



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA ELÉCTRICA – SISTEMAS ELECTRÓNICOS

INTEGRACIÓN DE UN MANIPULADOR ROBÓTICO DE 6 GRADOS DE
LIBERTAD CONTROLADO POR VOZ COMO ELEMENTO CENTRAL A UNA
CELDA DE MANUFACTURA EXPERIMENTAL CON MODO SENSORIAL DE
VISIÓN

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
Ing. ANGEL PADILLA SALAZAR

TUTOR PRINCIPAL
Dr. JUAN MARIO PEÑA CABRERA, INSTITUTO DE INVESTIGACIONES EN
MATEMÁTICAS APLICADAS Y EN SISTEMAS

MÉXICO, D. F. NOVIEMBRE 2013

JURADO ASIGNADO:

Presidente: DR. PÉREZ ALCÁZAR PABLO ROBERTO

Secretario: M. I. HARO RUÍZ LUIS ARTURO

Vocal: DR. PEÑA CABRERA JUAN MARIO

1^{er.} Suplente: DR. ACEVEDO CONTLA PEDRO

2^{d o.} Suplente: M. I. ÁLVAREZ CASTILLO JESÚS

Lugar donde se realizó la tesis: INSTITUTO DE INVESTIGACIONES EN MATEMÁTICAS APLICADAS Y EN SISTEMAS, UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

TUTOR DE TESIS:

DR. JUAN MARIO PEÑA CABRERA

FIRMA

(Segunda hoja)

Dedicatorias:

A mi compañera de viaje, Mayra González, por continuar a mi lado después de tantas batallas y por ayudarme a crear a ese pedacito de vida que está en tu panza. Te amo.

A Valentina Padilla González, te agradezco por escogerme como tu papá y por dejarme sin palabras para expresar lo que siento por tí. Te amo bolla.

A mis primeros amores, Georgina e Ismael, por seguirme tratando como su bebé. A Yessica, por ser la mejor hermana que podría desear. Y a mis lucecitas, Karina y Sonia, por alegrarme la vida con su presencia.

Gracias.

Agradecimientos:

- ❖ A la Universidad Nacional Autónoma de México, por permitirme crecer una vez más, tanto en lo personal como en lo profesional.
- ❖ A toda la gente del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), por todo su apoyo y amistad.
- ❖ Al Dr. Juan Mario Peña Cabrera, por su ayuda incondicional para lograr esta meta.
- ❖ A Pablo Roberto Pérez Alcázar, Luis Arturo Haro Ruíz, Pedro Acevedo Contla y Jesus Alvarez Castillo por el tiempo dedicado a la revisión de este trabajo.
- ❖ A la Coordinación de Estudios de Posgrado (CEP), por el apoyo económico brindado durante el último semestre de este posgrado.

Índice General

RESUMEN / ABSTRACT	1
CAPÍTULO 1, INTRODUCCIÓN	2
1.1 Justificación	2
1.2 Objetivo general	3
1.3 Objetivos específicos	3
1.4 Análisis del problema	4
1.5 Propuesta de solución	4
CAPÍTULO 2, ANTECEDENTES	5
2.1 Celdas de manufactura	5
2.2 Robots manipuladores	7
2.3 Reconocimiento automático de voz	13
CAPÍTULO 3, DESARROLLO E IMPLEMENTACIÓN	15
3.1 Manipulador robótico Labot Pro 5	15
3.2 Cinemática	17
3.3 Caracterización	22
3.4 Tarjeta de conversión USB~RS-232	27
3.5 Sistema de realimentación	29
3.6 Control manual	33

3.7	Control por voz	42
3.8	Modelo	55
3.9	Integración final	57
CAPÍTULO 4, PRUEBAS Y RESULTADOS EXPERIMENTALES		65
4.1	Repetibilidad	65
4.2	Reproducibilidad	66
4.3	Tablas de Consulta	70
4.4	Reconocimiento de voz	71
CAPÍTULO 5, CONCLUSIONES		73
5.1	Conclusiones.....	73
5.2	Trabajo futuro	74
REFERENCIAS		75
APÉNDICE A - GLOSARIO		79
APÉNDICE B – CÓDIGO FUENTE		82

Índice de Figuras

2.1	Esquema de una celda de manufactura flexible	6
2.2	Manipuladores industriales	7
2.3	Representación simbólica de las articulaciones robóticas	9
2.4	Configuraciones más comunes en los robots manipuladores	9
2.5	Espacio de trabajo de las configuraciones más comunes	11
2.6	Manipulador tipo codo	12
2.7	Estructura de la muñeca esférica	12
3.1	Manipulador robótico antropomórfico Labot Pro 5	16
3.2	Marcos de referencia para el robot articulado de 6-DOF	18
3.3	Dirección de rotación y rangos de movimiento Labot Pro 5	22
3.4	Espacio de trabajo restringido Labot Pro 5	23
3.5	Tarjeta de conversión USB-RS232	28
3.6	Diagrama de bloques del sistema de realimentación	29
3.7	Diagrama de flujo del algoritmo para el microcontrolador	31
3.8	Elementos del sistema de realimentación	32
3.9	Interfaz gráfica del fabricante	33
3.10	Interfaz gráfica propia	36
3.11	Diagrama de flujo de la rutina Principal	37
3.12	Diagrama de flujo de la subrutina Ejecutar	39
3.13	Diagrama de flujo de la subrutina Hombro_Ejecutar	40
3.14	Ejemplo de uso de la GUI para el control manual	42
3.15	Arquitectura sistema de reconocimiento de voz basado en HMM	43
3.16	Modelo de fonema basado en HMM	45
3.17	Forma de utilizar el software de reconocimiento de voz	47
3.18	Interfaz gráfica para el control por voz	49
3.19	Diagrama de flujo de la GUI del control por voz	51
3.20	Diagrama de flujo de la GUI del control por voz	52
3.21	Diagrama de flujo de la GUI del control por voz	53
3.22	Modelo del sistema de control por voz	55
3.23	Mesa de trabajo del manipulador robótico	57
3.24	Interfaz gráfica para el control manual y por voz	59
3.25	Interfaz gráfica para el control por voz	60
3.26	Diferentes posiciones del grupo de rutinas "Posición X"	61

3.27	Estructura de la celda de manufactura experimental	62
3.28	Circuito para el control de la banda transportadora	63
3.29	Celda de manufactura flexible experimental	64
4.1	Prueba para medir la repetibilidad	66
4.2	Numeración de las zonas en la prueba de reproducibilidad	67
4.3	Resultados de reproducibilidad con software del fabricante	68
4.4	Resultados de reproducibilidad con software propio	69

Índice de Tablas

3.1	Asignación de letras para transmisión de comandos	17
3.2	Parámetros de D-H para el robot articulado de 6-DOF	18
3.3	Características del manipulador robótico Labot Pro 5	24
3.4	Tablas de lookup que relacionan voltajes de los potenciómetros de referencia y ángulos con respecto a la horizontal	26
3.5	Tablas de lookup que relacionan ángulos con respecto a la horizontal y posiciones de los motores del manipulador	27
3.6	Modos de operación de ultra-baja potencia	30
3.7	Descripción de los componentes que conforman la GUI del programa "Control_Manual"	35
3.8	Descripción de los componentes que conforman la GUI del programa "Control_Voz"	48
3.9	Descripción de los comandos utilizados en el control por voz	50
4.1	Resultados de la prueba de repetibilidad	66
4.2	Resultados de la prueba del uso de tablas de Lookup	70
4.3	Resultados de la prueba de reconocimiento de comandos de voz	71
4.4	Resultados de la prueba sobre cómo afecta el ruido ambiental sobre el reconocimiento de voz	72

Resumen

En el presente trabajo de tesis se describe la integración de un manipulador robótico controlado por voz, a una celda de manufactura experimental con modo sensorial de visión que está siendo desarrollada dentro de una línea de investigación en robótica para manufactura en la Sección de Electrónica y Automatización del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas.

Las tareas de las que estará encargado el manipulador robótico serán las de transportar piezas dentro del espacio de trabajo mediante un control punto a punto; sin embargo, por la naturaleza propia del tipo de manipulador utilizado, será necesario implementar un sistema de realimentación que se encargue de decir al módulo de control el estado del manipulador en todo momento.

Por otro lado, el control por voz del manipulador robótico consistirá en darle una orden vocal para que realice la tarea o rutina correspondiente a dicha orden; de esta manera, el operador del brazo jamás estará en contacto directo con algún componente de la celda.

Abstract

In this thesis, the integration of a voice-controlled robotic manipulator to an experimental manufacturing cell is described. The cell is being developed within a research on manufacturing robotics inside the Institute for Research in Applied Mathematics and Systems (IIMAS) Electronics and Automation Department.

The tasks that the robotic manipulator has to take over are the transportation of pieces within the work envelope through a point-to-point control; however, by the very nature of the type of manipulator used, it will be necessary to implement a feedback system responsible of telling the control module the status of the manipulator at any moment.

On the other hand, the robotic manipulator voice control consists of the accomplishment of a specific routine for a given voice command, so that the manipulator operator will never be in direct contact with any of the cell components.

Capítulo 1

Introducción

En este primer capítulo se establecen los objetivos que buscan alcanzarse con esta tesis y la justificación del interés en formar parte del proyecto de investigación para el cual se hizo este trabajo. Además, se presenta un análisis del problema para después presentar una propuesta de solución.

1.1 Justificación

En la Sección de Electrónica y Automatización del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), existe una línea de investigación en robótica para manufactura que busca desarrollar una Celda de Manufactura Flexible Experimental con modo sensorial de visión^[1,2,3].

Actualmente, el mayor avance de la investigación se ha dado en la parte de los sistemas de visión artificial, encargados del reconocimiento de la posición y forma de objetos dentro de la celda; sin embargo, hace falta la integración del manipulador robótico que realice de manera robusta y confiable las acciones de transporte de piezas entre las máquinas que conformen la celda (cinta transportadora, inspección visual, centros de maquinado, etc.).

La integración del manipulador robótico dentro de la celda de manufactura resultará vital para aumentar la autonomía del sistema como una parte fundamental en su operación, permitiendo prescindir de una persona que transporte las piezas dentro del área de trabajo.

El sistema de realimentación del manipulador robótico será otro aporte importante a la integración en la celda, ya que ayudará a tener un control más robusto y preciso de las acciones que se realicen y, dado el caso, permitirá realizar las correcciones necesarias para un funcionamiento óptimo.

De la misma manera, al dotar a la celda de manufactura de un modo sensorial de reconocimiento de voz, se permitirá un aumento en la flexibilidad y en la autonomía del sistema, así como la exploración de las ventajas y desventajas que este tipo de tecnología pueda brindar al proceso de automatización de la celda.

Por último, cabe señalar que este proyecto es bastante completo en cuanto a la aplicación de muchos de los conocimientos adquiridos durante el posgrado y la licenciatura; incluyendo temas de robótica, programación, instrumentación, electrónica, estadística, etc.

1.2 Objetivo general

Preparar y adecuar un manipulador robótico, antropomórfico y controlado por voz, para integrarlo a una celda de manufactura flexible experimental y que se encargue de realizar acciones básicas de traslado de piezas dentro del espacio de trabajo.

1.3 Objetivos específicos

- Desarrollar una interfaz gráfica de usuario (GUI) para controlar manualmente el manipulador robótico;
- Desarrollar una interfaz gráfica de usuario (GUI) para controlar el manipulador por medio de la voz;
- Diseñar e implementar un sistema de realimentación que ayude a mejorar el funcionamiento del robot;
- Corregir los problemas de funcionamiento del manipulador;
- Reforzar la estructura de la celda de manufactura flexible experimental;
- Integrar el manipulador a la celda de manufactura;
- Implementar el control analógico para una banda transportadora (*conveyor belt*) e integrarla a la celda.

1.4 Análisis del problema

Una vez definidos los objetivos, se hace un análisis del problema para que se pueda desarrollar una propuesta de solución factible.

Lo que se busca hacer en este proyecto es integrar un brazo manipulador robótico a una celda de manufactura flexible experimental, que se encargue de transportar las piezas dentro del espacio de trabajo. El desempeño del manipulador deberá mejorarse lo más que se pueda, sin importar los problemas de fábrica, mecánicos o de diseño que se tengan.

Otro punto importante será el desarrollo de una interfaz gráfica para que el usuario pueda manejar el manipulador robótico de forma robusta y sencilla.

También se requiere implementar un control que utilice comandos de voz para operar diferentes aspectos del manejo del manipulador.

1.5 Propuesta de solución

La solución propuesta para mejorar el funcionamiento del manipulador consiste en implementar un sistema que brinde la posibilidad de tener realimentación a través de los voltajes de referencia de los potenciómetros del hombro y del codo del robot, ya que estos se encuentran expuestos y es muy fácil acceder a ellos. Este sistema se implementa mediante un microcontrolador, que básicamente tenga las siguientes prestaciones: varios puertos de entrada, conversores analógicos digitales, bajo consumo de corriente, entre otras.

Para desarrollar las interfaces de usuario para los dos tipos de control, manual y por voz, se propone usar el ambiente de desarrollo especializado para proyectos de instrumentación llamado Labview.

En cuanto al reconocimiento de voz, se propone utilizar una paquetería comercial, para evitar el desarrollo de un software propio, ya que esto excedería por mucho los límites de tiempo y conocimientos del proyecto.

Capítulo 2

Antecedentes

En este capítulo se desarrollan los conceptos principales utilizados durante el proyecto; así como el estado del arte para darle contexto al trabajo.

2.1 Celdas de manufactura

El avance en los métodos de *Manufactura* a través de la historia ha jugado un papel vital en el desarrollo de la tecnología actual, la cual permite un crecimiento en la economía de las naciones y por lo tanto una mejora en el nivel de vida de su población^[4].

Dentro del ambiente de la producción de bienes, muchas de las piezas producidas tienen ciertas semejanzas en su forma y método de manufactura; esto da lugar a una clasificación en grupos o familias de piezas similares entre sí y, por ende, a un proceso productivo más eficiente. Para explotar completamente las similitudes en las partes de una familia, la producción debe organizarse en *Celdas de Manufactura* que contengan una o varias máquinas especializadas en elaborar las partes. Dichas celdas son comúnmente clasificadas según el número de máquinas y el grado de automatización que tengan; siendo las llamadas *Celdas de Manufactura Flexible* las que presentan el mayor nivel de automatización^[5]. Estas celdas se definen como un grupo de estaciones de trabajo (usualmente máquinas *CNC*¹), interconectadas por un sistema automático de manejo, transporte y almacenamiento de materiales y controladas por un sistema de

¹ Control Numérico Computarizado (Computer Numerical Control): es un proceso asistido por computadora para controlar máquinas de uso general a través de instrucciones generadas por un procesador y almacenadas en un sistema de memoria o medio de almacenamiento^[6].

cómputo integrado^[4]. Para observar un ejemplo de celda de manufactura flexible, véase la Figura 2.1, donde se distinguen las diferentes partes que la componen.

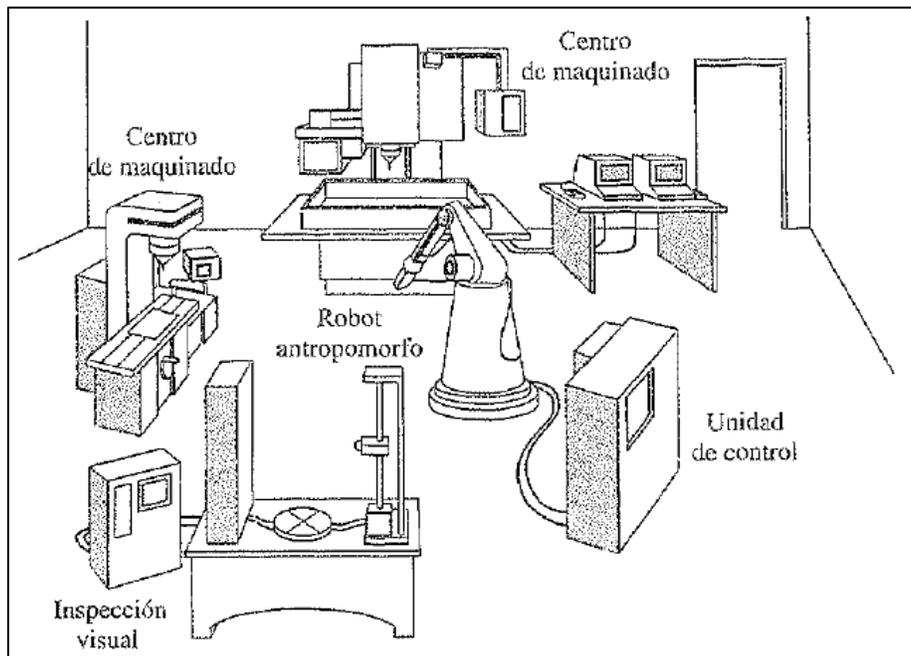


Figura 2.1 Esquema de una celda de manufactura flexible. Incluye dos máquinas herramienta, un sistema automatizado de inspección de piezas; un robot central que atiende a esas máquinas y un sistema de control que dirige todo el proceso.

En general, las celdas de manufactura flexible no están atendidas por humanos, por lo que su diseño y operación deben ser más precisos que los de otras celdas; debido a lo cual, se ha generalizado la idea de utilizar robots manipuladores con capacidades sensoriales y algoritmos inteligentes basadas en las últimas investigaciones y desarrollos en Inteligencia Artificial para lograr sistemas más robustos, auto-adaptables y parecidos al comportamiento humano^[7].

En la actualidad, mucha de la investigación sobre este tipo de celdas está más enfocada a desarrollar celdas flexibles para la enseñanza, con el objetivo de dotar a los alumnos de las habilidades interdisciplinarias necesarias para el diseño de los sistemas de manufactura en el ámbito de la industria^[8]. Otro enfoque actual sobre el uso de estas celdas es el desarrollo de sistemas de enseñanza para robots dentro de una línea de manufactura flexible, con el objeto de instruirlos para que puedan actuar de forma independiente, sin intervención humana^[9].

2.2 Robots manipuladores^[10]

Como se mencionó antes, las celdas de manufactura flexible no son atendidas por operadores humanos y necesitan un reemplazo altamente automatizado y adaptable a los cambios en las necesidades de los ámbitos de manufactura. De ahí la idea de implementar robots manipuladores dotados con los modos sensoriales más utilizados: reconocimiento de voz, visión y sensado de fuerza^[11].

Aunque el término robot se utiliza para nombrar a virtualmente cualquier sistema que sea más o menos autónomo, durante este trabajo se utilizará para nombrar solamente a los *manipuladores industriales* como los de la Figura 2.2; que en esencia son un brazo mecánico que opera bajo un control por computadora^[12]. En la actualidad, entre las empresas líderes en la fabricación de robots industriales se encuentran KUKA Robot Group^[13], ABB Group^[14], FANUC Robotics^[15] y Yaskawa Motoman Robotics^[16].

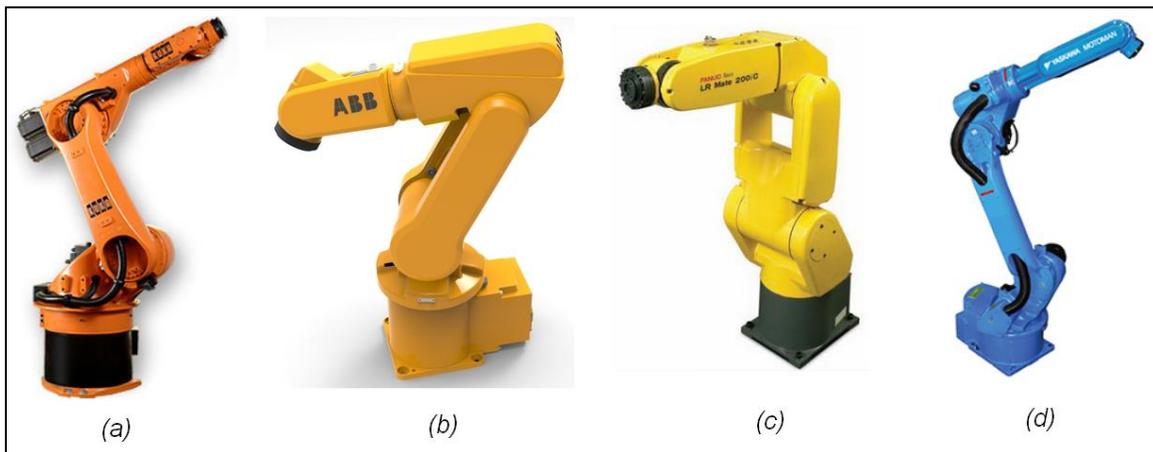


Figura 2.2 Manipuladores industriales. (a) KUKA KR 60-3. (b) ABB IRB 120. (c) FANUC LR Mate 200iC/5F. (d) MOTOMAN HP20D.

2.2.1 Componentes

- *Manipulador*: se trata del cuerpo mismo del robot formado por sus elementos estructurales.
- *Efecto final*: es la parte conectada al final del brazo (mano) que realiza las tareas requeridas. Puede ser desde la pinza más simple, que solo abra y cierre, hasta una mano antropomórfica que trate de imitar la mano humana.
- *Actuadores*: son los “músculos” que mueven la estructura del manipulador. Los más comunes son los servomotores, motores a pasos, actuadores hidráulicos o neumáticos, etc.

- *Sensores*: se utilizan para recolectar información del estado interno del robot. Algunos ejemplos son los sensores de fuerza, distancia, proximidad, los sistemas de visión e incluso sintetizadores de voz.
- *Controlador*: es el encargado de controlar los movimientos de los actuadores y coordinarlos con la realimentación de los sensores.
- *Procesador*: se encarga de hacer los cálculos de movimientos y velocidades y de supervisar las acciones coordinadas del controlador y los sensores. Por lo regular, el controlador y el procesador se encuentran en una misma unidad.

2.2.2 Grados de libertad

En un manipulador robótico, el número de articulaciones determina el número de *Grados de Libertad (Degrees-of-Freedom, DOF)*. En el espacio tridimensional, un objeto rígido tiene 6 grados de libertad: 3 para *posición* y 3 para *orientación*. Por lo tanto, un manipulador debería de contar con mínimo 6 DOF, ya que con menos de seis no podrá alcanzar cualquier punto dada una orientación cualquiera y con más de seis se tratará de un manipulador cinemáticamente redundante.

2.2.3 Articulaciones

En la teoría de los manipuladores robóticos^[17], éstos son considerados como elementos o *eslabones* (links) conectados mediante uniones o *articulaciones* (joints) para formar una *Cadena cinemática*. Dichas articulaciones pueden ser giratorias (de *Revolución*) o lineales (*Prismáticas*) y son denotadas por una *R* y una *P*, respectivamente; de ahí que el número y tipo de articulaciones se pueda usar para clasificarlos. En la Figura 2.3 se observan ambos tipos de articulaciones, sus representaciones y sus respectivos movimientos.

2.2.4 Configuraciones

Casi la totalidad de los manipuladores suelen clasificarse cinemáticamente en base a las tres primeras articulaciones del brazo y en la descripción por separado de la muñeca. La mayoría de los manipuladores caen dentro de alguna de las siguientes configuraciones, véase Figura 2.4: *Cartesiana* (PPP), *Cilíndrica* (RPP), *Esférica* (RRP), *Articulada* (RRR), *SCARA*² (RRP). En cuanto a la muñeca, la configuración más común es la *Esférica* (RRR). Cada una de las cinco configuraciones anteriores son robots con

² Selective Compliance Assembly Robot Arm.

elementos en serie; sin embargo, existe una sexta configuración de manipuladores conocidos como *manipuladores paralelos*, donde los elementos o eslabones están dispuestos en una cadena cinemática cerrada en vez de abierta como las otras configuraciones.

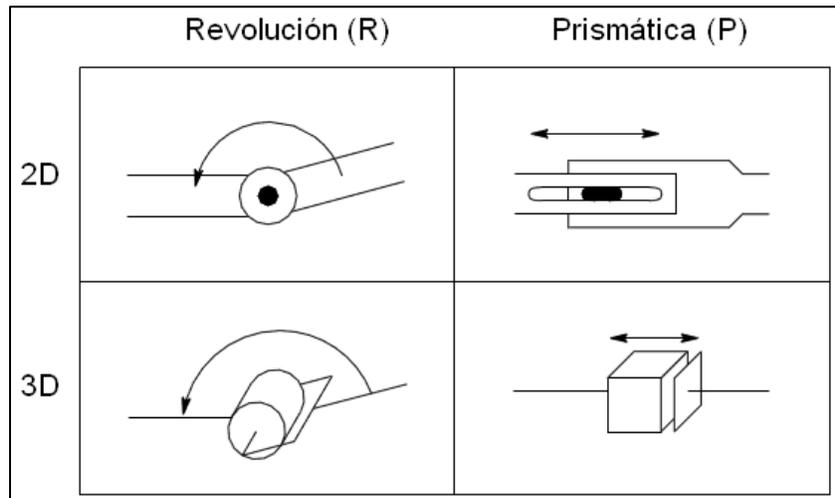


Figura 2.3 Representación simbólica de las articulaciones robóticas.

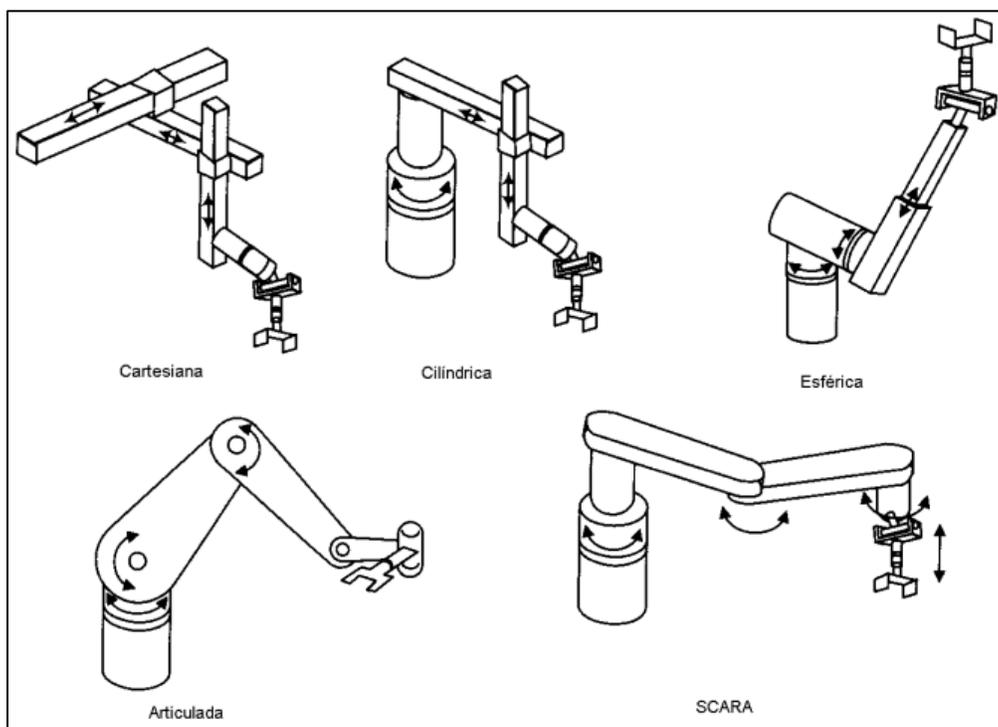


Figura 2.4 Configuraciones más comunes en los robots manipuladores.

2.2.5 Características

Para caracterizar las especificaciones del robot, se utilizan los siguientes conceptos:

- *Carga útil* (payload): es el peso que puede cargar sin exceder sus demás especificaciones.
- *Alcance* (reach): es la distancia máxima que puede alcanzar dentro de su espacio de trabajo.
- *Precisión* (accuracy): es una medida de cuán exactamente puede alcanzar el robot un punto dado dentro del espacio de trabajo.
- *Repetibilidad* (repeatability): es una medida de qué tan preciso es el manipulador para regresar a un mismo punto en repetidas ocasiones. Aunque el manipulador sea dirigido muchas veces al mismo punto, no siempre lo alcanzará; sin embargo, siempre estará dentro de un cierto radio del punto deseado, por lo tanto, al radio de un círculo formado por repetidos movimientos se le llama repetibilidad.

Debido a que la precisión y la repetibilidad son las características de funcionamiento más importantes para un manipulador robótico, existen normas internacionales sobre los procedimientos para obtenerlas. En la norma *ISO 9283:1998 Manipulating Robots – performance Criteria and Related methods* se definen las principales características de funcionamiento de los robots industriales, las formas de especificarlos, las recomendaciones sobre cómo realizar las pruebas para obtener dichas características e incluso la forma de presentar los informes de resultados^[18].

2.2.6 Espacio de trabajo

Al volumen total que abarca el órgano o efector final cuando el robot ejecuta todos los movimientos posibles se le conoce como *Espacio de Trabajo*. La Figura 2.5 muestra el espacio de las configuraciones más comunes.

2.2.7 Método de control

Existen dos métodos de control para robots: servo y no-servo. El control servo utiliza control en lazo cerrado para determinar su movimiento, esto permite que a diferencia del control no-servo, los robots tengan capacidades de ser multifuncionales y reprogramables. Dentro de los robots servo, existe otra clasificación de acuerdo al método que el controlador utiliza para guiar al órgano terminal. El tipo más simple de control en esta clase es el *Control Punto a Punto*, el cual consiste en enseñar al manipulador un conjunto discreto de puntos, pero sin control sobre la trayectoria entre éstos. Por otro

lado, en el *Control de Trayectoria Continua*, la trayectoria completa del efector final puede ser controlada.

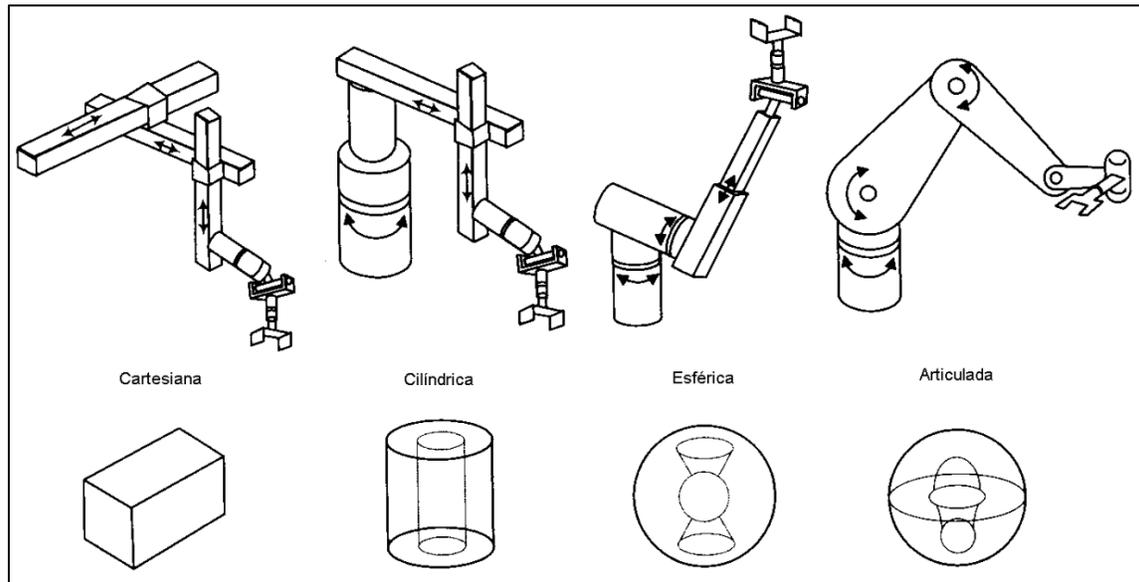


Figura 2.5 Espacio de trabajo de las configuraciones más comunes en los robots manipuladores.

2.2.8 Manipulador robótico articulado

Conocidos también como manipuladores rotacionales o *Antropomórficos*, al igual que el brazo humano tiene todas sus articulaciones de revolución, lo que lo convierte en la configuración más usada para robots industriales. Como se observa en la Figura 2.6, cuando el manipulador tiene los ejes de las articulaciones del hombro y el codo paralelos entre sí y perpendiculares con respecto al eje de la base, se le conoce como *manipulador tipo codo*.

Como se mencionó antes, un manipulador necesita de 3 grados de libertad para determinar su posición, lo cual se puede lograr con un *manipulador tipo codo*; sin embargo, para determinar la orientación, regularmente se utiliza otra configuración conocida como de *muñeca esférica*.

Configuración muñeca esférica (RRR)

El término muñeca se refiere a las articulaciones entre el brazo y la mano en la cadena cinemática y si todos los ejes de las articulaciones se intersecan en un punto en común se le llama *muñeca esférica* (véase la Figura 2.7). La importancia de la muñeca esférica radica en que simplifica el análisis cinemático, permitiendo desacoplar el posicionamiento y la orientación del efector final. Típicamente, el manipulador cuenta con tres DOF para la posición, producidos por las tres o más articulaciones del brazo. El número de grados de libertad para la orientación depende de los grados de libertad de la muñeca.

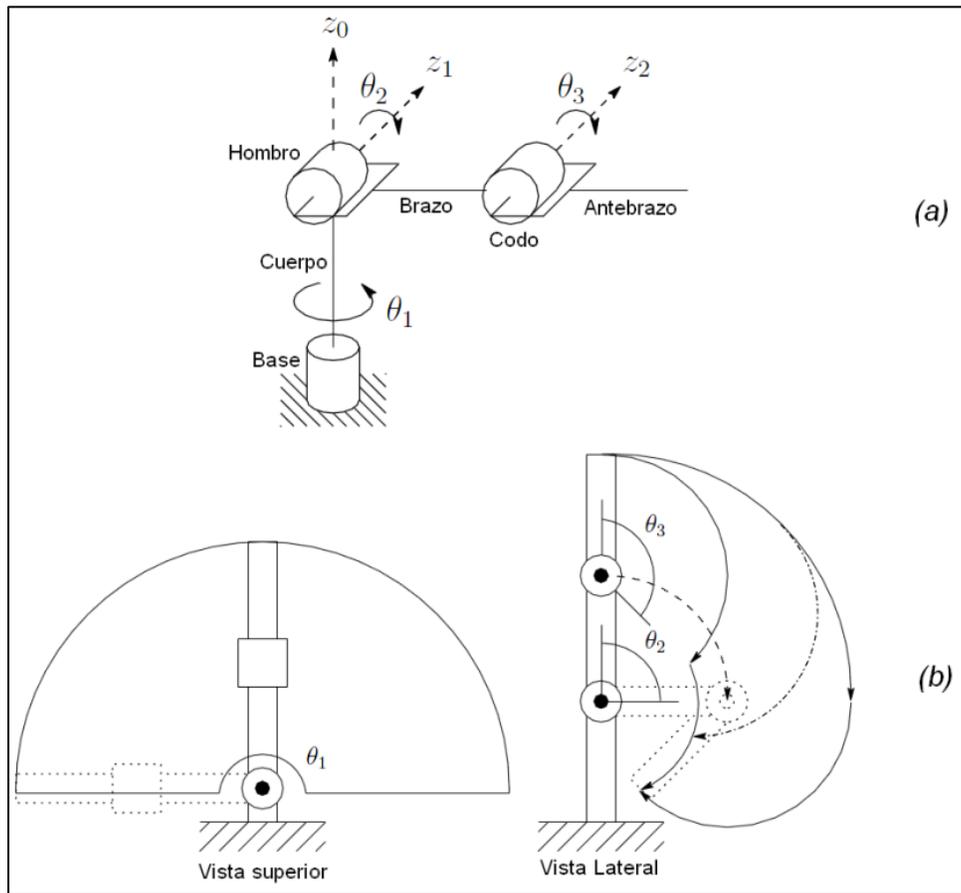


Figura 2.6 Manipulador tipo codo. (a) Estructura. (b) Espacio de trabajo.

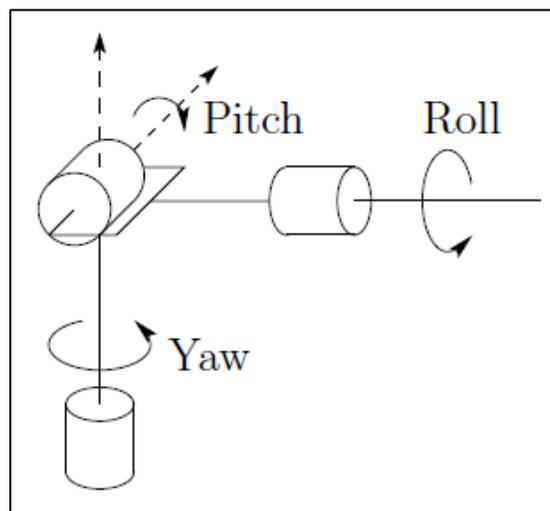


Figura 2.7 Estructura de la muñeca esférica.

2.3 Reconocimiento automático de voz

Una Interfaz de Usuario (UI) es un componente vital de cualquier producto manipulado por un usuario humano. El concepto principal de la robótica es el crear una máquina autónoma que pueda reemplazar el trabajo humano; sin embargo, para poder controlar al robot o proveerle una rutina de trabajo, el humano debe poder comunicarse con el robot, lo cual se logra por medio de una UI.

Desde los principios de la robótica, las herramientas preferidas para interactuar con una máquina eran las interfaces gráficas, los teclados, los joysticks, etc. Pero debido al avance de la tecnología, se han introducido nuevas formas de interactuar con las máquinas. En particular, las *interfaces de control por voz* juegan un rol muy importante debido a que la eficiencia de los sistemas de manufactura aumenta si la interfaz es más similar a la forma en que los humanos dan una orden (Lenguaje Natural) en un contexto social, de ahí que esta tecnología pueda ser asimilada relativamente fácil por los usuarios^[19].

A diferencia de los sistemas de reconocimiento de imágenes, los sistemas de Reconocimiento Automático de Voz para un ámbito industrial no han tenido el mismo nivel de desarrollo debido a los principales problemas que surgen en su aplicación: un ambiente industrial es sumamente ruidoso y los sistemas industriales no fueron diseñados para incorporar este tipo de características y por lo regular no cuentan con computadoras lo suficientemente potentes dedicadas específicamente a las interfaces humano-máquina (Human-Machine Interface, HMI)^[20]. Sin embargo, actualmente existen compañías líderes en el desarrollo de software de reconocimiento de voz, el cual cada vez cuenta con mejores tasas de precisión^[21].

Hoy por hoy existen trabajos de investigación sobre la interacción humano-robot en la UNAM y particularmente en el IIMAS con los trabajos del grupo “Golem”^[22]; sin embargo, dichos estudios se enfocan más en robots de servicio con más formas de interacción (visuales, táctiles, etc.) y no en una interacción para el ámbito industrial como lo es una celda de manufactura flexible.

Como concepto, el Reconocimiento Automático de Voz (Automatic Speech Recognition, ASR) consiste en procesar la señal de voz emitida por un ser humano para convertirla en texto; y como consecuencia, pueda ser entendido por una computadora.

Alrededor del mundo se han realizado trabajos de investigación^[23,24] sobre el reconocimiento de voz. En el año de 1952, en los laboratorios Bell, se creó la primera máquina capaz de reconocer cualquiera de los diez dígitos del 0-9 (Davis et al., 1952). Desde el año 1971 hasta 1976, la agencia del Departamento de Defensa de Estados Unidos DARPA (Defense Advanced Research Projects Agency) establece el programa SUR (Speech Understanding Research) para desarrollar un sistema computacional que pudiera entender el habla continua, del cual sobresalen los trabajos de la Universidad Carnegie Mellon “Harpy” capaz de reconocer 1,011 palabras con relativa precisión y “Byblos” de BBN Technologies^[25]. A partir de la década de los 1980’s, se cambia de

modelos intuitivos a un modelo estadístico que se volvería el más utilizado hasta la actualidad: el Modelo Oculto de Markov (HMM).

En la actualidad, existen un sin número de aplicaciones para reconocimiento de voz comerciales^[26], siendo “Dragon Naturally Speaking”^[27] de la empresa Nuance Communications Inc. uno de los reconocedores de voz más ampliamente utilizados. También existen aplicaciones de empresas tan grandes como Microsoft, con su reconocedor integrado en su sistema operativo “Windows 7”^[28] y el integrado en su sensor de movimiento para videojuegos “Kinect”^[29], y de Google con su aplicación para el sistema operativo móvil “Android” llamada “Google Voice”^[30].

En cuanto a los algoritmos que se usan para desarrollar herramientas de reconocimiento de voz se encuentra el llamado Dynamic Time Warping, el cual consiste en medir las similitudes entre dos secuencias (de imágenes, frases o gráficos) que pueden variar en tiempo o velocidad.

Otro algoritmo ampliamente utilizado hoy en día se basa en los *Modelos Ocultos de Markov*^[31], en los cuales se considera la probabilidad de que algún sonido desconocido sea o no una palabra.

Un último algoritmo es el basado en Redes Neuronales; sin embargo, actualmente existen trabajos que utilizan sistemas híbridos con Modelos Ocultos de Markov y Redes Neuronales^[32].

Capítulo 3

Desarrollo e Implementación

A lo largo de este capítulo se exponen de forma detallada las etapas que se desarrollaron y los pasos que se siguieron para implementar el proyecto.

3.1 Manipulador robótico Labot Pro 5

El elemento central de este trabajo es el brazo robótico que se integrará a la celda de manufactura experimental. Se trata de un manipulador robótico antropomórfico, semi-industrial, modelo Labot Pro 5, fabricado por la empresa mexicana *Robótica CRYA* y diseñado especialmente para propósitos de enseñanza^[33].

El manipulador Labot Pro 5 cuenta con 6 grados de libertad, incluyendo la pinza (*base, hombro, codo, muñeca, mano y efector final*); es manejado con un software propio del fabricante que utiliza control punto a punto; y se comunica con la PC por medio de un puerto serial. Los eslabones de aluminio del brazo son movidos por *Servomotores de Lazo Cerrado*, lo cual brinda una fuerza considerable^[34].



Figura 3.1 Manipulador robótico antropomórfico Labot Pro 5.

El control del manipulador es un *control punto a punto*, lo cual quiere decir que, para trasladarse a cualquier punto, el robot no toma en cuenta la trayectoria, puesto que lo único que importa es alcanzar el punto en cuestión; por lo tanto, si se requiere que el robot cubra cierta trayectoria, es necesario hacer que vaya a tantos puntos sobre la trayectoria como sea posible.

Para suministrar comandos al manipulador es necesario enviarle una *cadena de caracteres* formada por la letra mayúscula del motor que se quiere mover, según la relación de la Tabla 3.1, y un número entero entre 500 y 2500, equivalente a una posición dentro del rango de movimiento del motor. Por ejemplo, si se quiere mover el motor de la articulación del hombro a la mitad de su rango de posiciones, se le enviará la cadena de caracteres “B 1500”.

Articulación	Motor
Base	A
Hombro	B
Codo	C
Muñeca	D
Mano	E
Efector	F

Tabla 3.1 Asignación de letras para transmisión de comandos.

Resulta importante mencionar que, aunque el cambio mínimo dentro del rango de posiciones de los motores para el hombro y el codo es una unidad, en la práctica se observó que los motores solamente cambian de posición cada **20** unidades. Por ejemplo, idealmente dentro de un rango de movimiento de 500 a 2500 se tienen 2000 posiciones, sin embargo, en realidad se tienen solamente 100 posiciones.

3.2 Cinemática^[10,12,17]

Para el análisis cinemático del manipulador articulado, debe partirse de la suposición de que cada una de sus articulaciones tiene solamente un grado de libertad y la acción de cada una de ellas puede ser descrita por medio de un solo escalar real: el desplazamiento para el caso de una articulación prismática y el ángulo de rotación para una de revolución.

Cinemática directa

El objetivo de la cinemática directa consiste en calcular la posición y orientación del efector final del robot, si se conocen las variables de las articulaciones y su configuración; esto quiere decir que en cualquier momento se puede saber en dónde se encuentra el robot. Lo anterior se lleva a cabo utilizando matrices de transformación homogéneas, para relacionar el sistema (marco) de referencia del robot con el del efector final.

Para calcular las ecuaciones de la cinemática directa, se utiliza la convención de Denavit-Hartenberg (D-H), en la cual, cada transformación homogénea se representa como el producto de cuatro transformaciones básicas:

$$\begin{aligned}
 {}^{i-1}A_i &= R_{z,\theta_i} T_{z,d_i} T_{x,a_i} R_{x,\alpha_i} \\
 &= \begin{bmatrix} \cos \theta_i & -\sin \theta_i & 0 & 0 \\ \sin \theta_i & \cos \theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha_i & -\sin \alpha_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & 0 & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{3.1}$$

donde θ_i : ángulo, d_i : distancia (offset), a_i : longitud, α_i : ángulo entre dos ejes Z sucesivos.

El análisis para un manipulador de 6-DOF comienza con el establecimiento de los marcos de referencia, como lo ilustra la Figura 3.2.

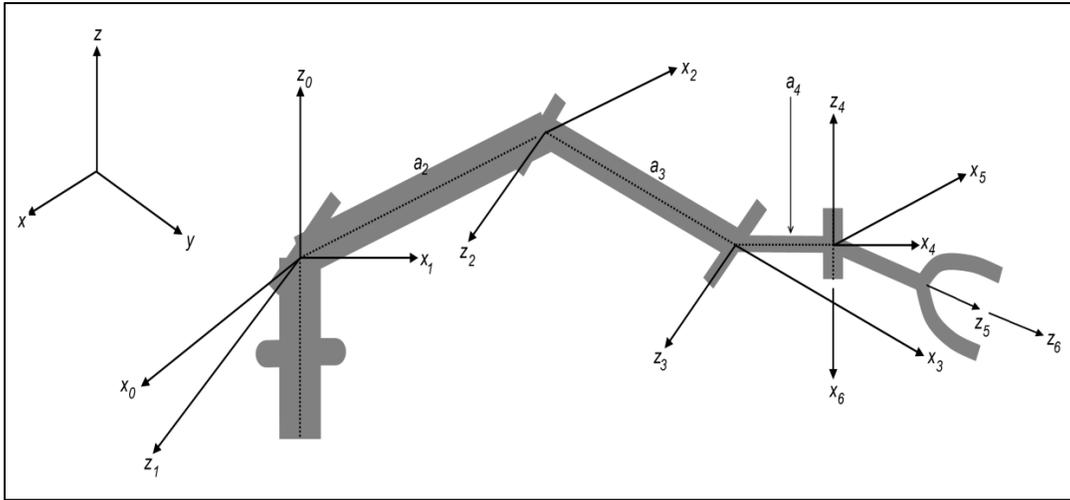


Figura 3.2 Marcos de referencia para el robot articulado de 6-DOF.

A continuación, se construye la Tabla 3.2 con los parámetros θ_i , d_i , a_i y α_i .

#	θ	d	a	α
0-1	θ_1	0	0	90
1-2	θ_2	0	a_2	0
2-3	θ_3	0	a_3	0
3-4	θ_4	0	a_4	-90
4-5	θ_5	0	0	90
5-6	θ_6	0	0	0

Tabla 3.2 Parámetros de D-H para el robot articulado de 6-DOF.

Para las matrices de transformación ${}^{i-1}A_i$ que corresponden a $i = 6$, se tiene:

$$\begin{aligned}
 {}^0A_1 &= \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^1A_2 &= \begin{bmatrix} C_2 & -S_2 & 0 & C_2 a_2 \\ S_2 & C_2 & 0 & S_2 a_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^2A_3 &= \begin{bmatrix} C_3 & -S_3 & 0 & C_3 a_3 \\ S_3 & C_3 & 0 & S_3 a_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^3A_4 &= \begin{bmatrix} C_4 & 0 & -S_4 & C_4 a_4 \\ S_4 & 0 & C_4 & S_4 a_4 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^4A_5 &= \begin{bmatrix} C_5 & 0 & S_5 & 0 \\ S_5 & 0 & -C_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^5A_6 &= \begin{bmatrix} C_6 & -S_6 & 0 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{3.2}$$

Para simplificar la solución final, se utilizan las siguientes simplificaciones: $\sin \theta_i = S_i$, $\cos \theta_i = C_i$ y las identidades trigonométricas:

$$\begin{aligned}
 \sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2 &= \sin(\theta_1 + \theta_2) = S_{12} \\
 \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 &= \cos(\theta_1 + \theta_2) = C_{12}
 \end{aligned} \tag{3.3}$$

Finalmente, la transformación total entre la base del robot (marco 0) y el efector es:

$$\begin{aligned}
 {}^0T_6 &= {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6 \\
 &= \begin{bmatrix} C_1(C_{234}C_5C_6 - S_{234}S_6) - S_1S_5C_6 & S_1(C_{234}C_5C_6 - S_{234}S_6) + C_1S_5C_6 & S_{234}C_5C_6 + C_{234}S_6 & 0 \\ C_1(-C_{234}C_5C_6 - S_{234}C_6) + S_1S_5C_6 & S_1(-C_{234}C_5C_6 - S_{234}C_6) - C_1S_5C_6 & -S_{234}C_5C_6 + C_{234}C_6 & 0 \\ C_1(C_{234}S_5) + S_1C_5 & S_1(C_{234}S_5) - C_1C_5 & S_{234}S_5 & 0 \\ C_1(C_{234}a_4 + C_{23}a_3 + C_2a_2) & S_1(C_{234}a_4 + C_{23}a_3 + C_2a_2) & S_{234}a_4 + S_{23}a_3 + S_2a_2 & 1 \end{bmatrix}^T
 \end{aligned} \tag{3.4}$$

Cinemática inversa

A pesar de que el análisis de la cinemática directa de un manipulador es útil, en la práctica, es más recurrente el problema de determinar el valor de cada variable de las articulaciones, dada una posición y orientación del efector final. Aunque existen varios métodos para resolver el problema de la cinemática inversa, como se trata de un manipulador articulado de 6-DOF con un brazo y una muñeca, resulta más sencillo utilizar el método de *desacoplamiento cinemático*, el cual, consiste en separar (desacoplar) el problema de la cinemática inversa en dos partes: cinemática inversa de la posición y cinemática inversa de la orientación.

Partiendo del modelo de matriz de transformación homogénea:

$$H = \begin{bmatrix} R & d \\ 0^T & 1 \end{bmatrix} \tag{3.5}$$

donde $R = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}$: matriz de orientación , $d = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$: vector de posición;

es posible expresar a la matriz (3.4) como:

$${}^0T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.6)$$

Para obtener los ángulos, se premultiplica a (3.4) y (3.6) por la matriz ${}^0A_1^{-1}$:

$${}^0A_1^{-1} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0A_1^{-1} [{}^0T_6] = {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6 \quad (3.7)$$

$$\begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6$$

$$= \begin{bmatrix} n_x C_1 + n_y S_1 & o_x C_1 + o_y S_1 & a_x C_1 + a_y S_1 & p_x C_1 + p_y S_1 \\ n_z & o_z & a_z & p_z \\ n_x S_1 - n_y C_1 & o_x S_1 - o_y C_1 & a_x S_1 - a_y C_1 & p_x S_1 - p_y C_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_{234} C_5 C_6 - S_{234} S_6 & -C_{234} C_5 C_6 - S_{234} C_6 & C_{234} S_5 & C_{234} a_4 + C_{23} a_3 + C_2 a_2 \\ S_{234} C_5 C_6 + C_{234} S_6 & -S_{234} C_5 C_6 + C_{234} C_6 & S_{234} S_5 & S_{234} a_4 + S_{23} a_3 + S_2 a_2 \\ -S_5 C_6 & S_5 S_6 & C_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.8)$$

Del elemento 3,4 de (3.8)³:

$$\theta_1 = \tan^{-1} \left(\frac{p_y}{p_x} \right) \quad y \quad \theta_1 = \pi + \tan^{-1} \left(\frac{p_y}{p_x} \right) \quad (3.9)$$

De los elementos 1,4 y 2,4 de (3.8):

$$\theta_3 = \tan^{-1} \left(\frac{S_3}{C_3} \right) \quad (3.10)$$

³ Para una discusión más detallada sobre la matemática inmersa en la obtención de las ecuaciones de la cinemática inversa, véase [12].

Debido a que las articulaciones 2, 3 y 4 son paralelas entre sí, la premultiplicación de las matrices ${}^1A_2^{-1}$ y ${}^2A_3^{-1}$ no entrega resultados útiles. A continuación, se debe premultiplicar por las inversas de 0A_1 hasta 3A_4 :

$${}^3A_4^{-1} {}^2A_3^{-1} {}^1A_2^{-1} {}^0A_1^{-1} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^3A_4^{-1} {}^2A_3^{-1} {}^1A_2^{-1} {}^0A_1^{-1} [{}^0T_6] = {}^4A_5 {}^5A_6$$

$$\begin{bmatrix} C_{234}(C_1n_x + S_1n_y) + & C_{234}(C_1o_x + S_1o_y) + & C_{234}(C_1a_x + S_1a_y) + & C_{234}(C_1p_x + S_1p_y) + \\ & +S_{234}n_z & +S_{234}o_z & S_{234}p_z - C_{34}a_2 - C_4a_3 - a_4 \\ C_1n_y - S_1n_x & C_1o_y - S_1o_x & C_1a_y - S_1a_x & 0 \\ -S_{234}(C_1n_x + S_1n_y) + & -S_{234}(C_1o_x + S_1o_y) + & -S_{234}(C_1a_x + S_1a_y) + & -S_{234}(C_1p_x + S_1p_y) + \\ & +C_{234}n_z & +C_{234}o_z & +C_{234}p_z + S_{34}a_2 + S_4a_3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_5C_6 & -C_5S_6 & S_5 & 0 \\ S_5C_6 & -S_5S_6 & -C_5 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(3.11)

Del elemento 3,3 de (3.11):

$$\theta_{234} = \tan^{-1} \left(\frac{a_z}{C_1a_x + S_1a_y} \right) \quad y \quad \theta_{234} = \pi + \tan^{-1} \left(\frac{a_z}{C_1a_x + S_1a_y} \right)$$
(3.12)

$$\theta_2 = \tan^{-1} \left[\frac{(C_3a_3 + a_2)(p_z - S_{234}a_4) - S_3a_3(p_x C_1 + p_y S_1 - C_{234}a_4)}{(C_3a_3 + a_2)(p_x C_1 + p_y S_1 - C_{234}a_4) + S_3a_3(p_z - S_{234}a_4)} \right]$$
(3.13)

$$\theta_4 = \theta_{234} - \theta_2 - \theta_3$$
(3.14)

De los elementos 1,3 y 2,3 de (3.11):

$$\theta_5 = \tan^{-1} \left[\frac{C_{234}(C_1a_x + S_1a_y) + S_{234}a_z}{S_1a_x - C_1a_y} \right]$$
(3.15)

Para obtener θ_6 se premultiplica (3.11) por ${}^4A_5^{-1}$ y se obtiene:

$$\theta_6 = \tan^{-1} \left[\frac{-S_{234}(C_1n_x + S_1n_y) + C_{234}n_z}{-S_{234}(C_1o_x + S_1o_y) + C_{234}o_z} \right]$$
(3.16)

3.3 Caracterización

Dos aspectos clave a considerar cuando se opera un manipulador robótico son la dirección de rotación de los ejes y su espacio de trabajo. Conocer a fondo ambos es necesario para evitar accidentes, tanto para el propio robot como para quien y lo que lo rodea.

3.3.1 Espacio de trabajo

Antes de calcular el espacio de trabajo, es prudente tener bien entendida la dirección de rotación y los valores reales de los rangos de los ejes del manipulador, ya que los que provee el fabricante resultaron erróneos, como se observa en la tabla de la Figura 3.3.

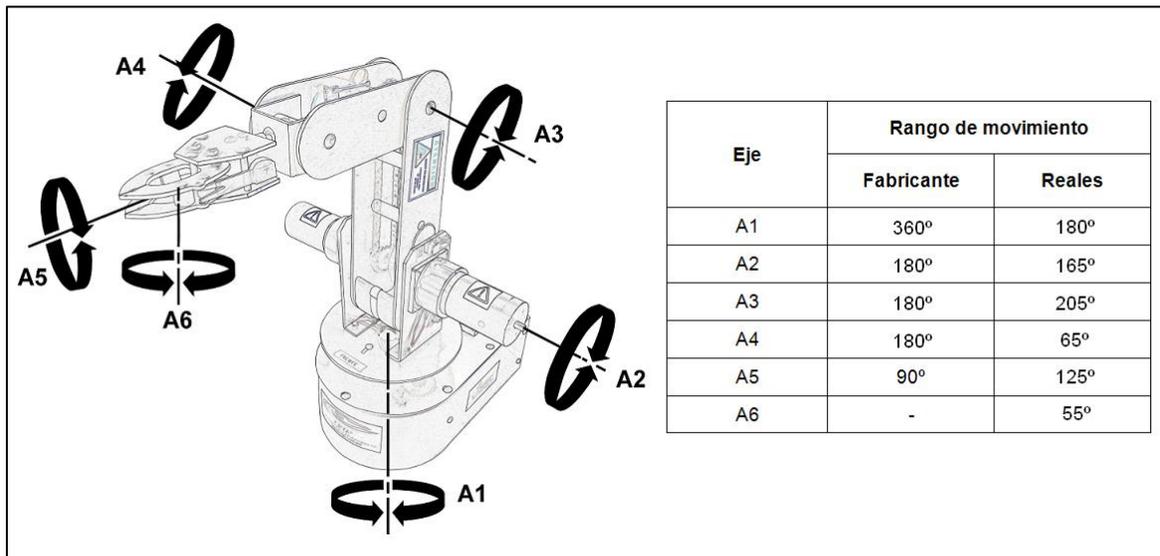


Figura 3.3 Dirección de rotación y rangos de movimiento de los ejes del manipulador robótico Labot Pro 5.

Calcular el espacio de trabajo del manipulador resulta vital para saber qué puntos se pueden alcanzar y para evitar los que puedan dañar la estructura del brazo. Para impedir esto último, se implementaron límites vía software en el manipulador utilizado, reduciendo los límites superiores e inferiores de los rangos de movimiento de la base, el hombro y el codo para impedir el choque entre estos últimos y con la mesa de trabajo. En la Figura 3.4 se puede ver el espacio de trabajo del robot.

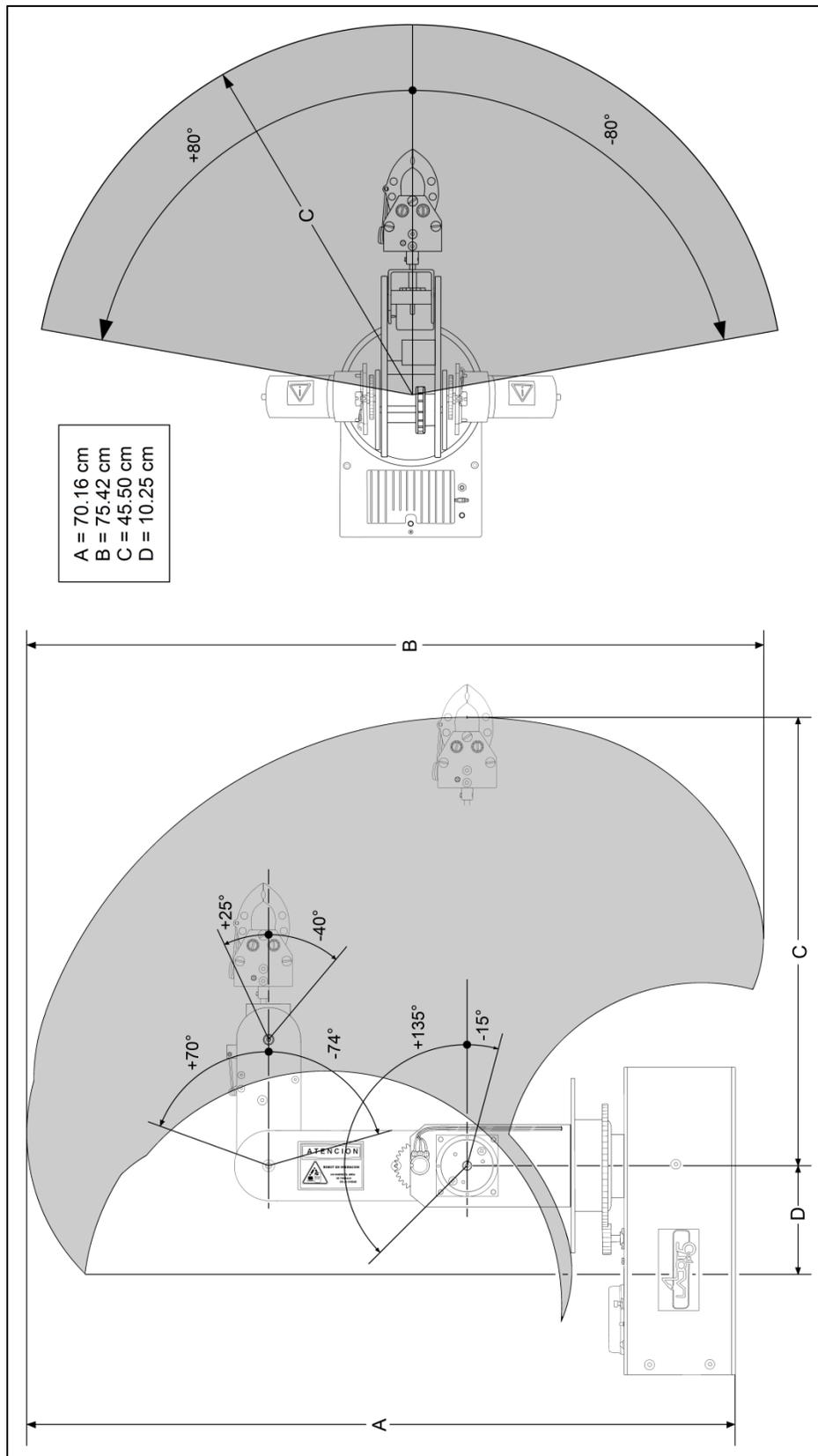


Figura 3.4 Espacio de trabajo restringido del manipulador robótico Labot Pro 5.

3.3.2 Características

Las características del manipulador robótico, mostradas en la Tabla 3.3, se obtuvieron por medio de la documentación del fabricante, excepto por el alcance máximo calculado experimentalmente; pero cabe hacer hincapié en que en capítulos posteriores se obtuvo experimentalmente el valor de la repetibilidad.

Labot Pro 5	
Alcance máximo	48.5 cm *
Carga máxima	800 g
Número de ejes	6, incluyendo la pinza
Tipo de control	Punto a punto
Peso	11 kg
Temperatura de operación	0 – 45 °C, al 40% de humedad
Repetibilidad	0.5 cm

Tabla 3.3 Características del manipulador robótico Labot Pro 5. * Valor obtenido experimentalmente.

3.3.3 Precisión, Repetibilidad y Reproducibilidad

Durante el uso constante del manipulador robótico, se detectó que al mandarlo a un mismo punto en repetidas ocasiones y sobre todo al reiniciarlo, éste no siempre caía en el mismo lugar, lo cual indica algunos problemas con la *repetibilidad* (0.5 cm, según el fabricante) pero graves problemas en la *reproducibilidad* ya que el robot no es capaz de reproducir adecuadamente una rutina si se le reinicia apagándolo o desconectándolo. Esto puede deberse a su parte mecánica o incluso a la lógica interna de su control.

Otro problema grave fue detectado cuando se observó que el manipulador no llega al mismo punto si parte desde arriba que si parte desde abajo, lo cual indica una baja en su *precisión*.

Para combatir estos problemas, es necesario implementar algo que mejore el desempeño del manipulador basándose en una relación cuyos valores sean constantes. Esta condición la cumplen bien los *voltajes de los potenciómetros de referencia*, ubicados en las articulaciones de los eslabones principales del manipulador (hombro y codo) y que sirven para crear una referencia con respecto a la horizontal. De esta manera, utilizando los voltajes de referencia y relacionándolos con sus respectivos ángulos, se pueden crear *Tablas de consulta o de lookup* para dejar de utilizar posiciones y comenzar a usar ángulos y voltajes para llevar al manipulador a un punto.

Para reforzar la idea de cambiar de posiciones a voltajes y ángulos, se detectó que en la interacción usuario–manipulador resulta más conveniente trabajar con los *ángulos* que trabajar con las *posiciones*; así por ejemplo, si queremos llevar al antebrazo a la mitad de un rango de 0° a 180° (equivalente a un rango de posiciones de 500 a 2500), es mejor y más entendible indicar en la interfaz gráfica “codo a 90° ” que “codo a 1500”.

Las tablas de lookup que relacionan los voltajes de referencia y los ángulos, se construyeron variando las posiciones del hombro y el codo y midiendo los voltajes de los potenciómetros y sus respectivos ángulos para obtener los resultados de la Tabla 3.4.

Sin embargo, como el manipulador solo acepta posiciones, se requieren tablas de lookup que relacionen *ángulos* con *posiciones*. Para esto, se realizó un estudio donde dado un ángulo, se mide el rango de voltajes y, utilizando las tablas de *voltajes vs. ángulos*, se infiere el valor de la posición que mejor se ajuste a dicho rango; véase los resultados en la Tabla 3.5. Cabe recordar que, como existe un problema donde la precisión depende de la dirección del movimiento, se crearon tablas cuando éste es hacia arriba y cuando es hacia abajo.

A continuación se detalla el procedimiento para obtener la tabla de lookup de *ángulos vs. posiciones*:

- Se lleva al brazo a la posición del extremo y poco a poco se mueve hacia arriba o hacia abajo hasta el otro extremo;
- En cada posición, se adquieren 500 muestras de voltaje y se obtienen sus máximos y mínimos;
- Para cada *ángulo* del rango de movimientos y tomando como referencia la tabla de *voltajes vs. ángulos*, se selecciona la *posición* que mejor se ajuste al rango de voltajes.

Un punto a recordar es que, debido a la estructura del manipulador utilizado, solamente se tiene acceso a los voltajes de referencia del *hombro* y del *codo*; sin embargo, con excepción de la base, éstas son las partes más importantes para trasladar al efector final a la mayoría de puntos del espacio de trabajo.

Otro punto muy importante para mejorar el desempeño del manipulador, es la utilización de un control de lazo cerrado usando el sistema de realimentación; sin embargo, este tema se aborda más adelante en esta tesis.

Hombro				Codo			
Voltaje [V]	Angulo [°]						
0.38	135	1.08	57	0.56	70	1.26	-4
0.41	132	1.1	54	0.58	68	1.27	-6
0.44	129	1.13	51	0.6	66	1.29	-8
0.47	126	1.16	48	0.62	64	1.31	-10
0.5	123	1.18	45	0.64	62	1.32	-12
0.53	120	1.21	42	0.67	60	1.34	-14
0.55	117	1.23	39	0.69	58	1.36	-16
0.58	114	1.26	36	0.71	56	1.38	-18
0.61	111	1.29	33	0.72	54	1.41	-20
0.64	108	1.31	30	0.75	52	1.42	-22
0.67	105	1.34	27	0.77	50	1.44	-24
0.7	102	1.36	24	0.79	48	1.46	-26
0.72	99	1.4	21	0.81	46	1.48	-28
0.75	96	1.42	18	0.83	44	1.5	-30
0.78	93	1.45	15	0.85	42	1.52	-32
0.8	90	1.48	12	0.86	40	1.54	-34
0.82	87	1.51	9	0.88	38	1.56	-36
0.84	84	1.53	6	0.9	36	1.58	-38
0.87	81	1.57	3	0.91	34	1.6	-40
0.89	78	1.6	0	0.93	32	1.62	-42
0.92	75	1.64	-3	0.95	30	1.64	-44
0.95	72	1.66	-6	0.97	28	1.66	-46
0.98	69	1.69	-9	0.99	26	1.68	-48
1	66	1.71	-12	1	24	1.7	-50
1.03	63	1.73	-15	1.02	22	1.71	-52
1.06	60			1.04	20	1.73	-54
				1.06	18	1.74	-56
				1.08	16	1.76	-58
				1.1	14	1.78	-60
				1.12	12	1.79	-62
				1.14	10	1.8	-64
				1.16	8	1.82	-66
				1.18	6	1.84	-68
				1.2	4	1.86	-70
				1.22	2	1.88	-72
				1.23	0	1.89	-74
				1.25	-2		

Tabla 3.4 Tablas de lookup que relacionan *voltajes* de los potenciómetros de referencia y *ángulos* con respecto a la horizontal.

Debido a que el empaque del módulo USB es de tipo SMD, se implementó en la tarjeta impresa de la Figura 3.5. La conexión del manipulador a la tarjeta se hace con un cable desde los pines Rx, Tx y GND del conector DB9 hembra; y la conexión de la tarjeta a la computadora se hace con un cable USB Tipo A-Tipo B.

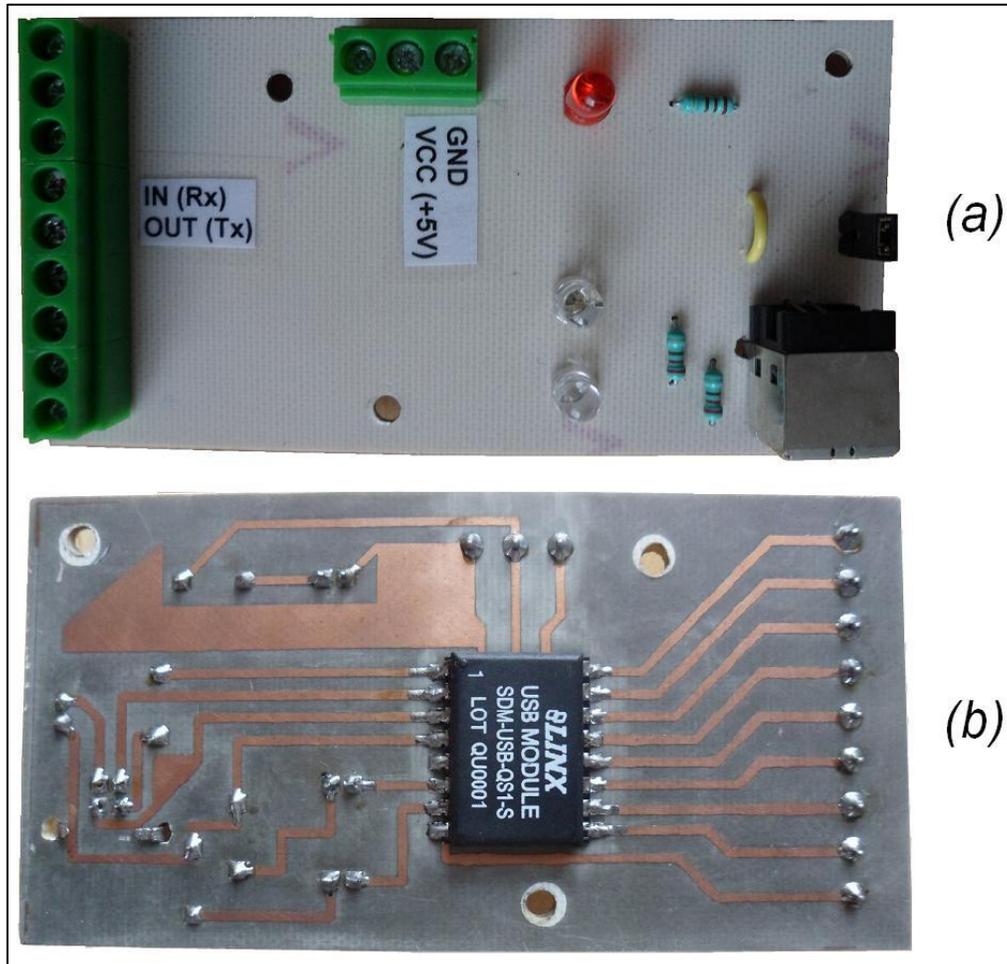


Figura 3.5 Tarjeta de conversión USB-RS232 basada en el módulo USB SDM-USB-QS-S. (a) Vista superior. (b) Vista inferior.

La tarjeta de conversión USB-RS232 se diseñó con las siguientes características:

- Auto alimentación desde la PC;
- Salida de +5 volts para alimentar otro circuito;
- LEDs testigos de transmisión y recepción;
- Los pines que no se utilizan para este proyecto, se conectaron a bornes con tornillo para que se encuentren disponibles para otras aplicaciones.

3.5 Sistema de realimentación

Por la naturaleza misma del control punto a punto, aunque los puntos queden programados en el sistema de control, no existe forma, además de la visual, de asegurarse que el brazo alcance dichos puntos; esto provoca que al integrarlo en la celda sea necesario un operador humano, lo cual se contrapone totalmente al concepto de celda de manufactura flexible. Por lo tanto, como lo que se busca es aumentar el grado de automatización dentro de la celda, se requiere implementar una forma de *realimentación* entre el control por computadora y el manipulador para conocer el estado del sistema en cualquier momento de forma remota.

Utilizando el principio de realimentación o de lazo cerrado^[36], la primicia del sistema es transmitir el ángulo deseado (de la base, el hombro o del codo) al manipulador desde la computadora y medir el voltaje del *potenciómetro de referencia* (véase la Figura 3.8a), el cual corresponde a un ángulo de su respectivo eslabón con relación a la horizontal. Una vez que se obtienen los voltajes, éstos deben ser digitalizados y transmitidos de vuelta a la computadora para ser comparados con la posición deseada al comienzo y si es el caso, rectificarla.

Cabe recordar que, debido a la estructura del robot, solamente se tiene acceso a los potenciómetros de referencia del hombro y del codo. El diagrama de bloques del sistema de realimentación se muestra en la Figura 3.6.

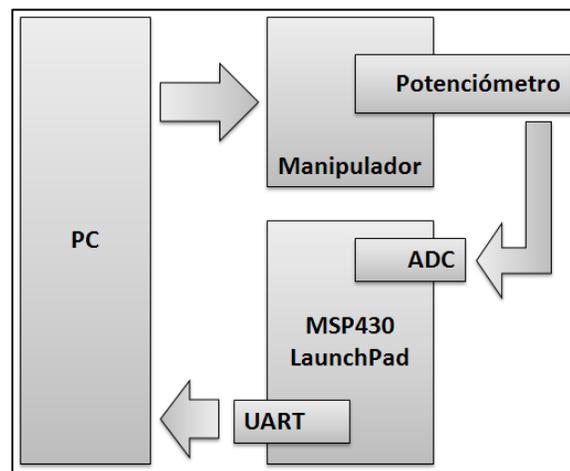


Figura 3.6 Diagrama de bloques del sistema de realimentación.

La primer parte del diseño del sistema de realimentación consiste en usar la PC para dar una orden al manipulador que contenga el ángulo que se desea que adopte el eslabón correspondiente al hombro o al codo. Para esto se usó el software del fabricante llamado “*Robodroid Software*”, donde es necesario escribir un número entero entre 500 y 2500 que equivale a una posición, y por ende a un ángulo con respecto a la horizontal, de uno de los eslabones del manipulador.

Para tomar los voltajes, digitalizarlos y enviarlos de vuelta a la PC, se utilizó una tarjeta de desarrollo experimental MSP-EXP430G2 LaunchPad^[37] basada en un microcontrolador de la familia MSP430x2xx de la empresa Texas Instruments. La principal razón por la que se seleccionó esta tarjeta fue que por un costo bajo brinda todo el hardware y software necesario para desarrollar un gran número de aplicaciones sin necesidad de otros elementos. Otra razón fue que la interfaz USB, además de servir para programar y depurar, proporciona también una conexión UART⁴ a 9600 baudios directa a la PC.

Algunas de las características de la tarjeta MSP-EXP430G2 LaunchPad son:

- Soporta microcontroladores de 14 y 20 pines de las familias MSP430G2xx y MSP430F20xx;
- Dos LEDs indicadores integrados para realimentación visual;
- Dos push-buttons para realimentación con el usuario y reset del dispositivo;
- Cristal de 32.768 kHz para otras aplicaciones.

El microcontrolador utilizado es el MSP430G2553 y su principal ventaja radica en que puede usarse para aplicaciones de ultra-baja potencia^[38] cambiando sus modos de operación, véase la Tabla 3.6. Para minimizar el consumo de energía, dichos modos de operación pueden apagar el CPU y los relojes del sistema; esto último provoca que el o los periféricos que utilicen tal reloj se deshabiliten y vuelvan a funcionar hasta salir de ese modo. Para salir de los modos de ultra-baja potencia, es necesario que se presente una interrupción.

Modo	Estado del CPU y Relojes	
	Activos	Inactivos
Activo	CPU y todos los relojes	-
LPM0	SMCLK, ACLK	CPU, MCLK
LPM1	SMCLK, ACLK	CPU, MCLK, DCO
LPM2	ACLK	CPU, MCLK, SMCLK, DCO
LPM3	ACLK	CPU, MCLK, SMCLK, DCO, generador de DC
LPM4	-	Todos los relojes

Tabla 3.6 Modos de operación de ultra-baja potencia. MCLK: Master Clock; SMCLK: Sub-System Master Clock; ACLK: Auxiliary Clock; DCO: Digitally Controlled Oscillator.

⁴ Universal Asynchronous Receiver-Transmitter (Transmisor-Receptor Asíncrono Universal).

Debido a que el microcontrolador funciona con un voltaje máximo de 3.6 Volts^[39], y los potenciómetros de referencia del manipulador utilizan voltajes de 0 a 5 Volts, se implementaron divisores de voltaje antes de la entrada de los convertidores analógico-digitales como se ve en la Figura 3.8b.

El algoritmo programado en el microcontrolador consiste básicamente en esperar, en modo de bajo consumo de energía, hasta recibir un caracter desde la computadora solicitando la lectura de los voltajes del hombro y el codo para después digitalizarlos y transmitirlos de vuelta hacia la PC. En la Figura 3.7 se muestra el diagrama de flujo del algoritmo.

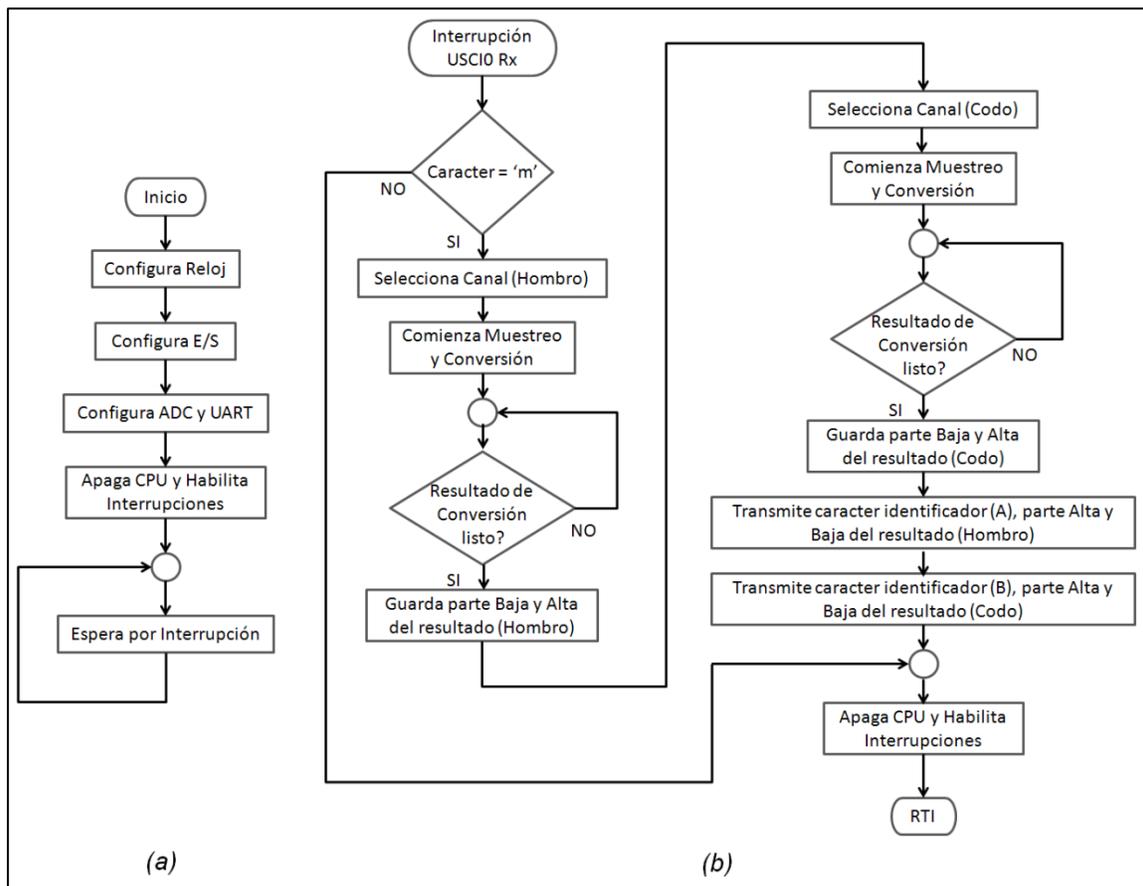


Figura 3.7 Diagrama de flujo del algoritmo para el microcontrolador. (a) Rutina principal. (b) Rutina de Servicio a Interrupción.

Para realizar las pruebas, la caracterización del manipulador o simplemente checar el estado del sistema de realimentación, se diseñó un programa que solicitara constantemente voltajes, al enviar un caracter hacia el microcontrolador, y que después recibiera y desplegara dichos voltajes; creándose una especie de “voltímetro virtual” como se muestra en la Figura 3.8c.

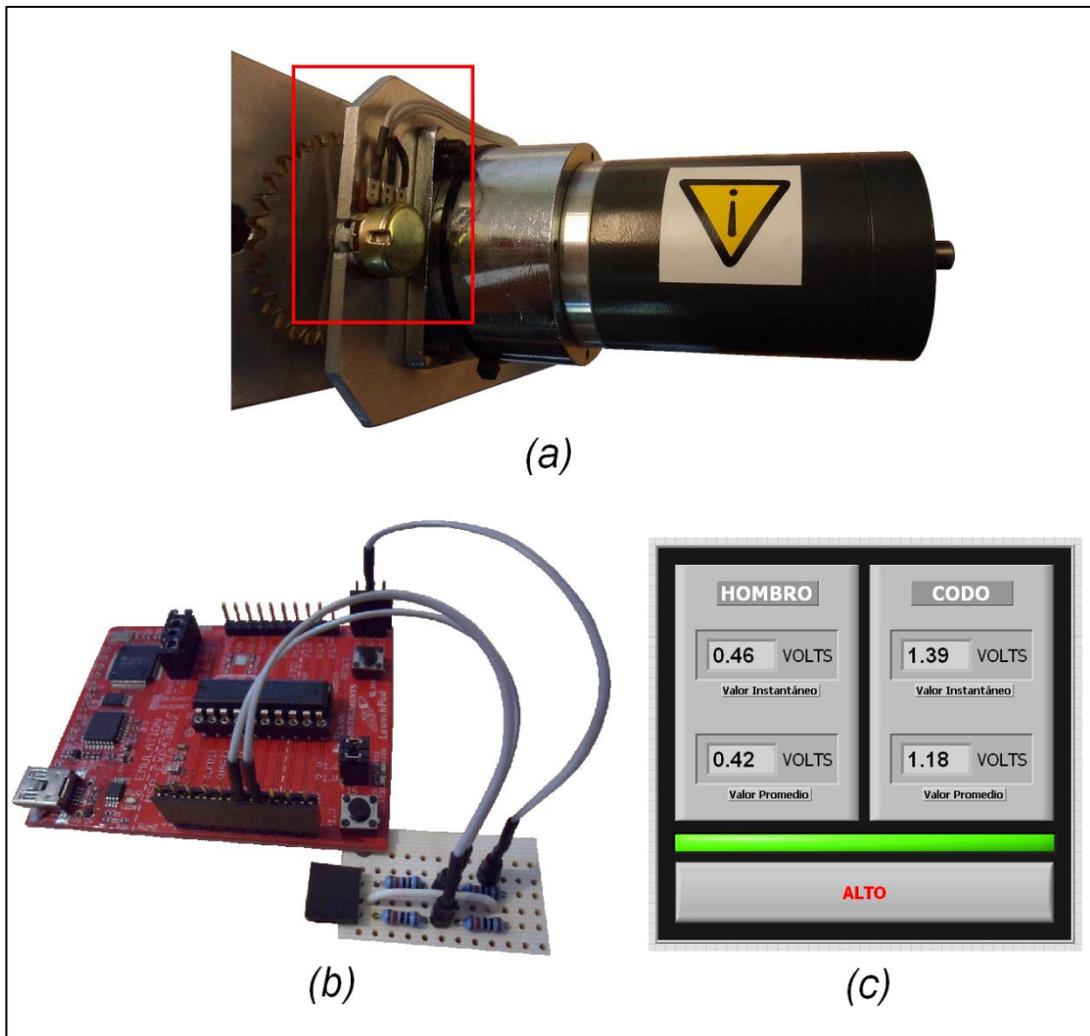


Figura 3.8 Elementos del sistema de realimentación. (a) Potenciometro de referencia. (b) Microcontrolador. (c) Voltímetro virtual.

3.6 Control manual

Como se menciona en los objetivos, el control del robot se hará por medio de dos maneras: *manual* y *por voz*.

El control manual consiste en manejar todos los aspectos del brazo desde una PC; desde mover todas sus partes, detectar problemas en su funcionamiento hasta trabajar con rutinas (crearlas, editarlas, etc.).

3.6.1 Interfaz gráfica de usuario

Una vez que se implementó el sistema de realimentación y se hizo la caracterización del manipulador, fue preciso hacer una *interfaz gráfica de usuario* (*Graphical User Interface, GUI*) adecuada a las necesidades del proyecto en lugar de utilizar la de la Figura 3.9 que el fabricante incluye con la compra del manipulador, la que además de tener limitaciones para el proyecto, tiene problemas que no permiten que se controle correctamente el robot.

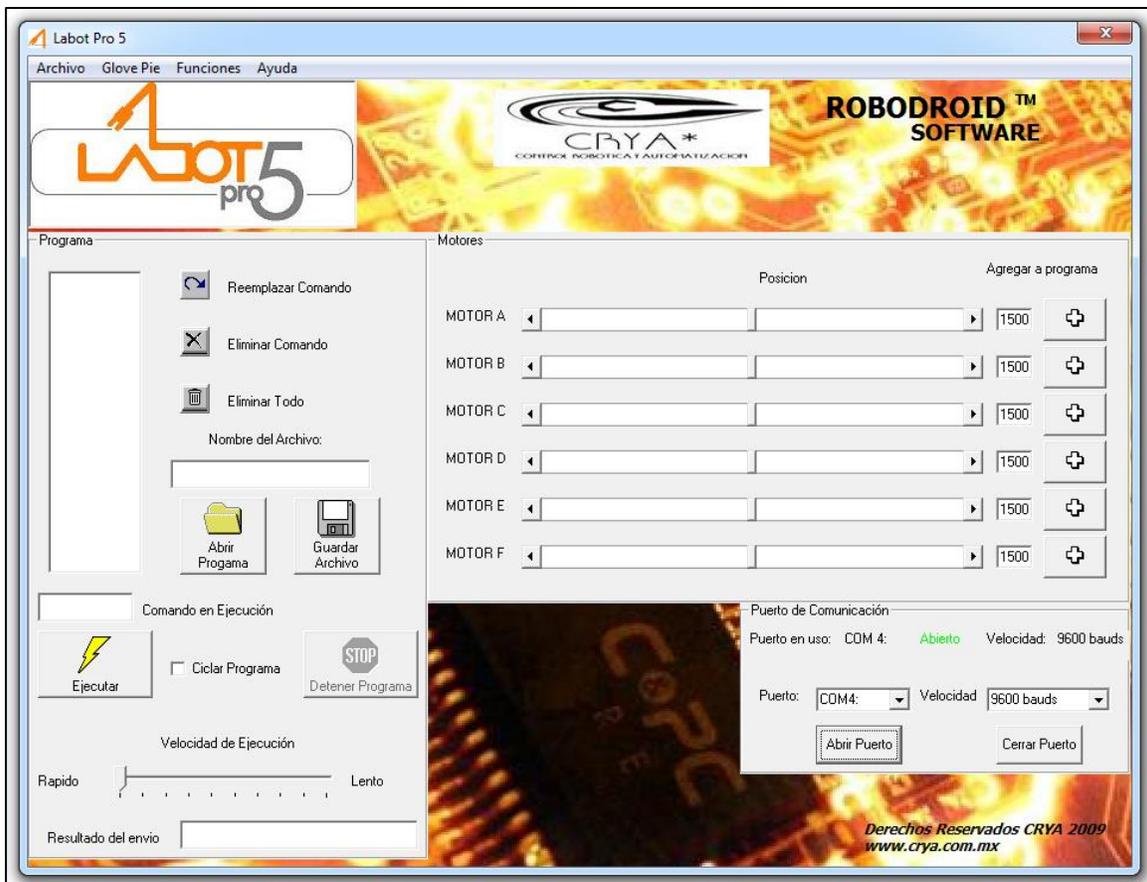


Figura 3.9 Interfaz gráfica del fabricante para el control del manipulador robótico Labot Pro 5.

La GUI mostrada en la Figura 3.10 fue programada dentro del entorno de desarrollo LabView, por ser un software especialmente diseñado para la instrumentación; además, ya que la interfaz gráfica consta en su mayoría de botones, lo mejor es utilizar una *programación por eventos*, en donde el cambio de estado de cada uno de los botones o controles en la GUI equivale a un *evento* y cada uno de los eventos debe contar con cierto código llamado *administrador*, el cual se ejecuta cuando ocurre un evento.

En la Tabla 3.7 se describen los elementos de la GUI del programa para controlar manualmente al manipulador “Control_Manual”. Donde se incluye cada uno de los 23 eventos, los respectivos administradores y sus correspondientes subrutinas asociadas.

#	Tipo	Evento	Administrador	
			Subrutinas asociadas	Descripción
1	Botón	Conectar	Conectar, Home	Conecta y lleva al manipulador a su posición inicial. Desconecta la comunicación entre la PC y el manipulador.
2	Indicador	-	-	Enciende cuando la PC está conectada al manipulador.
3	Botón	Stop	-	Desconecta la comunicación entre la PC y el manipulador y detiene la ejecución del programa.
4a	Control	Base	Base	Lleva a la base al ángulo seleccionado y enciende indicador si se encuentra en la posición inicial.
4b	Control	Hombro	Hombro	Lleva al hombro al ángulo seleccionado y enciende indicador si se encuentra en la posición inicial.
4c	Control	Codo	Codo	Lleva al codo al ángulo seleccionado y enciende indicador si se encuentra en la posición inicial.
4d	Control	Muñeca	Muñeca	Lleva a la muñeca al ángulo seleccionado.
4e	Control	Mano	Mano	Lleva a la mano al ángulo seleccionado.
4f	Control	Efector	Efector	Lleva al codo al ángulo seleccionado, mueve botón y enciende indicador si se encuentra Abierto o Cerrado.
5a	Botón	Home_Base	-	Lleva a la base a su posición inicial.
5b	Botón	Home_Hombro	-	Lleva al hombro a su posición inicial.
5c	Botón	Home_Codo	-	Lleva al codo a su posición inicial.
5d	Botón	Home_Muñeca	-	Lleva a la muñeca a su posición inicial.
5e	Botón	Home_Mano	-	Lleva a la mano a su posición inicial.

5f	Botón	Home_Efector	-	Lleva al efector a su posición inicial, enciende indicador y mueve botón a posición de Abierto.
6a	Indicador	-	-	Enciende cuando el hombro no llegó a la posición correcta en 5 segundos.
6b	Indicador	-	-	Enciende cuando el codo no llegó a la posición correcta en 5 segundos.
7	Botón	Home	Home	Lleva al manipulador a su posición inicial y enciende indicadores de posición de inicio.
8	Botón	Abrir/Cerrar_Efector	Abrir/Cerrar	Abre o cierra el efector final, mueve botón y enciende indicador si se encuentra Abierto o Cerrado.
9	Botón	Reset_Guardar	-	Reinicia la subrutina Guardar.
10	Botón	Guardar_Posicion	Guardar	Guarda la posición actual del manipulador en un archivo de texto seleccionado la primera vez que se guarda una posición y despliega aviso de confirmación.
11	Botón	Retardo	Retardo	Se habilita una vez que ha ocurrido el evento Guardar_posicion, guarda el retardo indicado en el control 11 en un archivo de texto y despliega aviso de confirmación.
12	Control	-	-	Permite seleccionar el valor del retardo (en segundos) que se guarda con el Botón 10.
13	Arreglo	-	-	Contiene la rutina que se puede ejecutar.
14	Botón	Cargar_Rutina	-	Carga una rutina seleccionada y despliega aviso de confirmación.
15	Botón	Editar_Rutina	-	Abre un archivo de texto utilizando un editor de textos y despliega aviso de confirmación.
16	Botón	Ejecutar_Rutina	Ejecutar, Base, Hombro_ejecutar, Codo_Ejecutar, Muñeca, Mano, Efector	Ejecuta la rutina cargada, enciende indicadores si se encuentra en la posición inicial y enciende indicadores si el hombro o el codo no llegaron correctamente a alguna posición de la rutina en 5 segundos.
17	Indicador	-	-	Enciende mientras se ejecuta una rutina.
18	Botón	Ciclar_Rutina	Ciclar, Base, Hombro_ejecutar, Codo_Ejecutar, Muñeca, Mano, Efector	Ejecuta continuamente la rutina cargada, enciende indicadores si se encuentra en la posición inicial y enciende indicadores si el hombro o el codo no llegaron correctamente a alguna posición de la rutina en 5 segundos.

Tabla 3.7 Descripción de los componentes que conforman la GUI del programa “Control_Manual”.

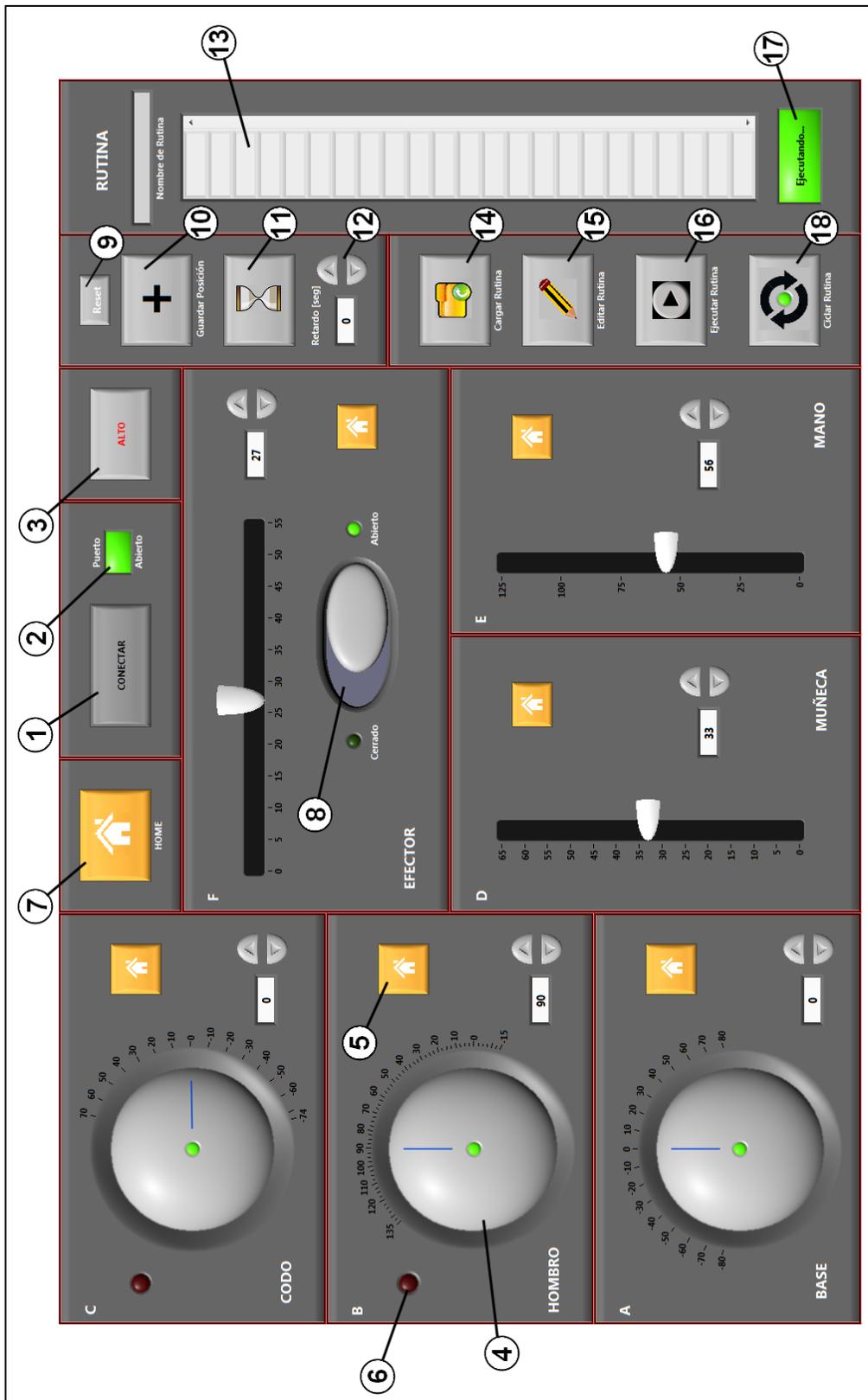


Figura 3.10 Interfaz gráfica propia para el control manual del manipulador robótico Labot Pro 5.

3.6.2 Algoritmo

Aunque en el programa hecho para controlar manualmente al manipulador “Control_Manual” se utilizan 15 subrutinas, solamente se explicarán las siete más significativas: Main, Conectar, Guardar, Retardo, Ejecutar, Hombro_Ejecutar y Codo_Ejecutar.

Rutina Principal (Main)

Como se observa en el diagrama de flujo de la Figura 3.11, la rutina carga las variables, las tablas de lookup y espera a que ocurra un evento.

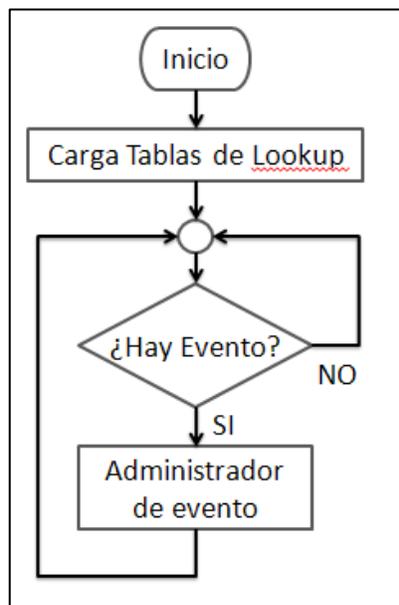


Figura 3.11 Diagrama de flujo de la rutina Principal.

Subrutina Conectar

Configura los parámetros de comunicación serial del puerto del manipulador y del sistema de realimentación (LaunchPad).

Baudaje:	9600 Bd
Bits de datos:	8
Paridad:	Ninguna
Bits de parada:	1.0
Control de flujo:	Ninguno

Subrutina Guardar (Posición)

Con el sistema de realimentación, se adquieren los voltajes del hombro y el codo. Solamente cuando es la primera vez que se ejecuta, es posible seleccionar el archivo de texto donde se guardarán las posiciones; sin embargo, si se requiere cambiar dicho archivo, basta presionar el botón “Reset” ubicado justo al lado del botón “Guardar Posición”.

➤ //Posición	Comentario invisible para la ejecución
➤ A	Base
➤ 0000	Ángulo Base
➤ B	Hombro
➤ 0090	Ángulo Hombro
➤ 0080	Voltaje Hombro
➤ 0820	Posición Hombro
➤ C	Codo
➤ 0000	Ángulo Codo
➤ 0123	Voltaje Codo
➤ 1180	Posición Codo
➤ D	Muñeca
➤ 0033	Ángulo Muñeca
➤ E	Mano
➤ 0056	Ángulo Mano
➤ F	Efactor
➤ 0027	Ángulo Efactor

Subrutina Retardo

Solamente disponible una vez que se ejecutó la subrutina Guardar.

➤ //Retardo	Comentario invisible para la ejecución
➤ T	Retardo
➤ 0001	Segundos de retardo

Subrutina Ejecutar

Ejecuta la rutina que esté cargada en memoria de la forma en que se indica en el diagrama de flujo de la Figura 3.12.

Es muy importante resaltar que esta subrutina solamente se utiliza cuando se ejecuta una rutina almacenada y no cuando se manejan los eslabones del robot en tiempo real.

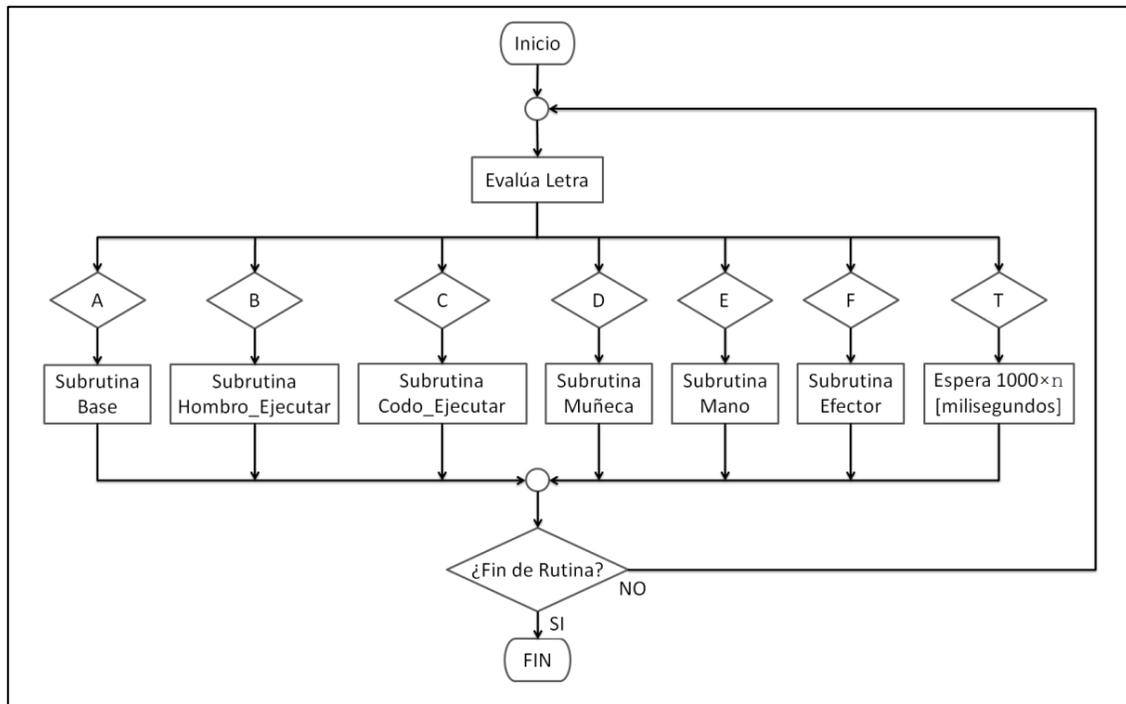


Figura 3.12 Diagrama de flujo de la subrutina Ejecutar.

Subrutina Hombro_Ejecutar y Codo_Ejecutar

Estas rutinas, análogas para el hombro y el codo, son una parte muy importante del programa debido a que implementan el control que, junto con el uso de las tablas de lookup y el cambio de posiciones a ángulos, ayuda a mejorar los problemas de precisión, repetibilidad y reproducibilidad.

El control, representado en la Figura 3.13, consiste en utilizar al sistema de realimentación para conocer el *voltaje de referencia real*, compararlo con el *voltaje (deseado)* almacenado en la rutina a ejecutar y obtener un error, el cual finalmente se compensará moviendo hacia un lado o hacia el otro el eslabón del robot correspondiente. A este control también se le integra una medida de seguridad que evita que, si por cualquier razón el eslabón no llega a la posición debida, después de 5 segundos se detiene el control y se enciende un indicador de aviso.

Aunque el control aplicado puede parecer un control en lazo cerrado, no existe una ejecución continua que busque que el error sea cero sino que solamente se aplica una vez la compensación de éste debido a que el brazo es muy lento en sus movimientos y tardaría demasiado tiempo en estabilizarse.

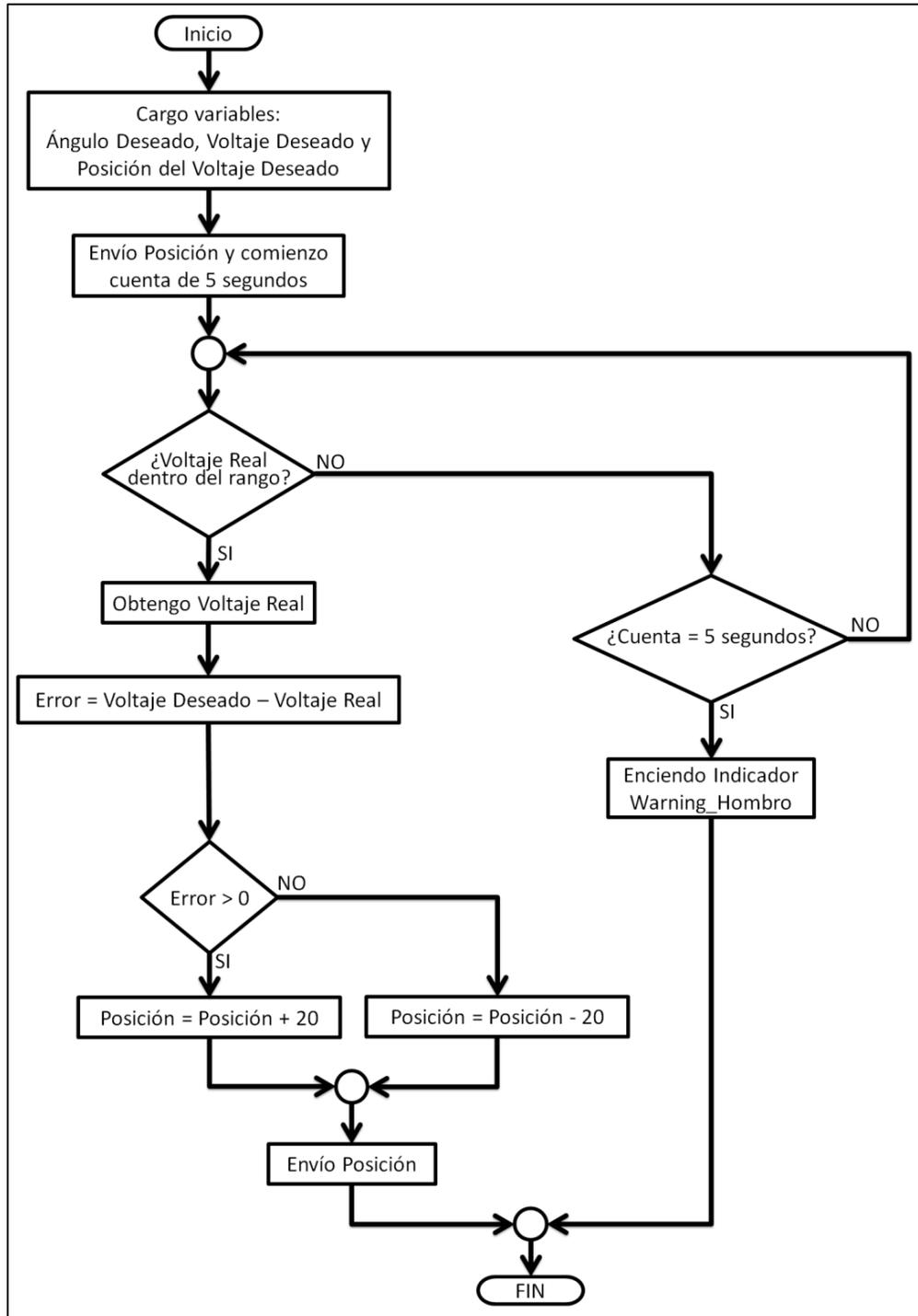


Figura 3.13 Diagrama de flujo de la subrutina Hombro_Ejecutar.

3.6.3 Ejemplo de uso

Para comprender mejor la forma en que funciona la GUI para el control manual del manipulador, a continuación se presenta un ejemplo completo sobre el manejo en tiempo real del manipulador para crear una rutina y después ejecutarla. El objetivo del ejemplo será crear una rutina que lleve al brazo desde la posición inicial hasta tomar con el efector final una pelota ubicada dentro del espacio de trabajo que se ve en la Figura 3.14a y por último soltar la pelota y regresar a la posición inicial:

1. Ejecutar el programa “Control_Manual”;
2. Presionar el botón “Conectar” para establecer la comunicación con el robot y llevarlo a la posición inicial;
3. Presionar el botón “Guardar Posición” para almacenar la posición en el archivo indicado;
4. Usar los controles de los eslabones (base, hombro, codo, muñeca y mano) para llevar al manipulador a la posición de la Figura 3.14b;
5. Presionar el botón “Guardar Posición” para almacenar la posición;
6. Seleccionar “1” con el control y presionar el botón “Retardo” para almacenar un retardo de 1 segundo;
7. Mover el efector a la posición de “Cerrado” para asir la pelota;
8. Seleccionar “1” con el control y presionar el botón “Retardo” para almacenar un retardo de 1 segundo;
9. Presionar el botón “HOME” para regresar a la posición inicial, abrir el efector y liberar la pelota;
10. Seleccionar “1” con el control y presionar el botón “Retardo” para almacenar un retardo de 1 segundo.

Para ejecutar la rutina recién creada simplemente basta con presionar el botón “Cargar Rutina”, seleccionar el archivo que contiene la rutina y presionar el botón “Ejecutar Rutina”.

Si se requiere editar la rutina creada, basta con presionar el botón “Editar Rutina” para que se abra automáticamente el editor de texto básico de Windows (Bloc de notas).

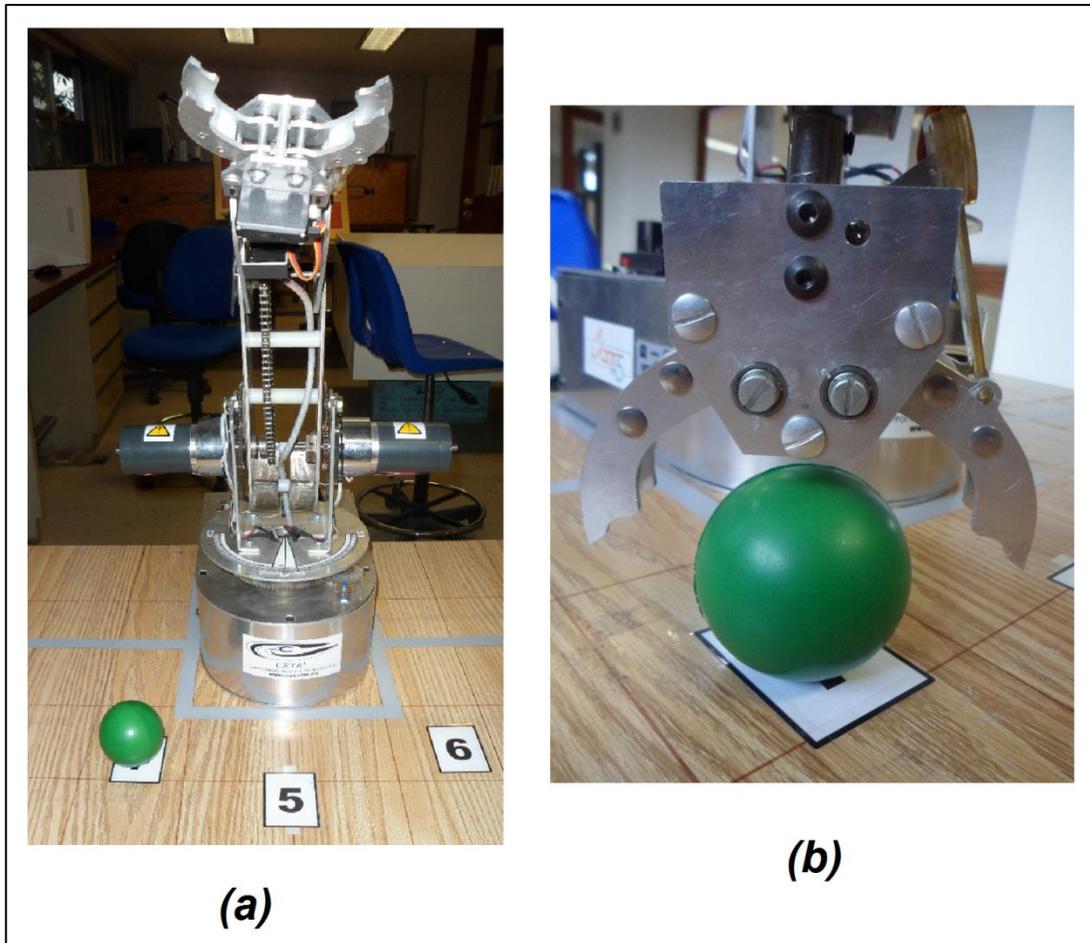


Figura 3.14 Ejemplo de uso de la GUI para el control manual. (a) Ubicación de la pelota dentro del espacio de trabajo. (b) Posición para asir la pelota.

3.7 Control por voz

Aparte del control manual, el manipulador robótico estará controlado por voz. El control por voz consiste tanto en manejar el propio software como en ejecutar rutinas previamente almacenadas usando el control manual.

3.7.1 Reconocimiento de voz

Debido a que crear un software que reconozca la voz implica demasiado trabajo y tiempo, la mejor opción es utilizar una paquetería comercial especializada en el reconocimiento de voz. El “Dragon Naturally Speaking” de la empresa Nuance, es uno de los programas líderes en el mundo sobre este tipo de reconocimiento; de ahí que fuera la elección para usar en el proyecto.

Sistema de reconocimiento de voz basado en Modelos Ocultos de Markov

El programa Dragon Naturally Speaking, utiliza la arquitectura de un sistema de reconocimiento basado en Modelos Ocultos de Markov (*Hidden Markov Models, HMM*)^[40] mostrada en la Figura 3.15:

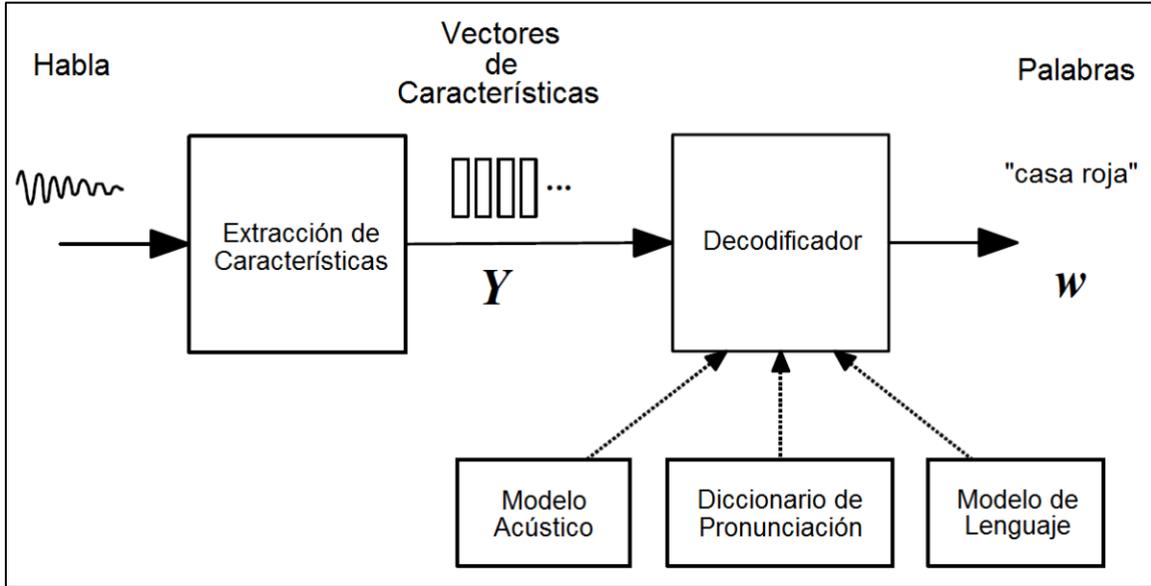


Figura 3.15 Arquitectura de un sistema de reconocimiento de voz basado en Modelos Ocultos de Markov.

En dicha arquitectura, la señal de entrada (habla) del micrófono se convierte en una secuencia de vectores acústicos de tamaño fijo $Y_{1:T} = y_1, \dots, y_T$ en un proceso llamado *extracción de características*. Entonces el decodificador trata de encontrar una secuencia de palabras $w_{1:L} = w_1, \dots, w_L$ que sea la más probable que haya generado Y :

$$\hat{w} = \arg_w \max\{P(w|Y)\}. \quad (3.17)$$

Sin embargo, como esto es difícil de modelar, se utiliza un equivalente:

$$\hat{w} = \arg_w \max\{P(Y|w)P(w)\}. \quad (3.18)$$

La probabilidad $P(Y|w)$ está determinada por un modelo acústico y $P(w)$ por un modelo de lenguaje. La unidad básica del sonido representado por el modelo acústico es el *fonema*. En particular, para el idioma español se requieren poco más de 20 fonemas y para el inglés 40.

Extracción de características

La primera etapa de un sistema de reconocimiento de voz consiste en extraer los componentes que identifiquen el contenido lingüístico de la señal de audio y a su vez descartar toda la demás información que contenga ruido, emoción, etc. De esta manera, lo que se busca es tener una representación compacta y precisa del fonema producido.

Típicamente, los vectores acústicos se calculan cada 25 ms utilizando algún esquema de codificación; de los cuales uno de los más utilizados es el basado en *Coefficientes Cepstrales en las Frecuencias Mel* (MFCCs).

A grandes rasgos, los MFCCs se obtienen por medio de los siguientes pasos^[41]:

- Aplicar a cada vector acústico la *Transformada de Fourier Discreta (DFT)* para obtener su espectro de potencias (módulo al cuadrado de la DFT);
- Calcular un banco de 20 a 40 filtros triangulares cuyas frecuencias centrales estén linealmente espaciadas según la escala mel, cuya fórmula de conversión está indicada por

$$M(f) = 1125 \ln \left(1 + f/700 \right); \quad (3.19)$$

- Obtener el logaritmo de las potencias en cada una de las frecuencias mel;
- Aplicar a la lista de logaritmos la *Transformada de Coseno Discreta (DCT)*.

Modelo acústico^[42]

El modelo acústico del sistema de reconocimiento, se compone de las representaciones estadísticas de cada uno de los fonemas que forman un idioma, llamadas *Modelos Ocultos de Markov (HMMs)* y cabe notar, que cada fonema tiene su propio HMM.

Con lo anterior en mente, es posible decir que cada palabra w puede ser descompuesta en una secuencia de K_w sonidos básicos llamados *fonemas base*. Esta secuencia es llamada pronunciación $q_{1:K_w}^w = q_1, \dots, q_{K_w}$; pero, para permitir la posibilidad de que se puedan tomar múltiples pronunciaciones, la probabilidad $p(Y|w)$ sería:

$$p(Y|w) = \sum_Q p(Y|Q)P(Q|w), \quad (3.20)$$

donde la sumatoria abarca todas las secuencias de pronunciación válidas para w , Q es una secuencia particular de pronunciaciones,

$$P(Q|w) = \prod_{l=1}^L P(q^{(w_l)}|w_l), \quad (3.21)$$

y donde cada $q^{(w_l)}$ es una pronunciación válida para la palabra w_l . Cabe recalcar que en la práctica, solamente existen muy pocas pronunciaciones alternativas para cada w_l , lo que hace a la sumatoria fácilmente manejable.

En la Figura 3.16, se ilustra un HMM de tres estados que representa cada uno de los fonemas base q , dada una secuencia de vectores Y . La probabilidad de llevar a cabo una transición del estado s_i al estado s_j está dada por la probabilidad de transición $\{a_{ij}\}$. Al entrar a un estado, se genera un vector de características utilizando la distribución asociada con el estado al que se entra $\{b_j(\cdot)\}$.

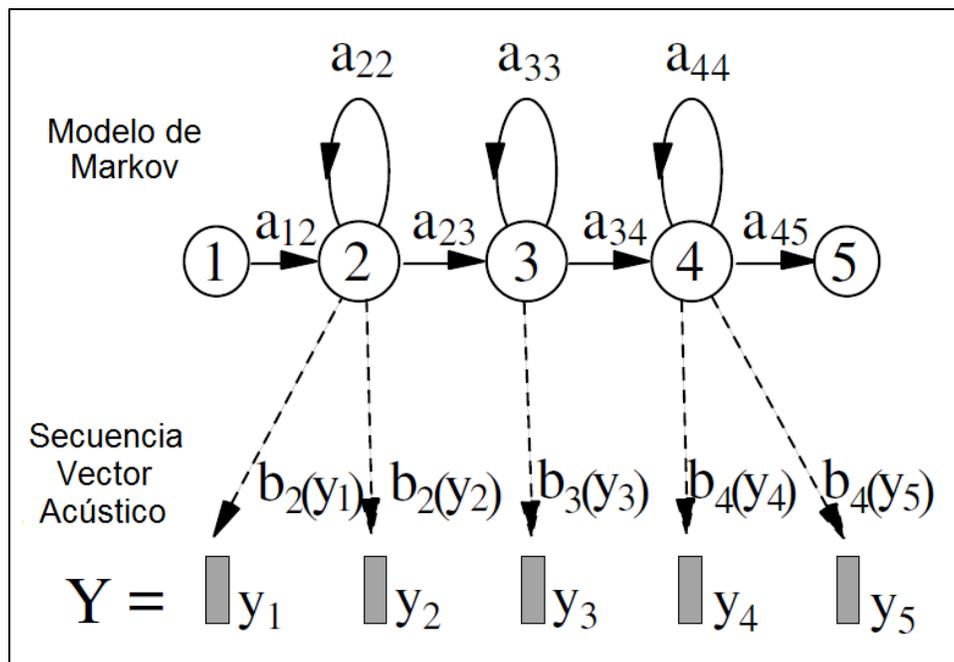


Figura 3.16 Modelo de fonema basado en HMM.

Una vez que se forma el HMM compuesto Q , formado al concatenar todos los fonemas base $q^{(w_1)}, \dots, q^{(w_L)}$ que lo constituyen, entonces la probabilidad acústica está dada por

$$P(Y|Q) = \sum_{\theta} p(\theta, Y|Q), \quad (3.22)$$

donde $\theta = \theta_0, \dots, \theta_{T+1}$ es una secuencia de estados a través del modelo compuesto Q ; θ_0 y θ_{T+1} corresponden a los estados de entrada y salida, que sirven para simplificar el proceso de concatenación.

Por último y, con base en un *diccionario de pronunciación*, se lleva a cabo la concatenación de los modelos de los fonemas para formar palabras.

Modelo de lenguaje^[43]

Dentro del decodificador del sistema de reconocimiento, el modelo de lenguaje brinda información sobre la probabilidad de que ocurra cierta palabra o secuencia de palabras.

La probabilidad de una secuencia de palabras $w = w_1, \dots, w_K$ requerida en (3.18), está dada por

$$P(w) = \prod_{k=1}^K P(w_k | w_{k-1}, \dots, w_1), \quad (3.23)$$

Para el reconocimiento de grandes vocabularios, la condición histórica de palabras en (3.23) usualmente es truncada a $N-1$ palabras para formar un modelo de lenguaje de *N-grama*

$$P(w) = \prod_{k=1}^K P(w_k | w_{k-1}, w_{k-2}, \dots, w_{k-N+1}), \quad (3.24)$$

donde típicamente N está en el rango de 2 a 4.

En el caso particular donde $N = 2$, el modelo recibe el nombre de bigrama; el cual, proporciona la probabilidad condicional de una palabra dada la palabra precedente. Por ejemplo, para la frase “tengo un auto rojo”, la probabilidad es calculada como:

$$P(\text{tengo, un, auto, rojo}) \approx P(\text{tengo} | \{s\})P(\text{un} | \text{tengo})P(\text{auto} | \text{un})P(\text{rojo} | \text{auto})P(\{s\} | \text{rojo})$$

Una vez que se tienen los datos, el decodificador busca a través de todas las secuencias de palabras posibles utilizando algoritmos de poda (*pruning*) para eliminar hipótesis improbables y así mantener la búsqueda manejable. Una vez que se alcanza el final de la frase, la secuencia de palabras más probable corresponde a la salida⁵.

⁵ Para una discusión más detallada sobre la matemática inmersa en un sistema de reconocimiento de voz basado en modelos ocultos de Markov, véase [40].

Siguiendo con la implementación del proyecto, además del software de reconocimiento de voz, es necesario tener un micrófono adecuado para la captura de voz desde la PC, para esto se seleccionó el modelo MIC-525 de la empresa mexicana Steren; el cual incorpora un filtro que elimina el ruido para tener mayor claridad en las grabaciones.

Ya instalado el programa Dragon versión 10.1, es necesario llevar a cabo la configuración inicial para que se optimicen los resultados de reconocimiento^[44]. Esta configuración consiste en dos pasos: configurar el micrófono para que funcione correctamente y entrenar al reconocedor para mejorar la precisión y entienda lo mejor posible el dictado.

La configuración del micrófono consiste en leer un texto para que la PC mida el volumen de voz mínimo que debe tener el usuario para un buen reconocimiento, así como la calidad del equipo que se usa. Antes de empezar el entrenamiento, debe elegirse el idioma y el país en que se encuentra, en este caso se habla español de México. A continuación, el entrenamiento consiste en leer varios textos, a la vez que el programa va marcando las palabras ya leídas o en su caso pida al usuario repetir la lectura para mejorar el porcentaje de éxito. Cabe recordar que en cualquier momento se puede volver al entrenamiento para mejorar los resultados.

Aunque la prestación principal del software de reconocimiento es convertir la voz humana en texto, también es posible que entienda comandos para controlar el sistema operativo de la PC donde se encuentra. Por ejemplo, es posible iniciar programas, abrir documentos y carpetas, abrir y cerrar menús, seleccionar botones y opciones e incluso mover el puntero del ratón.

Una vez configurado el programa, la forma de usarlo es posicionar el puntero del ratón en cualquier lugar donde sea posible introducir caracteres, p. ej. cualquier editor de texto, y comenzar a hablar directo al micrófono y a una distancia adecuada como se muestra en la Figura 3.17.

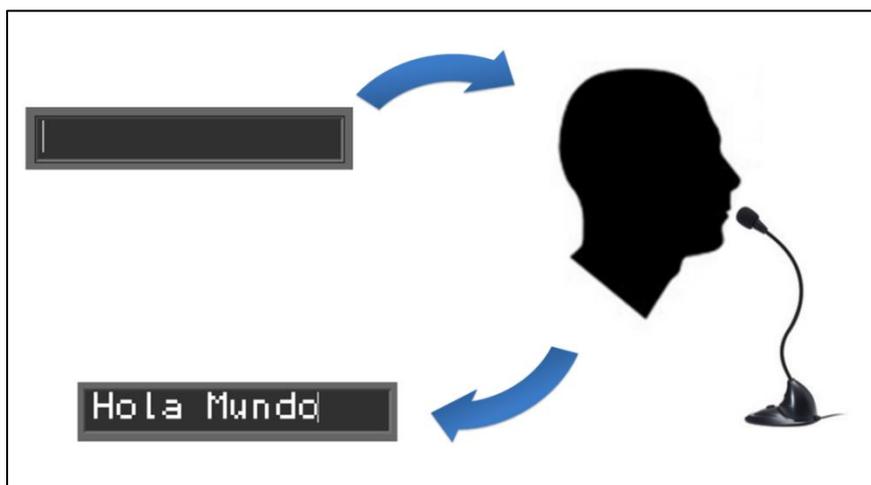


Figura 3.17 Forma de utilizar el software de reconocimiento de voz.

3.7.2 Interfaz gráfica de usuario

Una vez que se dominó el uso del software de reconocimiento de voz, es preciso hacer una interfaz gráfica de usuario (GUI) adecuada a las necesidades del proyecto. La GUI mostrada en la Figura 3.18, fue programada también dentro del entorno de desarrollo LabView y se utilizó una *programación estructurada* y no por eventos como en el caso de la desarrollada para el control manual. En la Tabla 3.8 se describen los elementos de la GUI del programa “Control_Voz”.

#	Tipo	Nombre	Descripción
1	Botón	Conectar	Conecta o Desconecta la comunicación entre la PC y el manipulador.
2	Indicador	Puerto Abierto	Enciende cuando la PC está conectada al manipulador.
3	Botón	Alto	Desconecta la comunicación entre la PC y el manipulador y detiene la ejecución del programa.
4	Control	Comando de voz recibido	Recibe el comando convertido desde el programa de reconocimiento de voz.
5	Indicador de Texto	Último comando recibido	Indica el comando último comando en el que se ejecutó alguna rutina.
6	Indicador de Texto	Estado	Indica el estado del sistema: Esperando Comando..., Desconectado, Conectado.
7	Indicador	Codo/Hombro	Enciende cuando el hombro/codo no llegó a la posición correcta en 5 segundos.
8	Indicador	LED Rojo	Indica cuando el sistema está ejecutando una rutina y no puede recibir nuevos comandos.
9	Indicador	LED Amarillo	Indica que el sistema está en fase de transición.
10	Indicador	LED Verde	El sistema está listo para recibir un comando.
11	Indicador	Ejecutar...	Enciende mientras se ejecuta una rutina.

Tabla 3.8 Descripción de los componentes que conforman la GUI del programa “Control_Voz”.

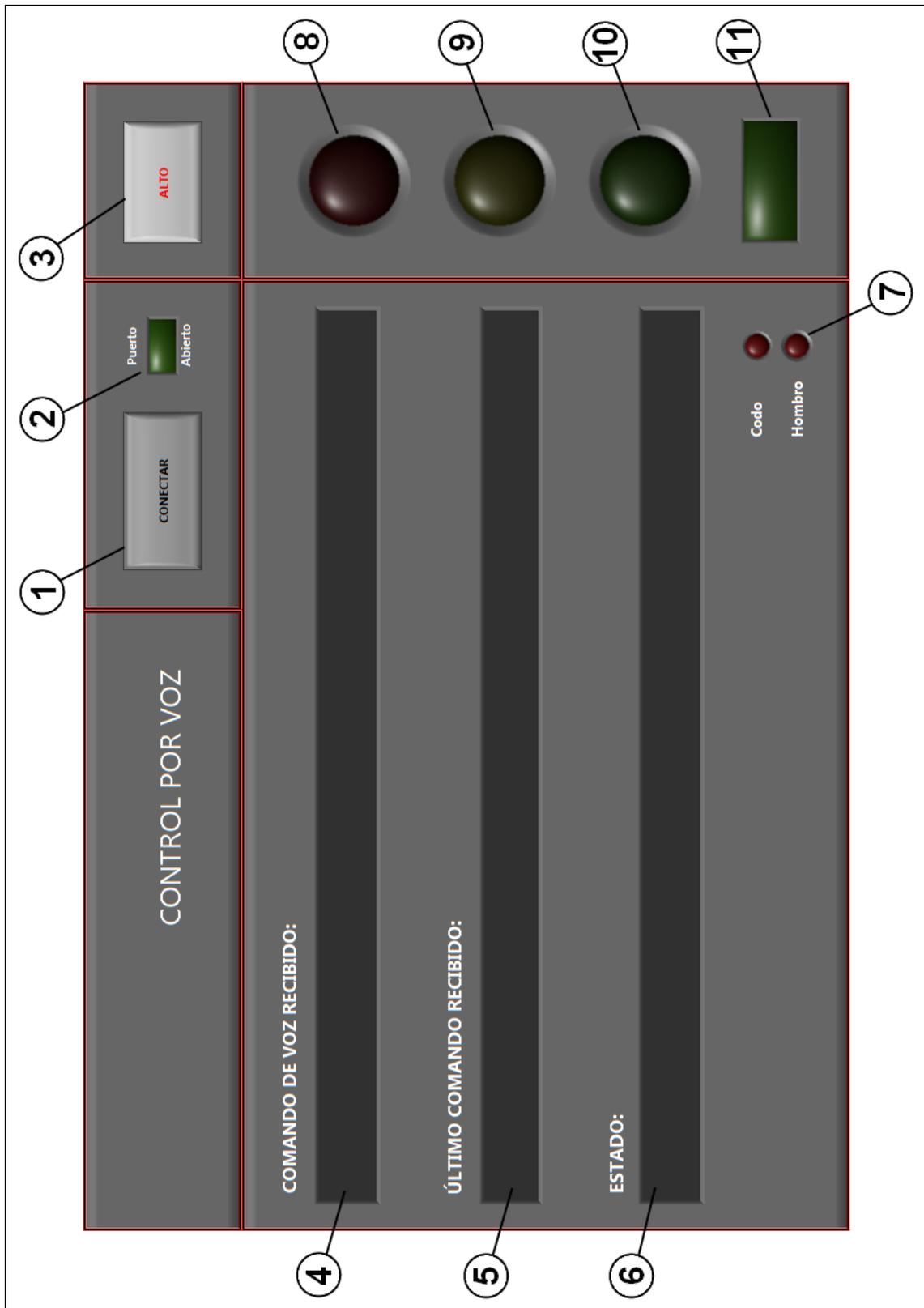


Figura 3.18 Interfaz gráfica para el control por voz del manipulador robótico Labot Pro 5.

Para controlar al manipulador con voz, se utilizaron los comandos descritos en la Tabla 3.9. En caso de ser diferentes a los comandos permitidos, p. ej., si el reconocimiento fue fallido, la misma GUI lo indicará oportunamente.

Comando	Descripción
CONECTA/R	Conecta la comunicación entre la PC y el manipulador.
DESCONECTA/R	Conecta la comunicación entre la PC y el manipulador.
ALTO/SALIR	Detiene el programa.
BRAZO	Indica al manipulador que en los próximos 2 segundos, espere un comando.
INICIO	Lleva al manipulador a la posición de inicio.
POSICIÓN X	Ejecuta la rutina que lleva al manipulador a la posición X del espacio físico de trabajo. ⁶
LEVANTA/ARRIBA	Eleva solamente el brazo.
ABRE/ABRIR PINZA	Abre el efector final al máximo.
CIERRA/CERRAR PINZA	Cierra el efector al mínimo.
RUTINA X	Ejecuta la rutina con el número X.

Tabla 3.9 Descripción de los comandos utilizados en el control por voz.

3.7.3 Algoritmo

Las Figuras 3.19, 3.20 y 3.21 forman el diagrama de flujo de la rutina del programa “Control_Voz”.

⁶ Véase la Figura 3.23.

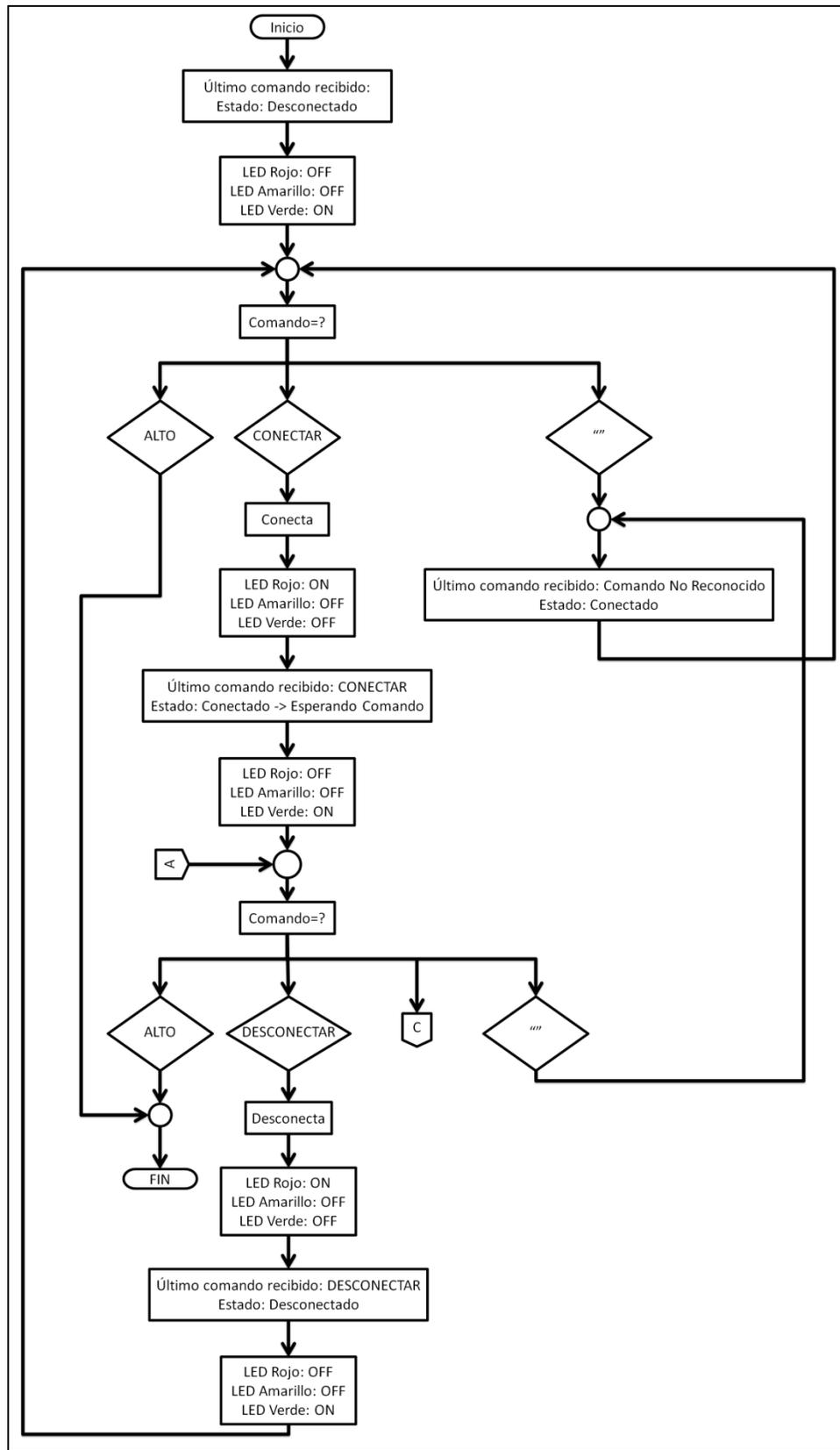


Figura 3.19 Diagrama de flujo de la GUI del control por voz.

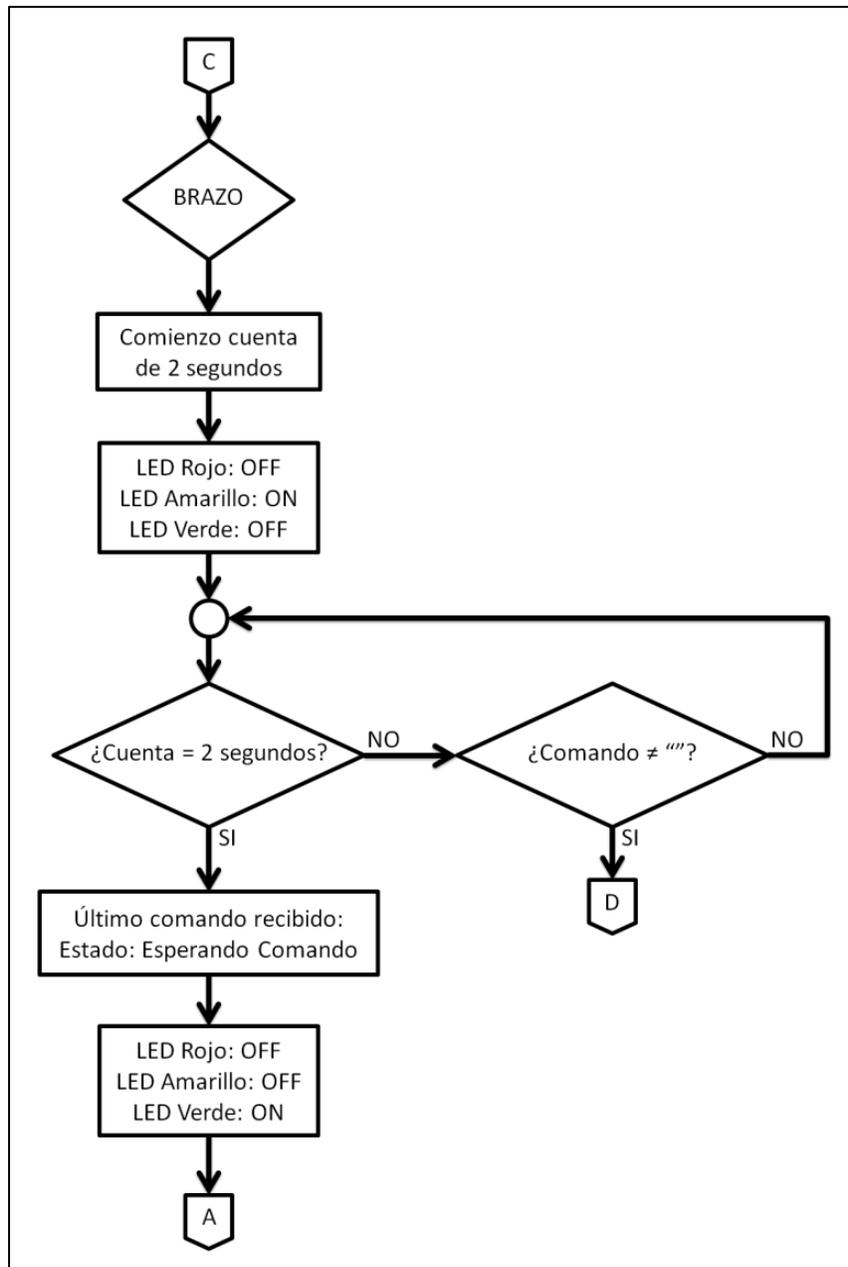


Figura 3.20 Diagrama de flujo de la GUI del control por voz (Continuación).

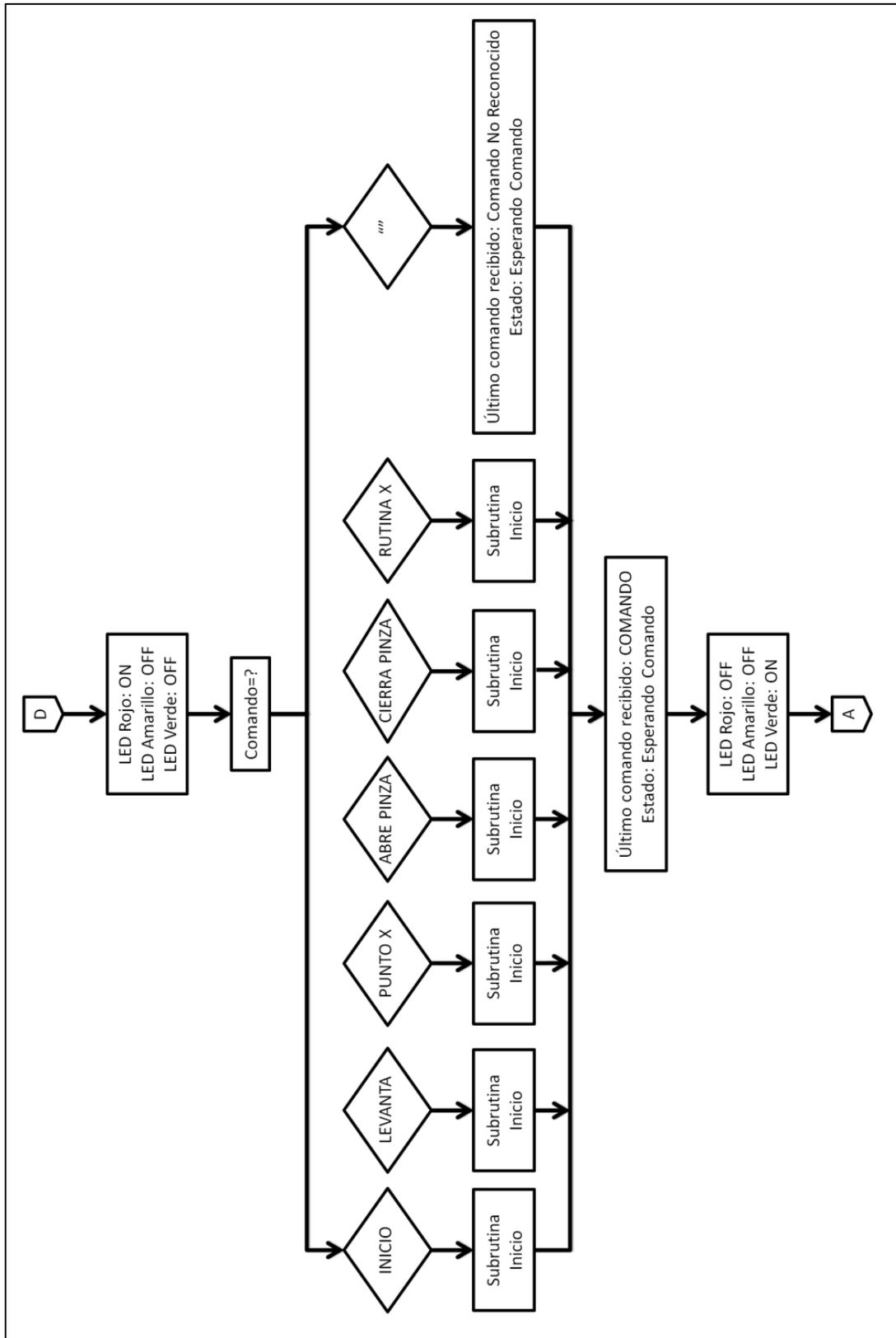


Figura 3.21 Diagrama de flujo de la GUI del control por voz (Continuación).

3.7.4 Ejemplo de uso

Para comprender mejor la forma en que funciona la GUI para el control por voz del manipulador, a continuación se presenta un ejemplo completo sobre el manejo del manipulador a base de ejecutar rutinas creadas con el control manual⁷. El objetivo del ejemplo será hacer que el robot transporte una pelota ubicada en la posición 7 hasta la posición 6 y regrese a su posición inicial.

1. Ejecutar el programa “Control_Voz”;
2. Decir el comando “CONECTAR” fuerte y claro ante el micrófono;
3. Decir el comando “BRAZO” fuerte y claro ante el micrófono;
4. Decir el comando “INICIO” fuerte y claro ante el micrófono;
5. Decir el comando “BRAZO” fuerte y claro ante el micrófono;
6. Decir el comando “ABRE PINZA” fuerte y claro ante el micrófono;
7. Decir el comando “BRAZO” fuerte y claro ante el micrófono;
8. Decir el comando “POSICIÓN 7” fuerte y claro ante el micrófono;
9. Decir el comando “BRAZO” fuerte y claro ante el micrófono;
10. Decir el comando “CIERRA PINZA” fuerte y claro ante el micrófono;
11. Decir el comando “BRAZO” fuerte y claro ante el micrófono;
12. Decir el comando “ARRIBA” fuerte y claro ante el micrófono;
13. Decir el comando “BRAZO” fuerte y claro ante el micrófono;
14. Decir el comando “POSICIÓN 6” fuerte y claro ante el micrófono;
15. Decir el comando “BRAZO” fuerte y claro ante el micrófono;
16. Decir el comando “ABRE PINZA” fuerte y claro ante el micrófono;
17. Decir el comando “BRAZO” fuerte y claro ante el micrófono;
18. Decir el comando “ARRIBA” fuerte y claro ante el micrófono;

⁷ Véase el Capítulo 3.6, Control Manual.

19. Decir el comando “BRAZO” fuerte y claro ante el micrófono;
20. Decir el comando “INICIO” fuerte y claro ante el micrófono;
21. Decir el comando “DESCONECTAR” fuerte y claro ante el micrófono;
22. Decir el comando “ALTO” fuerte y claro ante el micrófono;

3.8 Modelo

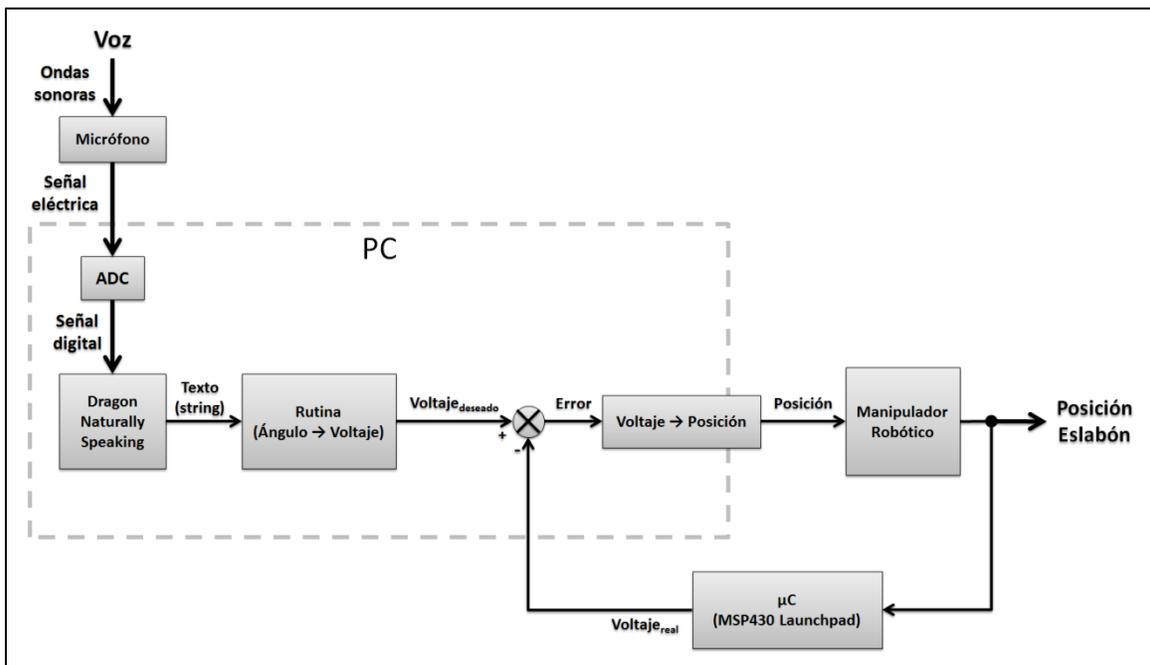


Figura 3.22 Modelo del sistema de control por voz del manipulador.

En la Figura 3.22 se ilustra el modelo de las etapas que conforman el sistema de control por voz del manipulador, desde la entrada de las ondas sonoras de la voz, hasta su correspondiente posición de salida de los eslabones del robot.

La descripción del modelo comienza con la señal de entrada del sistema, la cual, es un comando de voz compuesto de ondas sonoras.

Las ondas anteriores, entran a un transductor (micrófono) que se encarga de transformarlas en una señal eléctrica, correspondiente a una señal analógica variante en el tiempo. La frecuencia de la señal se encuentra en el rango de 50 a 20,000 [Hz], que es el rango normal del oído humano; sin embargo, para términos prácticos, es posible tomar un rango de frecuencias por debajo de los 3,400 [Hz], ya que es ahí donde se concentra la mayor parte de la información de la señal de voz^[45].

Una vez que la señal analógica sale del micrófono, ésta entrará a la tarjeta de audio de la PC, donde será digitalizada por el Convertidor Digital-Analógico (*ADC*). Esta etapa consiste en un *muestreo* de la señal a una frecuencia que cumpla con el teorema del muestreo de Nyquist, que dice que la tasa de muestreo debe ser superior al doble del ancho de banda de la señal muestreada (en este caso, $F_{max} = 48,000[Hz]$):

$$F_s > 2F_{max} \quad (3.25)$$

Terminado el muestreo, la señal analógica es ahora una señal en tiempo discreto y debe someterse a un proceso de *cuantificación*, esto es, que el valor de cada muestra de la señal se representa como un valor elegido de entre un conjunto finito de posibles valores. Finalmente, a cada valor se le asigna un número binario (*codificación*) y se llega a la señal digital de la voz.

Ya que se tiene la señal digital de la voz, se procede a la etapa de reconocimiento de voz, la cual se explicó en el capítulo 3.7.1 del presente trabajo y que arroja a su salida una cadena de caracteres correspondientes al comando de voz original.

La etapa siguiente, consiste en asignar a la cadena de caracteres la rutina correspondiente al comando que representa. En el caso del modelo descrito, se utilizó la rutina más básica, que consiste en mover cualquiera de los eslabones a un ángulo deseado. Para esto, la rutina se encarga de buscar en una tabla de lookup el voltaje (Tabla 3.4) correspondiente al ángulo deseado. Tal voltaje deseado, es el *set point* del lazo cerrado que forman las etapas siguientes.

Como toda configuración de lazo cerrado, una vez que se establece el set point, es necesario calcular el error, el cual se calcula con la siguiente ecuación:

$$e = V_{deseado} - V_{real} \quad (3.26)$$

Una vez calculado el error, es necesario transformar el voltaje deseado en una posición tal que el manipulador “entienda” hacia dónde se tiene que mover, para esto se busca en otra tabla de lookup (Tabla 3.5) la posición correspondiente al voltaje deseado y se le envía vía serie al manipulador.

Una vez que el manipulador comienza a moverse, entra en juego la etapa de realimentación descrita en el capítulo 3.5, el cual se encarga de medir el valor del voltaje real en el que se encuentra en ese momento el eslabón del robot y lo envía de vuelta hacia la PC, cerrándose así el lazo de control.

Por último, el lazo de control se detendrá una vez que el valor del error (3.26) caiga dentro de un rango preestablecido de la siguiente forma: En la Tabla 3.4 se encuentra el voltaje deseado, o en su defecto el número más cercano a éste y a continuación los valores próximo hacia arriba y próximo hacia abajo. Así, la salida del sistema será la posición del eslabón que le fue ordenada mediante un comando de voz.

3.9 Integración final

La integración final del proyecto abarca la unión tanto de las partes físicas como de las partes lógicas; incluyendo el espacio físico de todos los componentes (PC, manipulador y sistema de realimentación).

3.9.1 Espacio físico de trabajo

Para poder manejar el manipulador robótico y realizar todas las pruebas y rutinas de este proyecto, se le colocó en una mesa de trabajo, mostrada en la Figura 3.23, la cual cuenta con unas marcas tal que no se perdiera su posición con la constante vibración propia de los motores que conforman al robot. Esta mesa de trabajo también sirve de base a la celda de manufactura experimental.

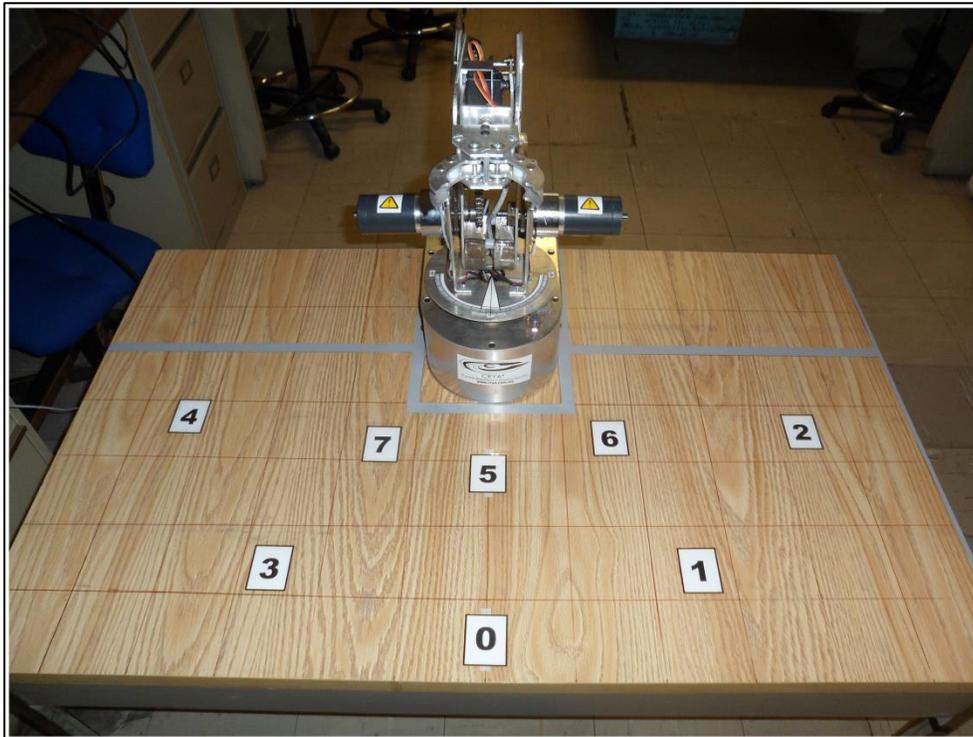


Figura 3.23 Mesa de trabajo del manipulador robótico Labot Pro 5.

3.9.2 Interfaz gráfica de usuario

Aunque ya se tienen dos interfaces gráficas para los diferentes tipos de control, resulta conveniente juntarlas en sola una interfaz, como la de la Figura 3.24, que brinde las mismas prestaciones pero que obviamente requiere de algunos cambios.

Ventajosamente, todas las prestaciones del control manual se mantuvieron sin cambios y solamente hubo variaciones leves en el control por voz. Primero que nada, la

forma de entrar al control por voz se hace mediante el botón marcado con un micrófono; esto hace que se abra la GUI de la Figura 3.25, que como se aprecia, ya no requiere del botón “Conectar” y del indicador “Puerto Abierto”; y el botón “Alto” ya no detiene el programa sino que solamente cierra la pantalla de la GUI del control por voz y le regresa el manejo a la GUI del control manual.

Utilizando la interfaz gráfica, fue posible crear todas las rutinas que se pueden ejecutar con el control por voz:

- **Posición Inicial**

Aunque no se trata de una rutina propiamente dicha, llevar al manipulador a su posición inicial consiste en alcanzar los siguientes ángulos en los eslabones: 0° para la base, 90° para el hombro y 0° para el codo con respecto a la horizontal, 33° para la muñeca que equivalen a 0° con respecto al eje del codo, 56° para la mano y 27° para el efector.

- **Posición X**

Este es un grupo de 8 rutinas que llevan al manipulador a los respectivos puntos marcados en la mesa de trabajo (véase Figura 3.23). En el caso de las posiciones Cero, Uno, Dos, Tres y Cuatro, la idea es dejar al efector final en la posición indicada en la Figura 3.26a para que logre agarrar una botella de plástico. Por otro lado, en el caso de las posiciones Cinco, Seis y Siete, la idea es dejar al efector final en la posición de la Figura 3.26b para que el manipulador pueda tomar una pelota de esponja.

- **Levanta**

Esta rutina consiste en elevar el brazo a cierta posición más arriba que cualquiera de las posiciones que se logran en las rutinas Posición X para lograr que el objeto que se agarra no se arrastre por la mesa y pueda ser transportado correctamente.

- **Rutina X**

Esta rutina puede ser cualquiera que se realice con el control manual; sin embargo, el sistema puede ejecutar hasta 10 rutinas diferentes (Rutina Cero hasta Rutina Nueve) utilizando el control por voz.

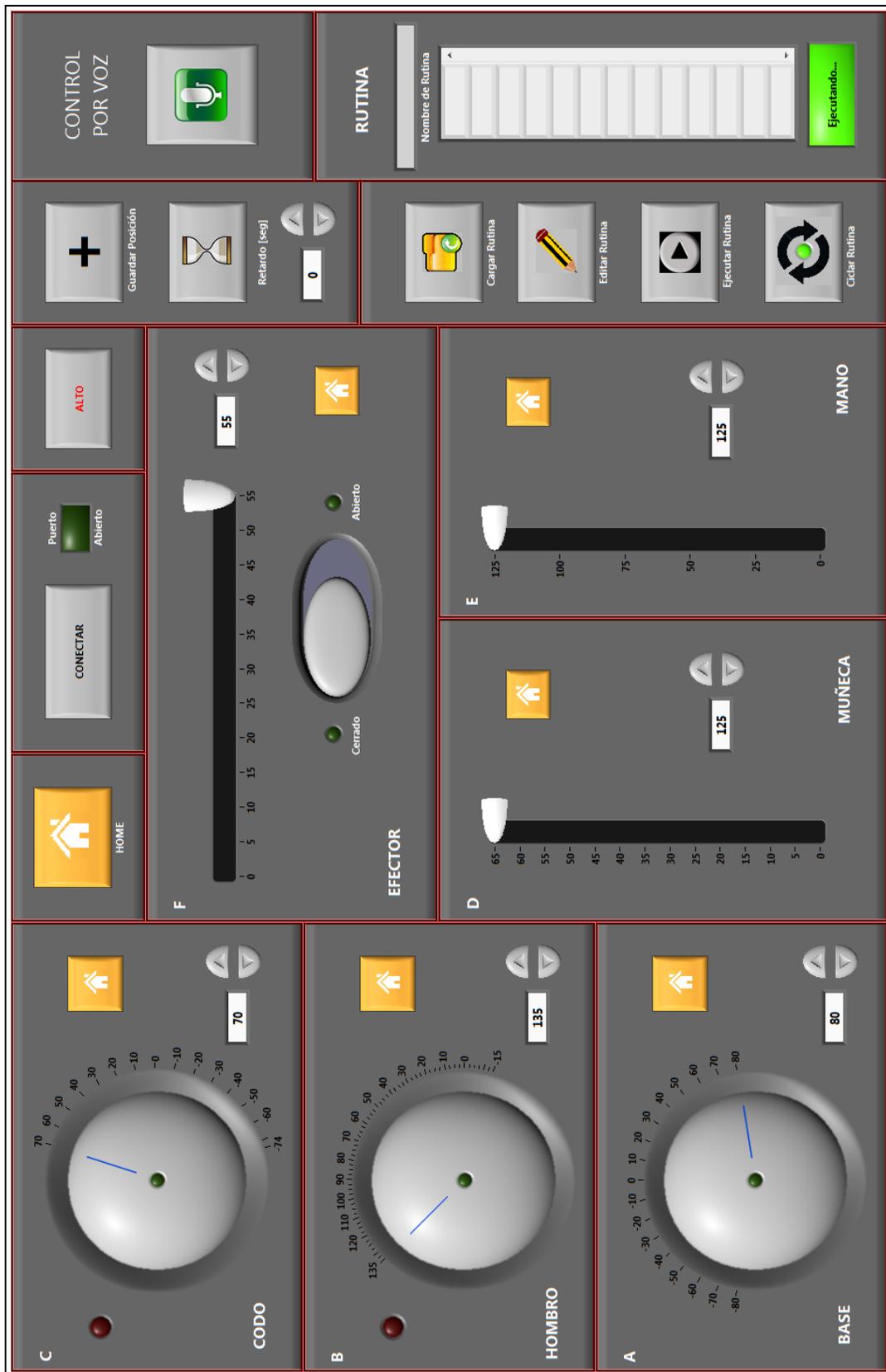


Figura 3.24 Interfaz gráfica para el control manual y por voz del manipulador robótico Labot Pro 5.

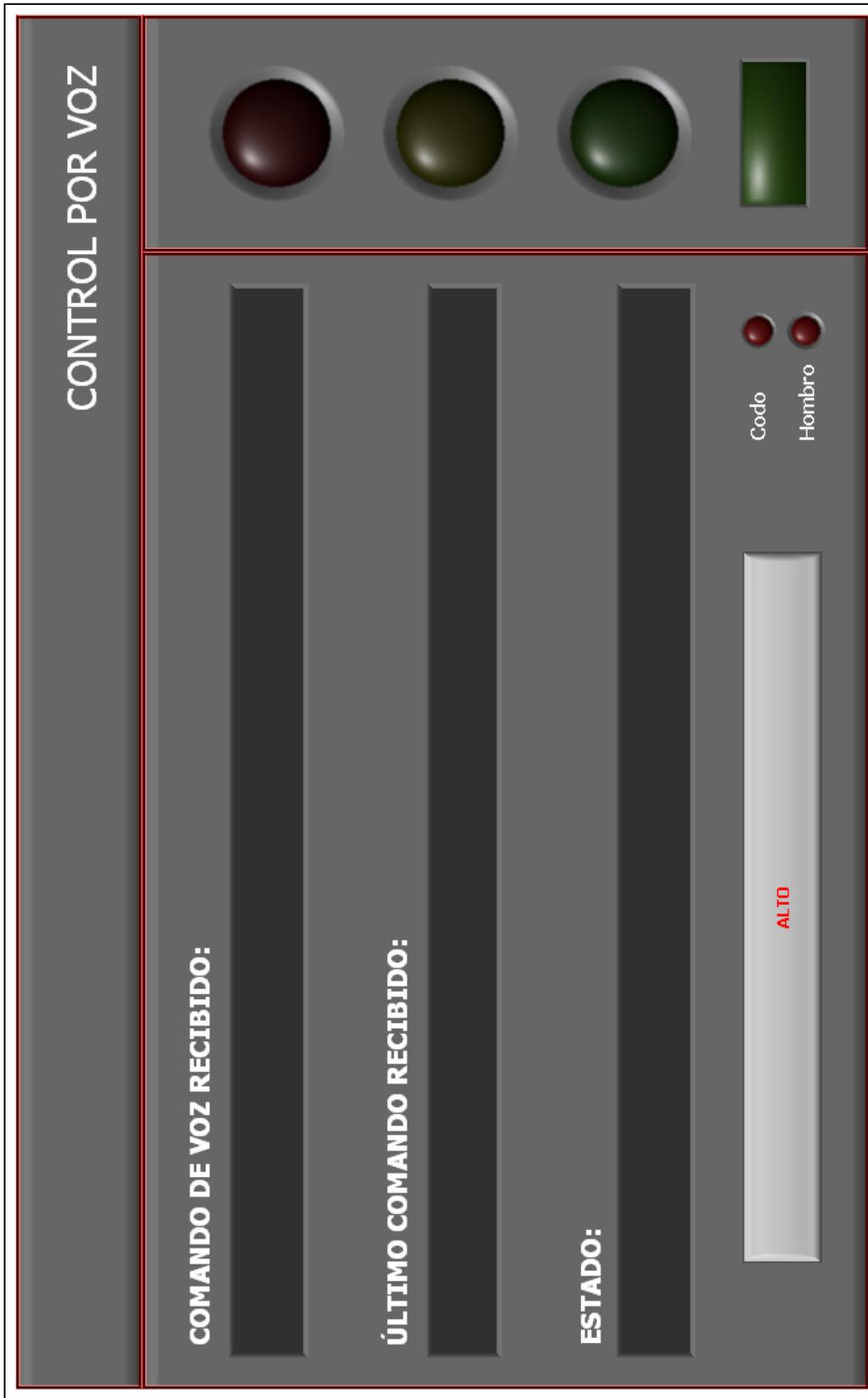


Figura 3.25 Interfaz gráfica para el control por voz.

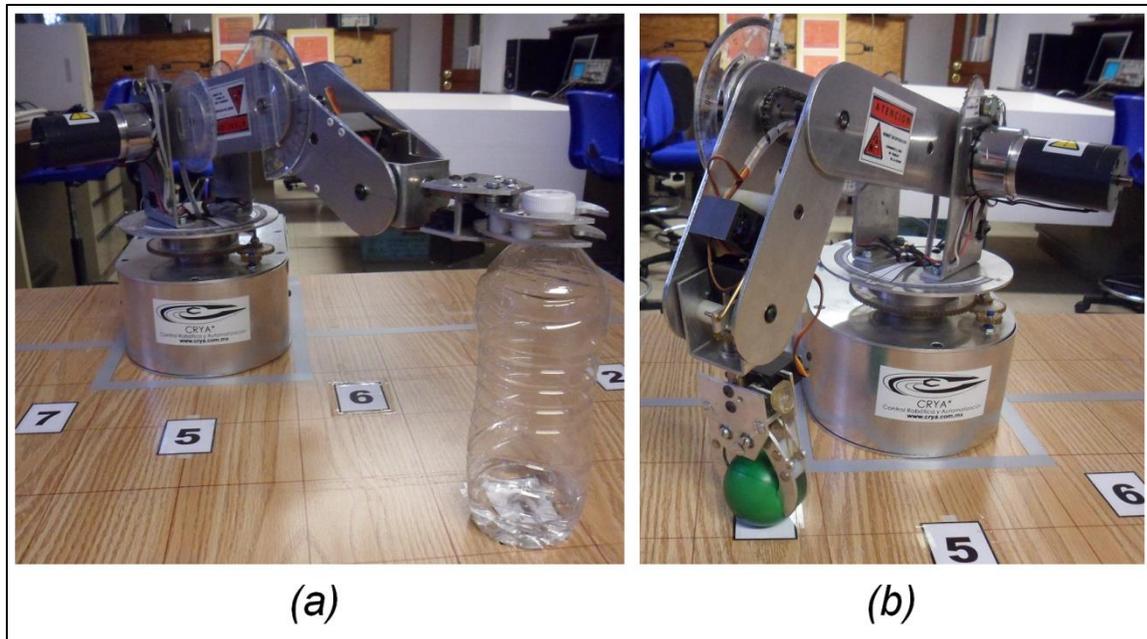


Figura 3.26 Diferentes posiciones del grupo de rutinas “Posición X”.

Cabe recordar que las rutinas “Posición X” y “Levanta” no contienen datos del efector; esto quiere decir que si se quiere tener la pinza abierta o cerrada, deberá indicársele a la interfaz con su correspondiente comando (ABRE/CIERRA PINZA).

3.9.3 Elementos

La celda de manufactura experimental con modo sensorial de visión que está siendo desarrollada, consta en la actualidad de los siguientes elementos: sistema posicionador para cámara monocular, banda transportadora, unidad de control y manipulador robótico antropomórfico. Todos los elementos, a excepción de la unidad de control, se encuentran contenidos en un sistema modular de aluminio ranurado que sostiene al sistema posicionador y que brinda la flexibilidad necesaria para cambiar su diseño de forma rápida y sin necesidad de mano de obra calificada; sin embargo, uno de los objetivos de este trabajo consiste en reforzar dicha estructura de aluminio.

Sistema posicionador para cámara monocular^[46]

Debido a que la celda de manufactura experimental basa su modo sensorial en la visión, este sistema posicionador se encarga de transportar una cámara web a través de dos grados de libertad lineal asociados a los ejes del plano cartesiano x y y . Tal posicionamiento se lleva a cabo utilizando motores eléctricos controlados por un microcontrolador y una PC.

Todo el sistema posicionador está sujetado por la estructura de aluminio ranurado de la Figura 3.27a; sin embargo, esta estructura carece de estabilidad y las cuatro columnas que la sostienen tienden a deslizarse hacia afuera y se corre el riesgo de la ruptura de algún componente.

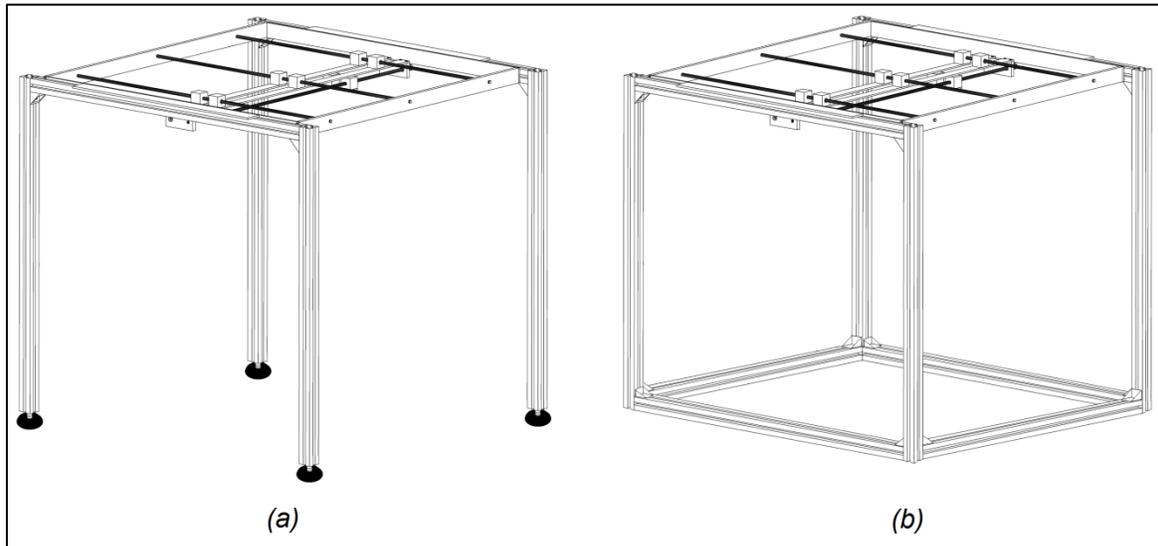


Figura 3.27 Estructura de la celda de manufactura experimental. (a) Antes del reforzamiento. (b) Después del reforzamiento.

Refuerzo en la estructura de aluminio

Debido a la gran inestabilidad de la estructura de aluminio que sostiene al sistema posicionador y que contendrá a todos los demás componentes de la celda, se necesitó reforzarla para evitar algún problema en el futuro. Dicho refuerzo consistió en agregar cuatro perfiles a modo de base que correspondieran al tamaño del sistema posicionador para que se formara una especie de cubo, como se observa en la Figura 3.27b. Tal forma geométrica brinda mucha más estabilidad a la estructura completa e incluso permitirá un aumento de peso en el sistema posicionador (p. ej. motores más grandes) sin comprometer los demás elementos de la celda.

Banda Transportadora

Un elemento muy útil dentro de una celda de manufactura es la banda transportadora o *conveyor belt*. Esta máquina consiste en una cinta continua entre dos o más poleas o tambores y sirve para llevar piezas dentro de la celda de una estación de trabajo a otra. En el caso de la celda experimental implementada, se cuenta con una banda transportadora sencilla que cuenta con un motor de DC que debe ser conectada a

una fuente de voltaje de 3.3 Volts; sin embargo, para mejorar las prestaciones y evitar el uso de una fuente de voltaje que genere un gasto extra, se implementó un control de velocidad y sentido de giro energizado directamente de la línea comercial.

Como se observa en la Figura 3.28, el control del motor de DC de la banda transportadora consta de dos transistores y un potenciómetro y es energizado por una fuente de voltaje bipolar de ± 5 Volts alimentada directamente de la línea comercial. La velocidad y el sentido de giro dependen de la posición en la que se encuentre el potenciómetro: si se encuentra completamente hacia la izquierda, la banda se moverá hacia ese lado a máxima velocidad; si se encuentra aproximadamente al 50%, el motor se mantiene en reposo y si se encuentra completamente hacia la derecha, la banda se moverá a máxima velocidad hacia la derecha.

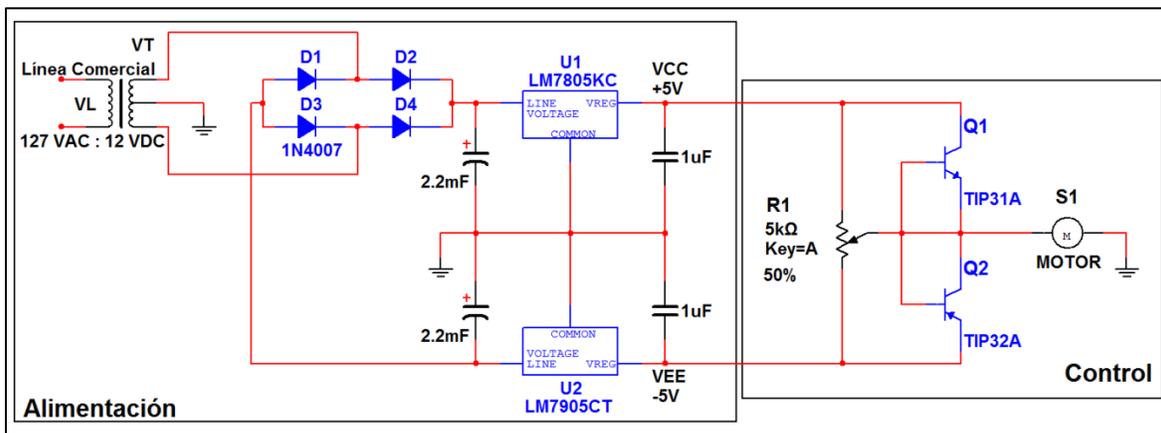


Figura 3.28 Circuito para el control de velocidad y sentido de giro de la banda transportadora.

Unidad de control

La unidad que controla los elementos de la celda de manufactura es una PC de escritorio con las siguientes características: Procesador Intel Core i3, 4 GB en RAM, 1 TB en ROM, Sistema Operativo Windows 7 Home Premium de 64 bits y LabVIEW Run-Time Engine 2011.

Por último, en la Figura 3.29 se pueden observar todos los elementos de la celda de manufactura flexible experimental.

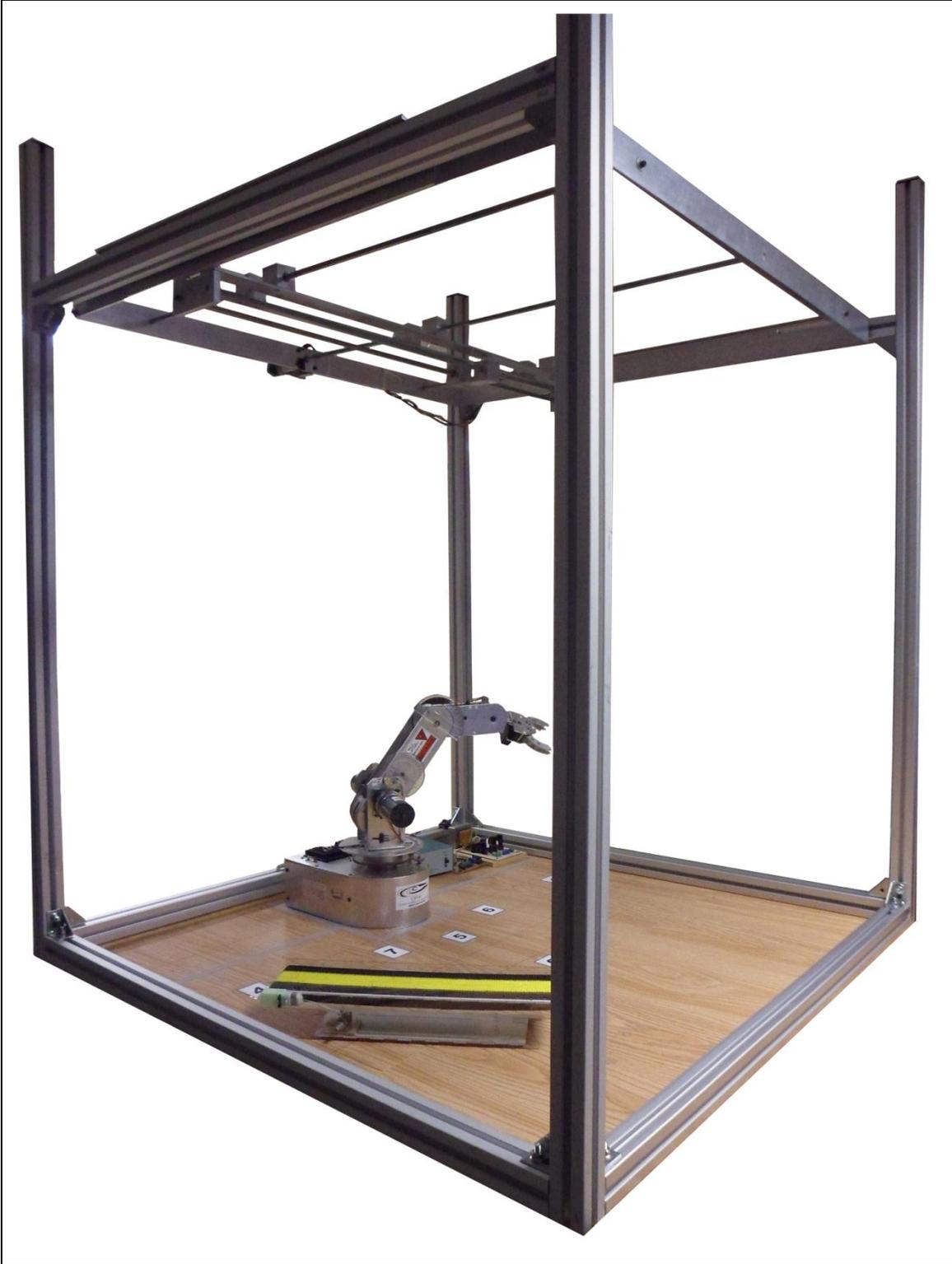


Figura 3.29 Celda de manufactura flexible experimental.

Capítulo 4

Pruebas y Resultados Experimentales

En este capítulo se explican las pruebas experimentales que se le hicieron a las diferentes partes del sistema y se exponen sus resultados.

4.1 Repetibilidad

Para comprobar el valor de la repetibilidad que indica el fabricante para su manipulador (0.5 cm) y también para comparar si existe alguna diferencia entre el software desarrollado para este proyecto y el propio del fabricante, se creó una prueba que consiste en llevar al manipulador desde su posición inicial, con un movimiento solamente sobre el plano vertical, hasta un punto fijo, marcado en una hoja de papel, en repetidas ocasiones. Para saber si se llega al punto en cuestión, se implementó en el efector final un señalador láser que apuntara directamente a la marca de dicho punto, como se ve en la Figura 4.1, y se tomó video de toda la prueba. Una vez terminadas las repeticiones, se observa el video para ubicar los puntos más alejados de la marca inicial, hacia ambos lados, y se mide la distancia entre estos para crear un círculo que los abarque; finalmente, el radio de ese círculo resulta ser el valor de la repetibilidad del manipulador.

Los resultados obtenidos se encuentran en la Tabla 4.1.

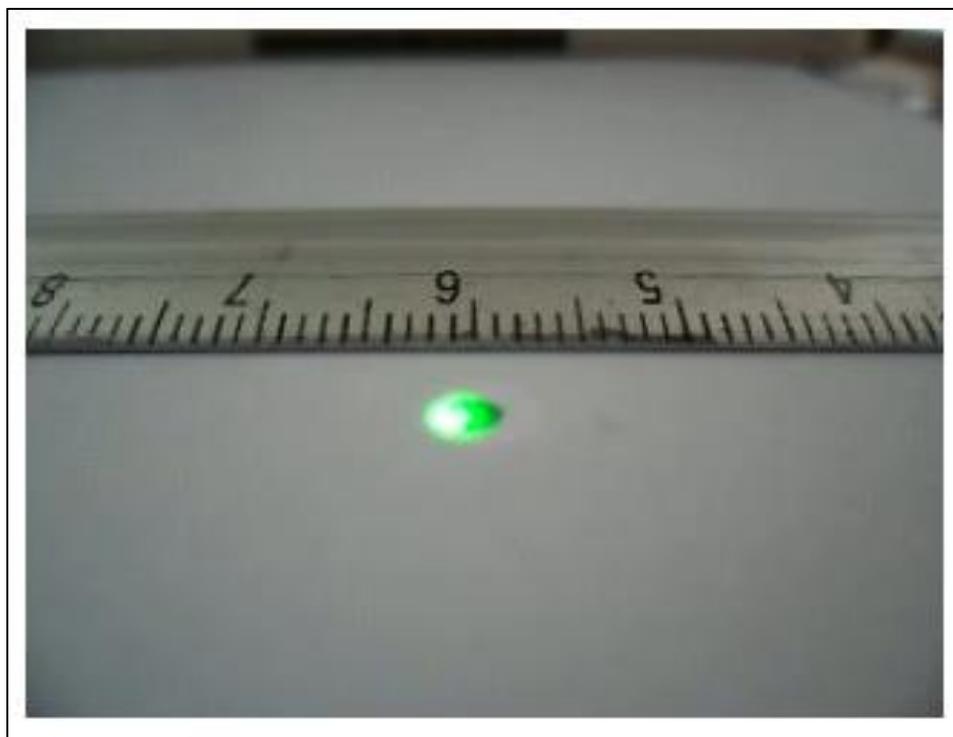


Figura 4.1 Prueba para medir la repetibilidad del manipulador.

Software	Tiempo de la prueba [min]	Número de muestras	Radio de repetibilidad [mm]
Fabricante	5	37	20
Proyecto	5	35	40

Tabla 4.1 Resultados de la prueba de repetibilidad.

4.2 Reproducibilidad

Para comprobar que el control desarrollado en este proyecto ayudó a combatir el grave problema de la reproducibilidad del manipulador, se realizó una prueba que consiste en marcar un *punto arbitrario* al que apunte el señalador láser y guardar la rutina que nos lleve desde la posición de inicio hasta ese punto. Lo siguiente es apagar el manipulador por completo y esperar unos minutos para volverlo a encender; una vez encendido se ejecuta la rutina guardada y se espera a que el apuntador láser llegue al mismo punto o por lo menos lo más cerca posible que la vez anterior. En esta prueba se tomaron 3 mediciones de cada uno de los 9 puntos (véase la Figura 4.2) *apagando* el manipulador y 1 cuarta muestra *desconectándolo*. Los resultados se observan en la Figura 4.3 para el software del fabricante y en la Figura 4.4 para el software propio.

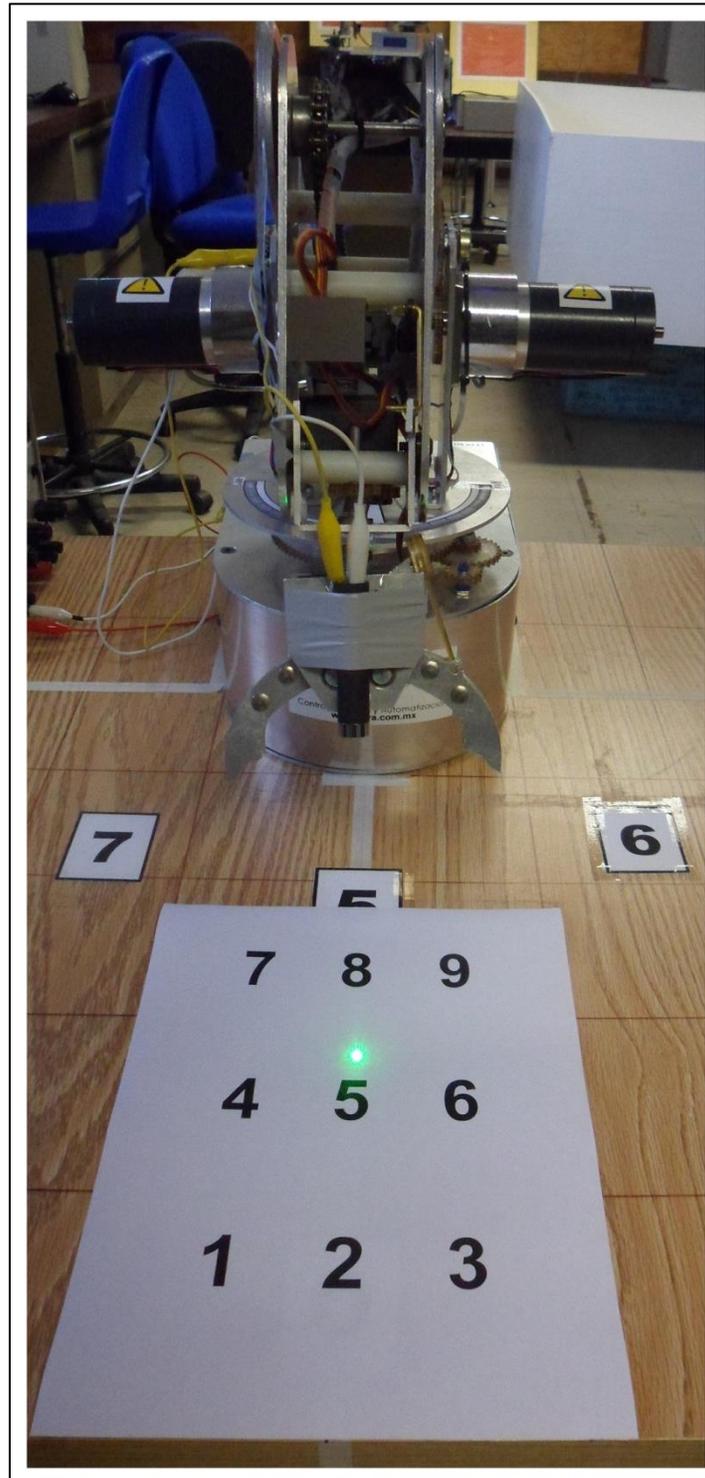


Figura 4.2 Numeración de las zonas de los 9 puntos usados en la prueba de reproducibilidad del manipulador.

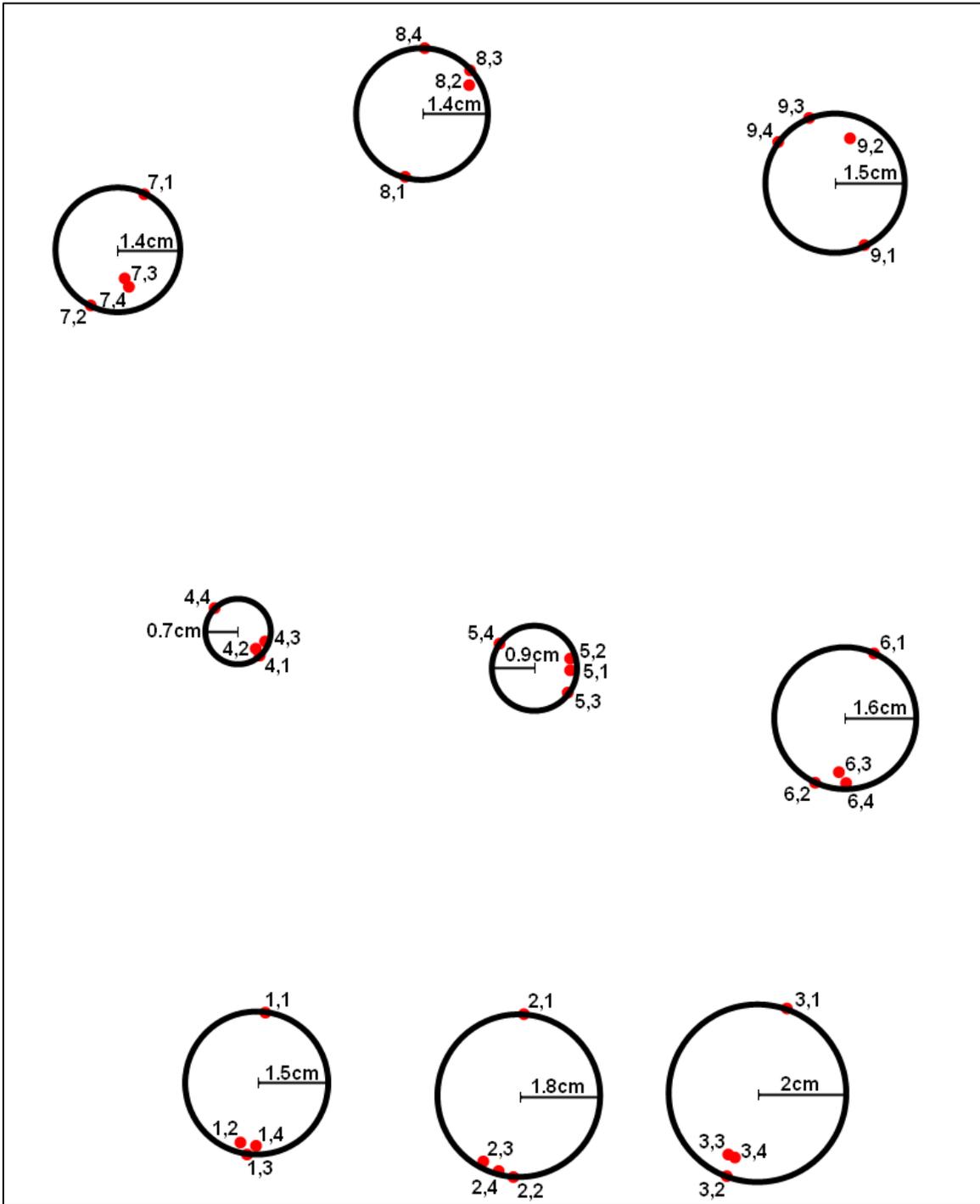


Figura 4.3 Resultados de la prueba de reproducibilidad utilizando el software del fabricante. Cada punto está numerado de la siguiente forma: número,muestra.

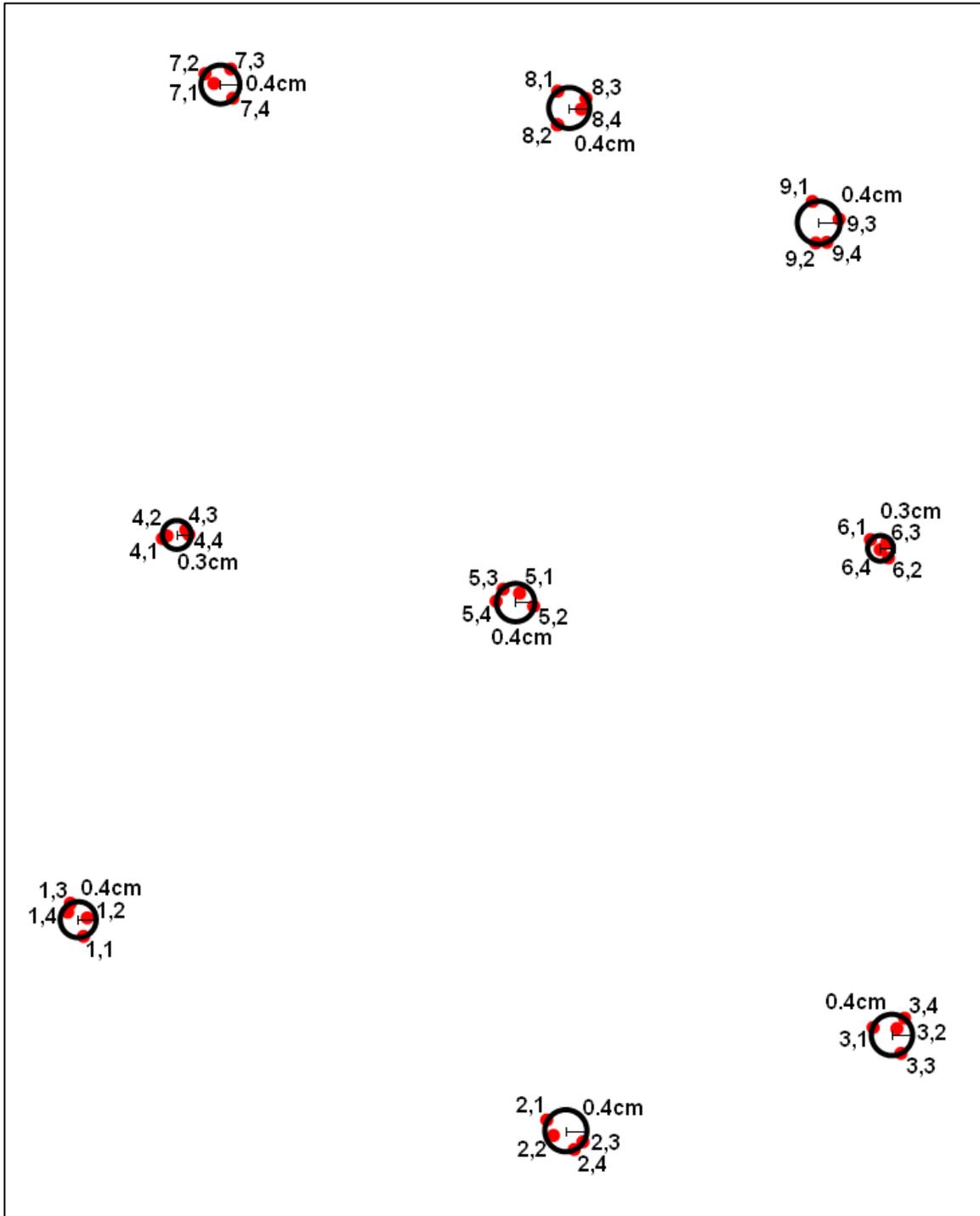


Figura 4.4 Resultados de la prueba de reproducibilidad utilizando el software propio. Cada punto está numerado de la siguiente forma: número,muestra.

4.3 Tablas de consulta

Para comprobar si el uso de las tablas de lookup funcionó, se llevó a cabo una prueba que consiste en mandar a un ángulo o posición fija, ya sea al hombro o al codo, desde un lado y desde el otro y observar qué tanta es la diferencia entre ellos. Para el caso donde no se utilizan tablas, las pruebas se llevan a cabo con el software del fabricante y se dan posiciones; en cambio, para el uso de tablas se utiliza el software propio y se dan ángulos.

Esta prueba toma en cuenta el grave problema que resulta de la especie de “histéresis” que tiene el manipulador, donde no es lo mismo acercarse al ángulo deseado desde un lado que desde el otro. Los resultados de la prueba se ven en la Tabla 4.2.

Sin tablas de Lookup							
Hombro				Codo			
Posición	Desde Arriba	Desde Abajo	Diferencia	Posición	Desde Arriba	Desde Abajo	Diferencia
700	111°	117°	6°	700	67°	64°	3°
1000	64°	60°	4°	1000	25°	23°	2°
1200	35°	31°	4°	1200	-2°	-5	3°
1340	16°	14°	2°	1400	-28°	-31	3°
1540	-8°	-10°	2°	1600	-53°	-57	4°
			Promedio: 3.6°				Promedio: 3.0°
Con tablas de Lookup							
Hombro				Codo			
Ángulo	Desde Arriba	Desde Abajo	Diferencia	Ángulo	Desde Arriba	Desde Abajo	Diferencia
108°	106°	110°	4°	60°	59	61	2°
69°	69°	72°	3°	30°	29	31	2°
39°	39°	38°	1°	0°	-1	0	1°
15°	15°	12°	3°	-30°	-31	-29	2°
-9°	-10°	-8°	2°	-60°	-62	-58	4°
			Promedio: 2.6°				Promedio: 2.2°

Tabla 4.2 Resultados de la prueba del uso de tablas de Lookup.

4.4 Reconocimiento de voz

Ya que el reconocedor de voz es una gran parte del buen funcionamiento del sistema, se realizó una prueba en la que se repitieron cada uno de los comandos de la interfaz para el control por voz para conocer qué tan acertado es su uso o si sería mejor optar por cambiar la palabra o decir una parecida. En la Tabla 4.3 se observan los resultados de la prueba.

Comando	Número de muestras	Número de aciertos	Porcentaje de acierto
<i>conectar</i>	50	46	92%
<i>conecta</i>	50	48	96%
<i>desconectar</i>	50	50	100%
<i>desconecta</i>	50	50	100%
<i>alto</i>	50	50	100%
<i>salir</i>	50	49	98%
<i>brazo</i>	50	50	100%
<i>inicio</i>	50	49	98%
<i>posición cero (a siete)</i>	40	39	97.5
<i>levanta</i>	50	49	98%
<i>arriba</i>	50	50	100%
<i>rutina cero (a nueve)</i>	50	48	96%
<i>abrir pinza</i>	50	50	100%
<i>cerrar pinza</i>	50	50	100%

Tabla 4.3 Resultados de la prueba de reconocimiento de comandos de voz.

Como la celda de manufactura está diseñada para ser usada dentro de un ambiente industrial, el ruido es un factor muy importante a tomar en cuenta al momento de utilizar el manipulador; razón por la cual se realizaron dos pruebas que arrojaran resultados sobre la viabilidad de este tipo de control dentro de este ambiente en particular y realizar los cambios o por lo menos tomar en cuenta las consideraciones necesarias para lograr los mejores resultados.

4.4.1 Ruido ambiental

Para conocer qué tanto afecta el ruido ambiental al reconocimiento de voz, se diseñó una prueba en la que se varía el volumen de un generador de ruido blanco^[47] desde un nivel bajo de 50 dB (ruido ambiental en el laboratorio), un nivel medio de 64 dB hasta un nivel alto de 71 dB. Después se repiten algunos de los comandos de la interfaz para el control por voz y finalmente se llena la Tabla 4.4 con los resultados.

Comando	Nivel de ruido	Número de muestras	Número de aciertos	Porcentaje de acierto
<i>conectar</i>	Bajo	10	10	100 %
	Medio	10	10	100 %
	Alto	10	NA	NA
<i>desconectar</i>	Bajo	10	10	100 %
	Medio	10	10	100 %
	Alto	10	NA	NA
<i>brazo</i>	Bajo	10	10	100 %
	Medio	10	9	90 %
	Alto	10	NA	NA
<i>inicio</i>	Bajo	10	10	100 %
	Medio	10	10	100 %
	Alto	10	NA	NA
<i>posición cero</i>	Bajo	10	10	100 %
	Medio	10	9	90 %
	Alto	10	NA	NA
<i>levanta</i>	Bajo	10	10	100 %
	Medio	10	9	90 %
	Alto	10	NA	NA
<i>rutina cero</i>	Bajo	10	10	100 %
	Medio	10	10	100 %
	Alto	10	NA	NA
<i>abrir pinza</i>	Bajo	10	10	100 %
	Medio	10	10	100 %
	Alto	10	NA	NA
<i>cerrar pinza</i>	Bajo	10	10	100 %
	Medio	10	8	80 %
	Alto	10	NA	NA

Tabla 4.4 Resultados de la prueba sobre cómo afecta el ruido ambiental sobre el reconocimiento de voz.

Capítulo 5

Conclusiones

5.1 Conclusiones

Con base en los resultados obtenidos en los experimentos, se demostró que el manipulador robótico integrado a la celda de manufactura flexible experimental es ya completamente operativo con el control manual y el control por comandos de voz.

A través de la implementación de las diferentes mejoras durante el proyecto, se observó que el valor que brinda el fabricante sobre la *repetibilidad* del manipulador (5mm) queda sobrado con respecto a los 2mm obtenidos con las pruebas; aunque cabe aclarar que la rutina de prueba solamente tomó en cuenta el movimiento del brazo y el antebrazo, por ser los eslabones de los que se tiene realimentación, y no el movimiento de todos los eslabones del manipulador.

Por otro lado, los resultados de las pruebas sobre *reproducibilidad* arrojaron que se logró corregir en gran medida el problema más grave del funcionamiento del robot, en el cual, el robot no era capaz de reproducir adecuadamente una rutina si se le reiniciaba, apagándolo o desconectándolo. Estos resultados permiten que una vez grabada una rutina, no será necesario hacer cambios que demoren el proceso de manufactura.

En cuanto a si el uso de *tablas de lookup* mejoró el funcionamiento o no, se observó que nos brinda aproximadamente un grado sexagesimal de mejora en la precisión; lo cual, en principio no parece muy significativo hasta que se toma en cuenta que, por ejemplo, para el antebrazo un grado de diferencia en el ángulo equivale a casi un centímetro de diferencia en la posición del efector final.

Con respecto a las pruebas sobre reconocimiento de voz, en la primera se obtuvo un porcentaje de acierto muy alto de todos los comandos usados en el control por voz; sin embargo, para evitar errores en el reconocimiento debe tomarse mucho en cuenta la distancia y la forma en que se habla al micrófono, ya que lo mejor es hablar directo al centro del micrófono a unos 2 o 3 centímetros de distancia.

Otro punto muy importante que hay que tener en cuenta en el reconocimiento de voz, es el ruido ambiental ya que se comprobó en la última prueba que el reconocedor deja de detectar palabras alrededor de los 70dB de ruido ambiental, lo cual equivale aproximadamente al ruido de una aspiradora; sin embargo, es importante recordar que el control del manipulador deberá de hacerse desde un cuarto de control y no in situ, ya que se perdería la parte de flexibilidad o de autonomía de la celda de manufactura.

Finalmente acerca del uso del manipulador, es vital que el usuario del manipulador robótico tenga entrenamiento básico previo para entender y conocer los detalles propios del sistema, los cuales pueden ser comprendidos con la lectura de este trabajo y después con la práctica constante y responsable.

5.2 Trabajo Futuro

Debido a que se trata de una celda de manufactura experimental, ésta siempre estará abierta a mejoras; por lo tanto, es preciso llevar un seguimiento a una parte tan importante como lo es el manipulador robótico y ya que el presente trabajo sentó las bases del sistema, lo siguiente es realizar rutinas cada vez más elaboradas de acuerdo con las necesidades de la propia investigación.

Otro punto que puede llevarse a cabo inmediatamente es agregar a la celda de manufactura un segundo manipulador (que ya ha sido adquirido), del mismo modelo que el usado en este trabajo, y crear rutinas para que puedan interactuar entre ellos. Aunado a esto, otro elemento que sería ideal agregar sería una máquina CNC para que, por ejemplo, un brazo transporte piezas desde la banda transportadora hasta la máquina CNC, ésta última haga su trabajo y finalmente el otro manipulador lleve las piezas terminadas hasta otra cinta que las lleve a almacenamiento; todo de manera automática y sin la participación de ninguna persona.

Una idea para el futuro de la celda de manufactura flexible experimental es hacer que con el sistema de reconocimiento de imágenes, implementado en FPGAs, que está siendo desarrollado actualmente, se pueda localizar un objeto seleccionado por el usuario y se envíen las coordenadas al manipulador para que éste calcule los ángulos que le permitan alcanzar la posición donde se encuentra el objeto y después lo traslade a un lugar elegido por el mismo usuario. Una vez logrado esto, el paso final en esta vertiente de la investigación, sería que las piezas estuvieran sobre la banda transportadora y que el manipulador fuera capaz de tomarlas en movimiento y luego llevarlas al destino deseado.

Referencias

- [1] Peña Cabrera, Mario et al. "*Object Location in Manufacturing Cells using Artificial Vision*". Electronics, Robotics and Automotive Mechanics Conference, 2009. CERMA '09.
- [2] Lopez Juarez, Ismael et al. "*A learning approach for on line object recognition tasks*". Computer Science, 2004. ENC 2004. Proceedings of the Fifth Mexican International Conference.
- [3] Peña Cabrera, Mario et al. "*Visión para Robots en Tareas de Ensamble*".
- [4] Groover, Mikell P. "*Fundamentals of Modern Manufacturing: Materials, Processes, and Systems*". 3^a ed. John Wiley & Sons, 2007.
- [5] Kalpakjian, Serope. "*Manufactura, Ingeniería y tecnología*". 4^a ed. Pearson Educación, México, 2002.
- [6] Madison, James. "*CNC Machining Handbook: Basic Theory, Production Data, and Machining Procedures*". 1^a ed. Industrial Press Inc. 1996.
- [7] Peña Cabrera, Mario et al. "*Manufactura Inteligente Utilizando Visión para Robots*". IFAC, 2004.
- [8] De Souza, Rafael et al. "*Development of an Automated Cell for the Teaching of Integrated Manufacturing Systems*". Technologies Applied to Electronics Teaching (TAE), 2012.
- [9] Calangiu, Gabriela et al. "*Expert System for Teaching Robots in a Flexible Manufacturing Line*". 15th IEEE International Conference on Intelligent Engineering Systems (INES), 2011.
- [10] Spong, Mark. "*Robot Modeling and Control*". 1^a ed.; John Wiley & Sons, 2005.
- [11] Peña Cabrera, Mario. "*Object Location in Manufacturing Cells using Artificial Vision*". Electronics, Robotics and Automotive Mechanics Conference, 2009.

-
- [12] Niku, Saeed B. “*Introduction to Robotics: Analysis, Control, Applications*”. 2^a ed.; John Wiley & Sons, 2011.
- [13] KUKA Robotics [en línea, Agosto 2013]. <http://www.kuka-robotics.com>
- [14] ABB en México [en línea, Agosto 2013]. <http://www.abb.com.mx>
- [15] FANUC Robotics [en línea, Agosto 2013]. <http://www.fanucrobotics.com.mx>
- [16] YASKAWA Motoman Robotics [en línea, Agosto 2013]. <http://www.motoman.com>
- [17] Rivera Dueñas, Juan C. “*Apuntes de Control de Robots Industriales*”. Facultad de Ingeniería, UNAM.
- [18] ABROX Metrología Industrial [en línea, Agosto 2013]. http://www.abrox.com/robotic_iso.htm
- [19] Kibria, Shafkat. “*Speech Recognition for Robotic Control*”. Master’s Thesis in Computing Science. Umeå University, Sweden. 2005.
- [20] Pires, Norberto. “*Robot-by-Voice: Experiments on Comanding an Industrial Robot Using the Human Voice*”. Industrial Robot, An International Journal, Emerald Group Publishing Limited, Volumen 32, Número 6, 2005.
- [21] Smart Planet; “*Say command: How speech recognition will change the world*” [en línea, Agosto 2013]. <http://www.smartplanet.com/blog/smart-takes/say-command-how-speech-recognition-will-change-the-world/19895>
- [22] Grupo Golem [en línea, Agosto 2013]. <http://golem.iimas.unam.mx/home.php?lang=es&sec=home>
- [23] Juárez Vázquez, Janet A. “*Construcción de un reconocedor de voz para el español de México con variación alofónica media*”. Facultad de Estudios Profesionales campus Aragón, UNAM. 2009.
- [24] Juang, B.H. “*Automatic Speech Recognition – A Brief History of the Technology Development*”. 2004.
- [25] Chow, Y et al. “*BYBLOS: The BBN continuous speech recognition system*”. Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '87.
- [26] PCWorld; *Speech Recognition Through the Decades : How We Ended Up With Siri* [en línea, Agosto 2013].
-

-
- http://www.pcworld.com/article/243060/speech_recognition_through_the_decades_how_we_ended_up_with_siri.html
- [27] Nuance; Dragon Speech Recognition Software [en línea, Agosto 2013]. <http://www.nuance.com/dragon/index.htm>
- [28] Set up Speech Recognition [en línea, Agosto 2013]. <http://windows.microsoft.com/en-US/windows7/Set-up-Speech-Recognition>
- [29] Reconocimiento de voz y charla en Kinect [en línea, Agosto 2013]. <http://support.xbox.com/es-ES/kinect/voice/speech-recognition>
- [30] Google Voice [en línea, Agosto 2013]. <https://www.google.com/voice/>
- [31] Soon Suck Jarng. "HMM Voice Recognition Algorithm Coding". International Conference on Information Science and Applications (ICISA), 2011.
- [32] Shing-Tai Pan et al. "Speech recognition via Hidden Markov Model and neural network trained by genetic algorithm". International Conference on Machine Learning and Cybernetics (ICMLC), 2010.
- [33] Robótica CRYA. "Especificaciones CRYA LABOT PRO 5". Robótica CRYA.
- [34] Robótica CRYA. "Manual del Sistema Labot Pro 5". Robótica CRYA.
- [35] LINX Technologies, Inc. "SDM-USB-QS-S USB Module Data Guide". LINX Technologies, Inc., 2008.
- [36] Ogata, Katsuhiko. "Ingeniería de Control Moderna". 1ª ed.; Prentice-Hall hispanoamericana. 1980.
- [37] Texas Instruments Inc. "MSP-EXP430G2 LaunchPad Experimenter Board User's Guide". Texas Instruments Inc. 2012.
- [38] Texas Instruments Inc. "MSP430x2xx Family User's Guide". Texas Instruments Inc. 2012.
- [39] Texas Instruments Inc. "MSP430G2x53/ MSP430G2x13 Data sheet: Mixed Signal Microcontroller". Texas Instruments Inc. 2012.
- [40] Gales, Mark. "The Application of Hidden Markov Models in Speech Recognition". Foundations and Trends in Signal Processing Vol. 1, No. 3, 2007.
-

- [41] Practical Cryptography [en línea, Octubre 2013].
<http://practicalcryptography.com/miscellaneous/machine-learning/guide-mel-frequency-cepstral-coefficients-mfccs>
- [42] VoxForge [en línea, Octubre 2013].
<http://www.voxforge.org/home/docs/faq/faq/what-is-an-acoustic-model>
- [43] Sphinx-4 [en línea, Octubre 2013]. <http://cmusphinx.sourceforge.net/sphinx4>
- [44] Nuance Communications Inc. “*Guía del Usuario, Dragon Naturally Speaking Versión 11*”. Nuance Communications Inc. 2010.
- [45] Joskowicz, José. “Codificación de Voz y Video”. Instituto de Ingeniería Eléctrica, Facultad de Ingeniería, Universidad de la República. Montevideo, Uruguay, Marzo 2013.
- [46] Becerril Rodríguez, Pablo. “*Sistema Posicionador para Cámara Monocular Utilizando Motores de paso y Microcontroladores para Implementarse en una Plataforma de Celda de manufactura Experimental*”. Facultad de Ingeniería, UNAM. 2012.
- [47] White Noise Player [en línea, Agosto 2013]. <http://www.whitenoiseplayer.com>
- [48] Fuentes Arderiu, X. et al. “*Bioquímica Clínica y Patología Molecular, Volumen I*”; 2^a ed. Editorial Reverté, 1998.
- [49] White, Curt M. “*Data Communications and Computer Networks: A Business User’s Approach*”. 7^a ed. Course Technology, Cengage Learning, 2013.
- [50] De Diego, María. “*Introducción a las Señales Aleatorias*”. 1^a ed.; Universidad Politécnica de Valencia, Servicio de Publicación. 1999.

Glosario

- **Manufactura** (*manufacturing*): proceso de convertir la materia prima en productos. Incluye el diseño del producto, la selección de la materia prima y la secuencia de procesos a través de los cuales será manufacturado el producto.
- **Celda de manufactura** (*manufacturing cell*): grupo de estaciones de trabajo (usualmente máquinas CNC), interconectadas por un sistema automático de manejo, transporte y almacenamiento de materiales y controladas por un sistema de cómputo integrado.
- **Celda de manufactura flexible** (*flexible manufacturing cell*): son las celdas de manufactura con el mayor nivel de automatización y por lo general no están atendidas por humanos.
- **Control Numérico Computarizado** (*Computer Numerical Control, CNC*): es un proceso asistido por computadora para controlar máquinas de uso general a través de instrucciones generadas por un procesador y almacenadas en un sistema de memoria o medio de almacenamiento.
- **Manipulador Industrial**: brazo mecánico que opera bajo un control por computadora.
- **Manipulador Industrial (componentes)**:
 - **Manipulador**: cuerpo mismo del robot formado por sus elementos estructurales.
 - **Efactor final**: parte conectada al final del brazo (mano) que realiza las tareas requeridas. Puede ser desde la pinza más simple que solo abra y cierre, hasta una mano antropomórfica que trate de imitar la mano humana.
 - **Actuador**: “músculos” que mueven la estructura del manipulador. Los más comunes son los servomotores, motores a pasos, actuadores hidráulicos o neumáticos, etc.
 - **Sensor**: recolecta información del estado interno del robot o se comunica con el entorno. Algunos ejemplos son los sensores de fuerza, distancia, proximidad, los sistemas de visión e incluso sintetizadores de voz.

- **Controlador:** controla los movimientos de los actuadores y coordinarlos con la realimentación de los sensores.
- **Procesador:** realiza los cálculos de movimientos y velocidades y supervisa las acciones coordinadas del controlador y los sensores. Por lo regular, el controlador y el procesador se encuentran en una misma unidad.
- **Grados de libertad** (Degrees-of-Freedom, DOF): número determinado por la cantidad de articulaciones del manipulador.
- **Cadena cinemática** (*kinematic chain*): unión de cuerpos rígidos por medio de articulaciones.
- **Carga útil** (*payload*): peso que puede cargar sin exceder sus demás especificaciones.
- **Alcance** (*reach*): distancia máxima que puede alcanzar dentro de su espacio de trabajo.
- **Precisión** (*accuracy*): medida de qué tan cerca puede llegar el robot a un punto dado dentro del espacio de trabajo.
- **Repetibilidad** (*repeatability*): medida de qué tanto el manipulador puede regresar a un punto enseñado con anterioridad.
- **Reproducibilidad** (*reproducibility*)^[48]: concordancia entre los resultados de mediciones sucesivas del mismo mesurando realizadas haciendo variar las condiciones de la medición. Las condiciones que pueden variar en un estudio de reproducibilidad son: el principio de medida, el método de medida, el observador, el instrumento de medida, el patrón de calibración, el lugar, las condiciones de uso y el periodo de tiempo en que se realizan las mediciones repetidas. Una vez seleccionadas las distintas condiciones, al conjunto de las mismas se le denomina *condiciones de reproducibilidad*.
- **Espacio de Trabajo** (*work envelope*): volumen total que abarca el órgano o efector final cuando el robot ejecuta todos los movimientos posibles.
- **Control punto a punto** (*point-to-point control*): consiste en enseñar al manipulador un conjunto discreto de puntos, pero sin tener control sobre la trayectoria entre éstos.
- **Control de trayectoria continua** (*continuous path control*): se tiene completo control sobre la trayectoria del efector final.
- **Reconocimiento Automático de Voz** (*Automatic Speech Recognition, ASR*): procedimiento que consiste en procesar la señal de voz emitida por un ser humano

para convertirla en texto; y como consecuencia, pueda ser entendido por una computadora.

- **Servomotor:** sistema electromecánico que utiliza la realimentación para controlar su posición final dentro de su rango de operación y mantenerse estable en dicha posición. Consta de un motor, un motorreductor (caja reductora) y un circuito de control.
- **Tabla de consulta** (*lookup table*): arreglo (o matriz) cuyos elementos pueden ser accedidos mediante índices y sirven para reemplazar tiempo de cómputo.
- **Baudaje** (*baud rate*)^[49]: número de veces por segundo que una señal cambia su valor.
- **Ruido Blanco**^[50]: señal aleatoria en la cual sus valores no guardan correlación estadística entre sí.

Código Fuente (MSP430 Launchpad)

```

1 /*****
2 *                               Realimentación PC-uC                               *
3 *                               *                                               *
4 * Espera en modo de bajo consumo de energía hasta recibir un caracter desde la *
5 * PC solicitando la lectura de los voltajes del hombro y el codo para después *
6 * digitalizarlos y transmitirlos de vuelta *
7 *                               *                                               *
8 *                               Ing. Angel Padilla Salazar                       *
9 *                               Maestría en Sistemas Electrónicos                *
10 *                               IIMAS, UNAM                                     *
11 *                               Febrero 2013                                    *
12 *                               CCS 5.3.0.00090                                 *
13 *****/
14
15 #include "msp430g2553.h"
16
17 //Constantes
18 #define LEDRojo      BIT0
19 #define RXD          BIT1
20 #define TXD          BIT2
21 #define LEDVerde    BIT6
22
23 //Variables
24 unsigned int HombroBajo, HombroAlto, CodoBajo, CodoAlto;
25 unsigned int i;
26
27 //Funciones
28 void ConfigUART(void);
29 void ConfigADC(void);
30 void Conversion(unsigned int);
31 void Transmision(void);
32
33
34 /*****
35 * Programa Principal
36 *****/
37
38 int main(void)
39 {
40     WDTCTL = WDTPW + WDTHOLD;           // Detengo el WDT (Watch Dog Timer).
41     BCSCTL1 = CALBC1_1MHZ;             // Configuro el DCO (Digitally-Controlled Oscillator), para
42     DCOCTL = CALDCO_1MHZ;             // para trabajar a 1MHz.
43
44     // Configuro Puertos
45     P1DIR |= LEDRojo + LEDVerde + TXD; // P1.0, P1.2 y P1.6 como salidas.
46     P1OUT &= ~(LEDRojo + LEDVerde);   // Apago LEDs.
47
48     P1SEL |= RXD + TXD;                // P1.1=RXD, P1.2=TXD.
49     P1SEL2 |= RXD + TXD;
50
51     ConfigADC();                       // Configuro ADC.
52
53     ConfigUART();                      // Configuro UART.
54
55     __bis_SR_register(CPUOFF + GIE);   // Apago CPU y Habilito Interrupciones.
56 }
57
58 |

```

```

59 /*****
60 * Configuración de Modulos (ADC y UART)
61 *****/
62
63 void ConfigADC()
64 {
65     ADC10CTL1 = INCH_4 + ADC10SSEL_3;      /* INCH_4    -> Input Channel Select: 4;
66                                           ADC10SSEL_3 -> Clock Source Select: SMCLK (Sub-System Master Clock).
67                                           */
68     ADC10CTL0 = SREF_1 + ADC10SHT_2 + REF2_5V + REFON + ADC100N;
69                                           /* SREF_1    -> Reference: Vr+=Vref+ y Vr-=Vss;
70     ADC10SHT_2 -> Sample&Hold Time: 16xADC10CLKs;
71     REF2_5V    -> Reference Generator Voltage: 2.5V;
72     REFON      -> Reference Generator ON;
73     ADC100N    -> ADC10 ON;
74     ADC10IE    -> Interrupt Enabled.
75                                           */
76     ADC10AE0 |= BIT4 + BITS;              // P1.4 y P1.5 como entradas del ADC.
77
78     __delay_cycles(1000);                // Delay que permite al Voltaje de Referencia estabilizarse.
79 }
80
81 void ConfigUART()
82 {
83     UCA0CTL1 = UCSSEL_3;                  // Clock Source Select: SMCLK (Sub-System Master Clock).
84     UCA0BR0 = 104;                        // 1MHz -> 9600 baudios
85     UCA0BR1 = 0;                          // 1MHz -> 9600 baudios
86     UCA0MCTL = UCBR51;                    // Modulation UCBR5x = 1      ??????????
87     UCA0CTL1 &= ~(UCSWRST);               // *Initialize USCI state machine*  ??????????
88     IE2 |= UCA0RXIE;                     // Habilito interrupción de RX
89 }
90
91
92 /*****
93 * Rutinas de Servicio a Interrupción (Interrupt Service Routine, ISR)
94 *****/
95
96 #pragma vector=USCIAB0RX_VECTOR
97 __interrupt void USCI0RX_ISR(void)
98 {
99     if(UCA0RXBUF == 'm')                  // El caracter recibido es 'm'?
100    {
101        P1OUT |= LEDVerde;                 // Enciendo LED (verde) testigo.
102        P1OUT &= ~(LEDRojo);
103
104        ADC10CTL0 &= ~(ENC);                // Deshabilito ADC para cambiar configuración.
105        ADC10CTL1 &= ~(INCH0);             // Canal 4.
106
107        ADC10CTL0 |= ENC + ADC10SC;        // Comienzo muestreo y conversión.
108
109        while (ADC10CTL1&ADC10BUSY);       // Resultado de conversión listo?
110
111        HombroBajo = ADC10MEM;              // Guardo parte Baja del resultado de la conversión.
112        HombroAlto = (ADC10MEM >> 8);     // Guardo parte Alta del resultado de la conversión.
113
114
115        ADC10CTL0 &= ~(ADC10IFG + ENC);    // Deshabilito ADC para cambiar configuración.
116        ADC10CTL1 |= INCH0;                // Canal 5.
117
118        ADC10CTL0 |= ENC + ADC10SC;        // Comienzo muestreo y conversión.
119
120        while((ADC10CTL1&ADC10BUSY));      // Resultado de conversión listo?
121
122        CodoBajo = ADC10MEM;                // Guardo parte Baja del resultado de la conversión.
123        CodoAlto = (ADC10MEM >> 8);       // Guardo parte Alta del resultado de la conversión.
124
125
126        while(!(IFG2&UCA0TXIFG));          // Transmiso caracter identificador.
127        UCA0TXBUF = 'A';
128        while(!(IFG2&UCA0TXIFG));          // Transmiso la parte Alta del resultado de la conversión.
129        UCA0TXBUF = HombroAlto;
130        while(!(IFG2&UCA0TXIFG));          // Transmiso la parte Baja del resultado de la conversión.
131        UCA0TXBUF = HombroBajo;
132        while(!(IFG2&UCA0TXIFG));          // Transmiso caracter identificador.
133        UCA0TXBUF = 'B';
134        while(!(IFG2&UCA0TXIFG));          // Transmiso la parte Alta del resultado de la conversión.
135        UCA0TXBUF = CodoAlto;
136        while(!(IFG2&UCA0TXIFG));          // Transmiso la parte Baja del resultado de la conversión.
137        UCA0TXBUF = CodoBajo;
138
139        P1OUT &= ~(LEDVerde);
140    }
141    else
142    {
143        P1OUT &= ~(LEDVerde);
144        P1OUT |= LEDRojo;                  // Enciendo LED testigo (rojo).
145    }
146
147    __bis_SR_register(GIE);               // Apago CPU y Habilito Interrupciones.
148 }
149

```