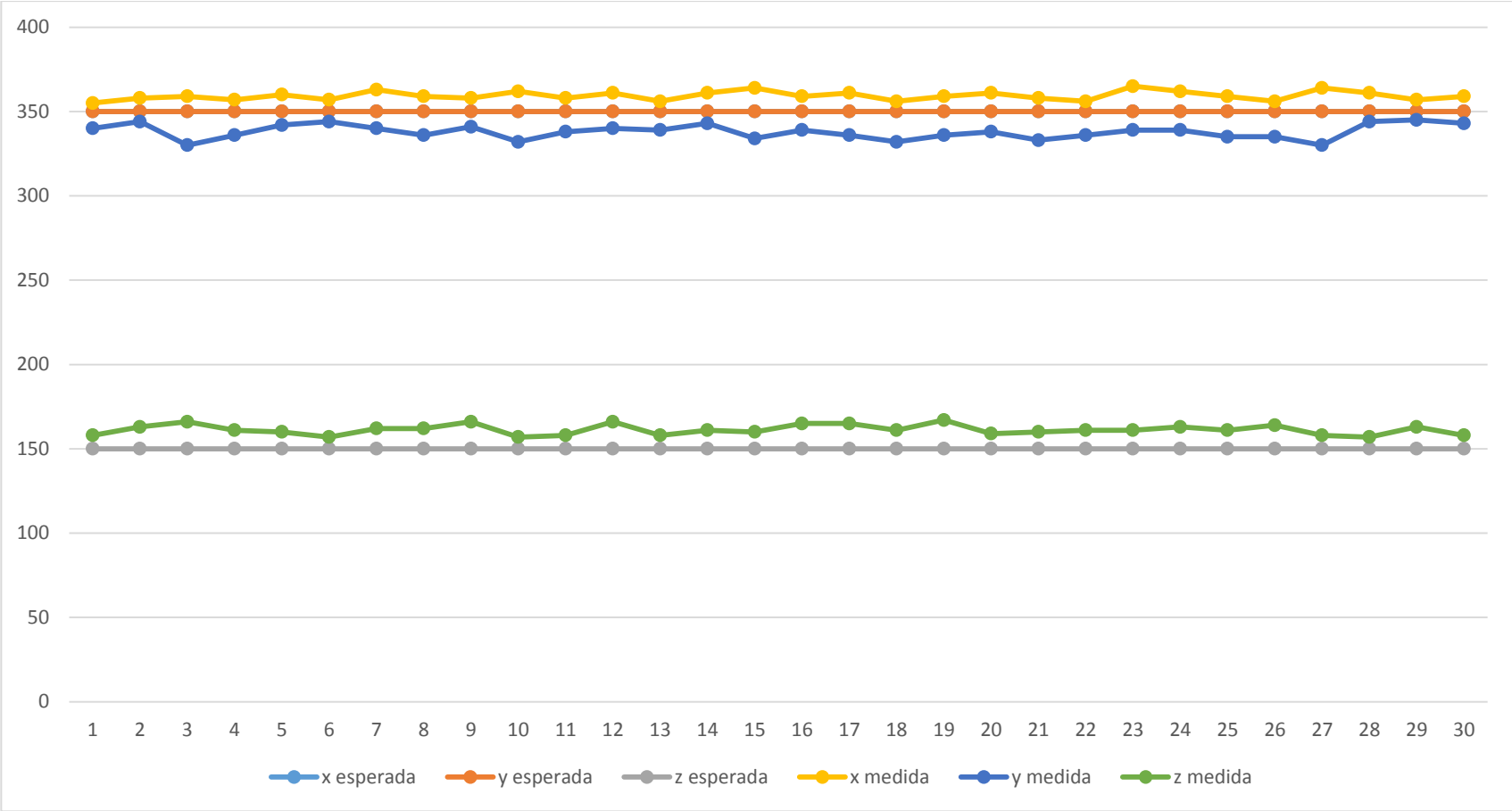


Figura 41. Mediciones de la posición alcanzada por el robot con el programa secuencial.

Los resultados de las 30 pruebas realizadas en la función secuencial, se muestran en la Tabla 13 y la Figura 41, donde el error máximo cometido por el robot utilizando dicha función fue de 12 mm en cada eje. Es necesario mencionar que aquí se combinan tanto la fuente de error del posicionamiento HOME y de la parte Mecánica. Esto quiere decir que existen errores de apreciación en el encoder del brazo, en los ángulos iniciales del robot y en los elementos que poseen desgaste en la parte mecánica, por lo tanto haciendo referencia a la Tabla 12 donde los valores máximos por el juego existente en la estructura del manipulador son de 7 mm en X, de 14 mm en Y, de 2 mm en Z, por lo cual determinamos que el error generado por el software en el eje X es 0 mm, en Y es 0 mm y en Z es 0,0001 mm.

4.5 Interpolación lineal

La interpolación lineal utiliza rutinas de cinemática inversa para su operación. La utilización de este programa exige la selección de dos puntos en el plano x-z y una altura y.

4.5.1 Datos recolectados

Para la prueba se escogieron los siguientes valores:

$P_1 = (450,0)$ mm

$P_2 = (400,0)$ mm

$y = 200$ mm

4.5.2 Análisis de resultados

Para el análisis de la interpolación lineal se hicieron 30 pruebas, con las cuales se conoció la precisión del manipulador Scorbot en cuanto a esta función. En la cual se realizaron líneas de 50 mm a una altura determinada. Los resultados de la prueba se muestran en la Tabla 14 y la Figura 42.

Tabla 14

Valores obtenidos en la prueba de exactitud de la interpolación lineal

Prueba	Valor esperado (mm)			Valor medido (mm)			Error absoluto (mm)			Error porcentual (%)		
	x inicial	**x final**	**altura y**	**x inicial**	**x final**	**altura y**	**x inicial**	**x final**	**altura y**	**x inicial**	**x final**	**altura y**
1	450	400	200	458	420	211	8	20	11	1,8	5	5,5
2	450	400	200	465	418	211	15	18	11	3,3	4,5	5,5
3	450	400	200	456	416	212	6	16	12	1,3	4	6
4	450	400	200	455	417	215	5	17	15	1,1	4,3	7,5
5	450	400	200	458	417	215	8	17	15	1,8	4,3	7,5
6	450	400	200	458	420	212	8	20	12	1,8	5	6
7	450	400	200	455	418	211	5	18	11	1,1	4,5	5,5
8	450	400	200	456	419	215	6	19	15	1,3	4,8	7,5
9	450	400	200	464	414	213	14	14	13	3,1	3,5	6,5
10	450	400	200	460	419	213	10	19	13	2,2	4,8	6,5
11	450	400	200	468	413	212	18	13	12	4	3,3	6
12	450	400	200	461	419	213	11	19	13	2,4	4,8	6,5
13	450	400	200	459	411	212	9	11	12	2	2,8	6
14	450	400	200	462	411	211	12	11	11	2,7	2,8	5,5
15	450	400	200	463	413	210	13	13	10	2,9	3,3	5
16	450	400	200	463	416	211	13	16	11	2,9	4	5,5
17	450	400	200	465	413	215	15	13	15	3,3	3,3	7,5
18	450	400	200	468	416	213	18	16	13	4	4	6,5
19	450	400	200	463	414	214	13	14	14	2,9	3,5	7
20	450	400	200	460	414	211	10	14	11	2,2	3,5	5,5
21	450	400	200	468	420	210	18	20	10	4	5	5
22	450	400	200	459	412	212	9	12	12	2	3	6
23	450	400	200	464	416	211	14	16	11	3,1	4	5,5
24	450	400	200	455	418	211	5	18	11	1,1	4,5	5,5
25	450	400	200	464	417	210	14	17	10	3,1	4,3	5

CONTINÚA

26	450	400	200	459	415	212	9	15	12	2	3,8	6
27	450	400	200	467	411	211	17	11	11	3,8	2,8	5,5
28	450	400	200	465	417	212	15	17	12	3,3	4,3	6
29	450	400	200	458	412	211	8	12	11	1,8	3	5,5
30	450	400	200	470	419	211	20	19	11	4,4	4,8	5,5

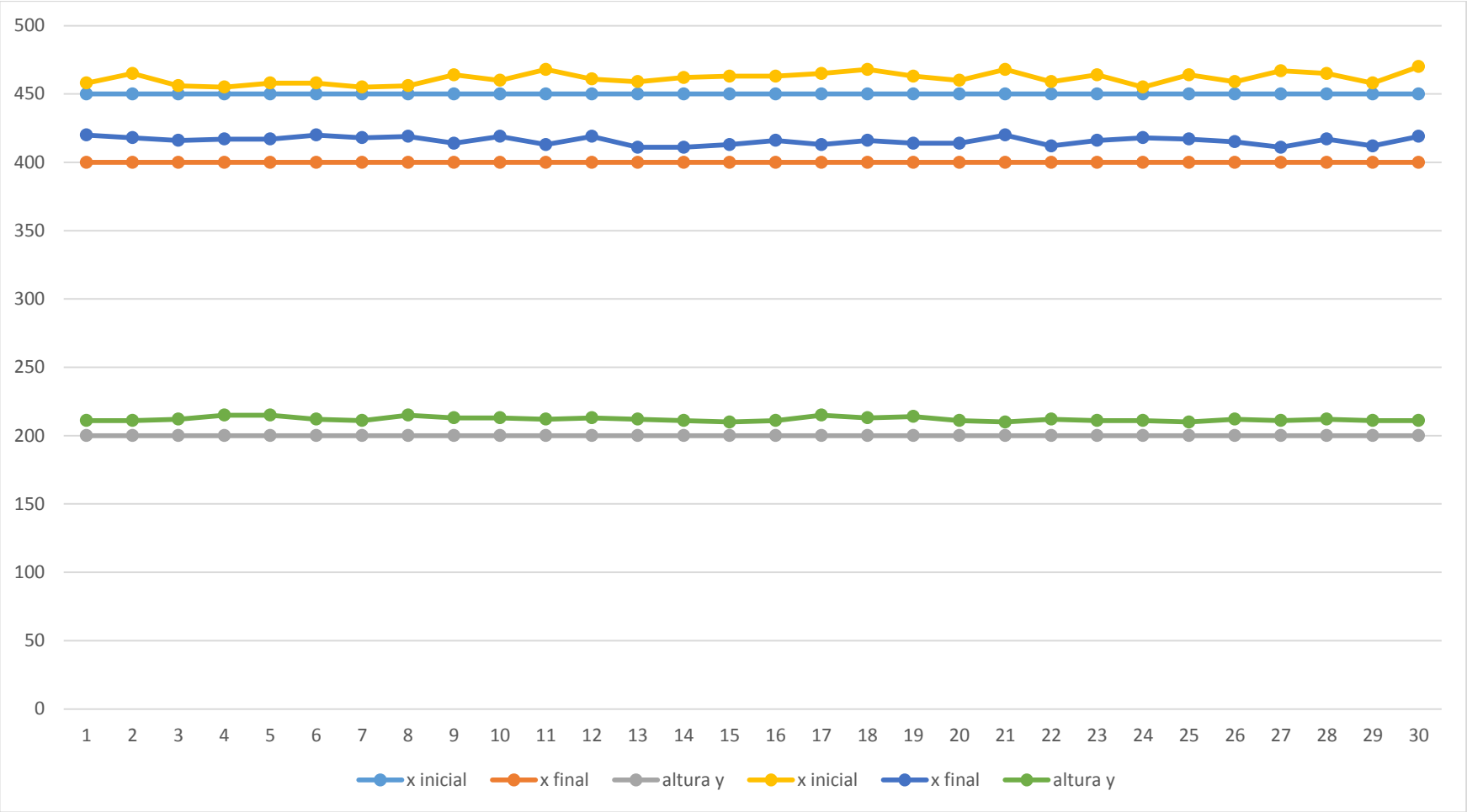


Figura 42. Mediciones de la posición alcanzada por el robot con el programa de interpolación lineal.

Los resultados de la prueba se muestran en la Tabla 14 y la Figura 42, en los cuales se aprecia que el máximo error cometido por el manipulador robótico al realizar la interpolación lineal fue 20 mm. Cabe mencionar el movimiento lineal es una sucesión de posiciones de cinemática inversa, cada una de las cuales acarrea su propio error y a esto se adhiere los errores provocados por la parte mecánica, por lo que es consecuente que este error sea superior a los anteriores, pero en tanto al error generado por el software sigue mintiéndose como máximo 0,0001 mm.

4.6 Interpolación circular

Para utilizar la interpolación circular el usuario debe ingresar los valores del centro de la circunferencia $\mathrm{C} = (\mathrm{h}, \mathrm{k})$, la altura y a la que se generará la trayectoria y el radio de la circunferencia R.

4.6.1 Datos recolectados

Las pruebas se realizaron con los siguientes valores:

$\mathrm{C} = (\mathrm{h}, \mathrm{k}) = (400{,}0)$

$r = 50$

$y = 200$

4.6.2 Análisis de resultados

Para el análisis de la interpolación circular se hicieron 30 pruebas, con lo cual se dio a conocer la precisión del manipulador Scorbot en cuanto a esta función. Para la cual se realizó la repetición de un círculo con radio de 50 mm a una altura determinada. La Tabla 15 muestra el listado con los valores obtenidos a lo largo de los ensayos realizados.

Tabla 15

Valores obtenidos en la prueba de exactitud de la interpolación circular

Prueba	Valor esperado (mm)			Valor medido (mm)			Error absoluto (mm)		
	h esperada	**D esperado**	**y esperada**	**h medida**	**D medido**	**y medida**	**h**	**D**	**altura y**
1	400	50	200	415	65	211	15	15	11
2	400	50	200	410	57	214	10	7	14
3	400	50	200	415	61	213	15	11	13
4	400	50	200	415	58	214	15	8	14
5	400	50	200	413	59	212	13	9	12
6	400	50	200	410	62	210	10	12	10
7	400	50	200	412	62	211	12	12	11
8	400	50	200	413	64	213	13	14	13
9	400	50	200	410	63	210	10	13	10
10	400	50	200	412	61	214	12	11	14
11	400	50	200	411	60	211	11	10	11
12	400	50	200	415	62	215	15	12	15
13	400	50	200	410	55	212	10	5	12
14	400	50	200	410	62	215	10	12	15
15	400	50	200	415	62	211	15	12	11
16	400	50	200	413	63	215	13	13	15
17	400	50	200	410	59	215	10	9	15
18	400	50	200	412	55	215	12	5	15
19	400	50	200	410	65	212	10	15	12
20	400	50	200	415	60	213	15	10	13
21	400	50	200	413	56	211	13	6	11
22	400	50	200	412	53	215	12	3	15
23	400	50	200	410	54	214	10	4	14
24	400	50	200	414	59	214	14	9	14
25	400	50	200	415	53	214	15	3	14
26	400	50	200	413	64	212	13	14	12
27	400	50	200	413	62	211	13	12	11
28	400	50	200	414	65	212	14	15	12
29	400	50	200	415	65	211	15	15	11
30	400	50	200	413	54	210	13	4	10

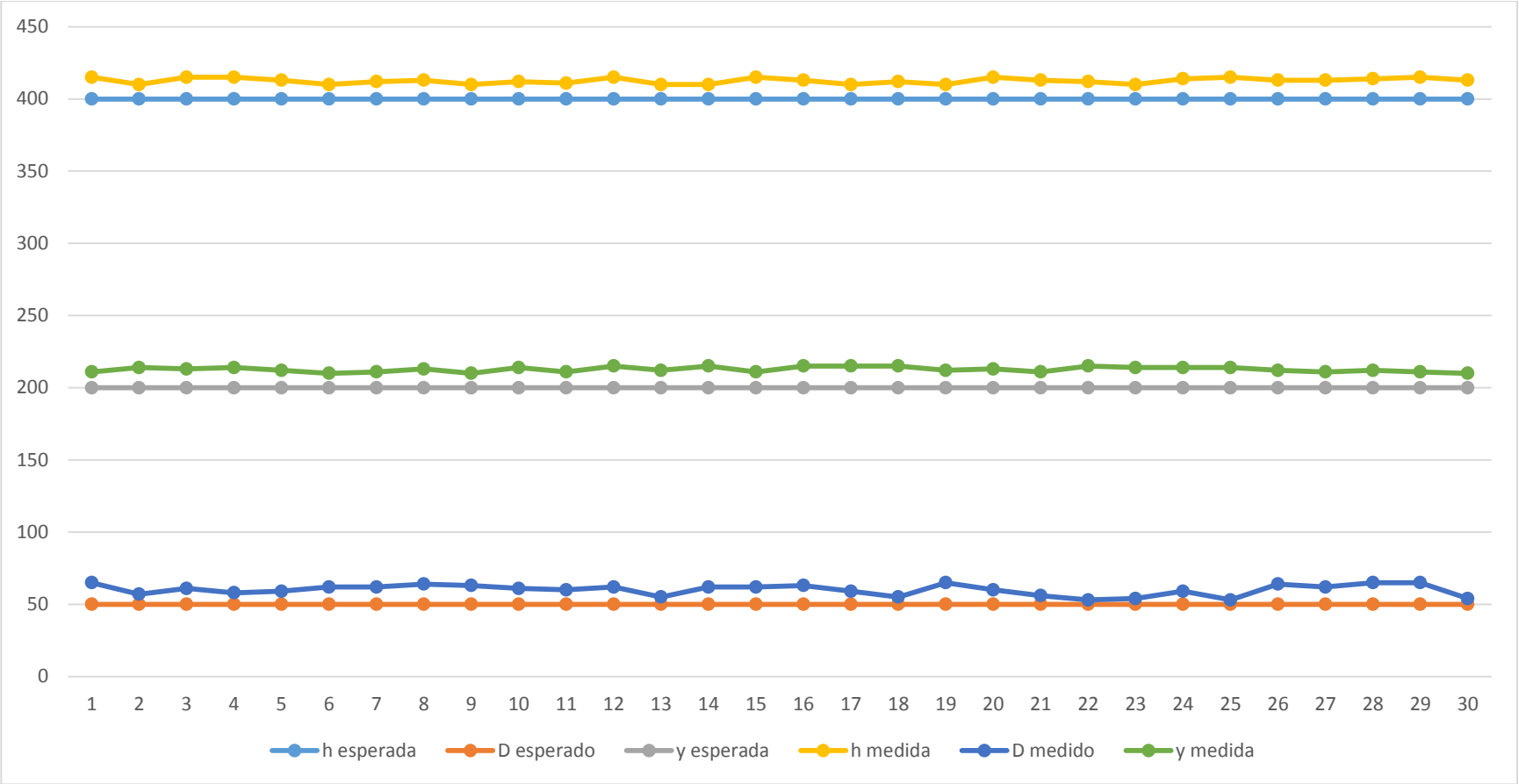


Figura 43. Mediciones de la posición alcanzada por el robot con el programa de interpolación circular.

La Tabla 15 y la Figura 43 muestran el máximo error cometido por el robot durante la interpolación circular, el mismo que fue 15 mm. La interpolación circular es una sucesión de cinemática inversa, cada una de las cuales acarrea el error de conteo de pulsos del brazo, el error en la posición HOME y el desgaste de elementos mecánicos propios de la estructura del Scorbot, provocando que el manipulador cabecee y no mantenga una altura uniforme durante la interpolación. Como agravante que permite justificar el aumento en el valor del error, es necesario mencionar que en este algoritmo se introducen las funciones trigonométricas SENO y COSENO que no permiten realizar un cálculo exacto de los respectivos puntos de la circunferencia. De igual forma que en la interpolación lineal se verifico que el software genera un error mínimo de 0,0001 mm, y este valor justamente satisface proyecto.

4.7 Análisis económico del proyecto

La Tabla 16 muestra el costo de implementación de este proyecto.

Tabla 16

Valores obtenidos en la prueba de exactitud de la interpolación circular

Número	Componente	Cantidad	Costo unitario USD	Costo total USD
1	Scorbot	1	850	850
2	Tarjeta Aduino	1	33	33
3	L298N	3	20	60
4	Fuente ATX	1	25	25
6	Caja metálica	1	50	50
5	Otros componentes electrónicos	-	-	50
7	Varios	-	-	50
			Total	1118

El costo de un Scorbot con el controlador original como el mostrado en la Figura 44 tiene un costo de 3300 dólares aproximadamente. El beneficio económico de la utilización de este controlador es 2182 dólares (Ebay, s.f.).

Figura 44. Scorbot con controlador original.

Fuente: (Ebay, s.f.)

4.8 Validación de la hipótesis

La hipótesis planteada para este proyecto es: ¿El diseño e implementación de un controlador genérico mediante software libre para el SCORBOT-ER le permitirá funcionar adecuadamente al manipulador? Validar la hipótesis implica comprobar que un controlador basado en una placa Arduino es capaz de gestionar la operación del brazo robótico Scorbot ER III.

La fase de pruebas incluyó el estudio del desempeño de todos los programas elaborados de los que se obtuvo un error máximo absoluto de 20 mm. Sin embargo, es necesario analizar además cómo varían los resultados en un mismo programa y entre programas para demostrar que todos siguen un mismo patrón y el error no se debe a fallos en la implementación de los algoritmos. Para ello es útil calcular la desviación máxima generada por cada programa; los resultados se muestran en la Tabla 17.

Tabla 17

Desviación estándar de los datos de desempeño del robot

Programa	Desviación estándar (mm)
Cinemática directa	1,64
Cinemática inversa	2,10
Home	1,59
Movimiento secuencial	2,18
Interpolación lineal	2,16
Interpolación circular	2,14

En la Tabla 17 se verifica que la desviación estándar es baja en todos los programas evaluados; además, los valores de distintos programas están próximos por cuanto el robot muestra comportamientos similares en todos los programas. Así puede concluirse que el robot, a pesar de presentar errores en la parte mecánica en la exactitud, tiene una precisión destacable.

Un trabajo similar a este fue el desarrollado por Holguin (2011) en el que se utilizó una placa Arduino UNO y un intérprete de C++ para controlar un Scorbot ER V. Los resultados alcanzados en dicho proyecto revelan errores de posicionamiento de hasta 50 mm recomendándose no utilizar el controlador para tareas de alta precisión.

Con un error mínimo de 0,0001 mm, obtenido por software y un error despreciable generado por parte de los componentes mecánicos que conforman el manipulador se puede afirmar que realizando un mantenimiento correctivo a la parte estructural mecánica es posible alcanzar una mayor precisión del sistema sin elevar significativamente el costo de implementación.

Por lo tanto se verifica que un controlador genérico para Scorbot, basado en una placa Arduino es capaz de gestionar la operación del brazo robótico con niveles aceptables de error.

4.9 Manual de usuario

El manual de usuario del controlador genérico desarrollado en este proyecto se muestra en el Anexo 2.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1 Conclusiones

- Se diseñó y construyó un controlador genérico para repotenciar el brazo robótico SCORBOT ER III que permite ejecutar las siguientes tareas: posicionamiento HOME, cinemática directa, cinemática inversa, movimiento secuencial, interpolación lineal e interpolación circular.

- La información técnica recopilada del estado del arte y la documentación del robot fueron de utilidad para conocer los requerimientos, ventajas y limitaciones del Scorbot y diseñar el controlador más adecuado para este equipo.

- Se realizó el dimensionamiento y selección del hardware que compone el controlador en base a los requerimientos de los actuadores y sensores del robot.

- Para el desarrollo de la HMI se seleccionó el software Python porque es de distribución libre, tiene alta funcionalidad y su sintaxis es fácil de aprender.

- Mediante la utilización de software libre, como VPython, es posible diseñar a medida el software de control ya que todas sus funciones y librerías están disponibles para uso público además de tener menor costo de operación ya que no se requieren licencias.

- En el posicionamiento HOME, cinemática directa, cinemática inversa y movimiento secuencial se alcanzó un error máximo permitido debido mayormente a desajustes mecánicos de la estructura del robot.

- La interpolación lineal e interpolación circular presentan errores mayores a las otras funciones debido principalmente a que en estas intervienen dos o más de las funciones primarias.

- El control del brazo está implementado sobre una tarjeta Arduino Mega que se comunica con la HMI a través del protocolo de comunicación Firmata; este protocolo es muy fácil de implementar y cuenta con amplia documentación.

- Como parte del desarrollo del controlador se diseñó y construyó una placa de distribución de voltaje que además cumple la función de transmitir las señales de los interruptores de posición HOME a la tarjeta madre.

- La tarea que requirió mayor atención fue la solución de la cinemática inversa ya que es la base para la mayoría de funciones implementadas y contiene cálculos matemáticos extensos.

- Con la implementación de este proyecto los estudiantes que tomen asignaturas relacionadas con robótica industrial tendrán la oportunidad de profundizar sus conocimientos sobre manipuladores.

- Se verificó que una fuente tipo ATX es factible de utilizar en proyectos electrónicos de baja potencia lo cual es beneficioso porque tienen excelente calidad de salida de voltaje y su costo no es elevado.

5.2 Recomendaciones

- En Proyectos posteriores se invita a revisar la parte mecánica del manipulador, para realizar un mantenimiento preventivo y correctivo de ser necesario, ya que la existencia de elementos mecánicos defectuosos producen errores en cuanto a los movimientos realizados por el robot.

- Para la implementación de proyectos similares se recomienda utilizar componentes de iguales características o superiores a la tarjeta Arduino MEGA, driver L298N y fuente de poder ATX de 750 W.

- Antes de operar el Scorbot con el controlador genérico se recomienda leer el manual de usuario del controlador a fin de evitar daños en el equipo.

- A fin de mejorar la funcionalidad del robot se recomienda el empleo de dispositivos con memoria de posición como potenciómetros de alta precisión y encoders absolutos.

- Es recomendable revisar el estado del robot y de las conexiones del controlador siempre que se vaya a iniciar una sesión de trabajo con el robot.

- En este proyecto se presentó la resolución geométrica de la cinemática inversa del robot; es recomendable incursionar en otros métodos de resolución de la cinemática inversa a fin de mejorar la experiencia del usuario cuando utilice este equipo.

- Para alcanzar un modelo matemático más cercano a la realidad, se recomienda para trabajos futuros resolver el modelo cinético del Scorbot.

BIBLIOGRAFÍA

- Barrientos, A. (2007). *Fundamentos de robótica.* Madrid: McGraw-Hill.
- Yague, J. (2013). *Control de un brazo robótico de cinco grados de libertad con Arduino* . Valladolid: Universidad de Valladolid.
- ESHED ROBOTEC. (1998). *Adjustin the home position of mechanical arm ER III.* ESHED ROBOTEC.
- Ferrolho, A. (2005). *Software development to control the Scorbot ER VII robot with a PC.* Viseu, PT: Superior School of Technology of the Polytechnic Institute of Viseu.
- Pattnaik, A. (2010). *Robotic arm control through human movement using accelerometers* . Rourkela: National Institute Of Technology, Rourkela.
- Rasouliha, M. (2004). *Computer controlled robot Arm* . Victoria, CA: University of Victoria.
- Rea, S. (2012). *Control de movimiento del manipulador de cinxo grados de libertad CRS-A255 por esqueletización.* Sangolquí: ESPE.
- Neto, P. (2010). *Accelerometer-based control of an industrial robotic arm.* Coimbra: IEEE.
- Tuntivivat, S. (2011). *Design and Construction and Motion Control of 6-Axis Robot Manipulator for Industrial Applications* . Pathumthani: Rajamangala University of Technology Thanyaburi.

LINKOGRAFÍA

- ADAFRUIT. (s.f.). *Raspberry Pi*. Recuperado el 12 de Septiembre de 2015, de ADAFRUIT: http://www.adafruit.com/pdfs/raspberrypi2modelb.pdf
- Albu, M. (s.f.). *Four quadrant full bridge choppers*. Recuperado el 10 de Enero de 2015, de Technical University of Iasi: http://www.euedia.tuiasi.ro/lab_ep/ep_files/Lab_no_20_c1.pdf
- ARDUINO. (s.f.). *Arduino Mega 2560*. Recuperado el 12 de Septiembre de 2015, de ARDUINO: https://www.arduino.cc/en/Main/ArduinoBoardMega2560
- ARDUINO. (s.f.). *Arduino Motor Shield*. Recuperado el 12 de Septiembre de 2015, de ARDUINO: https://www.arduino.cc/en/Main/ArduinoMotorShieldR3
- Automated motion systems. (s.f.). *Using shafts encoders in motion control*. Recuperado el 20 de Marzo de 2015, de Automated motion systems: http://www.automotsys.com.au/encodersmc.html
- Bar, F. (s.f.). *Building an Arduino Bluetooth Robot Car – Part 2: Controlling DC Motors*. Recuperado el 20 de Febrero de 2015, de Wordpress: https://filderbaer.wordpress.com/2014/10/15/building-an-arduino-bluetooth-robot-car-part-2-controlling-dc-motors/
- DIDEL. (s.f.). *Optical encoder*. Recuperado el 10 de Febrero de 2015, de DIDEL: http://www.didel.com/microkit/encoder/Encoder.html
- Dimension Engineering. (s.f.). *A beginner's guide to accelerometers*. Recuperado el 12 de Enero de 2015, de Dimension Engineering: http://www.dimensionengineering.com/info/accelerometers
- Dimension Engineering. (s.f.). *Sabertooth dual 5A motor driver*. Recuperado el 12 de Septiembre de 2015, de Dimension Engineering: http://www.dimensionengineering.com/products/sabertooth2x5
- Ebay. (s.f.). *ESHED ROBOTEC SCORBOT ER 4p Robot with Controller and Tech*. Recuperado el 10 de Febrero de 2016, de Ebay: http://www.ebay.com/itm/ESHED-ROBOTEC-SCORBOT-ER-4pc-Robot-with-controller-Tech-/321799207827?hash=item4aecba3b93:g:LO4AAOSw9N1Vl5gu
- ELECTRONILAB. (s.f.). *Driver dual para motores L298N*. Recuperado el 12 de Septiembre de 2015, de ELECTRONILAB: http://electronilab.co/tienda/driver-dual-para-motores-full-bridge-l298n/
- ELECTRONILAB. (s.f.). *Tutorial: Uso de driver L298N para motores DC y paso a paso con Arduino*. Recuperado el 8 de Mayo de 2015, de ELECTRONILAB: http://electronilab.co/tutoriales/tutorial-de-uso-driver-dual-l298n-para-motores-dc-y-paso-a-paso-con-arduino/

- FRITZING. (s.f.). *Parts*. Recuperado el 8 de Febrero de 2015, de Fritzing: http://fritzing.org/parts/
- Holguín, H. (2011). *Control de un brazo robótico teleoperado mediante acelerómetro.* México: Universidad Nacional Autónoma de México.
- Intelitek. (1995). *SCORBOT-ER III, User's manual*. Manchester, NH: Intelitek.
- Magnusglad. (s.f.). *TIP120 to control a DC motor using arduino*. Recuperado el 12 de Febrero de 2015, de Wordpress: https://magnusglad.files.wordpress.com/2013/03/tip120_dc.png
- Maldonado, D. (2013). *Estudio, diseño e implementación de un controlador (hardware y software) para tres grados de libertad del manipulador robótico CRS-A255.* Sangolquí: ESPE.
- Martínez, M. (2008). *Control de un brazo robótico con un sistema operativo en tiempo real.* Cantabria: Universidad de cantabria.
- MICROCHIP. (s.f.). *PIC18F2455/2550/4455/4550 datasheet*. Recuperado el 8 de Septiembre de 2015, de MICROCHIP: http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf
- National Instruments. (s.f.). *Pulse width modulation*. Recuperado el 10 de Marzo de 2015, de National Instruments: http://www.ni.com/tutorial/2991/en/
- OBrien, G. (s.f.). *The Scorbot ER-III Limit Switch & Encoder - Explained*. Recuperado el 24 de Noviembre de 2015, de Digital Circuitry: http://www.digital-circuitry.com/index.html
- Parallax. (s.f.). *Memsic 2125 Dual-axis Accelerometer*. Recuperado el 15 de Febrero de 2015, de Parallax: http://learn.parallax.com/KickStart/28017
- SERVOCITY. (s.f.). *Roboclaw 2x5A Motor Controller*. Recuperado el 12 de Septiembre de 2015, de SERVOCITY: https://www.servocity.com/html/roboclaw_2x5a_motor_controller.html#.VfQ1eRF_Oko
- Sparkfun. (s.f.). *Pulse width modulation*. Recuperado el 24 de Noviembre de 2015, de Learn Sparkfun: https://learn.sparkfun.com/tutorials/pulse-width-modulation
- Vex robotics. (s.f.). *Potentiometer*. Recuperado el 30 de Enero de 2015, de Vex robotics: http://www.vexrobotics.com/wiki/images/2/29/Potentiometer_Figure_1a.jpg

ANEXOS

DEPARTAMENTO DE ENERGÍA Y MECÁNICA

CARRERA DE INGENIERÍA EN MECATRÓNICA

CERTIFICACIÓN

Se certifica que el presente trabajo fue desarrollado por el señor: **Nelson Hernán Aldaz Saca**.

En la ciudad de Latacunga a los 07 días de marzo del 2016.

Ing. Fausto Acuña

DIRECTOR DEL PROYECTO

Aprobado por:

Ing. Vicente Hallo

DIRECTOR DE CARRERA

Dr. Juan Carlos Díaz

SECRETARIO ACADÉMICO