



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**USO DE UN CONTROLADOR CON
TECNOLOGÍA FPGA Y TIEMPO
REAL PARA EL SCORBOT-ER 4U**

TESIS

Que para obtener el título de
INGENIERO MECATRÓNICO

P R E S E N T A

EMMANUEL JUAN CIPRES TORRES

DIRECTOR DE TESIS

M.I. SERAFIN CASTAÑEDA CEDEÑO



Ciudad Universitaria, Ciudad de México, 2016



Índice

1	Introducción	1
1.1	Antecedentes	1
1.1.1	Robótica	1
1.1.2	Robot SCORBOT-ER 4u	5
1.1.3	Controller-USB	7
1.1.4	Trabajos anteriores	8
1.1.5	CompacRIO®	11
1.2	Objetivos	15
1.3	Alcances	15
2	Cinemática del Robot	16
2.1	Cinemática directa	16
2.1.1	Algoritmo de D-H aplicado al robot SCORBOT-ER 4u	17
2.2	Cinemática inversa	21
2.2.1	Cinemática de posición inversa	21
2.2.2	Cinemática de orientación inversa	24
3	Implementación	26
3.1	Programación en el chasis FPGA	28
3.1.1	Posicionamiento de articulaciones	28
3.1.2	Control PID	30
3.1.3	Generación de PWM	32
3.1.4	Lectura de interruptores	33
3.2	Programación del controlador <i>Real Time</i>	33
3.2.1	Posicionamiento de <i>HOME</i>	34
3.2.2	Cinemática inversa	35
3.2.2.1	Matrices homogéneas	35
3.2.2.2	Módulo de LabVIEW robotics	35
3.2.3	Perfil de trayectoria	37
3.2.4	Manipulación de variables del control PID	38
3.3	Programación en la PC	38



3.3.1	Interfaz principal	39
3.3.2	Interfaz simulación 3D	43
3.3.3	Interfaz monitores de variables	44
4	Pruebas y Resultados	46
4.1	Cinemática inversa	46
4.2	Simulación 3D	47
4.3	Funcionamiento del SCORBOT-ER 4u.	49
5	Conclusiones	56
6	Referencias	59
7	Anexos	62
	Anexo A. Conector del robot	
	Anexo B. Diagrama de conexiones	
	Anexo C. Cotizaciones de precios entre controladores	

1 Introducción

1.1 Antecedentes

Durante los últimos años, hablar de robótica es abordar un tema muy amplio gracias a las diversas aplicaciones que conlleva, debido a los grandes beneficios que ha brindado para el desarrollo de nuevas tecnologías tanto para la industria, universidades y actualmente en el hogar realizando tareas domésticas. Hoy en día, existen fábricas industriales completamente automatizadas gracias a los robots manipuladores. La robótica aun así, es un campo nuevo de la tecnología.

1.1.1 Robótica

La Robótica es “una disciplina científica que aborda la investigación y desarrollo de una clase particular de sistemas mecánicos, denominados robots manipuladores, diseñados para realizar una amplia variedad de aplicaciones industriales, científicas, domésticas y comerciales.” [1] [Barrientos]

El vocablo robot proviene de la palabra checa “robotas” que significa trabajo, y ha sido adoptado a varias lenguas incluido el español.

Existen una gran variedad de definiciones para el termino robot dependiendo del tipo de literatura, para este trabajo y su aplicación, se optó por la definición: “Un robot manipulador industrial es una máquina manipuladora con varios grados de libertad controlada automáticamente, reprogramable y de múltiple usos, pudiendo estar en un lugar fijo o móvil para su empleo en aplicaciones industriales” de la Federación Internacional de Robótica bajo la norma ISO/TR 8373.



Figura 1.1 Línea de ensamble de Volkswagen en Baunatal, Alemania. [E1]

De la misma forma que existe cierta controversia para definir lo que es un robot, también es el mismo problema al momento de clasificar los robots, una clasificación más formal es la que da la Asociación Francesa de Robótica Industrial (AFRI).

Tipo	Definición
Tipo A	Manipulador con control manual o telemando.
Tipo B	Manipulador automático con ciclos preajustados; regulación mediante fines de carrara o topes por PLC; accionamiento neumático, eléctrico o hidráulico.
Tipo C	Robot programable con trayectoria continua o punto a punto. Carece de conocimientos sobre su entorno.
Tipo D	Robot capaz de adquirir datos de su entorno, readaptando su tarea en función de estos.

Tabla1.1 Clasificación de los robots según la AFRI. [1][Barrientos]

Los robots manipuladores son dispositivos que han logrado aceptación en la industrial debido a su capacidad de llevar a cabo un número variado de tareas con rapidez y precisión, además de que pueden adaptarse a diversas situaciones con sólo modificar el programa que comanda los movimientos de la máquina. Su uso ha incrementado gracias a que permite reducir costos, aumentar la producción, calidad del producto y ahorrar espacio.

La gran variedad de los robots manipuladores está relacionada con el número de ejes que poseen y con su distribución. Pero a diferencia de otras máquinas, no es intuitivo predecir el desplazamiento final del brazo a partir del movimiento de sus ejes.

En todo robot manipulador se puede distinguir cuatro componentes: mecánico, hardware, control y software, y por ello se logra que los robots manipuladores sean versátiles, inteligentes, robustos y de excelente repetitividad.

Mecánicamente los manipuladores se arman y obtienen toda su flexibilidad a partir de un principio simple el cual es conectar varias barras a través de articulaciones, que normalmente son rotacionales o traslacionales. El conectar varias barras entre sí recibe el nombre de cadena cinemática debido a esto los robots manipuladores suelen ser:

➤ *Manipuladores Seriales*

Este tipo de robot industrial se construye conectando eslabones uno tras otro (en serie) y está conformado por una cadena cinemática abierta. Los manipuladores seriales son la configuración más conocida debido a la demanda industrial y tiene dos tipos de variedades: los brazos robóticos y las mesas cartesianas.

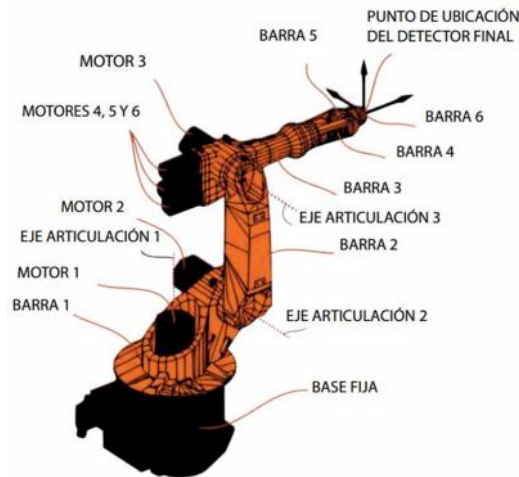


Figura 1.2 Robot Serial de seis grados de libertad KUKA KR-6 [E2]

➤ Manipuladores Paralelo

Se denomina manipulador paralelo general a aquél que posee un elemento terminal unido a un elemento base o fijo por medio de varias cadenas cinemáticas independientes. Dichos manipuladores son utilizados principalmente en aquellas aplicaciones donde sus características los hacen ideales para resolver aquellos problemas que se presentan con los robots serie.

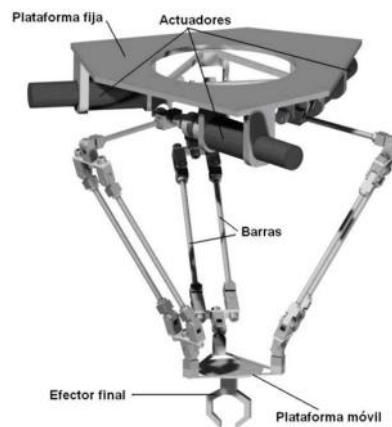


Figura 1.3 Robot Paralelo Delta UPB. [E2]

Los manipuladores paralelos son menos conocidos pero no menos útiles. El tipo más común son los robots delta, ampliamente utilizados en aplicaciones de posicionamiento rápido de objetos (*pick and place*), muy útiles en la industria alimenticia. Se caracterizan principalmente por sus altas velocidades y áreas de trabajo más reducidas que las de los robots seriales.

Teniendo en cuenta las definiciones entre manipuladores en serie y en paralelo, se muestran las ventajas y desventajas entre uno y otro en la Tabla 1.2

Robot Serie	Robot Paralelo
Estructura poco rígida	Arquitectura más rígida
Baja relación carga-peso del manipulador	Relación carga/peso muy superior
Robot más pesado	Robot ligero
Baja precisión.	Alta precisión.
Precisión del posicionamiento.	Los errores en las articulaciones afectan con el mismo orden de magnitud a la posición de la plataforma.
Los errores en las articulaciones crecen de elemento a elemento y se acumulan hasta el elemento terminal.	
Bajas velocidades y aceleraciones.	Altas velocidades y aceleraciones.
Peores características dinámicas	Mejora respuesta dinámica.
Cinemática simple.	Cinemática compleja.
Resolución analítica de las ecuaciones.	Alto tiempo de computación.
Facilidad en el lazo de control	Dificultad en el control.
Amplio espacio de trabajo	Espacio de trabajo más reducido
Técnica de calibración resuelta	Calibración compleja.
	Problema abierto
Posiciones singulares únicamente en el problema inverso	Posiciones singulares más complejas en problemas directo e inverso
Tecnología ya desarrollada	Nueva tecnología

Tabla 1.2 Características de los robot serie y paralelo. [E3]

➤ Manipuladores Híbridos

Además de los manipuladores ya mencionados con anterioridad (seriales y paralelos), existe otra rama de estudio de la robótica el cual combina estructuras seriales y paralelas para crear nuevas estructuras y es conocido como robótica híbrida.

Los robots híbridos son la combinación de dos o más estructuras robóticas que toman ventaja de las propiedades de cada estructura robótica [E4]. Al poder contar con manipuladores seriales y paralelos en una misma estructura se reducen las limitaciones de cada una de ellas.

Sukhan Lee y Sungbok Kim han propuesto una clasificación de los robots híbridos: paralelo-paralelo, serie-paralelo, paralelo-serie y serie-serie. [E4]

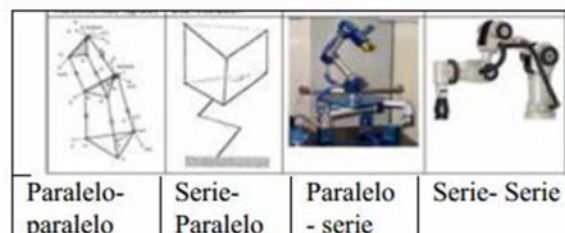


Figura 1.4 Clasificación de Robots Híbridos. [E4]

1.1.2 Robot SCORBOT-ER 4u

El robot SCORBOT-ER 4u que se encuentra en el Centro de Ingeniería Avanzada de la Facultad de Ingeniería de U.N.A.M., fue diseñado y desarrollado para emular las condiciones de un robot industrial. Tiene fines didácticos y está pensado de manera que los estudiantes sean capaces de visualizar los mecanismos de movimiento internos. [4][Intelitek]

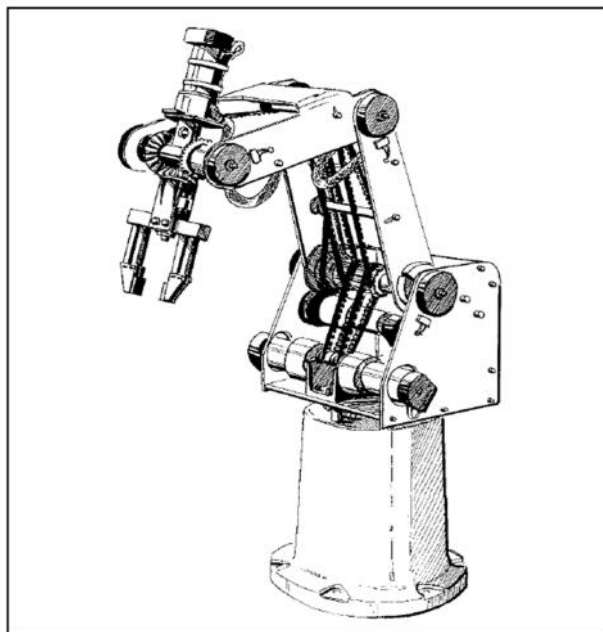


Figura 1.5 Robot SCORBOT-ER 4u. [4] [Intelitek]

El robot SCORBOT-ER 4u está compuesto por dos componentes principales: el brazo mecánico y el controlador electrónico.

- El brazo mecánico está constituido como brazo vertical articulado, de cinco grados de libertad (base, hombro, codo y dos movimientos de muñeca) y una pinza. Todas las articulaciones poseen un movimiento de articulación giratorio, a excepción de la pinza que posee un movimiento lineal o prismático de apertura o cierre.
- EL controlador es un microcomputador cuyo propósito general es realizar las tareas de control del brazo articulado. Este controlador será detallado posteriormente.

Las articulaciones mencionadas con anterioridad son accionadas por medio de motores de corriente directa fijados a la base del robot a excepción del motor de la pinza, dichos motores están acoplados de manera indirecta a las articulaciones; esto es, los motores están montados lejos de las articulaciones y el movimiento de las articulaciones se transmite por medio de bandas o engranes, y así se reduce el peso del brazo mecánico.

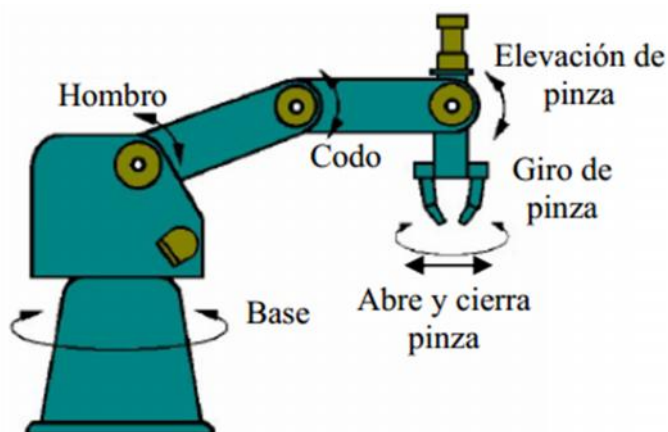


Figura 1.6 Ejes y Movimientos del SCORBOT-ER 4u. [E8]

# de Eje	Articulación	Movimiento	# de Motor	Límite
1	Cintura o Base	Rotación del cuerpo	1	$\pm 310^\circ$
2	Hombro	Sube y baja brazo	2	$+130^\circ/-35^\circ$
3	Codo	Sube y baja antebrazo	3	$\pm 130^\circ$
4	Pitch	Sube y baja pinza	4+5	$\pm 130^\circ$
5	Roll	Rota pinza	4+5	Ilimitado mecánicamente
6	Pinza	Apertura y cierre	6	75 mm sin gomas 65 mm con gomas

Tabla 1.3 Ejes, Movimientos y límites del SCORBOT-ER 4u (Cuando los motores 4 y 5 se mueven en direcciones opuestas la muñeca sube y baja. Cuando se mueven en la misma dirección la muñeca gira en sentido o en contra de las manecillas de reloj). [4] [Intelitek]

La longitud de los eslabones y los grados de rotación de las articulaciones limita el espacio de trabajo del robot.

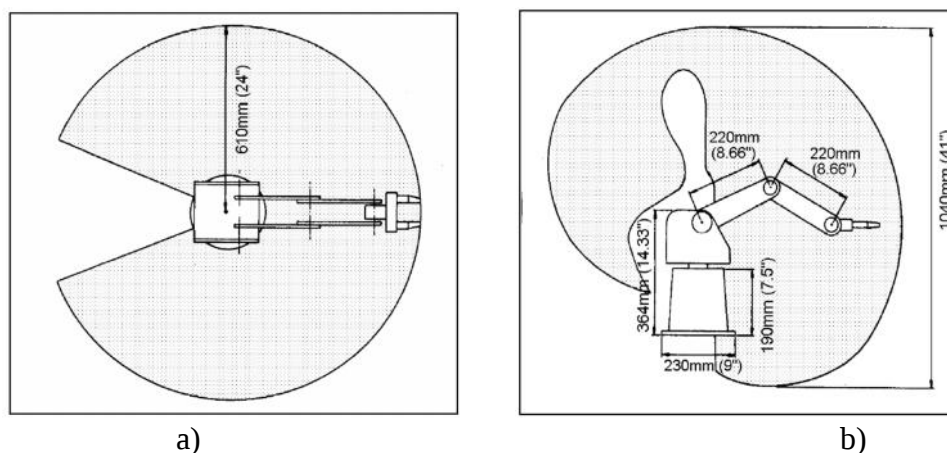


Figura 1.7 Rango de Operación: a) vista superior y b) vista lateral. [4] [Intelitek]

1.1.3 Controller-USB

El Controlador-USB es parte del sistema robótico SCORBOT, además de controlar al brazo robótico, también puede ser usado para controlar dispositivos periféricos motorizados.



Figura 1.8 Controlador-USB. [5][Intelitek]

El controlador-USB alimenta a los 6 motores del brazo robótico y además dos motores periféricos por medio de una fuente de alimentación de 24 V. También contiene circuitos electrónicos que generan una señal PWM para el control de los motores y se encarga de leer cada encoder de los motores y los finales de carrera de cada articulación.

En general, las principales características del controlador-USB son:

- Tipo de control: en Tiempo Real, Multi-Tareas, PID y PWM.
- Microcontrolador NEC V853 en la tarjeta de E/S.
- Dos canales de comunicación RS232.
- Programación opcional en *RoboCell 3D* y *Teach Pendant*.
- Fuente de alimentación para los motores de 12V.

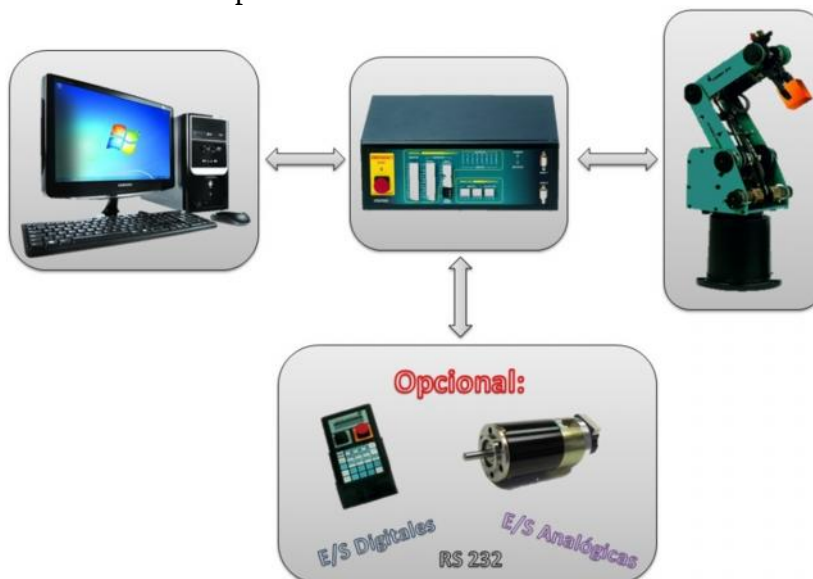


Figura 1.9 Conexiones posibles en el Controlador-USB.

1.1.4 Trabajos Anteriores

Para el desarrollo del este proyecto, se realizó una investigación sobre los trabajos hechos con los robots SCORBOT-ER en la Facultad de Ingeniería encontrando lo siguiente:

- *Movimiento de un robot teleoperado a través de sistemas de libre distribución por internet. [6][Contreras]*

El proyecto utilizó una computadora conectada a internet y auxiliada por los controladores del fabricante para poder ordenar las acciones que realizará el robot desde una computadora remota, empleando un programa en C para los comandos, mientras que el internet se manejó dentro del entorno de un servidor *Web*. Para el monitoreo de los movimientos del robot, se utilizó una cámara web que mostraba imágenes en tiempo real.

Los resultados del proyecto basado en el uso de los controladores del fabricante fueron:

Características	Problemas
Manejo remoto del robot por medio de Internet.	El encendido del controlador y de la energía para los motores del robot debe ser de manera manual y en el sitio donde se localiza el robot.
Mandar comandos al controlador por medio de comunicación serial.	Falta de control de errores entre el controlador y la comunicación serial.
Mandar comandos en lenguaje del controlador directamente al robot por medio de la página Web.	
Uso de software libre para la comunicación.	

Tabla 1.4 Pros y contras del proyecto “Movimiento de un robot teleoperado a través de sistemas de libre distribución por internet.”

- *Control de un brazo robot de cinco grados de libertad mediante un PLC. [7][Torres]*

El objetivo principal de este proyecto fue crear una solución viable para el reemplazo de los controladores originales de los robots SCORBOT-ER empleando un PLC y un microcontrolador PIC.

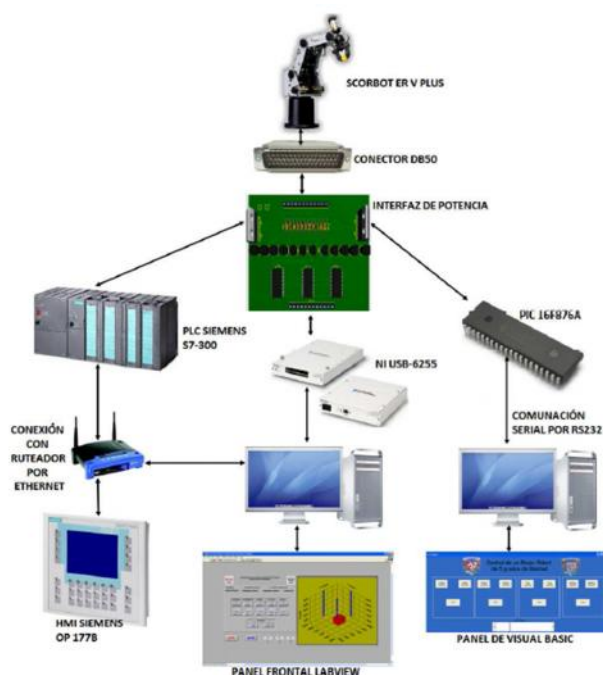


Figura 1.10 Representaciones de Interfaces. [7][Torres]

El proyecto utilizó dos métodos diferentes para operar al robot; “el primero se realizó usando un PLC Siemens S7-300 y una interfaz gráfica diseñada en LabVIEW utilizando una tarjeta NI USB-6255 para conocer la posición del robot. El segundo se realizó con microcontroladores PIC, controlados desde una interfaz creada en *Visual Basic*.” [7][Torres]

Los resultados del proyecto basados en el reemplazo de los controladores del fabricante por medio de un PLC y un microcontrolador PIC, fueron:

Características	Problemas
El uso de la tarjeta NI USB-6255 como contador de pulsos del encoder del robot.	Controlador que regula el voltaje de los motores al usar PLC.
Velocidad sin pérdida de información en la comunicación Ethernet entre el PLC y la PC.	Controlador que realice el control en lazo cerrado al usar PLC.
Flexibilidad en la manipulación y programación del microcontrolador.	Manejo de entradas y salidas en el uso del microcontrolador PIC.
Implementación de un control PID.	Limitaciones de velocidad en la comunicación serial entre el microcontrolador PIC y la PC.
Interfaz HMI.	
Solución de la cinemática del robot por medio de la PC.	

Tabla 1.5 Pros y contras del proyecto “Control de un brazo robot de cinco grados de libertad mediante un PLC.”

- *Control de un brazo Robótico Teleoperado mediante un Acelerómetro. [8]
[Helleman]*

El proyecto desarrollo la teleoperación de una plataforma robótica prediseñada como dispositivo esclavo (SCORBOT-ER V Plus controlado por un microcontrolador ATmega1280). El cual propuso un sistema de medición que utiliza un acelerómetro embebido (Acelerómetro BMA180 y microcontrolador ATmega328 como tarjeta de adquisición) en una tarjeta inalámbrica como dispositivo maestro.

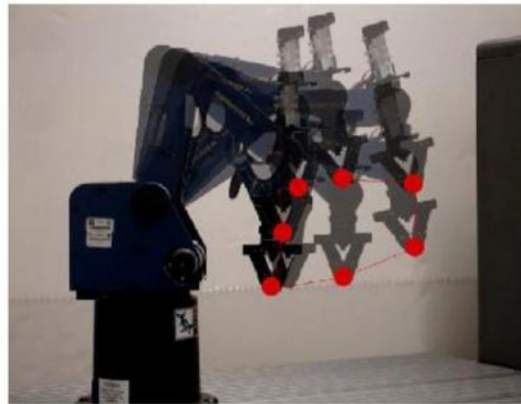


Figura 1.11 Trayectorias del robot. [8] [Helleman]

Los resultados del proyecto basados en el reemplazo de los controladores del fabricante por medio de un por un microcontrolador ATmega1280, fueron:

Características	Problemas
Flexibilidad en la manipulación y programación del microcontrolador. Solución de la cinemática del robot por medio de la PC. Repetitividad.	Retrasos en el movimiento del robot SCORBOT-ER V Plus. Movimientos imprecisos y bruscos debido al control implementado (no se consideró la dinámica del robot). Pérdida de información en el sensado de la posición de los motores. Pérdida de tiempo en el movimiento mecánico del robot.

Tabla 1.6 Pros y contras del proyecto “Control de un brazo Robótico Teleoperado mediante un Acelerómetro.”

- *Teleoperación de un SCORBOT-ER 4u empleando un PAMCA. [9][Rementeria]*

El proyecto consto de “la teleoperación de un SCORBOT-ER 4u vía internet empleando un Pantógrafo Maestro de Cadena Abierta (PAMCA) con la función de reproducir movimientos puntuales y trayectorias. El proceso se diseñó una secuencia específica, en la

cual tiene como inicio la ubicación de la posición del pantógrafo y termina en la reproducción de dicho movimiento por parte del esclavo.” [9][Rementeria]

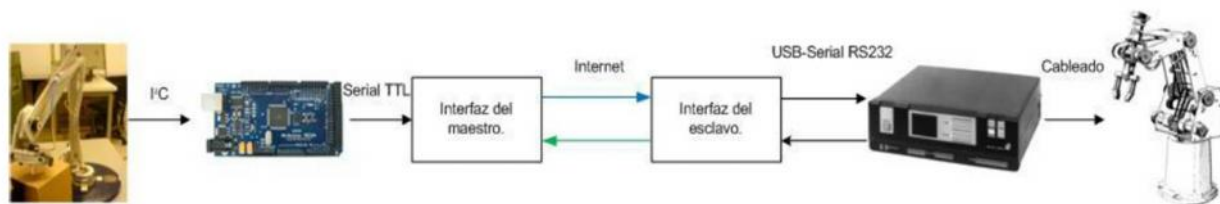


Figura 1.12 Visualización general del proyecto “Teleoperación de un SCORBOT-ER 4u empleando un PAMCA” [9] [Rementeria]

Los resultados del proyecto basados en el uso del el SCORBOT-ER 4u con el “Controller-A” del SCORBOT V:

Características	Problemas
Control correcto. Repetitividad. Precisión. Elementos de seguridad. Programación en el controlador de trayectorias en lenguaje ACL.	Lentitud al procesar operación para la realización de la teleoperación. Limitación para realizar movimiento multiejes y proceso en paralelo.

Tabla 1.7 Pros y contras del proyecto “Teleoperación de un SCORBOT-ER 4u empleando un PAMCA”.

1.1.5 CompacRIO®

CompacRIO® (cRIO) es un sistema embebido y reconfigurable de control y adquisición. La arquitectura robusta del hardware del sistema *CompacRIO*® incluye módulos de E/S, un chasis FPGA reconfigurable y un controlador embebido. Además, *CompacRIO*® se programa con herramientas de programación gráfica de NI LabVIEW® y puede usarse en una variedad de aplicaciones de control y monitoreo embebidos. [E1O]



Figura 1.13 Arquitectura de un CompacRIO®

El controlador en tiempo real proporciona un comportamiento fiable, predecible y se destaca en matemáticas de punto flotante y el análisis, mientras que el FPGA sobresale en tareas más pequeñas que requieren lógica de alta velocidad y el momento preciso.

Controlador en Tiempo Real

El controlador en tiempo real contiene un procesador que ejecuta de forma fiable y determinista aplicaciones de LabVIEW *Real-Time* y ofrece control multinivel, el trazado de la ejecución, a bordo de registro de datos y comunicación con los periféricos.

En la actualidad, el tiempo de procesamiento de un sistema es muy importante para la toma de decisiones y un correcto monitoreo, muchas veces este procesamiento es realizado por las computadoras con un sistema operativo instalado (como Windows) y esto podría retrasar la ejecución del programa debido a: ejecución de antivirus, actualización de gráficos, y así sucesivamente logrando así un fallo del sistema.



Figura 1.14 cRIO-9074 Controlador en Tiempo Real. [E10]

Un Sistema Operativo en Tiempo Real (RTOS, por sus siglas en inglés *Real-Time Operating System*) es diseñado para la ejecución de muchos procesos y aplicaciones en tiempo preciso, además de ofrecer confiabilidad. Específicamente, un RTOS ayuda a implementar:

- ✓ Realizar tareas en un periodo de tiempo del peor caso garantizado.
- ✓ Dar prioridad cuidadosamente las diferentes secciones de su programa.
- ✓ Ejecutar bucles con casi el mismo tiempo en cada iteración.
- ✓ Reinicio automáticamente si el programa deja de funcionar.
- ✓ Robustez de hardware.

Chasis FPGA reconfigurable

El chasis FPGA reconfigurable es el centro de la arquitectura del cRIO. Está conectado directamente a las entradas y salidas para que tenga un acceso rápido a las entradas y

salidas de cada módulo, este diseño beneficia a la respuesta del sistema debido a que no existirán retrasos de control.

Los arreglos de compuertas programables en campo (FPGAs, por sus siglas en inglés *Field Programmable Gate Array*) son chips de silicio reprogramables. Un FPGA es un chip programable compuesto de tres componentes básicos: bloques lógicos (compuertas lógicas, multiplicadores, flip-flops, etc), interconexiones programables y bloques de entradas y salidas.

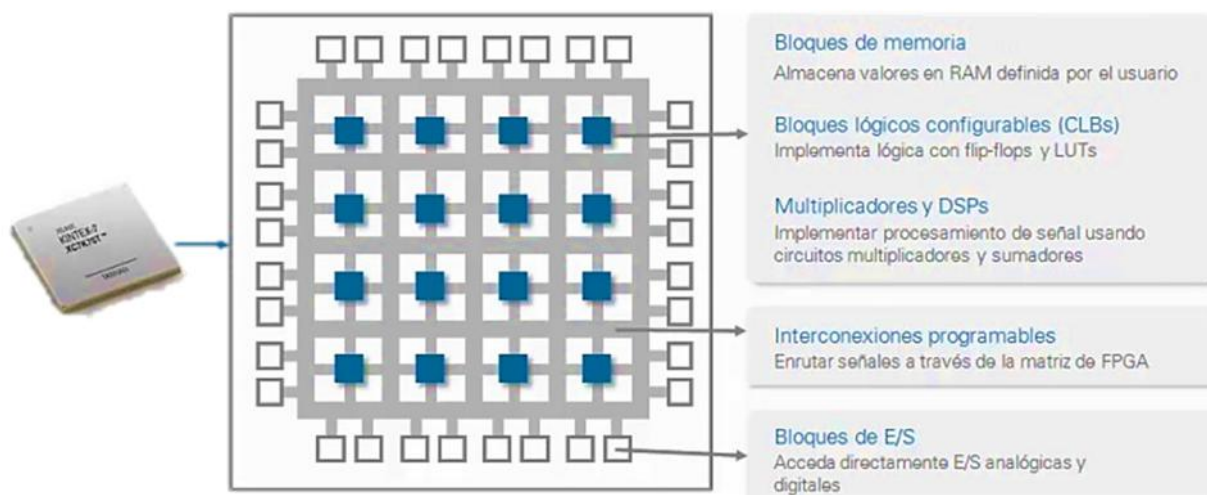


Figura 1.15 Arquitectura de un chip FPGA. [E9]

Ross Freeman, inventó el primer FPGA en 1985. El uso de los FPGA en las industrias es gracias a que los FPGAs combinan lo mejor de los circuitos integrados de aplicación específica (ASICs) y sistemas basados en procesador.

Los FPGAs ofrecen flexibilidad en software de poder reprogramarlo y en el hardware ejecutar tareas en forma paralela, tener fiabilidad y una respuesta muy rápida. Las aplicaciones comunes de los FPGA son: control de alta velocidad, adquisición de datos personalizada, protocolos de comunicación digital y procesamiento y reducción de datos en la tarjeta.

Módulos de E/S

Módulos E / S contienen aislamiento, circuitos de conversión, acondicionamiento de señales y conectividad integrada para la conexión directa a sensores / actuadores industriales. Se puede elegir más de 70 módulos de E/S para un cRIO y así conectarse con cualquier sensor o actuador.



Figura 1.16 Diferentes tipos de Módulos de E/S. [E10]

El Centro de Ingeniería Avanzada de la Facultad de Ingeniería de U.N.A.M., cuenta con el sistema embebido “NI cRIO-9073” el cual contiene un controlador tiempo real y un chasis FPGA reconfigurable de ocho slots, además se utilizarán módulos de E/S “NI 9505” los cuales servirán de etapa de potencia y lectura de encoder de cada motor y un módulo de E/S digitales de ocho canales “NI 9401” para la lectura de los interruptores de cada articulación. Dichos modelos cuentan con las siguientes características:

NI cRIO-9073 [E14]

- Sistema robusto y embebido de monitoreo y control.
- Procesador industrial en tiempo real de 266 MHz para control, registro de datos y análisis.
- Chasis FPGA con compuertas de 2M y 8 ranuras, para temporización, control y procesamiento de E/S personalizados.
- Puerto Ethernet 10/100Base-T; puerto serial RS232 para conexión a periféricos.
- Rango de temperatura de operación de -20 a 55 °C; una sola entrada de suministro de potencia de 19 a 30 VDC.

NI 9505 [E15]

- Módulo de Drive Servo de DC de Escobillas con Puente H Completo.
- Corriente continua hasta 5 A a 40 °C (o 1 A a 70 °C) a 30 V - para mayor potencia se le puede agregar el accesorio NI 9931.
- Se puede utilizar datos desde el sensor de corriente para tiempo de muestreo flexible y filtro de la corriente del motor.
- Drive de servo motor con puente H completo con una interfaz de codificador y sensor de corriente integrados.
- Contiene la característica de configuración y rendimiento del arreglo de compuerta programable en campo (FPGA).

NI 9401 [E16]

- 8 canales, E/S digitales de 100 ns de alta velocidad.
- E/S digital tipo sinking/sourcing de 5 V/TTL.

- Aislamiento CAT I de 60 VDC.
- Conector D-SUB de 25 pines estándar en la industria.
- Temperatura de operación de -40 °C a 70 °C, 5 g de vibración, 50 g de impacto.
- Se puede programar para tres configuraciones: 8 entradas digitales, 8 salidas digitales o 4 entradas digitales y 4 salidas digitales.

1.2 Objetivos

El objetivo principal de esta tesis es proponer, diseñar e implementar una solución viable para el reemplazo del controlador original del robot SCORBOT-ER 4u, por medio de dispositivos que permitan ejecutar los procesos en tiempo real, en paralelo y además que permita la libertad de programar diferentes técnicas de control.

1.3 Alcances

Los alcances del presente trabajo se limitarán a los siguientes puntos:

- Programación del sistema embebido *CompacRIO*® tanto de su controlador en tiempo real y del chasis FPGA reconfigurable para la lectura y generación de señales de cada módulo.
- Realización de una interfaz hombre-máquina (HMI, por su siglas en inglés *Human-Machine Interface*) en LabVIEW®, para el monitoreo del funcionamiento y variables del proceso.
- Caracterización de los motores del robot SCORBOT-ER 4u para el control de cada una de las articulaciones y la pinza.
- Realización de los diagramas de conexión implementados en este trabajo entre el robot SCORBOT-ER 4u y el *CompacRIO*® (específicamente en los módulos de E/S).

2 Cinemática del Robot

La cinemática del robot se encarga del estudio del movimiento del mismo con respecto a un sistema de referencia. Por lo tanto, la cinemática se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo, y en particular por las relaciones entre la posición y orientación del extremo final del robot con los valores que toman sus coordenadas articulares. [1] [Barrientos]

2.1 Cinemática directa

Para resolver el problema cinemático directo del robot SCORBOT-ER 4u, se optó por utilizar las matrices de transformación homogéneas por Denavit-Hartenberg, o DH.

En general, a cada eslabón de un robot de n grados de libertad, se le puede asignar un sistema de referencia solidario a él y, al utilizar transformaciones homogéneas, es posible representar las rotaciones y traslaciones relativas entre los distintos eslabones que componen a cada robot. A cada matriz de transformación homogénea que representa la posición y orientación relativa entre los sistemas asociados a dos eslabones consecutivos del robot se suele denominar matriz ${}^{i-1}A_i$. Del mismo modo, se puede representar de forma total o parcial la cadena cinemática que forma a cada robot con 0A_n como la matrices resultante del producto de las matrices ${}^{i-1}A_i$ con i desde 0 hasta n .

En esta convención, cada transformación homogénea A_i es representada como un producto de cuatro básicas transformaciones:

$$A_i = Rotación_{z,\theta_i} Traslación_{z,d_i} Traslación_{x,a_i} Rotación_{x,\alpha_i} \quad (2.1)$$

$$A_i = \begin{bmatrix} c_{\theta_i} & -s_{\theta_i}c_{\alpha_i} & s_{\theta_i}c_{\alpha_i} & a_i c_{\theta_i} \\ s_{\theta_i} & c_{\theta_i}c_{\alpha_i} & c_{\theta_i}s_{\alpha_i} & a_i s_{\theta_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Donde las cuatro variables θ_i , a_i , d_i y α_i son parámetros asociados al eslabón i .

1. $Rotación_{z,\theta_i} \rightarrow$ Rotación alrededor del eje z_{i-1} un ángulo θ_i .
2. $Traslación_{z,d_i} \rightarrow$ Traslación a lo largo de z_{i-1} una distancia d_i .
3. $Traslación_{x,a_i} \rightarrow$ Traslación a lo largo de x_i una distancia a_i .
4. $Rotación_{x,\alpha_i} \rightarrow$ Rotación alrededor del eje x_i un ángulo α_i .

El robot SCORBOT-ER 4u cuenta con los siguientes parámetros:

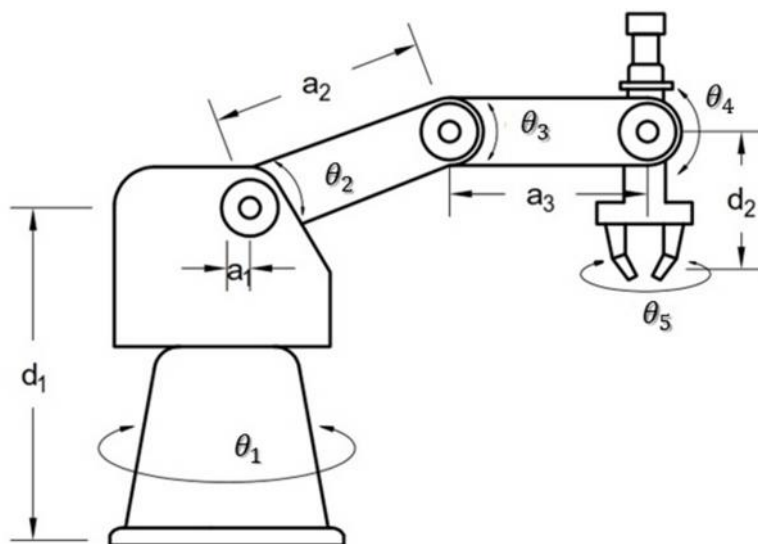


Figura 2.1 Parámetros definidos del robot SCORBOT-ER 4u.

	34.9 [cm]
	14.5 [cm]
	1.2 [cm]
	22.2 [cm]
	22.2 [cm]

Tabla 2.1 Valores de los parámetros.

2.1.1 Algoritmo de D-H aplicado al robot SCORBOT-ER 4u

D-H 1 Se enumeran los eslabones comenzando con 1 y acabando con. La base del robot se considera eslabo 0.

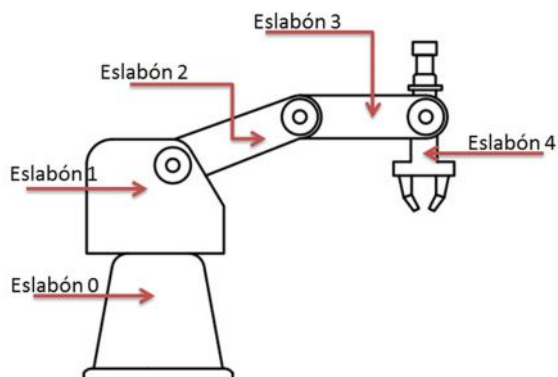


Figura 2.2 Identificación de Eslabones.

D-H 2 Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad y acabando en n).

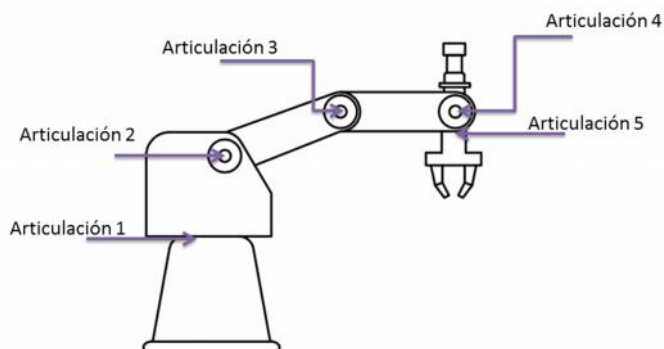


Figura 2.3 Identificación de articulaciones.

D-H 3 Localizar el eje de cada articulación. Si esta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.

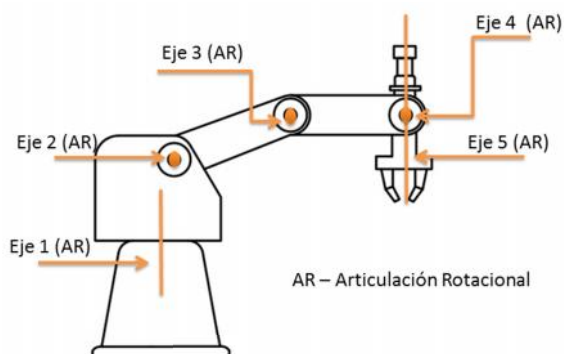


Figura 2.4 Ejes de cada articulación y tipo de articulación.

D-H 4 Para i de 0 a $n-1$, situar el eje Z_i sobre el eje de la articulación $i+1$.

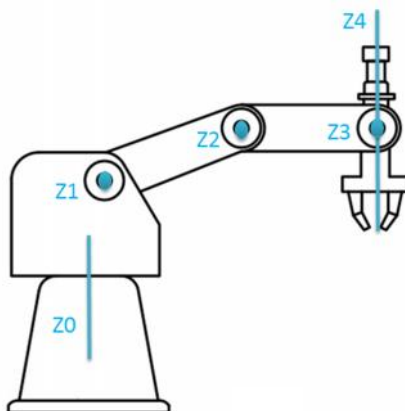


Figura 2.5 Ubicación de los ejes "Z".

D-H 5 Situar el origen del sistema de la base (S_0) en cualquier punto del eje Z_0 . Los ejes X_0 e Y_0 se situarán de modo que formen un sistema dextrógiro con Z_0 .

D-H 6 Para i de 1 a $n-1$, situar el sistema (S_i) (solidario al eslabón i) en la intersección del eje Z_i con la línea normal común a Z_{i-1} y Z_i . Si ambos ejes se cortasen se situaría (S_i) en el punto de corte. Si fuesen paralelos (S_i) se situaría en la articulación $i+1$.

D-H 7 Situar X_i en la línea normal común a Z_{i-1} y Z_i .

D-H 8 Situar Y_i de modo que forme un sistema dextrógiro con X_i y Z_i ,

D-H 9 Situar el sistema (S_n) en el extremo del robot de modo que Z_n coincida con la dirección de Z_{n-1} y X_0 sea normal a Z_{n-1} y Z_n .

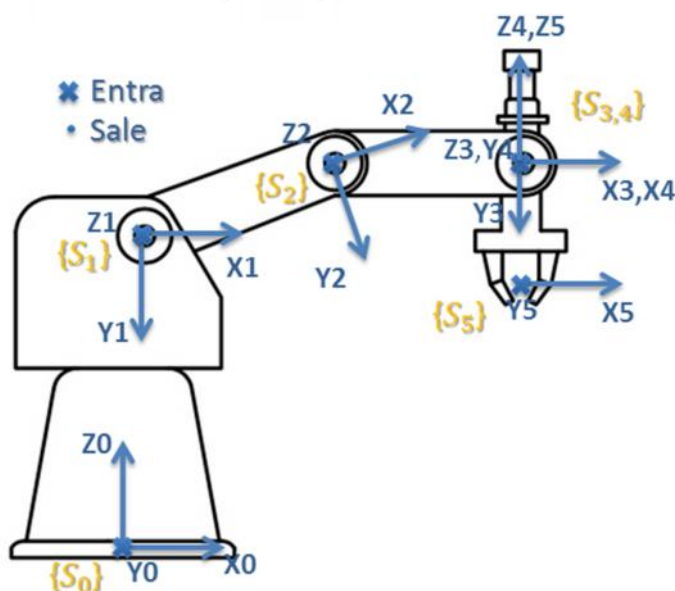


Figura 2.5 Asignación de ejes de referencia.

Una vez plasmados los sistemas de referencia del robot, los pasos del algoritmo D-H 9 al D-H 13 sirven para obtener los parámetros de D-H.

i	a_i	α_i	d_i	θ_i
1	a_1	α_1	d_1	θ_1
2	a_2	α_2	0	θ_2
3	a_3	α_3	0	θ_3
4	a_4	α_4	0	θ_4
5	a_5	α_5	0	θ_5

Tabla 2.2 Parámetros D-H para el robot SCORBOT-ER 4u.

D-H 14 Obtener las matrices de transformación ${}^{i-1}A_i$.

$${}^0A_1 = \begin{bmatrix} c\theta^1 & 0 & s\theta^1 & a_1 s\theta^1 \\ s\theta^1 & 0 & c\theta^1 & d_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$${}^1A_2 = \begin{bmatrix} c\theta^2 & s\theta^2 & 0 & a_2 c\theta^2 \\ s\theta^2 & c\theta^2 & 0 & a_2 s\theta^2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

$${}^2A_3 = \begin{bmatrix} c\theta^3 & s\theta^3 & 0 & a_3 c\theta^3 \\ s\theta^3 & c\theta^3 & 0 & a_3 s\theta^3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.5)$$

$${}^3A_4 = \begin{bmatrix} c\theta^4 & 0 & -s\theta^4 & 0 \\ s\theta^4 & 0 & c\theta^4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.6)$$

$${}^4A_5 = \begin{bmatrix} c\theta^5 & -s\theta^5 & 0 & 0 \\ s\theta^5 & c\theta^5 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

D-H 15 Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot $T = {}^0A_1 \cdot {}^1A_2 \cdot \dots \cdot {}^{n-1}A_n$.

$$T = {}^0A_1 + {}^1A_2 + {}^2A_3 + {}^3A_4 + {}^4A_5 \quad (2.8)$$

$$\begin{bmatrix} \cos[\theta_1] \cos[\theta_{234}] \cos[\theta_5] - \sin[\theta_1] \sin[\theta_5] & -\cos[\theta_5] \sin[\theta_1] - \cos[\theta_1] \cos[\theta_{234}] \sin[\theta_5] & \cos[\theta_1] \sin[\theta_{234}] & \cos[\theta_1] (a_1 + a_2 \cos[\theta_2] + a_3 \cos[\theta_{23}] + d_2 \sin[\theta_{234}]) \\ \cos[\theta_{234}] \cos[\theta_5] \sin[\theta_1] - \cos[\theta_1] \sin[\theta_5] & \cos[\theta_1] \cos[\theta_5] - \cos[\theta_{234}] \sin[\theta_1] \sin[\theta_5] & \sin[\theta_1] \sin[\theta_{234}] & \sin[\theta_1] (a_1 + a_2 \cos[\theta_2] + a_3 \cos[\theta_{23}] + d_2 \sin[\theta_{234}]) \\ -\cos[\theta_5] \sin[\theta_{234}] & \sin[\theta_{234}] \sin[\theta_5] & \cos[\theta_{234}] & d_1 + d_2 \cos[\theta_{234}] - a_2 \sin[\theta_2] - a_3 \sin[\theta_{23}] \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

D-H 16 La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación del extremo referido a la base en función de las n coordenadas articulares.

$$T = \begin{bmatrix} R_{3 \times 3} & P_{3 \times 1} \\ f_{1 \times 3} & 1 \times 1 \end{bmatrix} = \begin{bmatrix} \text{matriz de} & \text{vector de} \\ \text{rotación} & \text{posición} \\ \text{transformación} & \text{escalado} \\ \text{de perspectiva} & \end{bmatrix}$$

Por lo tanto, el vector de posición quedaría expresado con las siguientes ecuaciones:

$$P_x = \cos[\theta_1](a_1 + a_2 \cos[\theta_2] + a_3 \cos[\theta_2 + \theta_3] + d_2 \sin[\theta_2 + \theta_3 + \theta_4]) \quad (2.9)$$

$$P_y = \sin[\theta_1](a_1 + a_2 \cos[\theta_2] + a_3 \cos[\theta_2 + \theta_3] + d_2 \sin[\theta_2 + \theta_3 + \theta_4]) \quad (2.10)$$

$$P_z = d_1 + d_2 \cos[\theta_2 + \theta_3 + \theta_4] - a_2 \sin[\theta_2] - a_3 \sin[\theta_2 + \theta_3] \quad (2.11)$$

2.2 Cinemática inversa

La cinemática inversa se encarga de encontrar los valores que deben adoptar las coordenadas articulares del robot $\theta = [\theta_1, \theta_2, \dots, \theta_n]^T$ para que el efector final se posicione y oriente en una determinada localización espacial.

Para la resolución de la cinemática inversa, existen diversos métodos: por medio de métodos numéricos, por medio de la matriz de transformaciones homogéneas y métodos geométricos. En este trabajo ya que el robot cuenta con 5 grados de libertad, se utilizara el método de desacoplo cinemático, en el cual como su nombre lo dice desacopla el problema cinemático inverso en dos simples problemas: cinemática de posición inversa y cinemática de orientación inversa.

2.2.1 Cinemática de posición inversa

Se obtiene los primeros ángulos que resuelven la posición θ_1, θ_2 y θ_3 .

$${}^0A_3 = {}^0A_1 * {}^1A_2 * {}^2A_3 \quad (2.12)$$

$${}^0A_3 * n = \begin{bmatrix} \cos[\theta_1] \{ a_1 + a_2 \cos[\theta_2] + a_3 \cos[\theta_2 + \theta_3] \} & \sin[\theta_1] \{ a_1 + a_2 \cos[\theta_2] + a_3 \cos[\theta_2 + \theta_3] \} \\ \sin[\theta_1] \{ a_1 + a_2 \cos[\theta_2] + a_3 \cos[\theta_2 + \theta_3] \} & \cos[\theta_1] \{ a_1 + a_2 \cos[\theta_2] + a_3 \cos[\theta_2 + \theta_3] \} \\ d_1 - a_2 \sin[\theta_2] - a_3 \sin[\theta_2 + \theta_3] & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.13)$$

Donde $n = \{0, 0, 0, 1\}$.

$${}^3A_5 = {}^3A_4 * {}^4A_5 \quad (2.14)$$

$${}^3A_5 * n = \begin{bmatrix} q_2 \sin[\theta_4] & 0 \\ -d_2 \cos[\theta_4] & 0 \\ 0 & 1 \end{bmatrix} \quad (2.15)$$

Se extrae los primeros tres elementos de (2.13)

$${}^0E_3 = \begin{bmatrix} C_1 C_2 C_3 + a_2 C_1 S_2 + a_3 C_1 S_2 S_3 + d_1 \\ C_1 C_2 S_3 + a_2 C_1 C_3 + a_3 C_1 C_3 S_3 + d_1 \\ C_1 S_2 C_3 + a_2 S_2 + a_3 S_2 S_3 + d_1 \\ a_1 + d_1 - a_2 S_2 - a_3 S_2 S_3 \end{bmatrix} \quad (2.16)$$

Se despeja 0A_1 de (2.12) y siendo el primer término de la igualdad

$${}^0A_1 = {}^0A_2 {}^2A_3 \quad (2.17)$$

Y el segundo miembro de la igualdad

$${}^0A_1 = \text{Inversa} [{}^0E_3] \quad (2.18)$$

Se define la matriz del efector final

$$T_{1m} = \begin{bmatrix} nx & ox & ax & Pxm \\ ny & oy & ay & Pym \\ nz & oz & az & Pzm \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.19)$$

Se calcula

$$L1\theta_1 = {}^1A_0 + T_{1m} + n \quad (2.20)$$

$$L1\theta_1 = \begin{bmatrix} -a_1 + Pxm C_1 + Pym S_1 \\ d_1 - Pzm \\ Pym C_1 - Pxm S_1 \\ 1 \end{bmatrix} \quad (2.21)$$

Se iguala el termino (1, 3) de (2.13) y (2.17), se observa que se puede utilizar la expresión de ArcTan en función de seno y coseno. Por lo tanto

$$\theta_1 = \text{ArcTan}[Pxm, Pym] \quad (2.22)$$

Debido al desacoplo cinemático

$$\theta_1 = \text{ArcTan}[Pxm, Pym] \quad (2.23)$$

$$Pxm = Px - (d_2)ax \quad (2.24)$$

$$Pym = Py - (d_2)ay \quad (2.25)$$

Sustituyendo (2.23) y (2.24) en (2.22)

$$\theta_1 = \text{ArcTan}[Px - (d_2)ax, Py - (d_2)ay] \quad (2.26)$$

Ahora se despeja 1A_2 de (2.12) obteniendo el nuevo primer miembro de la igualdad

$${}^1A_2 = {}^1A_3 {}^3A_4 \quad (2.27)$$

Y el segundo miembro de la igualdad

$${}^2A_0 = \text{Inversa} [{}^0A_1 * {}^1A_2] \quad (2.28)$$

Se multiplica (2.28) por $(T1m * n)$ para obtener

$${}^2A_0 = \begin{bmatrix} -a_2 \cos[\theta_2] + (a_1 - Pzm) \cos[\theta_2] + (a_1 - Pxm) \cos[\theta_1] - Pym \sin[\theta_1] \sin[\theta_2] \\ (a_1 - Pzm) \sin[\theta_2] + (a_1 - Pxm) \sin[\theta_1] - Pym \cos[\theta_1] \sin[\theta_2] \\ Pxm \sin[\theta_1] \\ 1 \end{bmatrix} \quad (2.29)$$

Se utiliza los términos (1, 1) y (1, 2) de (2.27) y (2.29)

$$LD2x = a_2 \cos[\theta_2] \quad (2.30)$$

$$LD2y = a_2 \sin[\theta_2] \quad (2.31)$$

$$L12x = (a_1 - Pxm) \cos[\theta_1] - Pym \sin[\theta_1] \quad (2.32)$$

$$L12y = (a_1 - Pzm) \sin[\theta_1] - Pym \cos[\theta_1] \quad (2.33)$$

Se agrupan términos semejantes, mando a a_2 del derecho de la igualdad y reemplazo

$$\begin{aligned} (-a_1 + Pxm \cos[\theta_1] + Pym \sin[\theta_1]) &\rightarrow A \\ (d_1 - Pzm) &\rightarrow B \end{aligned}$$

Por lo tanto

$$LD2xs = A + \cos[\theta_2] \quad ; \quad B + \sin[\theta_2] \quad (2.34)$$

$$LD2ys = A + \cos[\theta_2] \quad ; \quad B + \sin[\theta_2] \quad (2.35)$$

$$L12xs = a_2 \cos[\theta_2] + B + \sin[\theta_2] \quad (2.36)$$

$$L12ys = a_2 \sin[\theta_2] + B + \cos[\theta_2] \quad (2.37)$$

Elevando al cuadrado ambas expresiones y sumándolas

$$S2I = L12x^2 + L12ys^2 \quad (2.38)$$

$$S2D = LD2x^2 + LD2ys^2 \quad (2.39)$$

Resolviendo la igualdad $S2I = S2D$

$$= \frac{((a_1)^2 - (a_2)^2 - (a_3 + d_1 - Pzm)(a_3 - d_1 + Pzm) + (Pxm \cos[\theta_1] + Pym \sin[\theta_1])(-2a_1 + Pxm \cos[\theta_1] + Pym \sin[\theta_1]))}{2a_2a_3} \cos[\theta_3] \quad (2.40)$$

Se utiliza la identidad trigonométrica

$$\sin[\theta_3] = \sqrt{1 - (\cos[\theta_3])^2} \quad (2.41)$$

Debido a lo anterior, y sustituyendo (2.23), (2.24) y (2.25) en (2.40), se puede obtener θ_3

$$\theta_3 = \text{Arctan}(\text{Cos}[\theta_3], \text{Sin}[\theta_3]) \quad (2.42)$$

Ahora ya obtenido el valor de θ_3 , se resuelven las igualdades siguientes

$$L12xs - LD2xs = 0 \quad (2.43)$$

$$L12ys - LD2ys = 0 \quad (2.44)$$

En el Cual el resultado es

$$\begin{aligned} & \frac{\text{Cos}[\theta_2]}{(d_1 - Pzm)^2 + (-a_1 + Pxm\text{Cos}[\theta_1] + Pym\text{Sin}[\theta_1])^2} \\ & = \frac{(-(a_2 + a_3\text{Cos}[\theta_3])(a_1 - Pxm\text{Cos}[\theta_1] - Pym\text{Sin}[\theta_1]) + a_3(d_1 - Pzm)\text{Sin}[\theta_3])}{(d_1 - Pzm)^2 + (-a_1 + Pxm\text{Cos}[\theta_1] + Pym\text{Sin}[\theta_1])^2} \quad (2.45) \\ & \frac{\text{Sin}[\theta_2]}{(a_1)^2 + (d_1 - Pzm)^2 + (Pxm\text{Cos}[\theta_1] + Pym\text{Sin}[\theta_1])(-2a_1 + Pxm\text{Cos}[\theta_1] + Pym\text{Sin}[\theta_1])} \\ & = \frac{(d_1 - Pzm)(a_3\text{Cos}[\theta_3]) + a_3(a_1 - Pxm\text{Cos}[\theta_1] - Pym\text{Sin}[\theta_1])\text{Sin}[\theta_3]}{(a_1)^2 + (d_1 - Pzm)^2 + (Pxm\text{Cos}[\theta_1] + Pym\text{Sin}[\theta_1])(-2a_1 + Pxm\text{Cos}[\theta_1] + Pym\text{Sin}[\theta_1])} \quad (2.46) \end{aligned}$$

Debido a lo anterior, y sustituyendo (2.23), (2.24) y (2.25) en (2.45) y (2.46), se puede obtener θ_2

$$\theta_2 = \text{Arctan}(\text{Cos}[\theta_2], \text{Sin}[\theta_2]) \quad (2.47)$$

2.2.2 Cinemática de orientación inversa

Se obtiene la matriz de 3x3 que nos proporciona la orientación de (2.14)

$$R35Iz = \begin{bmatrix} \text{Cos}[\theta_5] & \text{Sin}[\theta_5] & 0 \\ -\text{Sin}[\theta_5] & \text{Cos}[\theta_5] & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.48)$$

También, se obtiene la matriz de 3x3 que nos proporciona la orientación de (2.12) y se obtiene su matriz inversa

$$R3I = \begin{bmatrix} \text{Cos}[\theta_1] & \text{Sin}[\theta_1] & 0 \\ -\text{Sin}[\theta_1] & \text{Cos}[\theta_1] & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.49)$$

Se obtiene la matriz de orientación del efector final de (2.19)

$$R_{05} = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix} \quad (2.50)$$

Se multiplica (2.49) y (2.50)

$$R_{35D} = R_{3I} * R_{05} \quad (2.51)$$

Debido al gran contenido del resultado de R_{35D} , no se mostrara en este apartado.

Al igualar (2.48) y (2.51) el término (3, 2), se puede obtener θ_5

$$\cos[\theta_5] = o_y + \cos[\theta_1] * o_x + \sin[\theta_1] \quad (2.52)$$

Y por medio de la identidad trigonométrica (2.41)

$$\sin[\theta_5] = \sqrt{1 - (o_y + \cos[\theta_1] * o_x + \sin[\theta_1])^2} \quad (2.53)$$

Debido a las ecuaciones anteriores, se puede obtener θ_5

$$\theta_5 = \text{ArcTan}(\cos[\theta_5], \sin[\theta_5]) \quad (2.54)$$

Ahora, al igualar (2.48) y (2.51) el término (2, 3), se puede obtener θ_4

$$\cos[\theta_4] = a_z + \cos[\theta_2 + \theta_3] * (a_x + \cos[\theta_1] * a_y + \sin[\theta_1] \sin[\theta_2 + \theta_3]) \quad (2.55)$$

$$\sin[\theta_4] = \sqrt{1 - (a_z + \cos[\theta_2 + \theta_3] * (a_x + \cos[\theta_1] * a_y + \sin[\theta_1] \sin[\theta_2 + \theta_3]))^2} \quad (2.56)$$

Debido a las ecuaciones anteriores, se puede obtener θ_4

$$\theta_4 = \text{ArcTan}(\cos[\theta_4], \sin[\theta_4]) \quad (2.57)$$

En resumen, las ecuaciones finales son:

$$\begin{aligned} \theta_1 &= \text{ArcTan}[P_x - (d_2)ax, P_y - (d_2)ay] \\ \theta_2 &= \text{ArcTan}(\cos[\theta_2], \sin[\theta_2]) \\ \theta_3 &= \text{ArcTan}(\cos[\theta_3], \sin[\theta_3]) \\ \theta_4 &= \text{ArcTan}(\cos[\theta_4], \sin[\theta_4]) \\ \theta_5 &= \text{ArcTan}(\cos[\theta_5], \sin[\theta_5]) \end{aligned}$$

3 Implementación

El controlador CompacRIO® nos permite trabajar de forma conjunta con una computadora, la cual en muchas ocasiones es ocupada como monitorio del proceso o tarea a desarrollar (caso de la presente tesis). También permite trabajar de manera independiente, debido al controlador integrado en tiempo real y el chasis FPGA reconfigurable que contiene, dichos elementos trabajaran con el último proyecto guardado.

De manera muy general, el *cRIO-9073* contiene dos puertos: Ethernet y RS-232. A través del puerto Ethernet se realiza el enlace con una PC, este enlace permite la transferencia de información entre la plataforma cRIO y la PC, también permite la programación del controlador integrado en tiempo real y el chasis FPGA reconfigurable. El controlador integrado de tiempo real es el encargado de administrar los dos puertos ya mencionados, además de poder comunicarse con la tarjeta de estado sólido (SSD, por sus siglas en ingles), el cual almacena el último proyecto guardado. La conexión entre el controlador de tiempo real y el chasis FPGA se realiza vía un bus interconexión de componentes periféricos (PCI, por sus siglas en ingles) de alta velocidad. Por último, el FPGA es el encargado de interactuar con los módulos de E/S, y a vez estos con los actuadores o sensores.

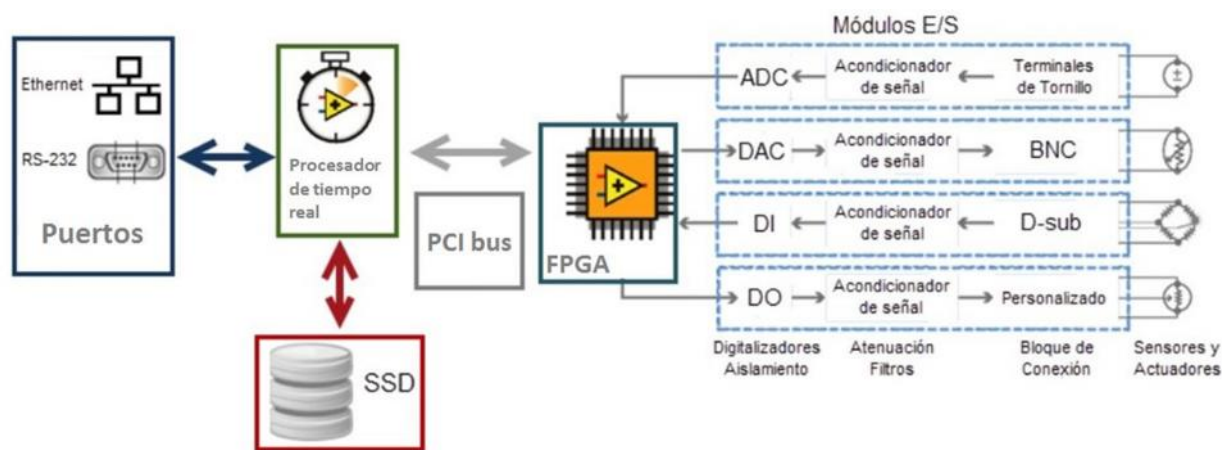


Figura 3.1 Esquema de la plataforma CompactRIO®. [E17]

Como ya se había mencionado, el robot SCORBOT-ER 4u cuenta con 6 motores para el movimiento de sus articulaciones, debido a ello, se realizó el siguiente diagrama general de trabajo el cual incluye las funciones principales a realizar por el *CompacRIO®*.

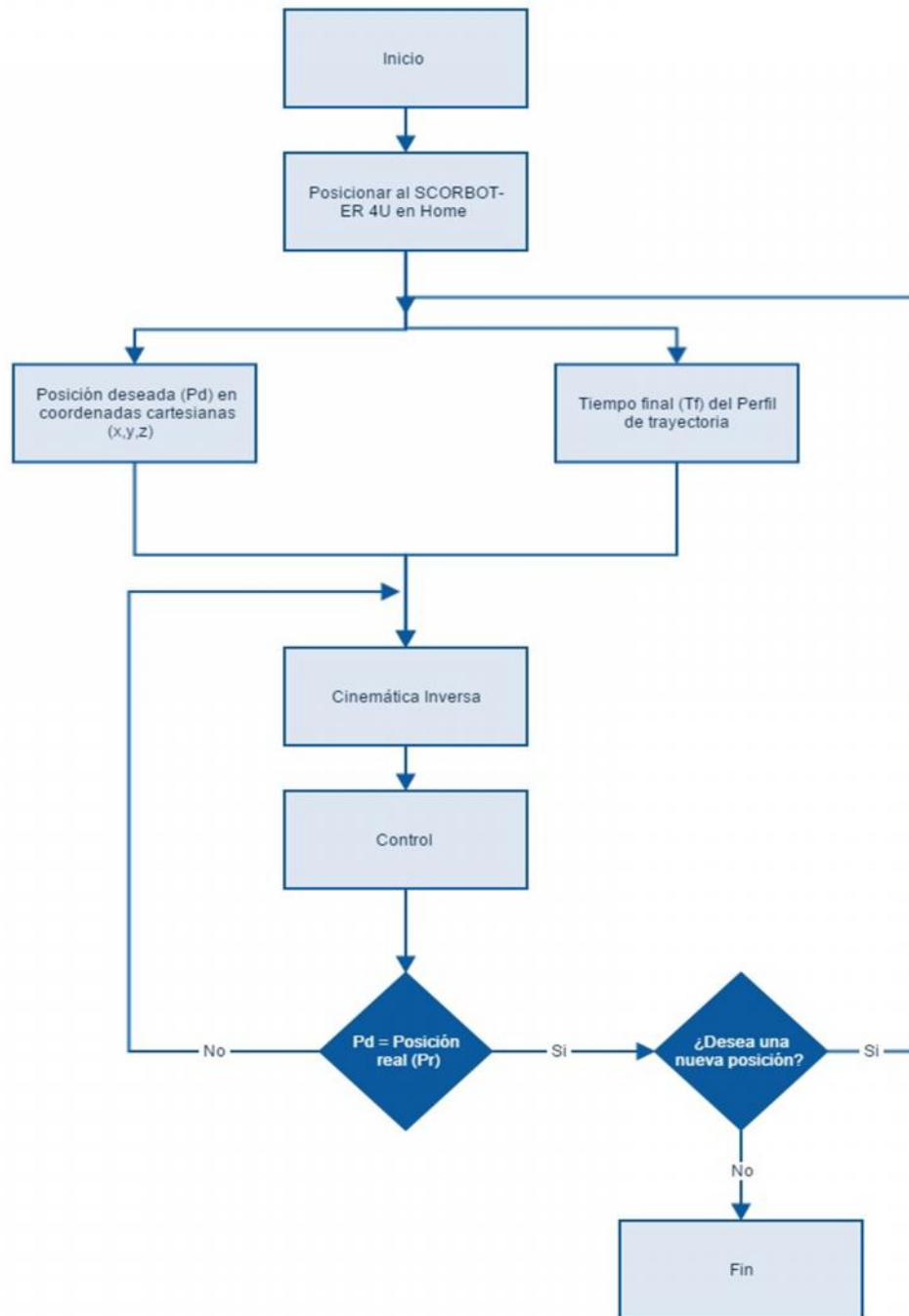


Figura 3.2 Diagrama general de trabajo y funciones principales de CompactRIO®.

La plataforma *cRIO-9073* es un chasis que contiene 8 slots de los cuales se ocuparon los 6 primeros slots para conectar los módulos de E/S *NI 9505*, a cada módulo se le asignará un motor de el robot SCORBOT-ER 4u para realizar la tarea de lectura de encoder y etapa de potencia. También se utilizará un slot en el cual se conectará el módulo *NI 9401* el cual se encargará de la lectura de los interruptores de cada articulación (en total 5 interruptores).

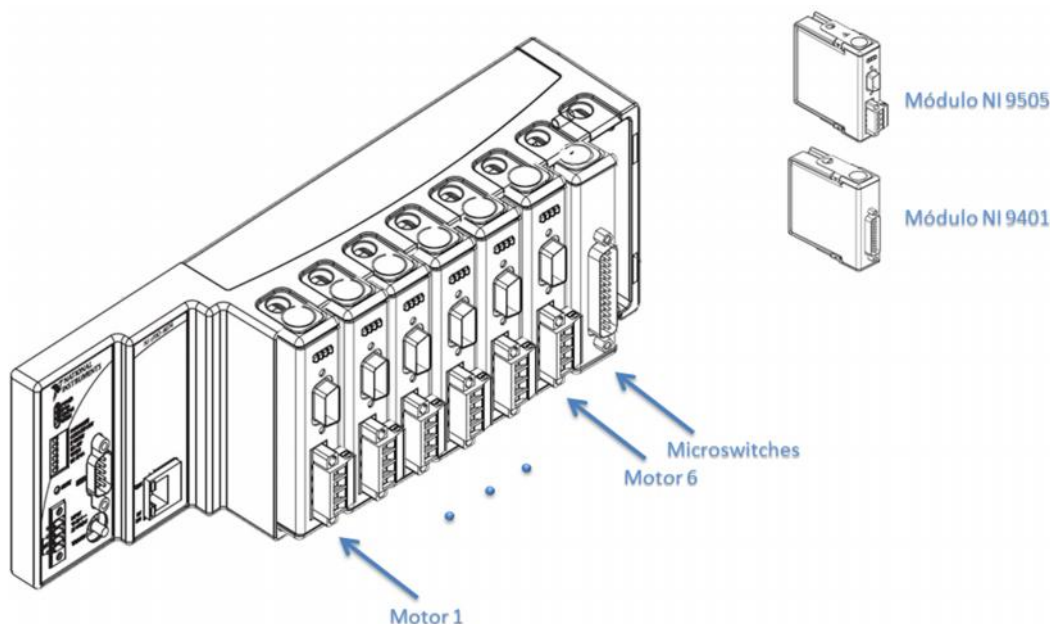


Figura 3.3 Distribución de los módulos de E/S en el cRIO 9073.

3.1 Programación en el chasis FPGA

En esta sección del *CompacRIO*® se elaboró un programa el cual es el encargado de realizar las siguientes funciones:

- Posicionamiento de articulaciones por medio de las lecturas de encoders.
- Generación de PWM para cada motor.
- Ejecución del control PID.
- Lectura de interruptores de cada articulación para el posicionamiento de *HOME*.

3.1.1 Posicionamiento de articulaciones

Los motores del SCORBOT-ER 4u son de la marca Pittman, funcionan con una tensión de 12 [Voltios] de CD. Su consumo de corriente varía entre 0.5 y 1 [Ampere], dependiendo de la carga que se ejerce sobre el motor.

Estos motores cuentan con un encoder eléctrico-óptico e incremental. Cuando un eje del robot se mueve, el encoder genera una serie alternada de pulsos altos y bajos, el número de pulsos es proporcional al movimiento del eje, además la secuencia de pulsos indica el sentido de giro. Cada encoder cuenta con las siguientes terminales, las cuales están ubicadas en el conector principal del robot que irán conectadas al módulo de E/S NI 9505.

Pin	Función
2	Fototransistor (P_0)
5	Fototransistor (P_1)

11	Voltaje de LED (V_{LED})
33	Tierra (GND)

Tabla 3.1 Terminales de encoder del motor 1 ubicadas en el conector del robot.

Para la lectura de cada encoder incremental, se utilizó una codificación X4. Este es un tipo común de codificación, donde cada conjunto de pulsos de cuadratura del encoder incrementa o decrementa la posición de la cuenta por 4.

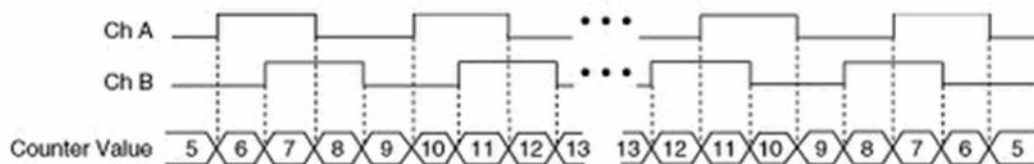


Figura 3.4 Ejemplo de codificación X4. [E18]

Por lo tanto se realizó el siguiente diagrama de bloques:

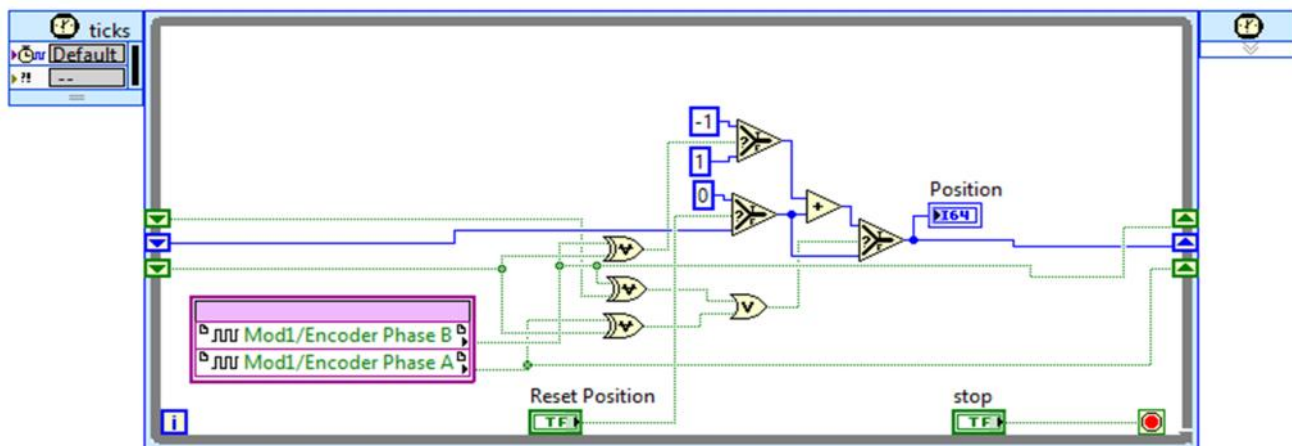


Figura 3.5 Estimación de la Posición.

Dónde:

Encoder Phase A → Pulso de Fototransistor (P_1)

Encoder Phase B → Pulso de Fototransistor (P_0)

Position → Cuenta acumulada

Reset Position → Asignar el valor de 0 en la cuenta acumulada

stop → Detener el ciclo

Para obtener la relación de pulsos por cada grado $[\circ]$, se pretendió calcularlas teóricamente ya que el robot cuenta con transmisión de movimiento en algunas articulaciones por medio de juego de engranes y bandas, pero debido a que en el manual del robot no especifica el número de dientes de los engranes y los diámetros de las poleas para el movimiento de las

bandas y la dificultad de desarmar el robot, se optó por utilizar un método experimental, por medio de la ayuda del software de fábrica del robot “SCORBASE”.

La SCORBASE contiene una ventana en la cual muestra la posición en grados de cada articulación y la cuanta de pulsos del encoder al realizar un movimiento.

☒ XYZ:	X(mm):	169.03	Y(mm):	0.00	Z(mm):	504.33	Elev.(g):	-63.55	Giro(g):	0.00						
☒ Ejes:	Base:	0.00	Brazo superior:	-120.27	Brazo inferior:	95.03	Elev.	88.81	Giro	0.00						
☒ Contador de encoders:	1:	0	2:	0	3:	0	4:	0	5:	0	6:	0	7:	0	8:	0

Figura 3.6 Ventana de SCORBASE donde muestra ángulos y contadores de los pulsos de encoder.

En primer lugar, en la ventana de SCORBASE se colocó las articulaciones en donde la cuenta de los encoders sea 0, una vez hecho esto, se realizó en cada articulación un movimiento de un grado para así tener como resultado cuantos pulsos cuenta el programa y así obtener dicha relación de pulsos por cada grado. Para cerciorarnos de que los valores obtenidos sean correctos, se realizaron varias pruebas del mismo movimiento y cada valor fue el igual a la primera prueba.

Movimiento	Relación pulsos x grado
Base (motor 1)	141.463
Hombro (motor 2)	113.899
Codo (motor 3)	113.571

Tabla 3.2 Relación pulsos x grado de los primeros tres movimientos.

Ya que el movimiento de elevación y giro de la muñeca es realizado por los motores 4 y 5, el encoder del motor 4 será utilizado para conocer los grados del movimiento de elevación y el encoder del motor 5 para el movimiento de giro de la muñeca.

Movimiento de muñeca	Relación pulsos x grado
Elevación (motor 4)	28.87
Giro (motor 5)	28.75

Tabla 3.3 Relación pulsos x grado de los movimientos de la muñeca.

3.1.2 Control PID

El Control Proporcional-Integral-Derivativo (PID) es el algoritmo de control más común en la industria y ha sido universalmente aceptado en la industria de control. El controlador calcula un valor de error como la diferencia entre una señal medida y una señal de mando mejor conocido como *SetPoint* o valor de punto de ajuste. El controlador consiste en tres términos:

- *Término Proporcional* produce un valor de salida que es directamente proporcional al valor actual del error. El valor puede ser ajustado multiplicado el error con una constante k_p llamada la ganancia proporcional.
- *Término Integral* es la suma del error instantáneo en el tiempo y le da el valor acumulado que deberían haberse corregido previamente. El error acumulado se multiplica entonces por la ganancia integral k_i y se añade a la salida del controlador.
- *Término Derivativo* retrasa la velocidad de cambio de la salida del controlador. El control derivativo se usa para reducir el sobrepaso producido por el componente integral y mejorar la estabilidad del controlador-proceso combinado.

La salida del controlador PID es el resultado de la suma de los términos proporcional, integral y derivativo que poder ser expresado como:

$$u^n = k_p e^n + k_i \sum_{i=0}^n e_i \Delta t + k_d \left(\frac{e^n - e_{n-1}}{\Delta t} \right) \quad (56)$$

Donde e_n es el error entre el valor del *setpoint* y el valor medido, u_n es la salida del controlador, n es el tiempo discreto y ΔT es el tiempo de muestreo. En el caso de esta tesis la salida del controlador será una salida de PWM.

Es muy importante recalcar, que este tipo de controladores no requiere necesariamente un modelo matemático del proceso, ya que los modelos de sintonización se aplican a través de los ensayos, guiados solamente por la respuesta dinámica observada en el sistema de control. Por lo tanto, en la presente tesis el método de sintonización usado fue el experimental apoyado en la respuesta de cada articulación y en la siguiente tabla.

Término	Efecto
Proporcional ($\frac{1}{k_p}$)	Decremento en el tiempo de levantamiento, incremento en el sobrepaso y $e_{n=0}$ constante.
Integral ($\frac{1}{k_i}$)	$e_{n=0}$, incremento del sobrepaso, incremento del tiempo de asentamiento.
Derivativo ($\frac{1}{k_d}$)	$e_{n=0}$ poco cambio, decremento del sobrepaso y tiempo de asentamiento.

Tabla 3.4 Efectos de los términos del PID en el valor de la salida. [12][Castañeda]

La programación del control PID en el FPGA, se realizó mediante el apoyo del ejemplo llamado “Using Discrete PID” que contiene LabVIEW® al instalar el modulo “Módulo LabVIEW FPGA”. La programación está dividida en cuatro partes:

- Interrupción: Se establece una interrupción la cual condiciona al programa a seguir con el control y da el inicio de un nuevo ciclo de control.

- Entrada: se condiciona la variable del proceso para el control, dependiendo del valor del control “simulate?”, si es *TRUE* se envía el valor la cuenta del encoder de la articulación y si es *FALSE* se envía el valor del control “input 0”.
- Controlador: se realiza el control con la variable de proceso establecida en la etapa de entrada y el valor del control “setpoint”. Aquí también se definen las constantes del control (k_p , k_i y k_d) y rango de salida en este caso del ciclo de trabajo PWM.
- Salida: Se envía el valor del ciclo de trabajo para el programa que generara el PWM el cual se le aplicara al motor en cuestión.

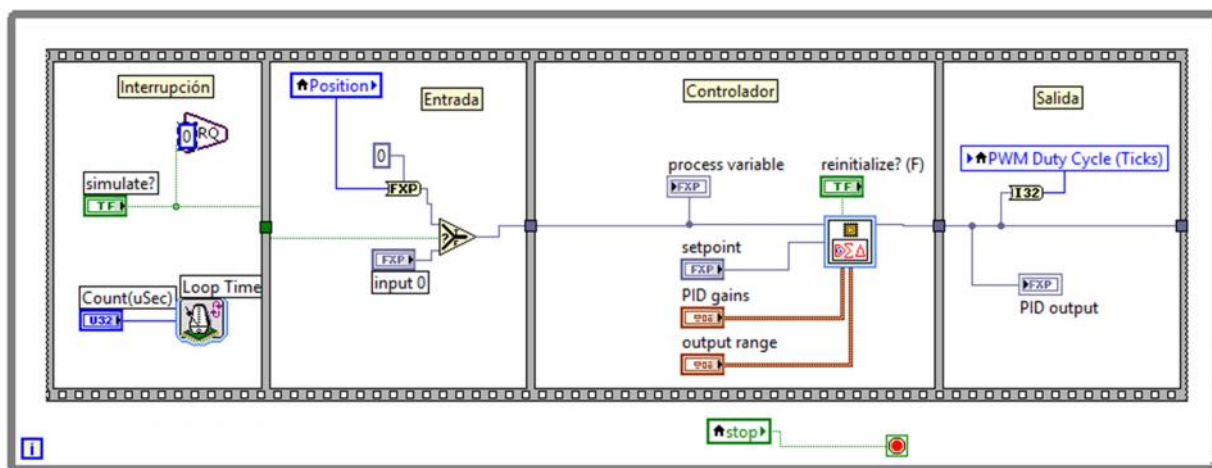


Figura 3.7 Etapa de Control PID.

3.1.3 Generación de PWM

Para la generación de PWM se utilizó la programación de [10][Romero]. Lo único que se le modificó, fue una condición la cual nos permite decidir entre el uso del PWM obtenido de la etapa del control PID o el valor de un control numérico “PWM1”, el cual será asignado por el usuario.

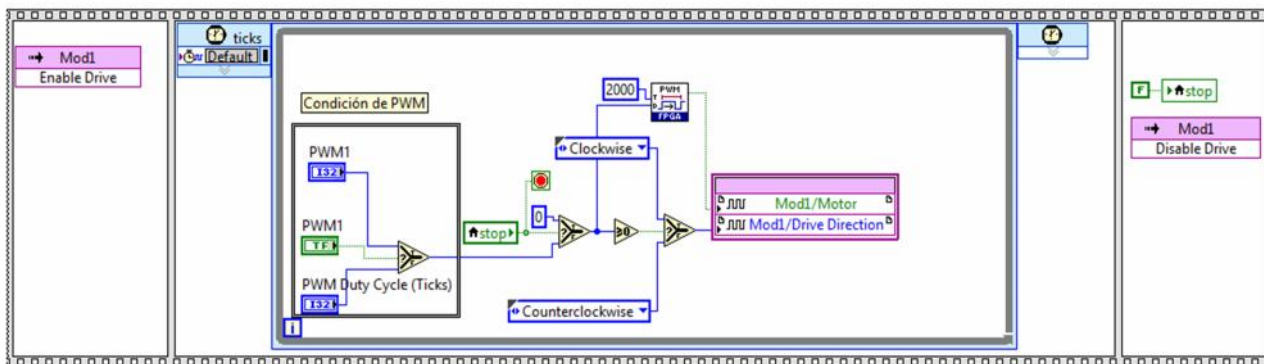


Figura 3.8 Generación de PWM.

3.1.4 Lectura de interruptores

Para la lectura de los sensores, se colocó un indicador a cada sensor el cual si esta encendido indica que la articulación no se encuentra en la posición de *HOME*.

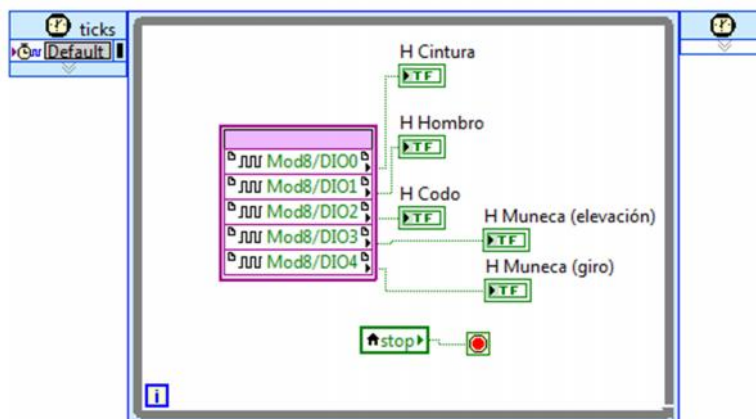


Figura 3.9 Lectura de interruptores.

Los casos de programación explicados con anterioridad, fueron aplicados a cada módulo de E/S implicado en el posicionamiento en el robot. Para el caso del *gripper*, simplemente se utilizara el código de la generación del PWM ya que para su abertura y cierre se le programo una rutina la es un control por tiempo debido a la tarea que realizara el robot.

3.2 Programación del controlador *Real Time*

En esta sección se elaboró un programa el cual es el encargado de realizar las siguientes funciones:

- Rutina de Posicionamiento de *HOME*
- Cinemática Inversa
- Perfil de Trayectoria
- Manipulación de variables del control PID

La programación en esta sección es fundamental y de gran importancia, sin menos preciar a la programación en el chasis FPGA y en la PC, ya que en esta parte se realiza el posicionamiento de *HOME* el cual es el punto de partida para el manejo de ángulos de cada articulación, la cinemática inversa la cual influye en las variables del control, la sincronización entre el Controlador *Real Time* y el chasis FPGA para una correcta ejecución de programas y el acceso a las variables del FPGA.

La programación se realizó en modo de secuencia:

1. Iniciar referencia al *Real Time* con el programa FPGA.
2. Posicionamiento de *HOME*.

3. Inicializar parámetros del robot para la Cinemática Inversa.
4. Cinemática Inversa con perfil de trayectoria.
5. Envío de las variables del Control PID.
6. Cerrar referencia hacia el programa FPGA.

Al ejecutar el programa, los puntos 1, 2, 3 y 6 se realizarán una sola vez, los puntos 4 y 5 se ejecutarán cíclicamente ya que sus parámetros serán modificados por el usuario.

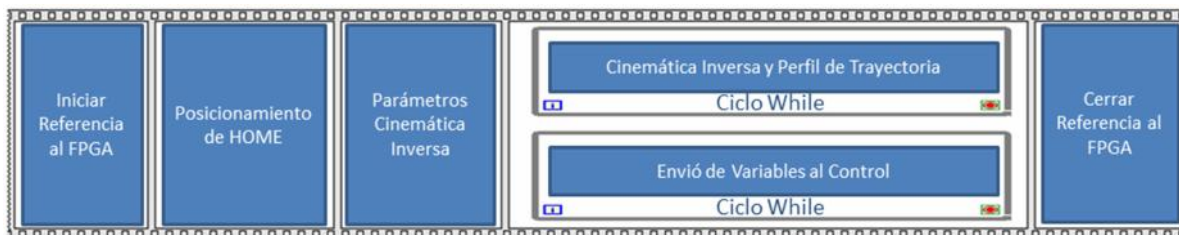


Figura 3.10 Secuencia de Funciones

Para poder hacer uso de las variables del FPGA en el *Real Time*, se requiere una función para poder escribir en los controles y leer los indicadores, por lo tanto se utilizó la función “*Read/Write Control (FPGA Interface)*”. Mientras que para el intercambio de datos entre el *Real Time* y la PC, se utilizarán las variables globales las cuales serán del tipo de dato doble y booleano.

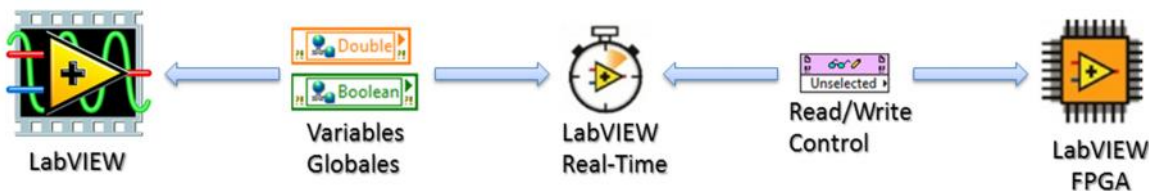


Figura 3.11 Medios de Transferencia de datos

3.2.1 Posicionamiento de HOME

Se programaron dos métodos diferentes de posicionar al robot en *HOME*; el primero, por medio de una rutina de movimientos la cual se realizó en el controlador Real Time para no tener pérdida de información ya que es el más importante debido a que al inicio del programa la posición del robot es desconocida, y el segundo a partir de coordenadas cartesianas.

Las funciones principales de la rutina de movimientos, son la búsqueda de los interruptores por medio de la asignación de un valor a los PWM de cada motor y resetear la cuenta de la posición de los motores para que esta tenga el valor de 0. Debido a que el controlador en tiempo real no trabaja en forma paralela como el FPGA, la búsqueda del *HOME* de cada motor se realiza un motor a la vez.

Variable	Valor Numérico
Coordenadas Cartesianas	
Pos. X	16.5459 [cm]
Pos. Y	0 [cm]
Pos. Z	50.169 [cm]
Orientación	
Rot. X	0 [°]
Rot. Y	26.43 [°]
Rot. Z	-180 [°]

Tabla 3.5 Coordenadas cartesianas y grados de orientación en la posición de Home

3.2.2 Cinemática inversa

Para la cinemática inversa se implementaron dos métodos diferentes para su resolución, el cual el primero se resuelve por medio de matrices homogéneas (se explica en el capítulo 2) y el segundo se utiliza un *subVI* proporcionado por el módulo de “Robotics”.

3.2.2.1 Matrices homogéneas

En este método se ingresaron las ecuaciones finales de cada ángulo por medio de la función “Formula Node” de LabVIEW®, el cual su programación es similar a la del lenguaje C.

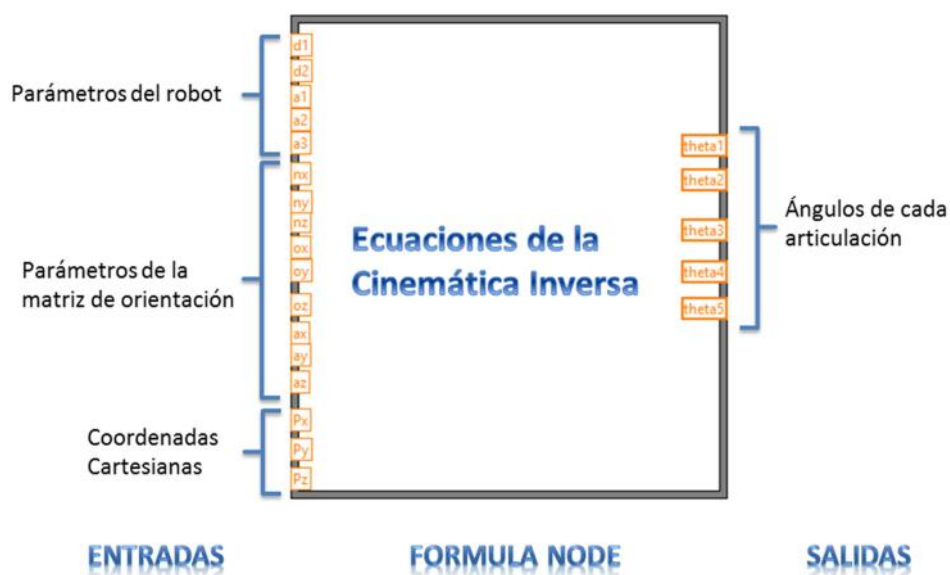


Figura 3.12 Programación de la CI por medio de Formula Node.

3.2.2.2 Módulo de LabVIEW Robotics

El módulo de *Robotics* nos proporciona la solución de la cinemática inversa de diferentes configuraciones de manipuladores seriales, en los cuales incluye la de un manipulador “5R

Type 1” que es la misma configuración al robot SCORBOT-ER 4u con cinco grados de libertad de los cuales dos son realizados en el efector final.

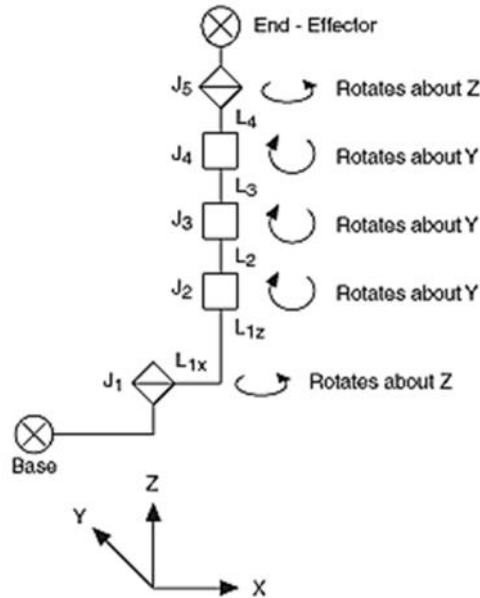


Figura 3.13 Configuración 5R Type 1. [E19]

Al hacer la comparación de los parámetros y movimientos de las articulaciones de cada configuración, se obtienen las siguientes similitudes:

SCORBOT-ER 4u	5R Type 1
Parámetros	
	L_{1z}
	L_4
	L_{1x}
	L_2
	L_3
Movimientos	
Cintura	J_1 - Rotates about Z
Hombro	J_2 - Rotates about Y
Codo	J_3 - Rotates about Y
Muñeca (elevación)	J_4 - Rotates about Y
Muñeca (giro)	J_5 - Rotates about Z

Tabla 3.6 Similitudes entre configuraciones.

Para realizar la implementación, primeramente se tiene que definir que configuración se va a utilizar por medio del subVI “Initialize Serial Arm”, el cual es un polimorfismo y se definen los valores de los parámetros del robot. Para la resolución de la cinemática inversa se utiliza el subVI “Analytical Inverse Kinematics”, el cual al ser conectado a “Initialize Serial Arm” automáticamente se configura para dicha solución. Este subVI contiene una

entrada la cual es la matriz de transformación con los valores de la orientación y la posición asignados, y dos indicadores, el primero es un arreglo que nos indica la posición de las articulaciones en radianes y el segundo es un booleano que nos indica si las coordenadas cartesianas ingresadas y la orientación definida se encuentran en el espacio de trabajo del robot.

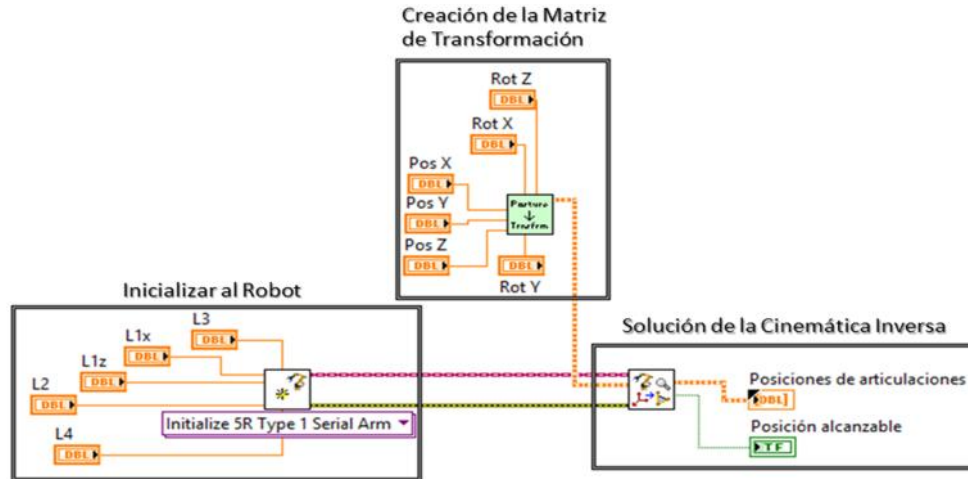


Figura 3.14 Programación de la CI por medio del Módulo de Robotics.

3.2.3 Perfil de trayectoria

En cuanto al movimiento del robot, se pretende un desplazamiento restringido en el cual al inicio y final de una trayectoria, el robot se desplace a bajas velocidades. Por lo tanto, se utilizó trayectorias parametrizadas por un polinomio de quinto orden, con coeficientes constantes, que cumplan que al inicio y final de la operación la velocidad sea cero, y que la aceleración este definida en cualquier punto de la trayectoria.

La trayectoria que seguirá el robot será una línea recta en el espacio, las ecuaciones que describen dicho movimiento son:

$$\text{El robot se mueva a una línea recta en} \quad (3.1)$$

$$x = x_1 + (x_2 - x_1) \left(\frac{t}{t_f} \right)^3 - 15 \left(\frac{t}{t_f} \right)^4 + 6 \left(\frac{t}{t_f} \right)^5 \quad (3.2)$$

$$y = y_1 + (y_2 - y_1) \left(\frac{t}{t_f} \right)^3 - 15 \left(\frac{t}{t_f} \right)^4 + 6 \left(\frac{t}{t_f} \right)^5 \quad (3.3)$$

Dónde:

Punto inicial $\rightarrow P1(x_1, y_1, z_1)$.

Punto final $\rightarrow P2(x_2, y_2, z_2)$.

Polinomio de quinto orden $\rightarrow P(t) = 10 \left(\frac{t}{t_f} \right)^3 - 15 \left(\frac{t}{t_f} \right)^4 + 6 \left(\frac{t}{t_f} \right)^5$

Tiempo final $\rightarrow t_f$

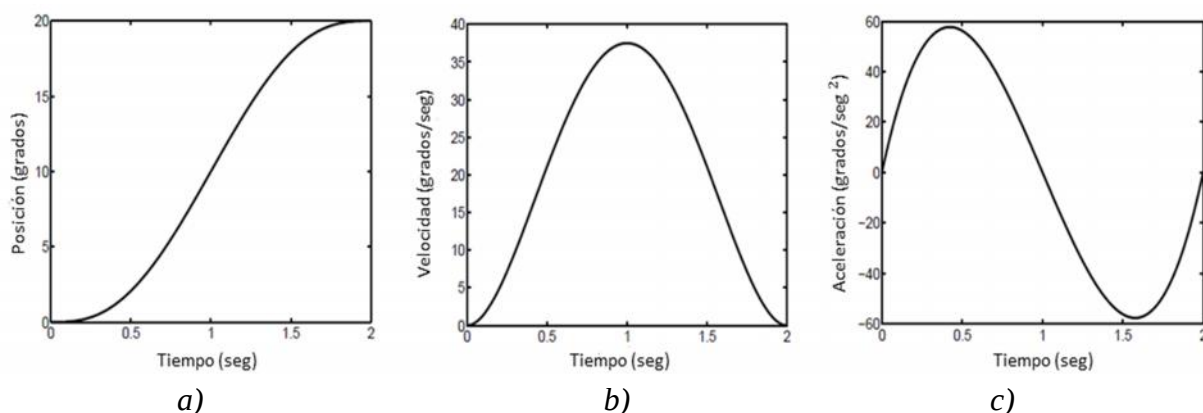


Figura 3.15 Graficas del perfil quintico: a) posición, b) velocidad y c) aceleración.

3.2.4 Manipulación de variables del control PID

Como ya se explicó en el punto 3.1.2, todo el proceso de control se realiza en el FPGA, lo único que se programó en este apartado fue la manipulación de las variables:

- *Setpoint*: el valor es asignado por la cinemática inversa automáticamente.
- Constantes del control (k_p , k_i y k_d) son modificadas dependiendo de la respuesta deseada y los parámetros de diseño.

3.3 Programación en la PC

Para la finalización del control total del robot, el único proceso que nos hace falta es ingresar las coordenadas cartesianas y la orientación del efector final por el usuario, por lo cual dicha programación se llevó a cabo en esta sección.

Debido a que es muy importante la comunicación entre el usuario y el proceso a realizar, se optó por la realización de una interfaz HMI la cual ayudara al usuario principalmente a controlar y monitorear las variables principales del robot. Para el desarrollo de la interfaz, se planteó que el usuario pueda realizar las siguientes funciones:

- Ingreso de las coordenadas cartesianas y la orientación deseada.
- Monitorear las variables del proceso.
- Realizar una secuencia de movimientos.
- Visualizar una simulación 3D del robot SCORBOT-ER 4u.

Por lo tanto, se creó una interfaz principal, en la cual se tiene acceso directo al robot físico y dos interfaces emergentes (secundarias), para la visualización 3D y monitoreo de variables. Al ejecutar la interfaz principal, el usuario podrá elegir la opción de trabajo ya sea con el robot físico o la visualización 3D del robot. La interfaz monitoreo de variables, se tendrá acceso únicamente cuando se esté trabajando con el robot físico, esta interfaz trabajara en paralelo con la interfaz principal.

3.3.1 Interfaz principal

La interfaz principal tiene dos tareas asignadas, las cuales son:

- Realizar una secuencia de movimientos: Esta tarea está programada para que el robot realice como máximo cinco (5) movimientos, el primer movimiento se realizará desde la posición actual del robot hasta la posición escogida por el usuario. Cada movimiento se efectuará por medio de un perfil de velocidad.
- Guardar posiciones del robot: Para esta apartado, el usuario podrá mover por medio de controles la posición del robot y también tendrá acceso a la orientación, una vez conseguido el punto deseado, podrá guardar dicha posición y así incluirla posteriormente en una secuencia de movimientos. En cuanto a las posiciones guardadas, no existe un límite alguno pero solamente se utilizarán las primeras cinco (5) posiciones en la secuencia de movimientos y cada que se ingrese a esta tarea, el contador de posiciones se reiniciará a la posición uno (1).

El diagrama funcional de esta interfaz es el siguiente:

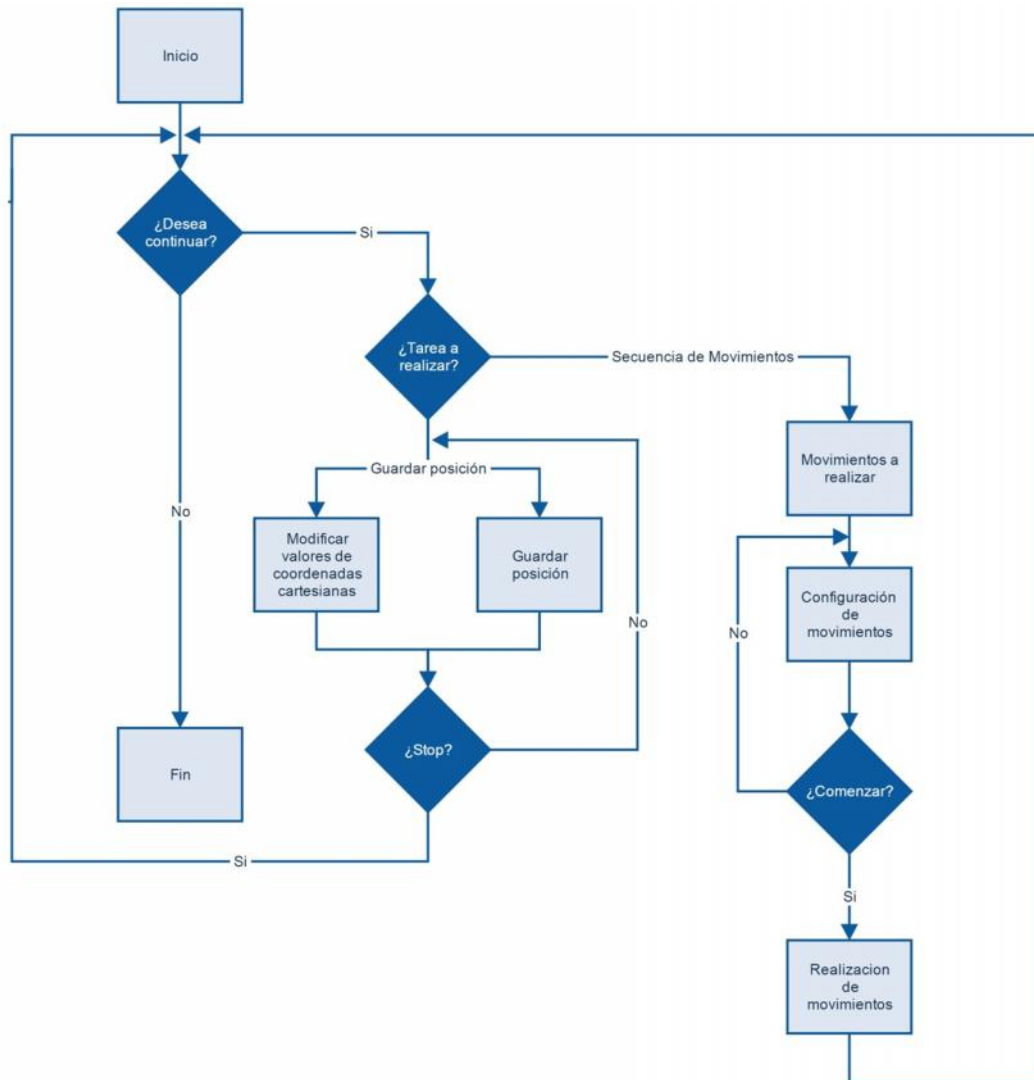


Figura 3.16 Diagrama de trabajo de la interfaz principal.

Para la creación de esta interfaz se utilizaron funciones básicas de LabVIEW™:

Controles Numéricos: Se utilizaron para que el usuario ingrese los valores de las coordenadas cartesianas y los grados de rotación para definir la orientación del efector final, a su vez estos valores son enviados por medio de variables globales a la cinemática inversa programada en el Real Time. Los límites de cada control se definieron con la ayuda del espacio del trabajo del robot.

Control de Cadenas: En este control el usuario ingresa el valor del tiempo final del perfil de velocidad.

Controles Booleanos: Se tienen seis (6) controles disponibles para el usuario dependiendo de la etapa del proceso en que se encuentre.

- Abrir y Cerrar *gripper*: Estos dos controles al seleccionarlos dan inicio a una rutina para la apertura o el cierre del *gripper*. La rutina se realiza por un control por tiempos, la cual se le asigna un PWM predeterminado al motor durante un tiempo predeterminado, la única diferencia entre cada rutina es el signo del valor de PWM.
- Guardar Posiciones: Este control hace el registro de los valores de las coordenadas, grados de la orientación y el tiempo final del perfil de velocidad por medio de una rutina condición.
- Comenzar: Da inicio a la secuencia de movimientos.
- Stop: Se utiliza para finalizar la tarea de guardar posiciones.
- Home: Al seleccionar este control, automáticamente el robot regresara a la posición de home debido a que a los controles numéricos se les asignan los valores de la tabla 3.5.

Controles Enumerados: Se tienen cinco (5) controles, los cuales aparecerán dependiendo de los movimientos que el usuario desee realizar. En estos controles se podrá elegir la posición final (previamente guardadas) de cada movimiento.

Indicador Matricial: Se indican en forma de lista las posiciones guardas con sus respectivos valores de coordenadas y el tiempo final.

Indicador Booleano: Se cuenta con un (1) indicador.

- Alcanzable: Muestra si las coordenadas cartesianas y orientación seleccionadas están dentro del espacio de trabajo del robot.

Cabe mencionar que algunos controles podrán ser usados en unas tareas y en otras no.

		Secuencia de Movimientos	Guardar Posiciones
Controles Numéricos	Pos. X		•
	Pos. Y		•
	Pos. Z		•
	Rot. Y		•
	Rot. Z		•
Controles Booleanos	Abrir/Cerrar	•	•
	Guardar Posición		•
	Comenzar	•	
	Stop	•	•
	Home		•

Controles Enumerados	Movimiento 1	•	
	Movimiento 2	•	
	Movimiento 3	•	
	Movimiento 4	•	
	Movimiento 5	•	
	Indicador Matricial	•	•
Indicador Booleano	Alcanzable		•

Tabla 3.7 Disponibilidad de Controles e Indicadores.

Una vez ejecutado el programa la interfaz se muestra en la siguientes dos figuras.



Figura 3.17 Interfaz mientras se ejecuta una secuencia de movimientos.



Figura 3.18 Interfaz mientras se guardan posiciones.

3.3.2 Interfaz simulación 3D

Esta interfaz es similar a la interfaz principal con el mismo diagrama de funcionamiento pero la manipulación de las variables se le aplicara a la simulación. La simulación se realizó gracias al módulo de *Robotics*, por lo tanto no se cuenta con un *gripper* que se pueda abrir o cerrar, además, se optó por no agregar en esta interfaz el control para manipular la rotación en Z ya que no se alcanza a apreciar el movimiento debido al tamaño establecido de la simulación.

Al seleccionar la opción de no continuar, el ciclo terminara y se cerrara automáticamente la interfaz actual para regresar a la interfaz principal.

Para un correcto funcionamiento tanto de la interfaz simulación 3D y la interfaz principal, se agregaron mensajes emergentes para el usuario, los cuales servirán de guía para un correcto funcionamiento.

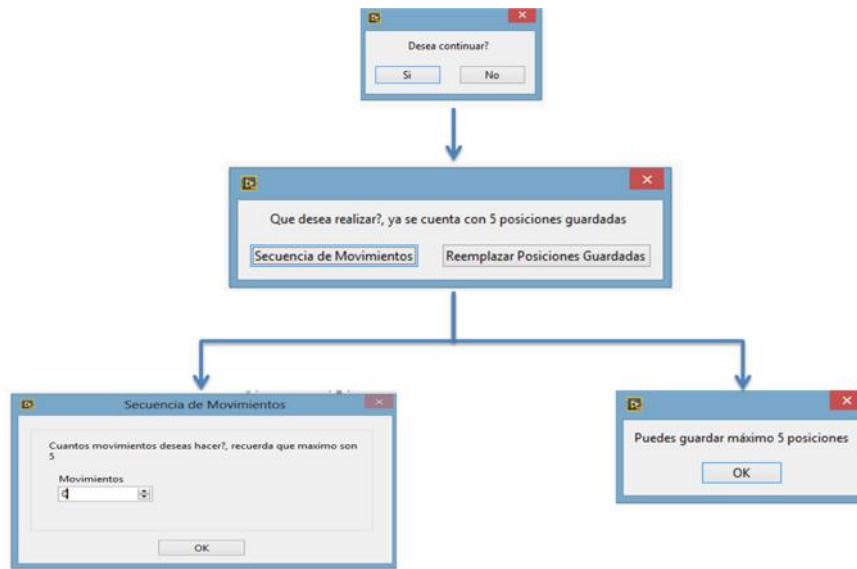


Figura 3.19 Secuencia de mensajes emergentes.

El funcionamiento de esta interfaz se mostrara en el siguiente capítulo.

3.3.3 Interfaz monitoreo de variables

Esta interfaz, se creó exclusivamente para que el usuario monitoree el comportamiento del control PID con lo cual se podrá visualizar lo siguiente:

- Gráficas de la respuesta del PID aplicado a cada motor.
- Los valores del PWM aplicados.

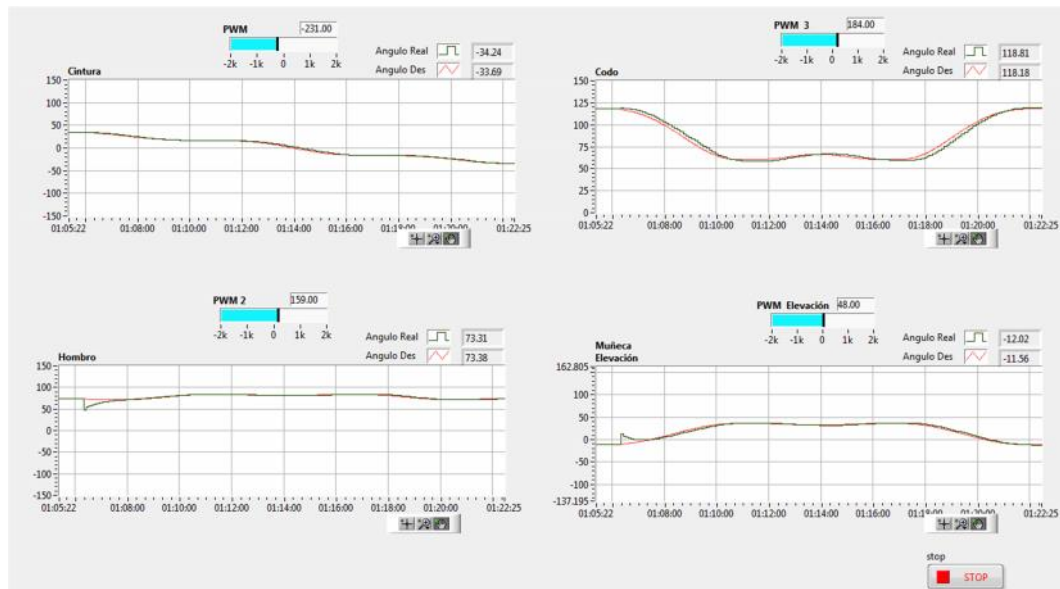


Figura 3.20 Interfaz monitoreo de variables en funcionamiento.

Finalmente, la arquitectura del árbol de este proyecto se muestra en la siguiente figura.

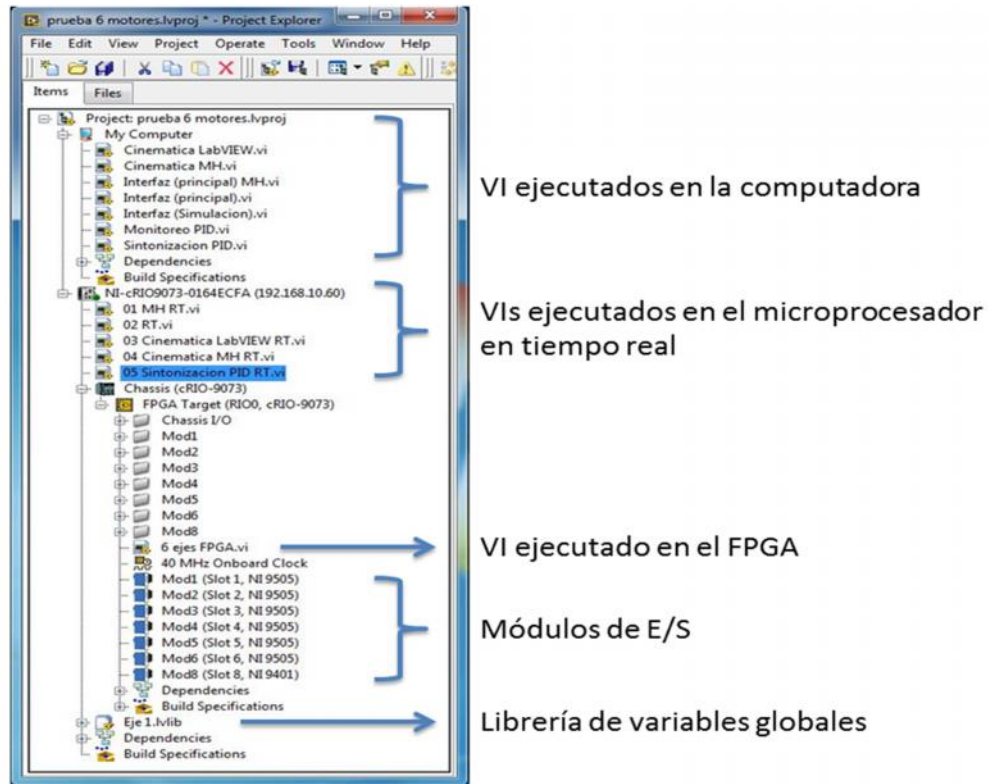


Figura 3.18 Estructura del proyecto.

4 Pruebas y Resultados

4.1 Cinemática inversa

En el capítulo anterior, se mencionó que se tienen dos métodos para la resolución de la cinemática inversa: matrices homogéneas y por medio de un *subVI* incluido en el módulo de *Robotics*. Debido a esto, para comprobar que la configuración programada (usando el módulo de LabVIEW®) puede ser aplicada al robot SCORBOT-ER 4u, se realizó una comparación entre ambos métodos por medio de una secuencia de cinco (5) movimientos, utilizando un perfil de trayectoria entre cada movimiento.

Los puntos seleccionados en el espacio de trabajo del robot son los siguientes:

Posición	X[cm]	Y[cm]	Z [cm]	Tiempo Final [ms]
Posición 1	17.339	0	37	5000
Posición 2	15	0	30	5000
Posición 3	15	0	10	5000
Posición 4	40	0	10	5000
Posición 5	40	0	30	5000

Tabla 4.1 Posiciones guardadas.

Para el inicio de la secuencia, se seleccionó a la posición 1 como posición inicial de la secuencia y los movimientos se definieron para que el robot formara un rectángulo en el plano XZ, por lo tanto, la secuencia fue la siguiente:

Movimiento	Posición Final
Movimiento 1	Posición 2
Movimiento 2	Posición 3
Movimiento 3	Posición 4
Movimiento 4	Posición 5
Movimiento 5	Posición 2

Tabla 4.2 Secuencia de movimientos.

Ya que en el método de LabVIEW® se puede manipular la orientación del efector final y eso influye en el cálculo de la cinemática inversa, se asignaron los valores correspondientes a los controles para tener la misma orientación que el método resuelto por matrices homogéneas.

Los valores de ángulo de cada articulación que resulta de ingresar en cada método las coordenadas cartesianas ya definidas son:

Ángulo	Posición 1		Posición 2		Posición 3	
	MH	LabVIEW®	MH	LabVIEW®	MH	LabVIEW®
0	0	0	0	0	0	0
0.1	-14.337	-14.3773	-12.576	-12.5763	59.9066	59.9066
0.2	117.141	117.141	135.504	135.504	134.192	134.192
0.3	77.236	77.2361	57.073	57.0727	-14.0984	-14.0984
0.4	0	0	0	0	0	0
0.5	0	0	0	0	0	0

Tabla 4.3 Comparación de las primeras tres posiciones definidas.

Ángulo	Posición 4		Posición 5	
	MH	LabVIEW®	MH	LabVIEW®
0	0	0	0	0
0.1	79.7907	79.7907	50.2905	50.2905
0.2	50.4284	50.4284	51.6247	51.6247
0.3	49.781	49.7809	78.085	78.0848
0.4	0	0	0	0
0.5	0	0	0	0

Tabla 4.4 Comparación de la posición 4 y 5.

Debido a que las únicas variaciones que se observan están en diezmilésimas y en el presente trabajo no es objetivo primordial tener una gran precisión, se comprueba que se puede aplicar este método de solución al robot SCORBOT-ER 4u, por lo tanto, las siguientes pruebas se realizarán aplicando el método incluido en el *subVI* del módulo de *Robotics*, solo por así convenir a la programación.

4.2 Simulación 3D

Para comprobar el funcionamiento completo de la simulación 3D, se verificó lo siguiente:

1. Correcto almacenamiento de posiciones y movimiento de la simulación, ver figura 4.2.
2. Realización de una secuencia de cinco (5) movimientos, se inició de un punto de partida cualquiera y se prosiguió con la siguiente secuencia: posición 1, posición 2, posición 3, posición 4 y terminando con la posición 5.

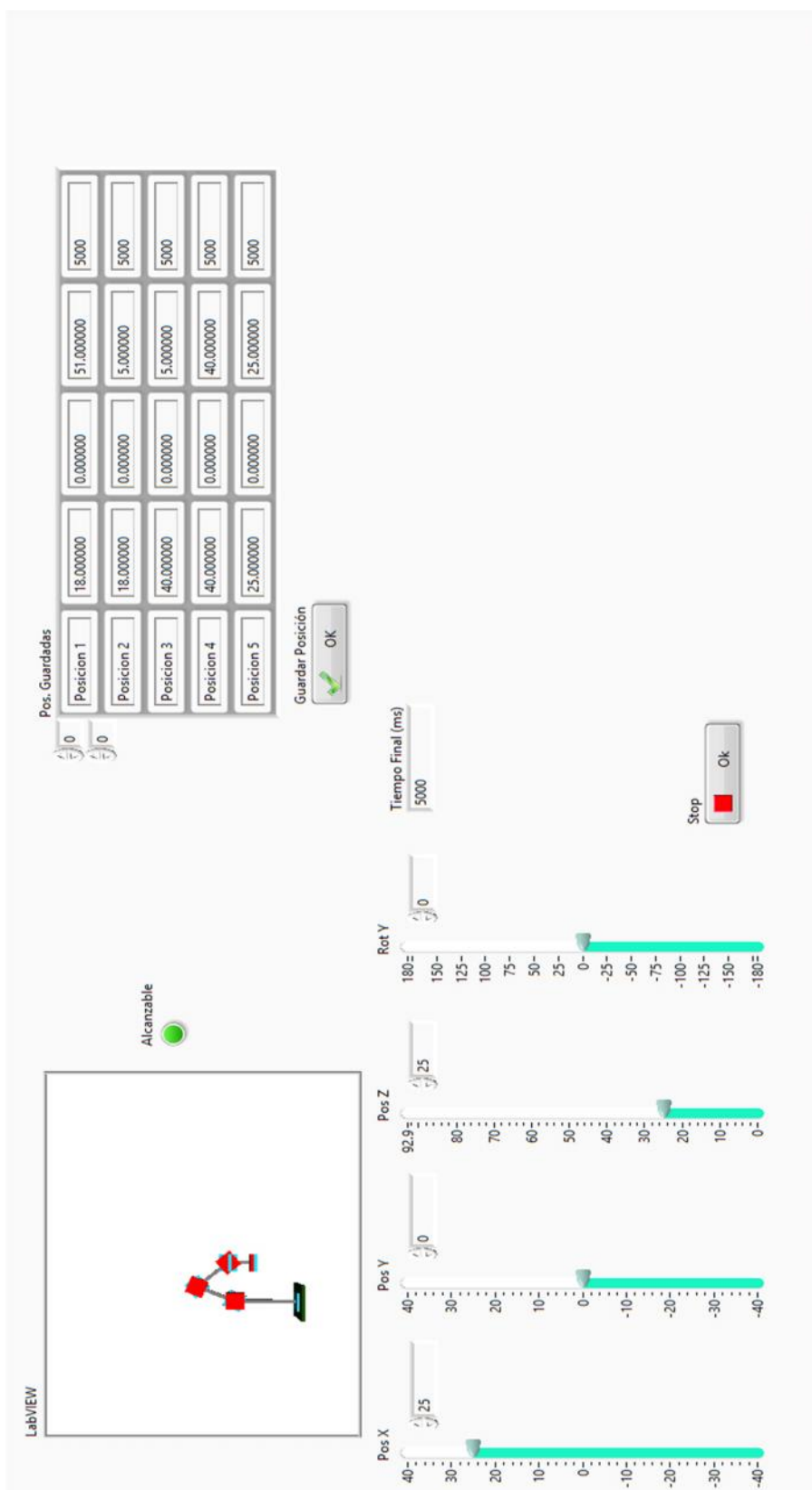


Figura 4.2 Almacenamiento de posiciones.

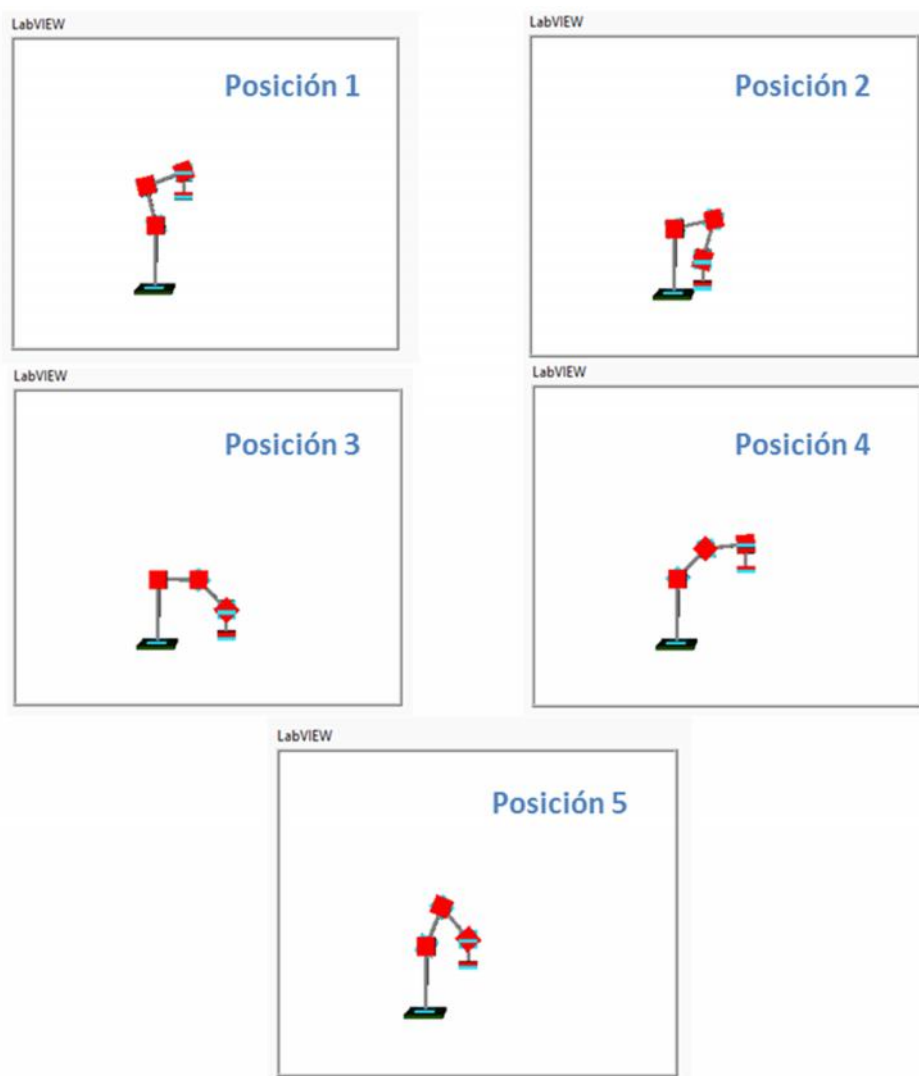


Figura 4.3 Secuencia movimientos en simulación 3D.

4.3 Funcionamiento del SCORBOT-ER 4u.

Antes de realizar algún movimiento con el robot se realizó la obtención de las constantes (k_p , k_i y k_d) del control PID, dicha sintonización, se realizó por medio de múltiples ensayos fijando los siguientes constantes para cada motor:

Constante	Valor			
	Cintura	Hombro	Codo	Muñeca (elevación)
k_p	4.45	4.1	4.4	4.9
k_i	0.0095	0.01	0.01	0.01
k_d	0.001	0.001	0	0.0001

Tabla 4.5 Constantes de cada control PID.

Ya definidas las constantes, se le ingresó una función escalón con el mismo valor a cada movimiento del motor para poder visualizar y analizar su respuesta.

Cintura

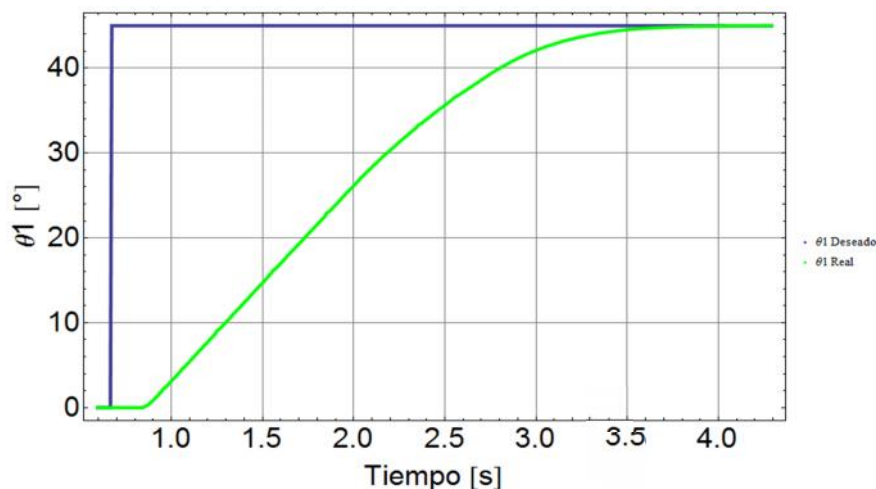


Figura 4.4 Respuesta de θ_1 .

Valor de función escalón	45°
Repuesta	Sobre amortiguada
Tiempo de asentamiento 5%	2.415 [s]
Zona muerta	0.163 [s]

Tabla 4.6 Análisis de respuesta de θ_1 .

Hombro

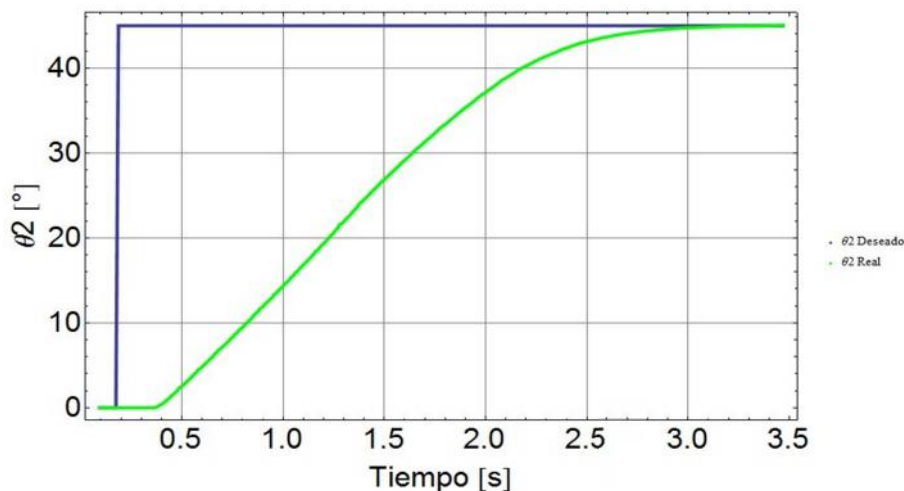


Figura 4.5 Respuesta de θ_2 .

Valor de función escalón	45°
Repuesta	Sobre amortiguada
Tiempo de asentamiento 5%	2.224 [s]

Zona muerta	0.172 [s]
-------------	-----------

Tabla 4.7 Análisis de respuesta de θ_2 .

Codo

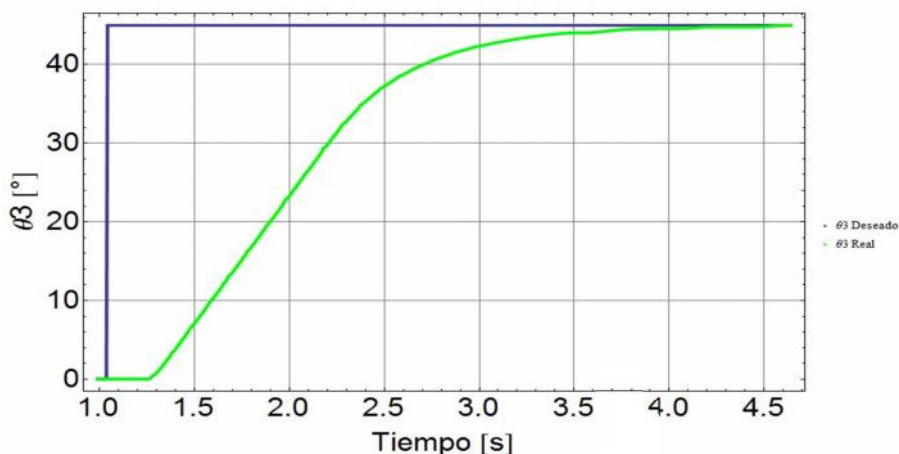


Figura 4.6 Respuesta de θ_3 .

Valor de función escalón	45°
Repuesta	Sobre amortiguada
Tiempo de asentamiento 5%	1.983 [s]
Zona muerta	0.210 [s]

Tabla 4.8 Análisis de respuesta de θ_3 .

Muñeca (elevación)

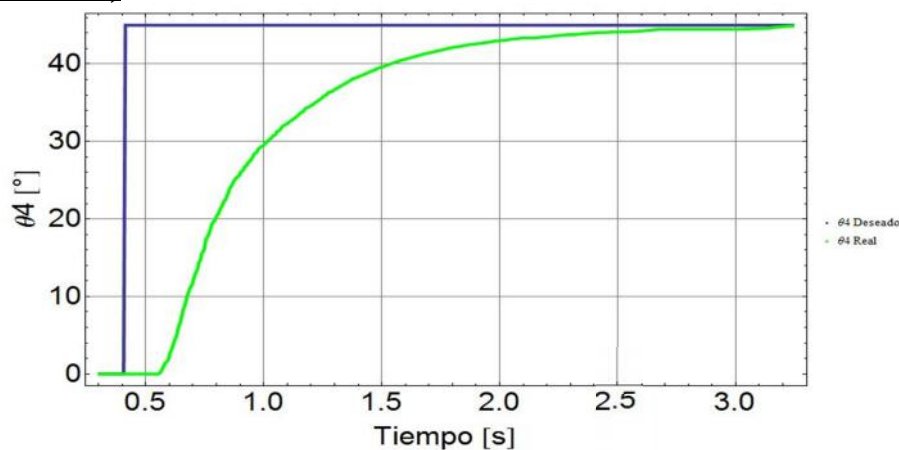


Figura 4.7 Respuesta de θ_4 .

Valor de función escalón	45°
Repuesta	Sobre amortiguada
Tiempo de asentamiento 5%	1.452 [s]
Zona muerta	0.139 [ms]

Tabla 4.9 Análisis de respuesta de θ_2 .

Al comparar cada una de las respuestas de cada control, se determina que influye mucho el peso de cada articulación en la respuesta de los movimientos (motores) ya que el tiempo de asentamiento disminuye debido a la posición de cada articulación, esto es, el movimiento de cadera ejecutado por el motor 1, tiene que cargar de todo el peso del robot mientras en el movimiento de elevación de muñeca ejecutado por los motores 4 y 5, solo cargan el peso del efector final.

El siguiente punto, es comprobar el funcionamiento del robot físico ya en conjunto, por lo tanto, se ejecutaron tres secuencias de movimientos, en las cuales, el robot dibujo una figura geométrica; cada secuencia se realizó en diferente plano (plano XY, plano YZ y plano XZ) y así se verificó lo siguiente:

1. Cinemática Inversa por medio del posicionamiento del robot en el espacio.
2. Cumplimiento del perfil de trayectoria en el tiempo establecido.

Así mismo, las secuencias de movimientos también se realizaron con el controlador de fábrica “Controlador-USB” para monitorear su funcionamiento y desempeño.

Cabe recalcar, que para el inicio de las tres secuencias, se escogió a la posición de Home como punto de partida y en los tres casos, la secuencia de movimientos es la siguiente:

Movimiento	Posición Final
Movimiento 1	Posición 1
Movimiento 2	Posición 2
Movimiento 3	Posición 3
Movimiento 4	Posición 4
Movimiento 5	Posición 1

Tabla 4.10 Secuencia de movimientos para las pruebas físicas.

Secuencia de Movimiento 1 (Cuadrado en el plano XY)

Posición	X[cm]	Y[cm]	Z [cm]	Tiempo Final [ms]
Posición 1	15	-10	5	5000
Posición 2	35	-10	5	5000
Posición 3	35	10	5	5000
Posición 4	15	10	5	5000
Posición 5	17.339	0	51.5506	5000

Tabla 4.11 Posiciones guardadas de la secuencia 1.

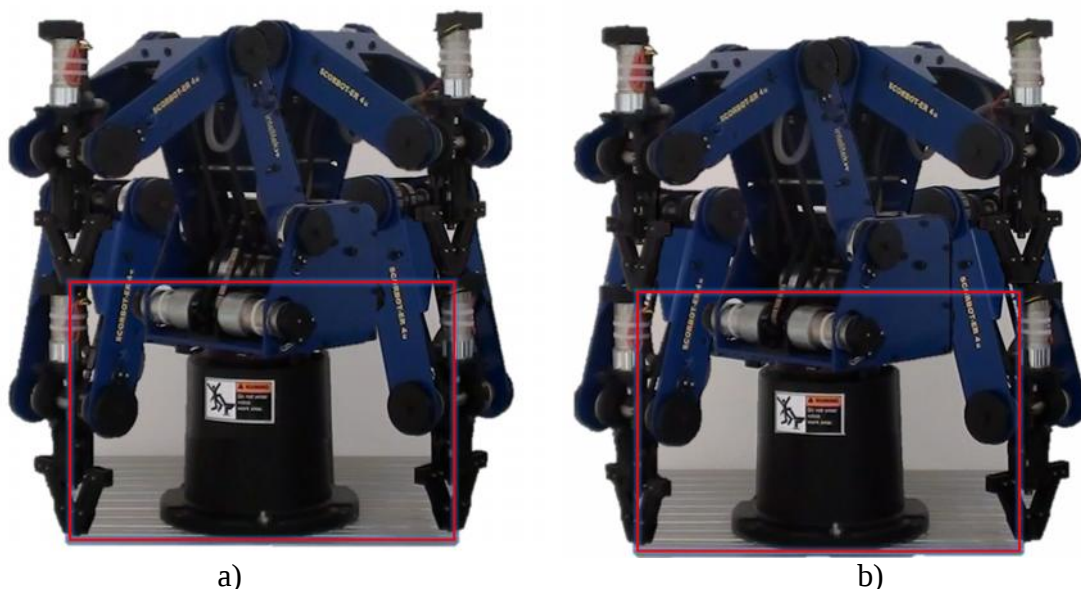


Figura 4.8 Secuencia realizada por: a) Controlador-USB b) CompacRIO®.

Secuencia de Movimiento 2 (Rectángulo en el plano YZ)

Posición	X[cm]	Y[cm]	Z [cm]	Tiempo Final [ms]
Posición 1	20	-20	30	5000
Posición 2	20	-20	3	5000
Posición 3	20	20	3	5000
Posición 4	20	20	30	5000
Posición 5	17.339	0	51.5506	5000

Tabla 4.12 Posiciones guardadas de la secuencia 2.

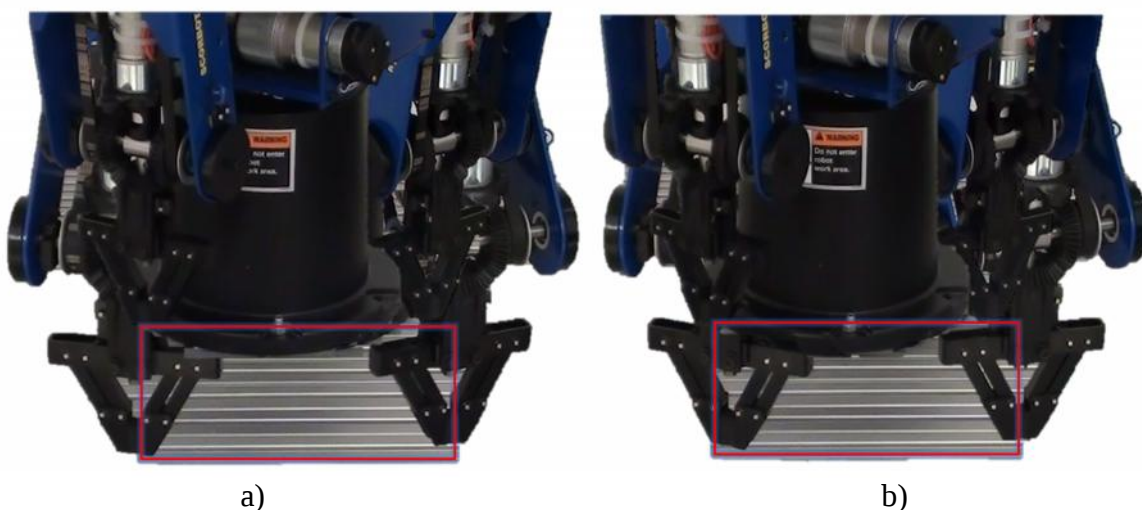


Figura 4.9 Secuencia realizada por: a) Controlador-USB b) CompacRIO®.

Secuencia de Movimiento 3 (Rombo en el plano XZ)

Posición	X[cm]	Y[cm]	Z [cm]	Tiempo Final [ms]
Posición 1	30	0	50	5000
Posición 2	20	0	30	5000
Posición 3	30	0	10	5000
Posición 4	40	0	30	5000
Posición 5	17.339	0	51.5506	5000

Tabla 4.13 Posiciones guardadas de la secuencia 3.

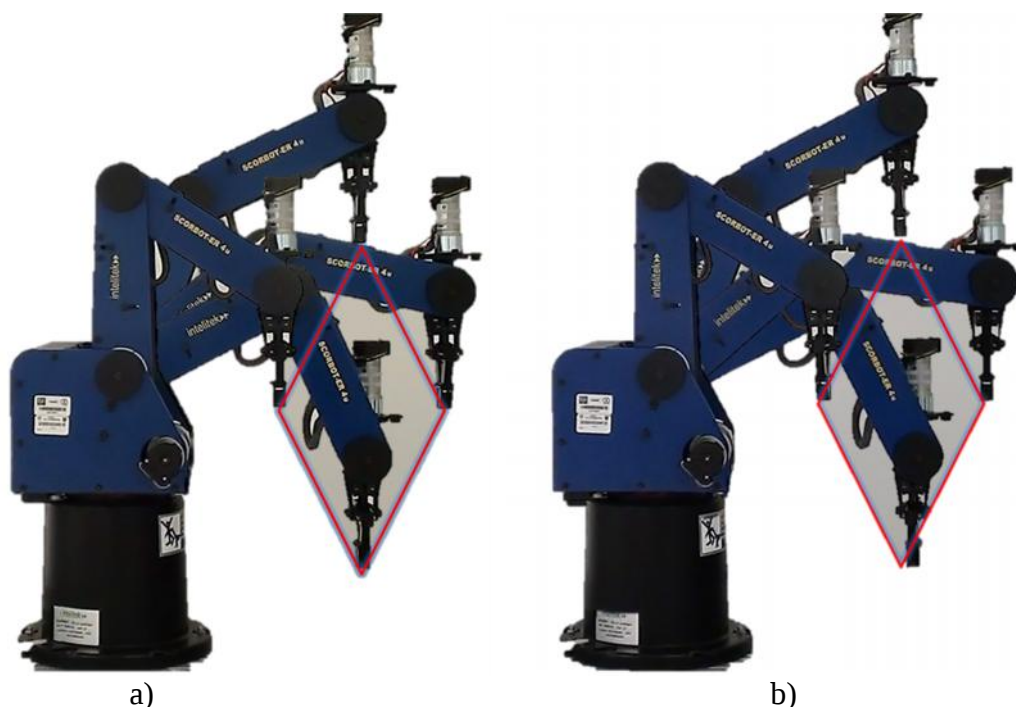


Figura 4.10 Secuencia realizada por: a) Controlador-USB b) CompacRIO®.

Al comparar las imágenes de cada controlador, se muestra un ligero desfase de puntos en el controlador propuesto, lo cual se debe a una acumulación de errores influyendo principalmente a la medición de cada articulación y el cálculo de pulsos de encoder por cada grado, ya que se realizó la estimación de esos manualmente.

En la última secuencia, se realizó el análisis de trayectoria solamente con el *CompacRIO®* debido a que en el controlador de fabrico no se cuenta con esa opción.

Perfil de trayectoria en X

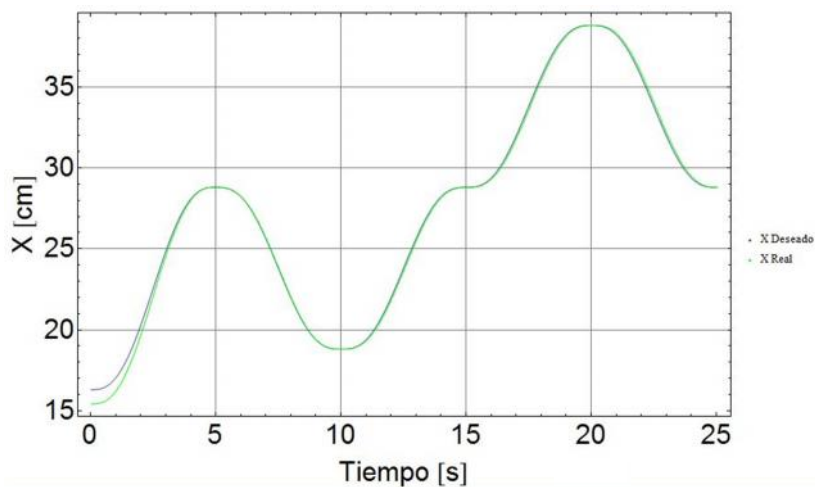
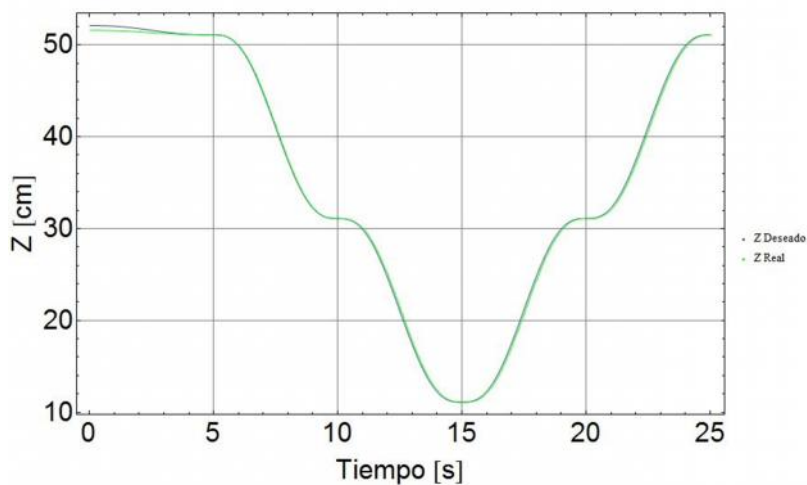


Figura 4.11 Perfil de trayectoria de X en el efector final.

Entre el punto deseado y el punto real se tiene un $3.1251 \times 10^{-5}\%$ de error absoluto.

Perfil de trayectoria en Z



Entre el punto deseado y el punto real se tiene un 0.06189% de error absoluto.

5 Conclusiones

El reemplazo de un controlador y de una plataforma de trabajo como tal es un caso difícil que lleva mucho tiempo de trabajo y de estudio. En este capítulo se evalúan los puntos tratados en el presente trabajo y se realizarán las comparaciones.

Como primer punto a tratar es el hacer funcionar al robot con la tecnología que nos proporciona el *CompacRIO*® a la par con sus módulos de E/S: un chasis reconfigurable FPGA y un controlador en Tiempo Real. Este objetivo se cumplió satisfactoriamente, ya que se logra el movimiento del robot por medio del dispositivo *NI 9505* que es el encargado de realizar la lectura del posicionamiento a partir de la lectura de encoders, y lograr el giro de cada motor por medio de la generación de un PWM.

La programación utilizada para el posicionamiento (encoders e interruptores), es completamente viable, ya que nunca existe pérdida de información debido a la velocidad de ejecución de una iteración del FPGA, que es de 1ns por ciclo, además de siempre conseguir el mismo valor en un movimiento determinado como se logra ver en las pruebas de perfil de trayectoria. Cabe resaltar que estos instrumentos de potencia son especializados para esta tarea, lo que hace más fácil su uso.

En cuanto a las técnicas de control, se implementó un control PID, en el cual sus constantes pueden ser modificadas o se tiene la opción de programar otra técnica de control (control por retro de estados, control óptimo, etcétera) en comparación con el controlador de fábrica.

La programación de la cinemática inversa se cumple. El desfase que existe entre cada posición como se explicó en el capítulo 4, es debido a la acumulación de errores de medición, pero los valores arrojados por la cinemática inversa son de alta confiabilidad. Una de las ventajas de utilizar el *subVI* que proporciona LabVIEW, es que no se cuentan con restricciones en la manipulación de la orientación del efector final y el método de matrices homogéneas, está limitado a resolverla con una orientación asignada y está limitada a funcionar solamente a no más de 45 centímetros en el eje Z, pero para este trabajo cumplen los dos métodos.

La implantación de una interfaz de simulación, fue de gran ayuda para realizar pruebas previas antes de implementarlas en el robot físico, aunque está limitada a realizar las tareas implementadas en este trabajo, comparándolo con la plataforma del robot nuestra interfaz cumple en cierta manera ya que logra realizar las funciones básicas del robot. En cuanto a la interfaz de monitores la propuesta iguala a la plataforma del robot, ya que se puede monitorear los PWM de cada motor y las gráficas del comportamiento de los motores.

En cuanto al procesamiento de instrucciones entre un controlador y otro, es mucho mejor el propuesto, ya que el *NI cRIO-9073* maneja un ciclo de reloj de 266 MHz, además de contar un chasis reconfigurable FPGA y el Controller-USB maneja un microcontrolador NEC V853 con un ciclo de 33 MHz.

En cuanto al precio entra cada controlador es el siguiente:

Solución propuesta	\$	87,723.00
Controlador-USB	\$	89,402.77

Al comparar los precios, son casi iguales resultando ser más económico utilizar mi solución propuesta, además de también no estar limitados a las funciones de una plataforma, ya que el entorno de programación de LAbVIEW® te brinda muchas aplicaciones y formas de trabajo además de seguir una programación en paralelo, en cambio la plataforma SCORBASE de programación del controlador-USB, sigue una programación en escalera y está limitado a sus funciones.

Los puntos a mejorar de este proyecto serian tratar de conseguir las medidas exactas de cada articulación (ya que en los manuales no se proporciona esta información) para disminuir ciertos errores de posicionamiento que se existen comparando con el otro controlador, cabe resalta que se le pidió a la empresa fabricante del robot y al proveedor en México estos datos, pero nunca se tuvo respuesta de los mismos. También se disminuiría esos errores haciendo un análisis más profundo a los motores y líneas de transmisión para obtener el valor real de los pulsos de encoder por cada grado.

Ya que en este trabajo se limitan los movimientos de la muñeca (elevación y giro), estos es, que no se pueden ejecutar los movimientos al mismo tiempo debido a la programación implementada, lo que se podría realizar aquí seria encontrar la relación entre cada encoder y los grados de cada movimiento simultaneo de los dos motores para lograr su correcto posicionamiento.

La comunicación es un punto importante en este trabajo, el uso de las variables globales es correcto y funciona pero depende mucho de la sincronización de los tiempos entre la PC y el controlador en tiempo real, ya que si la computadora no tenía un temporizador mayor al del controlador existía un tiempo de retraso en la ejecución del tarea asignada. En este caso se propone utilizar el método de pilas y colas para lograr una correcta ejecución de tareas si necesidad de estar sintonizando los programas. Entre la comunicación del controlador RT y el chasis FPGA no existe ningún problema ya que es instantánea y actúa directamente con el estado de los controles e indicadores. Entre estas plataformas, lo único que se sincronizo fue el PID pero se realizó para tener un correcto funcionamiento sin pérdida de información.

La programación en general es correcta y tiene resultados satisfactorios, pero a mi punto de vista se puede mejorar en muchos aspectos ya que son muy grandes las aplicaciones que se le pueden hacer a la tecnología *CompacRIO*®. En nuestro caso, se programó la lectura de encoders e interruptores, generación de PWM y el control PID en el chasis FPGA, pero es factible implementar la programación de la cinemática inversa y el perfil de trayectoria en esta sección, aumentando el rendimiento del controlador.



Una vez mejorando esto, se podría alcanzar aplicaciones a nivel industrial tanto al robot como al controlador; sistemas de paletizado, empacadoras, soldadura, etcétera.

6 Referencias

Referencias bibliográficas

1. [Barrientos] BARRIENTOS, Antonio C., Fundamentos de Robótica, Capítulos 1, 2, 3 y 4, 2ª Edición, McGraw-Hill, México 2007.
2. [Tsai] TSAI, Lung-Wen, Robot Analysis: The Mechanics of Serial and Parallel Manipulators, Chapter 1 & 2, 1st Edition, John Wiley & Sons, Inc., USA 1999.
3. [Spong] SPONG, Mark W., HUTCHINSON, Seth, M. VIDYASAGAR, Robot Modeling and Control, Chapter 1 & 3, 1st Edition, John Wiley & Sons, Inc., USA 2005.
4. [Intelitek] INTELITEK, SCORBOT-ER 4u: User Manual, Catalog #100343 Rev. B, Intelitek Inc., Manchester 2001.
5. [Intelitek] INTELITEK, Controller-USB: User Manual, Catalog #100341 Rev. D, Intelitek Inc., Manchester 2003.
6. [Contreras] CONTRERAS Vara, Carlos Enrique, Movimiento de un robot teleoperado a través de sistemas de libre distribución por internet, Tesis para obtener el título de Ingeniero Mecánico Electricista, Capitulo 4, México: UNAM, Facultad de Ingeniería 2002, 114 h.
7. [Torres] TORRES Miranda, Gabriel, MATUS Pérez, Mario Alberto, Control de un brazo robot de cinco grados de libertad mediante un PLC, Tesis para obtener el título de Ingeniero Eléctrico Electrónico, Capitulo 1 y 8, México: UNAM, Facultad de Ingeniería 2010, 120 h.
8. [Hellemann] HELLEMANN Holguín, Herbert, SÁNCHEZ Chávez, Daniel, Control de un brazo robótico teleoperado mediante un acelerómetro, Tesis para obtener el título de Ingeniero Mecatrónico, Capitulo 3 y 14, México: UNAM, Facultad de Ingeniería 2011, 177 h.
9. [Rementeria] REMENTERIA López, Juan Antonio, Teleoperacion de un SCORBO-ER 4u empleando un PAMCA, Tesis para obtener el título de Ingeniero Mecatrónico, Capitulo 1 y 5, México: UNAM, Facultad de Ingeniería 2011, 180 h.
10. [Romero] ROMERO Jaimes, Osvaldo, Implementación de una red neuronal artificial en un FPGA para el control de un motor de CD, Tesis para obtener el título de Ingeniero Mecatrónico, Capitulo 5, México: UNAM, Facultad de Ingeniería 2014, 95 h.
11. [Martínez] MARTÍNEZ Zamudio, Patricio, Análisis Cinemático de un Manipulador Paralelo Híbrido Tipo Delta, Tesis para obtener el grado de Maestro en Ingeniería campo de conocimiento – Mecatrónica, Capitulo 3, México: UNAM, Facultad de Ingeniería 2009, 160 h.
12. [Castañeda] CASTAÑEDA Cedeño, Serafín, Curso de Diseño Mecatrónico, Facultad de Ingeniería, UNAM, Semestre 2015-1.



Referencias electrónicas

1. <http://www.geceka.com/skynet-desperto-robot-de-ensamble-mata-a-empleado-de-volkswagen/>
Consulta: Octubre de 2015
2. http://www.metalactual.com/revista/29/soldadura_roboticos.pdf
Consulta: Febrero de 2015
3. http://www.ehu.eus/compmech/welcome/doc/dissertation_Salgado.pdf
Consulta: Febrero de 2015
4. http://somim.org.mx/articulos2010/memorias/memorias2013/pdfs/A2/A2_303.pdf
Consulta: Febrero de 2015
5. <http://proton.ucting.udg.mx/materias/robotica/>
Consulta: Febrero de 2015
6. http://www.unicauca.edu.co/ai/publicaciones/ISAShow_Vivas.pdf
Consulta: Febrero de 2015
7. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.192.8171&rep=rep1&type=pdf>
Consulta: Febrero de 2015
8. <http://www.dspace.espol.edu.ec/bitstream/123456789/6906/1/Gu%C3%ADa%20de%20Laboratorio%20del%20Brazo%20Robot%20Scorbot%20ER%204u.pdf>
Consulta: Febrero de 2015
9. <http://www.ni.com/webcast/3414/es/>
Consulta: Marzo de 2015
10. <http://www.ni.com/pdf/products/us/fullcriodevguide.pdf>
Consulta: Octubre de 2015
11. <http://www.ni.com/pdf/manuals/374211f.pdf>
Consulta: Octubre de 2015
12. <http://www.ni.com/pdf/manuals/374068f.pdf>
Consulta: Octubre de 2015



13. <http://www.ni.com/pdf/manuals/374639e.pdf>
Consulta: Octubre de 2015
14. <http://sine.ni.com/nips/cds/view/p/lang/es/nid/205621>
Consulta: Octubre de 2015
15. <http://sine.ni.com/nips/cds/view/p/lang/es/nid/202711>
Consulta: Octubre de 2015
16. <http://sine.ni.com/nips/cds/view/p/lang/es/nid/208809>
Consulta: Octubre de 2015
17. <https://www.ni.com/compactrio/esa/whatis.htm>
Consulta: Septiembre de 2015
18. <http://www.ni.com/tutorial/3921/en/>
Consulta: Septiembre de 2015
19. http://zone.ni.com/reference/en-XX/help/372983D-01/lvrobogsm/robo_arm_definition/#5r
Consulta: Septiembre de 2015

7 Anexos

Anexo A. Conector del robot

El robot SCORBOT-ER 4u contiene de fábrica un conector DB50 tipo macho, el cual esta cableado hacia los elementos del robot tales como: sensores de posicionamiento (Micro Switch), sensores de cada encoder y motores.

Cableado del robot SCORBOT-ER 4u				
Señal de Brazo Robot				Terminal del Conector DB50
Eje	Motor	Encoder	Microsw.	Pin #
1	+			50
	-			17
2	+			49
	-			16
3	+			48
	-			15
4	+			47
	-			14
5	+			46
	-			13
Gripper	+			45
	-			12
1		GDN		33*
		P ₁		5
		V _{led}		11
		P ₀		2
2		GDN		32*
		P ₁		21
		V _{led}		27
		P ₀		1
3		GDN		31*
		P ₁		4
		V _{led}		10
		P ₀		36
4		GDN		30*
		P ₁		20
		V _{led}		26
		P ₀		35

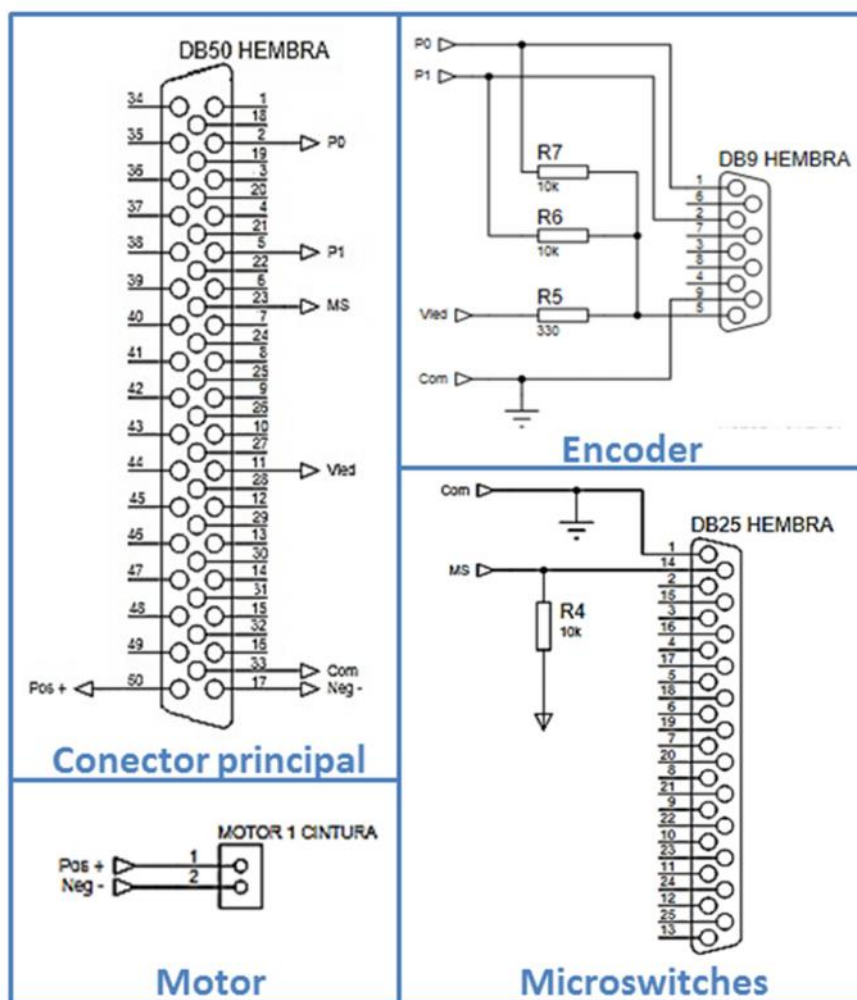


5	GDN	29*
	P_1	3
	V_{led}	9
	P_0	18
Gripper	GDN	28
	P_1	19
	V_{led}	25
	P_0	34
1	GDN	33*
	MS	23
2	GDN	32*
	MS	7
3	GDN	31*
	MS	24
4	GDN	30*
	MS	8
5	GDN	29*
	MS	6

Anexo B. Diagrama de conexiones

Para una mayor facilidad de conexión entre los módulos y el conector DB50 del robot, se utilizó un conector DB50 tipo hembra; el cual esta cableado con terminales de los motores, cada sensor de posicionamiento se conectó a un circuito pull up y a su vez las señales de interés a un conector DB25, y para la lectura de los encoders, se utilizaron 6 conectores DB9 tipo hembra para la lectura de encoder de cada motor ya que los módulos “NI 9505” tienen un conector DB9 tipo macho con las terminales de lectura.

Se muestra solamente el diagrama de conexiones implicadas en el movimiento de la cintura.





Anexo C. Cotizaciones de precios entre controladores

Cotización del proyecto.

Producto	Cantidad	Precio
NI cRIO-9073	1	\$ 35,370.00
NI 9505	6	\$ 47,331.00
NI 9401	1	\$ 5,022.00
Total		\$ 87,723.00

Esta cotización se realizó en día 17 de marzo de 2016.

UNIVERSIDAD AUTONOMA DE MEXICO
FACULTAD DE INGENIERIA
DEPARTAMENTO DE MECATRONICA
At'n: MTRO SERAFIN CASTAÑEDA
Tel:
e-mail:

P R E S E N T E

Estimado

De acuerdo a su solicitud, me permito poner a su consideración la propuesta siguiente:

PART	CANTIDAD	CONCEPTO	PRECIO UNITARIO SIN I.V.A. (M.N.)	PRECIO TOTAL SIN I.V.A. (M.N.)
1	PAQ	Controlador para Robot ER4 Marca: Intelitek Modelo: 009234 El controlador USB proporciona funciones avanzadas de control para el SCORBOT-ER 4u brazo robótico. Totalmente integrado con SCORBASE (control robótico) y el software de programación RobotCell (software de simulación 3D) Dimensiones (L x W x H) 315 mm x 223 mm x 117 mm 12.4 "x 8.8" x 4.6 " Peso 7 kg (15,4 libras) Temperatura de funcionamiento 0 ° - 55 ° C 32 ° - 131 ° F Tipo de control Tiempo real; PID; PWM Conductores ejes servo 8 ejes para robot y periféricos Microcontroladores Todas las funciones, NEC V853 RISC de 32 bits microcontrolador Comunicación Conexión USB para PC RS232 integrado para enseñar colgante Entradas 8 entradas digitales: PNP / NPN (alto / bajo) configurable, 0-24VDC max 4 entradas analógicas (resolución de 8 bits): voltaje de entrada 0-10V Salidas 8 salidas digitales: 1-4: relés 10A máx. 5-8: disipador / fuente configurables colectores abiertos Sink: 15VDC, 0.5A max. para cada salida Fuente: 15VDC, 50mA máx. para todas las salidas combinadas 2 salidas analógicas: (resolución de 8 bits): voltaje de salida 0-10 V, 20 mA máx Requerimientos de energía 110-220VAC (+ 15% -10%), 2A max, 50-60Hz Fuentes de alimentación internas	\$ 77,071.35	\$ 77,071.35

UNIVERSIDAD AUTONOMA DE MEXICO
FACULTAD DE INGENIERIA
DEPARTAMENTO DE MECATRONICA
At'n: MTRO SERAFIN CASTAÑEDA
Tel:
e-mail:

P R E S E N T E

Estimado

De acuerdo a su solicitud, me permito poner a su consideración la propuesta siguiente:

		Servo: 24V (dependiendo de la tensión de entrada y carga) Digital: 5V, + 15V, -12V Fuente de alimentación del usuario 12 VDC 0.1A max. Indicadores LED Poder principal bicolor Encendido Verde: La comunicación con PC: Rojo: No hay comunicación con PC: Intermitente: tiempo de espera de comunicaciones USB Motores Verde: Encendido Rojo: Emergencia Entradas digitales Verde Salidas digitales Rojo Características de seguridad Certificado de cumplimiento de seguridad de la CE; Interruptor de emergencia en el panel frontal del controlador / enseñar colgante / remoto Protección contra cortocircuitos Conductor de protección corte de energía Corte de energía del motor en caso de fallo de comunicación Impacto, la protección de carrera de software Vigilancia de hardware para cada eje protege contra fallas de software		
--	--	--	--	--

Sub Total	\$ 77,071.35
I.V.A.	\$ 12,331.42
Total:	\$ 89,402.77

Precios en Moneda Nacional

UNIVERSIDAD AUTONOMA DE MEXICO
FACULTAD DE INGENIERIA
DEPARTAMENTO DE MECATRONICA
At'n: MTRO SERAFIN CASTAÑEDA
Tel:
e-mail:

P R E S E N T E

Estimado

De acuerdo a su solicitud, me permito poner a su consideración la propuesta siguiente:

CONDICIONES COMERCIALES

1. **Vigencia:** Esta cotización tendrá una vigencia de **30 días naturales**, a partir de la fecha de la presente.
2. **Precios:** Los precios aquí cotizados son en Moneda Nacional y solo se contempla en paquete.
3. **Condiciones de pago:** 50% de anticipo al realizar su solicitud por escrito y el 50% a la entrega de los productos. Los precios aquí cotizados **incluyen suministro en las instalaciones del instituto y entrega, así como capacitación en el uso básico del controlador.**
4. **Entrega:** Los productos tienen un tiempo de entrega de 6 a 9 semanas, a partir de la fecha de recepción de su orden de compra.
5. **Garantía:** Garantía del buen funcionamiento de los equipos 12 meses
➤ **Nota: la garantía no será válida por mal uso o mal manejo de los mismos y si los sellos de garantía se encuentran alterados.**

Atentamente

ANGEL MORENO SUAZO
Coordinador de
Proyectos