



**UNIVERSIDAD NACIONAL AUTÓNOMA
DE MÉXICO**

FACULTAD DE INGENIERÍA

**DISEÑO, CONSTRUCCIÓN, CONTROL Y
OPERACIÓN DE UN BRAZO MANIPULADOR**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

P R E S E N T A N:

**AGUILAR MARTÍNEZ JEANNETE GUADALUPE
TREJO SIMÓN HERIBERTO**

**DIRECTOR DE TESIS
ING. ROMÁN VICTORIANO OSORIO COMPARÁN**



MÉXICO, D. F.

2006



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

AGRADECIMIENTOS

A Dios, por darme la fortaleza en los momentos más difíciles de mi vida como Universitaria.

Gracias intensamente a mis padres Rafael Aguilar Huitron y Yolanda Martínez Ortega, por su invaluable y constante sacrificio. Gracias Mamá por tus desvelos y consejos. Gracias Papá por aguantarme tanto tiempo y por creer en mí, que por ustedes he llegado hasta donde estoy.

Gracias hermanos Roberto, Junior y a mi sobrina Lluvia por estar conmigo en aquellos momentos en que el estudio y el trabajo ocuparon mi tiempo y esfuerzo.

También un agradecimiento muy especial a Heriberto que durante bastante tiempo tuvo la paciencia suficiente para apoyarme profundamente, para darme su comprensión, su cariño y su amor. Que sin su ayuda esto no hubiera sido posible, gracias por ser parte importante en el logro de esta meta.

Será difícil olvidar los maravillosos momentos que pase con mis amigos de la facultad. Gracias Lalo, Karina, Lulú, Nayeli, Luis, Alfredo, Diana, Fátima, Christian, Juan Carlos, Sergio, Alex y Raúl por haber compartido esos formidables momentos.

Cuatro grandes amigos Maty, Charlangas, Brisa y Gustavo por su amistad brindada durante tanto tiempo.

Por ultimo quiero dar gracias a todas aquellas personas que de una u otra manera han compartido su vida durante el transcurso de la mía, mi más sincero agradecimiento por su comprensión, estímulo y ayuda.

Jeannete Guadalupe Aguilar Martínez

A mis padres, Genoveva y Jesús, por su gran esfuerzo y apoyo durante toda mi vida, pero sobre todo por su gran amor y entrega incondicional. Su espíritu de sacrificio me ha llevado a ser una persona con valores, comprometido a alcanzar cada una de mis metas. Gracias por estar conmigo.

A mis hermanos, Yanet Alejandra, Allan David y Deysi Fabiola por su amor y comprensión; cada uno de ustedes representan un estímulo para superarme en la vida.

A mis sobrinos Abril, Santiago y Paulina, quienes trajeron alegría a mi vida.

A Jeannete G. Aguilar, por brindarme tu confianza y amor, por estar a mi lado y ser parte fundamental en este proyecto. Tu paciencia y apoyo han marcado el camino para culminar esta meta juntos. Gracias.

A todos ellos, por que siempre estarán en mi corazón, con respeto y admiración.

Heriberto Trejo Simón

A la Universidad Nacional Autónoma de México, por la formación académica que nos ha entregado. Es un orgullo pertenecer a esta institución.

A todos nuestros profesores, por compartirnos sus conocimientos.

Al Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), y Centro de Manufactura Avanzada (CDM), por el apoyo brindado en este proyecto.

A nuestro director de tesis, Ing. Román Osorio Comparán, por su orientación y apoyo.

Gracias

Jeannete y Heriberto

INDICE

INDICE

| | |
|----------------------------------------------------------------|-----------|
| Capítulo 1. Introducción..... | 1 |
| 1.1 Componentes de un robot..... | 2 |
| 1.2 Definición del problema..... | 4 |
| 1.2.1 Antecedentes..... | 4 |
| 1.2.2 Planteamiento del problema..... | 4 |
| 1.2.3 Objetivos..... | 6 |
| 1.2.3.1 Objetivo general..... | 6 |
| 1.2.3.2 Objetivos específicos..... | 6 |
| 1.3 Análisis del problema..... | 6 |
| 1.3.1 Especificaciones para el diseño del brazo..... | 7 |
| 1.3.1.1 Dimensión de los eslabones..... | 7 |
| 1.3.1.2 Actuadores..... | 11 |
| | |
| Capítulo 2. Diseño mecánico..... | 13 |
| 2.1 Modelado cinemático del sistema..... | 14 |
| 2.1.1 Descripción de la posición..... | 14 |
| 2.1.2 Rotación en el sistema cartesiano..... | 15 |
| 2.1.3 Transformaciones homogéneas..... | 17 |
| 2.1.4 Representación de Denavit – Hartenberg..... | 17 |
| 2.1.5 Parámetros Denavit – Hartenberg para el sistema..... | 20 |
| 2.1.6 Modelo cinemático del sistema..... | 21 |
| 2.2 Simulación del modelo..... | 23 |
| | |
| Capítulo 3. Diseño electrónico..... | 26 |
| 3.1 Etapa de sensado..... | 27 |
| 3.1.1 Sensor de posición del sistema..... | 27 |
| 3.1.2 Sensor de color de objetos..... | 28 |
| 3.2 Etapa de microcontrolador..... | 30 |
| 3.3 Circuito electrónico para el electroimán..... | 32 |
| 3.4 Etapa de potencia..... | 33 |
| | |
| Capítulo 4. Diseño del software de control..... | 35 |
| 4.1 Interfaz de usuario..... | 37 |
| 4.2 Control de posición..... | 40 |
| 4.2.1 Algoritmo de control de posición..... | 40 |
| 4.2.2 Implementación del algoritmo de control de posición..... | 45 |

| | |
|-----------------------------------------------------------------------|-----|
| Capítulo 5. Implementación física del Manipulador | 46 |
| 5.1 Eslabón 1..... | 47 |
| 5.2 Eslabón 2..... | 50 |
| 5.3 Eslabón 3 y 4..... | 52 |
| 5.4 Eslabón 5..... | 53 |
| 5.5 Ensamble final..... | 55 |
| | |
| Capítulo 6. Resultados y conclusiones | 56 |
| 6.1 Prueba del control de posición..... | 57 |
| 6.1.1 Trayectoria lineal..... | 57 |
| 6.2 Funcionamiento..... | 60 |
| 6.3 Repetibilidad..... | 62 |
| 6.4 Conclusiones..... | 63 |
| 6.4.1 En base al algoritmo de control por posición..... | 63 |
| 6.4.2 En base al diseño mecánico..... | 63 |
| 6.4.3 En base al diseño electrónico..... | 63 |
| 6.4.4 En base al sistema en general..... | 64 |
| 6.4.5 Trabajo futuro..... | 64 |
| | |
| Apéndices | |
| Apéndice A “Principales configuraciones de un brazo manipulador”..... | 65 |
| Apéndice B “Software utilizado”..... | 69 |
| Apéndice C “Código Fuente”..... | 77 |
| Apéndice D “Hojas de especificación técnica”..... | 83 |
| Apéndice E “Planos de fabricación”..... | 93 |
| | |
| Lista de figuras | 106 |
| | |
| Lista de tablas | 108 |
| | |
| Referencia bibliográfica | 109 |

CAPÍTULO 1

Capítulo 1 Introducción

El hombre desde hace siglos ha diseñado y fabricado máquinas capaces de realizar tareas similares a las acciones realizadas por el cuerpo humano. Su principal aplicación se encuentra en tareas con procesos repetitivos, actividades en donde se requiere gran esfuerzo, o bien actividades en donde la precisión es primordial. Este tipo de máquinas comúnmente son llamadas robots.

La palabra “robot” proviene del checo, y es usada por primera vez por el escritor Karel Capek a principios del siglo veinte para referirse, en sus obras, a máquinas con forma de humanoide. Un poco después, durante la segunda guerra mundial, Isaac Asimov, un famoso escritor de ciencia ficción volvió a utilizar el termino “robótica” en sus libros.

Cuando es utilizada la palabra robot, en diversas ocasiones se imagina a máquinas con apariencia humana capaces de tener sentimientos y sensaciones del mundo real, y que pueden interactuar de manera dinámica [OA2001]. Pero la realidad es que existen máquinas de diferentes formas, usos y configuraciones, dependiendo de su aplicación.

1.1 Componentes de un robot

En la *figura 1.1* se muestra el esquema de un robot (brazo manipulador). En está se pueden identificar los siguientes componentes: sistema mecánico, actuadores, fuente de alimentación, sensores, herramienta final, además de la computadora que generalmente aloja al sistema de control.

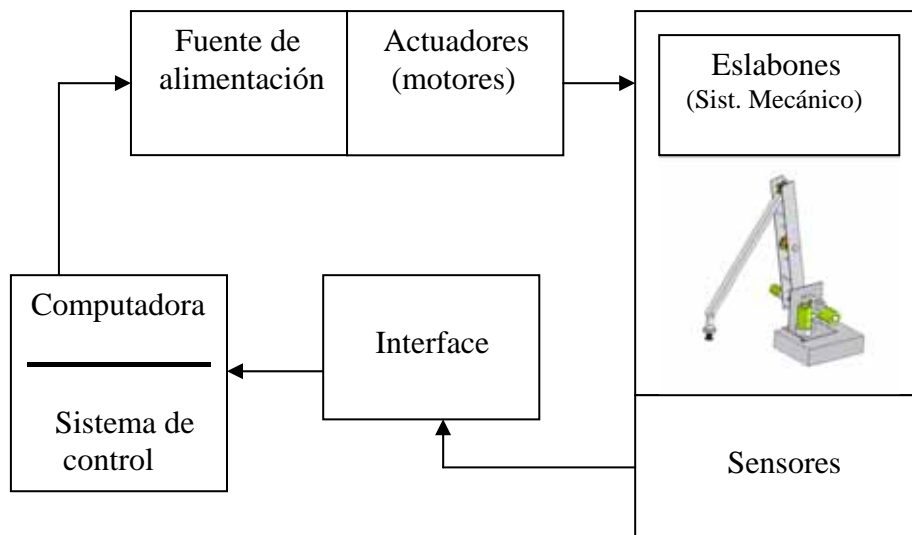


Figura 1.1 “Robot y su interacción con el entorno”

Dentro del sistema mecánico se encuentra el brazo, que es el elemento mecánico que posiciona en el espacio al elemento terminal (herramienta final). Los elementos rígidos del brazo (eslabones) están relacionados entre sí mediante articulaciones (grados de libertad), las cuales pueden ser giratorias (rotacionales) o traslacionales (prismáticas).

El grado de libertad de un robot depende del número de parámetros independientes que están involucrados para fijar la posición de la herramienta final. El número de grados de libertad suele coincidir con el número de eslabones de la estructura mecánica (cadena cinemática). Comúnmente, cada unión posee seis parámetros independientes, tres para su posicionamiento y tres para su orientación. En el caso de la figura 1.1 se tiene un sistema de cuatro uniones, cada una de las uniones se encuentran limitadas en cinco parámetros, y solo permiten el parámetro de rotación, para lograr una estructura articulada, mejor conocida como, configuración antropomórfica [OA2001].

El sistema de control o controlador, es el dispositivo que tiene como tarea regular los movimientos de los elementos del sistema, realizar cálculos matemáticos, procesos de información, y mantiene el registro de los sensores de posición o velocidad del sistema.

El sistema de control se puede implementar de distintas formas. Esto involucra a parámetros como posición, velocidad, fuerza, etc. Por ejemplo, el control de posición busca ubicar al elemento terminal, llevándolo a una posición final deseada. Esta puede ser siguiendo una trayectoria definida o posicionándose punto a punto.

La función de los actuadores es mover las articulaciones y eslabones a través de poleas, engranes, cadenas, etc. La clasificación de actuadores se realiza de acuerdo al tipo de energía que demandan: hidráulicos: se utilizan cuando se requiere una gran capacidad de carga, sumado a una precisa regulación de velocidad; neumáticos, emplean el aire comprimido como fuente de energía y son utilizados para controlar movimientos rápidos pero no de mucha precisión; y los eléctricos, que son los más utilizados debido a que su control es sencillo y preciso, también por las ventajas que representa la utilización de la energía eléctrica para su funcionamiento.

El elemento terminal del sistema es una herramienta acoplada según la necesidad, ésta es la encargada de completar la tarea del brazo manipulador. En determinadas ocasiones es conveniente que su peso y tamaño sean reducidos.

Los robots de última generación pueden interactuar con el medio de una manera más inteligente, es decir, pueden relacionarse eficazmente con su entorno de trabajo y tomar decisiones de una manera inmediata, adaptándose a las circunstancias en cada momento, todo esto gracias a los distintos sensores que les brindan información de posición, velocidad, aceleración, temperatura, presión, detección de objetos, etc.

1.2 Definición del problema

1.2.1 Antecedentes

Ante la necesidad de contar en México con una planta industrial productiva y competitiva a nivel internacional en el área de Robótica, se instauró una nueva carrera para formar profesionistas en el área de Ingeniería Mecatrónica en la Facultad de Ingeniería (FI) de la Universidad Nacional Autónoma de México (UNAM). En esta disciplina se relacionan áreas como control, computación, electrónica y mecánica, tomando de éstas la teoría y sintetizando metodologías para la solución de problemas de ingeniería. También vino a fortalecer las investigaciones de otras dependencias de la Universidad, como es el caso del Departamento de Ingeniería de Sistemas Computacionales y Automatización (DISCA) sección Electrónica y Automatización (EA) del Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas (IIMAS), el Instituto de Ingeniería y el Centro de Manufactura Avanzada (CDM). Dentro de estas instituciones se han venido desarrollando diversos proyectos mecatrónicos, como son: el diseño, control y fabricación de manos mecánicas, robots móviles, procesos de manufactura y automatización, etcétera. Estos proyectos tienen cuya finalidad el estudio, diseño e implantación de nuevas tecnologías y nuevos métodos de investigación que ayuden al adelanto tecnológico y científico que necesita nuestro país.

Por lo anterior nació la inquietud de desarrollar como proyecto de tesis “El diseño, construcción, y control de un brazo manipulador” que realice tareas específicas (detallas en el planteamiento del problema). Este trabajo servirá como material de enseñanza dentro de las actividades académicas (cursos, laboratorios, tesis, etc.) de las instituciones antes mencionadas, espacios en donde se apoyo totalmente el proyecto.

1.2.2 Planteamiento del problema

Para definir con mayor detalle la funcionalidad que debe presentar el manipulador, debemos mencionar que como una de las aplicaciones de este proyecto en tiempo y forma, además de servir como material de enseñanza dentro de las actividades académicas de ingeniería de la UNAM, está la de representar a nuestra Universidad en los distintos eventos de minirobotica que se llevan a cabo a lo largo del país, en diferentes cedes universitarias , siendo el más importante el organizado por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) en la ciudad de Querétaro. Por lo anterior la función principal del robot será seleccionar y tomar objetos, así como su traslado de un lugar a otro. Dichos objetos se encontrarán distribuidos aleatoriamente sobre un tablero de 400 [mm] x 400 [mm]. El brazo se encargará de tomar las piezas y dirigirlas a su respectivo contenedor, todo esto debe ser de manera autónoma. El manipulador puede seleccionar las piezas en cualquier orden, y colocarlas en su contenedor correspondiente conforme al color de la misma (blanco ó negro). Se dispone de un total de 10 piezas de las cuales 5 serán de color negro y 5 de color blanco.

Las piezas se colocan dentro del tablero con una distribución geométrica predeterminada, siendo la separación mínima entre una pieza y otra de 25 [mm] en todas las direcciones.

Por otro lado existen algunas otras especificaciones que se mencionan a continuación:

Tablero de 400 [mm] por lado de color gris. El tono de gris será similar al obtenido con una mezcla de 50% de pintura blanca mate y 50% de pintura negra mate.

Los contenedores tienen medidas interiores de 200 [mm], de cada lado y 100 [mm] de altura, los cuales se colocan como lo indica la *figura 1.2* centrados con respecto a la longitud del tablero, con sus caras paralelas a las caras del tablero y colocados a tal distancia que el hueco inicie a 20 [mm] del tablero. La posición de los contenedores, de cada lado del tablero, es fija y sin posibilidad de movimiento.

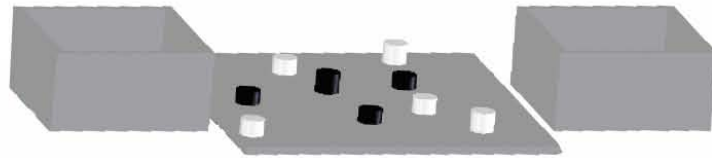


Figura 1.2 “Disposición de las fichas en el tablero y sus contenedores”

El robot se coloca a un costado del tablero, equidistante a ambos contenedores, como lo muestra la *figura 1.3*. La distancia del centro del robot al tablero, medida sobre el eje de simetría del arreglo, será libre. Una vez fijada la posición del prototipo no podrá ser movido una vez iniciada su rutina.

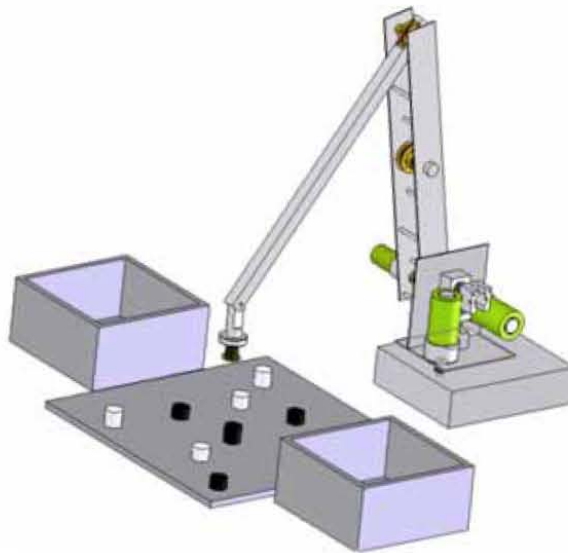


Figura 1.3 “Escenario principal”

El robot puede realizar desplazamientos tridimensionales por encima de todo el tablero. Es importante no olvidar que el área total del desplazamiento del robot deberá ser tal que contenga el área del tablero y las áreas de los dos contenedores en la posición dispuesta.

El robot puede tener uno o más puntos de apoyo, y puede realizar movimientos prismáticos o rotacionales en cualquiera de sus articulaciones.

Las piezas u objetos a desplazar son cilíndricas, con medidas de 25 [mm] de diámetro y altura variable entre 15 [mm] y 25 [mm], con fierro de al menos 5 [mm] de espesor en la parte superior y sin perforaciones. Cada una de las piezas tendrá un peso máximo de 15 gramos.

1.2.3 OBJETIVOS

1.2.3.1 Objetivo general

“Diseñar, construir y controlar un brazo manipulador de tres grados de libertad, capaz de tomar y trasladar objetos de un punto a otro, dentro de un espacio de trabajo bien definido”.

1.2.3.2 Objetivos específicos

1. Obtener el modelo cinemático del sistema.
2. Diseñar el control electrónico
3. Diseñar el software de control
4. Implementación física del sistema

1.3 Análisis del problema

Una vez planteada la definición formal del problema, así como los objetivos a desarrollar, podemos extraer las siguientes funcionalidades que el manipulador debe cubrir:

Seleccionar y tomar objetos cilíndricos, así como su traslado de una posición a otra en el espacio.

1. Carga y descarga de objetos.
2. Reconocimiento del color del objeto
3. Exactitud y repetibilidad

Considerando estos requerimientos y analizando las opciones de configuraciones posibles mostradas en el apéndice A “Principales configuraciones de un brazo manipulador”, todas podrían cumplir adecuadamente con el punto 1. Si se requiere eliminar opciones a partir de las desventajas, las configuraciones cartesiana, esférica y cilíndrica, pese a implicar un control más sencillo, su espacio de trabajo es reducido en comparación con el que presentaría un manipulador del tipo antropomórfico. Debido a esta valoración se propone la construcción de un brazo con configuración antropomórfica para resolver el problema, además que todas sus articulaciones son rotacionales, y éstas son más sencillas de fabricar.

1.3.1 Especificaciones para el diseño del brazo

1.3.1.1 Dimensión de los eslabones

Para definir la longitud de los eslabones del manipulador se deben de tomar en cuenta las dimensiones del área de trabajo, esto con la finalidad de que la herramienta final pueda ser posicionada en cada uno de los puntos de dicho espacio. Los eslabones 1 y 2, corresponden a los ejes rotacionales del primero y segundo grado de libertad del robot, y sus dimensiones no afectan de una manera directa al alcance que puede tener el brazo, sin embargo la dimensión de los eslabones 3 y 4 sí influyen de manera directa, debido a que son los elementos de mayor tamaño en la estructura (observar *figura 1.4*). Por lo anterior, se calculó, de una manera aproximada, la longitud mínima de debe de tener dichos eslabones para cubrir en su totalidad el tablero mencionado anteriormente.

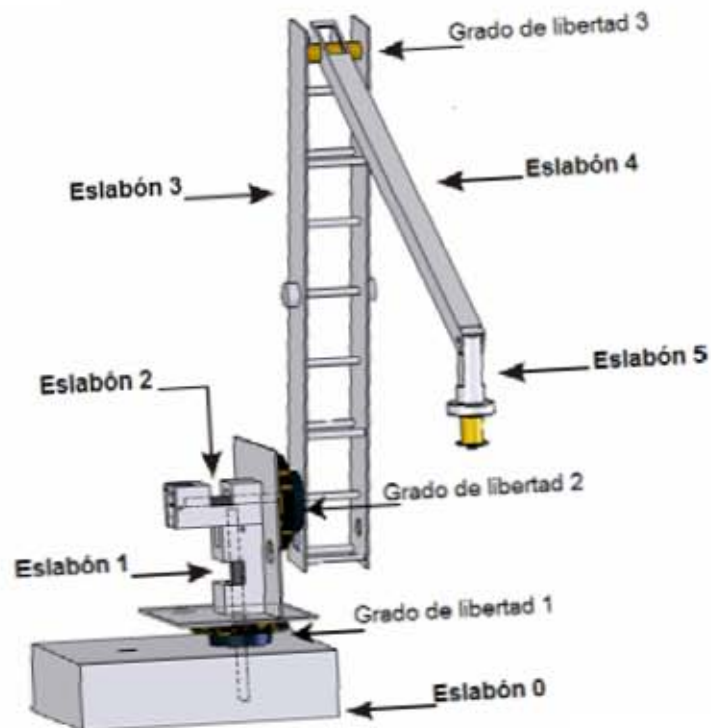


Figura 1.4 “Eslabones del sistema brazo manipulador”

Para obtener dichas magnitudes, se tomó como referencia la longitud entre cualquiera de las dos esquinas superiores del tablero y el centro del eslabón encargado de proporcionar el giro del brazo, llamado “eslabón 1” o cintura (ver *figura 1.4* y *1.5*). Esta longitud puede hallarse tomando en cuenta que el centro del eslabón 1 se colocará a una distancia de 130 [mm] del centro del tablero (ver *figura 1.5*), entonces dicho punto se encontrará a una distancia de 200 [mm] en la dirección “x” y de 530 [mm] en la dirección “y” (observar *figura 1.5*). Si formamos un triángulo rectángulo con tales magnitudes, obtenemos un hipotenusa cuya magnitud sería de 566.48 [mm].

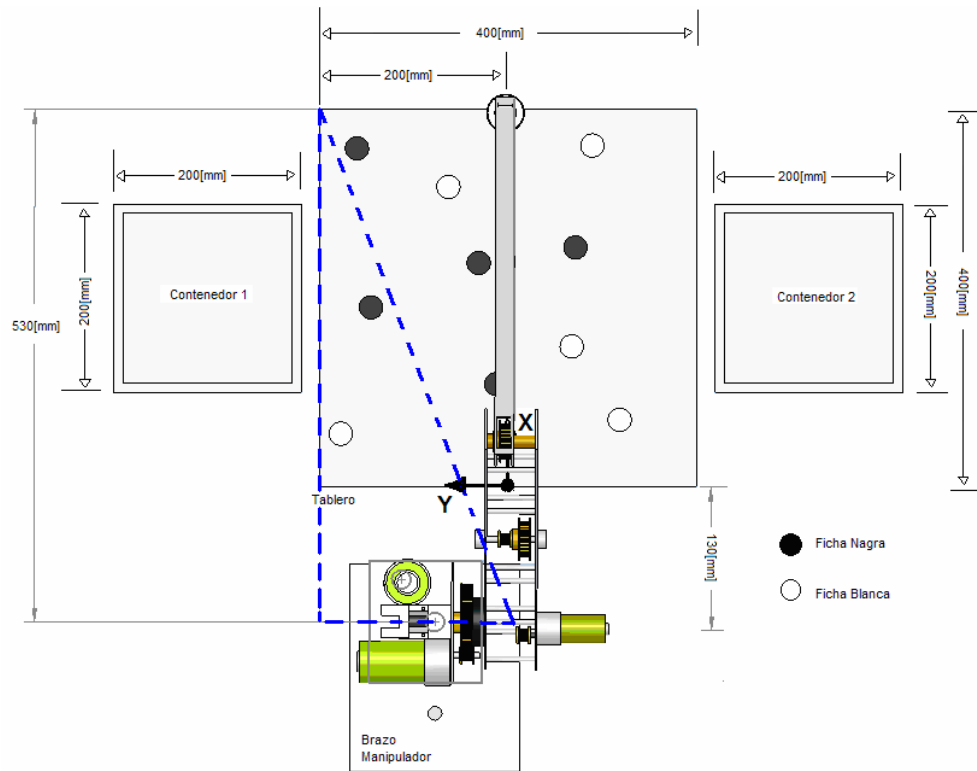


Figura 1.5 “Dimensiones del escenario”

Ahora bien, si la medida de la hipotenusa es de 566.48 [mm], podríamos decir que las longitudes de los eslabones deberán ser, como mínimo, de 283.24 [mm]. Esto es cuando dichos eslabones estén completamente extendidos. Sin embargo, en dicho escenario el brazo se encontraría demasiado forzado, logrando con ello alguna situación indeseada, por lo tanto la longitud de los eslabones 3 y 4 deberá ser mayor a 283.24 [mm].

Cabe señalar que la construcción del brazo se realizó a partir de elementos de fácil adquisición y fabricación, así como también de material disponible en el laboratorio de EA. De los elementos mostrados en la *tabla 1.1*, la banda dentada larga, acoplada con sus respectivas poleas y ejes de soporte (ver capítulo 5 “Implementación física del manipulador”), es quién le proporcionará la transmisión al eslabón número 4, su alcance es de aproximadamente 360 [mm]. Por lo tanto, se tomará inicialmente esta medida como longitud de los eslabones 3 y 4.

| Descripción | Material | Ventajas | Observaciones | Unidad |
|-------------------------------|----------|----------------------------------------------------|--------------------------------------|--------|
| Placa 22x18 [cm] | Aluminio | Bajo costo, ligero, maquinable, fácil adquisición. | Parte superior de la base | 1 |
| Placa 22x5.8 [cm] | Aluminio | Bajo costo, ligero, maquinable, fácil adquisición. | Caras laterales de la base | 2 |
| Placa 18x5.8 [cm] | Aluminio | Bajo costo, ligero, maquinable, fácil adquisición. | Cara frontal y trasera de la base | 2 |
| Cople con diámetro de 5 [cm]. | Nylamid | Maleabilidad, durabilidad, rigidez, económico. | Brinda acoplamiento entre eslabones. | 2 |

| | | | | |
|-------------------------------------------|-------------|----------------------------------------------------|--------------------------------------------------------------------------------------|---|
| Poleas dentadas con diámetro de 8 [cm]. | Nylamid | Maleabilidad, durabilidad, rigidez, económico. | Utilizadas para la transmisión del primer y segundo grado de libertad. | 2 |
| Poleas dentadas con diámetro de 4 [cm]. | Nylamid | Maleabilidad, durabilidad, rigidez, económico. | Utilizadas para la transmisión del tercer grado de libertad. | 2 |
| Poleas dentadas con diámetro de 1.5 [cm]. | Nylamid | Maleabilidad, durabilidad, rigidez, económico. | Acopladas en los actuadores. | 4 |
| Lamina de 12x13 [cm] | Aluminio | Bajo costo, ligero, maquinable, fácil adquisición. | Soporte inferior para montar la transmisión del primer grado de libertad. | 1 |
| Lamina de 12x14 [cm] | Aluminio | Bajo costo, ligero, maquinable, fácil adquisición. | Soporte inferior para montar la transmisión del segundo grado de libertad | 1 |
| Bloque | Aluminio | Bajo costo, ligero, maquinable, fácil adquisición. | Soporte para potenciómetros y eje rotacional de primer y segundo grados de libertad. | 2 |
| Placas 45x5 [cm] | Aluminio | Bajo costo, ligero, maquinable, fácil adquisición. | Utilizadas para formar la estructura de placas paralelas del eslabón 3 | 2 |
| Perfil cuadrado de ¾" | Aluminio | Bajo costo, ligero, maquinable, fácil adquisición. | Utilizado para formar el eslabón 4. | 1 |
| Elemento terminal | Polietileno | Económico, fácil adquisición, ligero y rígido. | Utilizado para formar el eslabón 5 o herramienta final. | 1 |
| Banda dentada corta | Plástico | Resistente, durable, no deformable | Transmisión de potencia para el primer y segundo eslabón. | 2 |
| Banda dentada larga | Plástico | Resistente, durable, no deformable | Transmisión de potencia para el tercer eslabón. | 2 |
| Eje rotacional | Acero | Resistente, durable, no deformable | Ejes rotacionales del primer y segundo grado de libertad | 2 |

Tabla 1.1 “Materiales utilizados para la construcción del robot”

Para la construcción del eslabón 1 y 2, que representan los ejes de rotación del primer y segundo grado de libertad, se utilizaron barras de acero, y sus dimensiones son de 170 [mm] y 65 [mm] respectivamente (observar apéndice A “Plano de fabricación” y capítulo 3 “Implementación física del manipulador”).

Para sostener la herramienta final, el eslabón 5 se maquina con trozos de polietileno y se sujetó con una articulación libre en el extremo del eslabón 4 (ver *figura 1.6*). “De esta manera es posible ahorrarse el uso de un cuarto actuador, manteniendo la vertical por efecto gravitatorio en todo momento” [ES2004].

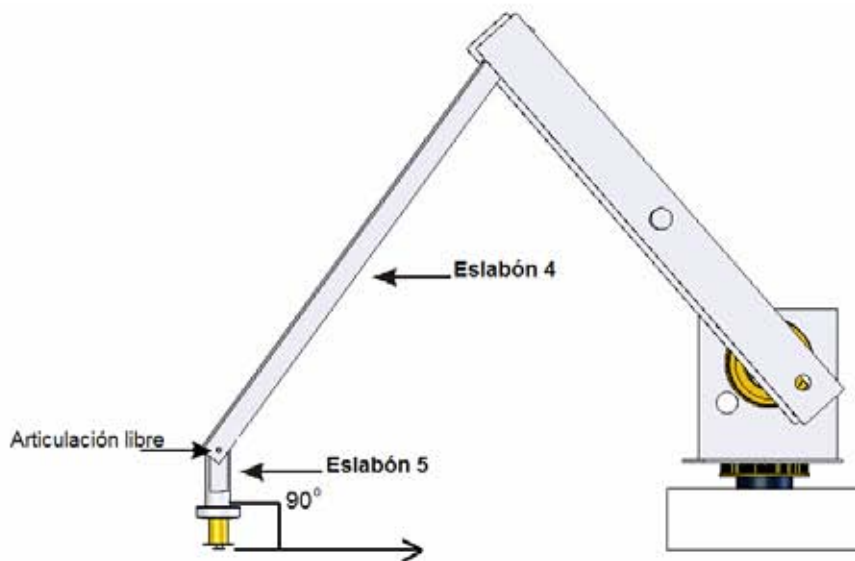


Figura 1.6 “Eslabón 5 unido al eslabón 4 mediante una articulación libre”

Las características que hay que tener en cuenta para el diseño del elemento terminal son: capacidad de carga, fuerza de agarre, geometría, dimensiones de los objetos que debe manejar, tipos de movimientos que debe realizar, tipo de alimentación (neumática, eléctrica, hidráulica) y características de la superficie de contacto. Dado que nuestra tarea es relativamente sencilla, puesto que las fichas no pesan más de 15 [gr] y su diámetro será de 20 [mm], se optó por colocar en el extremo del último enlace del manipulador un electroimán con la finalidad que redujera el grado de complejidad en el control del brazo, ver *figura 1.7*



Figura 1.7 “Electroimán”

Para aproximar las dimensiones del eslabón que sostendrá la herramienta final, el electroimán puede medirse como un cilindro con 278 [mm] de diámetro y 350 [mm] de altura. El eslabón que sostendrá el electroimán se muestra a continuación:

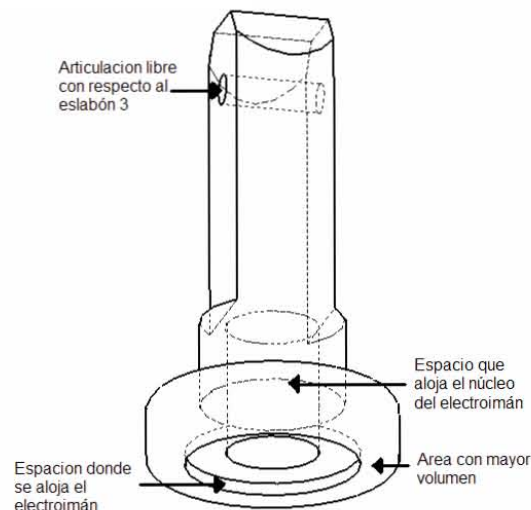


Figura 1.7 “Quinto eslabón o elemento terminal”

El orificio de la parte superior tiene como finalidad alojar los rodamientos que permitan lograr una articulación libre con respecto al eslabón 4. En el área de mayor volumen, en la parte inferior, se sujetará el electroimán. Entre el centro del orificio superior y el extremo del electroimán se tendrán aproximadamente 100 [mm]. Las dimensiones completas de todos los eslabones pueden verse en el apéndice E “Planos de fabricación”.

1.3.1.2 Actuadores

Los actuadores generan las fuerzas o pares necesarios para animar la estructura mecánica. Se utilizan tecnologías de tipo hidráulico, neumático, o bien motores eléctricos, pero en la actualidad se ha extendido el uso en gran medida de estos últimos y en particular motores de corriente directa (CD), servos controlados y motores a pasos.

Un motor eléctrico convierte energía eléctrica en energía mecánica. Los motores de CD son comúnmente usados en brazos manipuladores y robots móviles de baja escala, debido a que éstos pueden energizarse con baterías portátiles de CD; además se pueden encontrar en una gran variedad de tipos y tamaños. Para las necesidades de un robot, un motor de CD usualmente maneja muy altas velocidades y muy bajos torques. Para compensar estas características, se suelen utilizar sistemas de reducción por medio de engranes.

Un pequeño o mediano sistema de engranes puede ser agregado al eje del motor o puede adquirirse un motor con el engrane ya incluido junto con una cubierta del motor. Este tipo de motores son llamados motores con cabeza reductora y como ya se mencionó son muy utilizados para construir robots pequeños y medianos en cuanto a tamaño. Los motores con cabeza reductora están normalmente basados en imanes permanentes sin rotor de hierro para ser lo más ligero posible.

Otro tipo de motores de corriente directa muy usados son los servo motores, este tipo de actuador es más bien un ensamble que incorpora un motor de CD, una reducción, límites de giro, un potenciómetro para retro-alimentar posición y en muchos casos velocidad.

Para nuestra aplicación utilizamos motores de CD con cabezal reductor, debido a sus características citadas anteriormente, aparte de que permiten acoplar de una manera sencilla elementos que pueden sensar la posición, como un decodificador óptico o potenciómetro.

La funcionalidad de nuestra aplicación no requiere de momentos de inercia considerables, una razón más por la cual se opto por utilizar motores de corriente directa. Por otro lado es relativamente sencillo implementar un control de velocidad por medio de señales tipo PWM (“Pulse Width Modulation” por sus siglas en ingles). A continuación se muestran las características de los actuadores utilizados para este proyecto:

Para el primer y segundo grado de libertad se utilizaron motores de 24V con las siguientes características:

| Velocidad [r.p.m] (sin carga) | Velocidad nominal [r.p.m] | Par nominal o torque [N.m] |
|-------------------------------|---------------------------|----------------------------|
| 123 | 106 | 5.92 |

Para el tercer grado de libertad se utilizo un motor, igualmente de 24V, con las siguientes características:

| Velocidad [r.p.m] (sin carga) | Velocidad nominal [r.p.m] | Par nominal o torque [N.m] |
|-------------------------------|---------------------------|----------------------------|
| 162 | 129 | 3.44 |

CAPÍTULO 2

Capítulo 2 Diseño mecánico

2.1 Modelado cinemático del sistema

El análisis cinemático de un brazo mecánico consiste en obtener un modelo que nos permita determinar la dirección o posicionamiento del elemento terminal (herramienta final del sistema) a partir de las variables de cada unión, o viceversa. Una parte importante de esta sección es el uso de la representación Denavit – Hartenberg (DH) para facilitar la obtención del modelo [SM1989].

Existen dos caminos para poder representar cinemáticamente a un sistema mecánico. La primera es conocida como modelado directo y consiste en determinar la posición de la herramienta final, conociendo la longitud de cada eslabón y los ángulos entre ellos. En el segundo método, modelado inverso, se conoce la posición del elemento terminal y las longitudes de los eslabones, con ello podemos determinar los ángulos en cada articulación. Para resolver la problemática planteada en el capítulo uno (“Planteamiento del problema”), en donde se especifica que la ubicación (coordenadas) de las fichas en el tablero es conocida, el método más adecuado es la obtención del modelo inverso, utilizando para ello la formulación DH.

2.1.1 Descripción de la posición

La manipulación a través de robots implica el movimiento de cuerpos rígidos y herramientas a través del espacio. Por tal motivo, es necesario representar de una manera adecuada las posiciones y orientaciones de los componentes que forman un robot. Para lograr esto se debe de emplear un sistema de referencia fijado al origen de nuestro objetivo con tres vectores en posición ortogonal (*ver figura 2.1*). El sistema se puede asignar arbitrariamente. Sin embargo, por conveniencia el origen de este sistema comúnmente se elige en un centro geométrico, o también puede colocarse en un centro de gravedad, o en una de esquinas del los cuerpos.

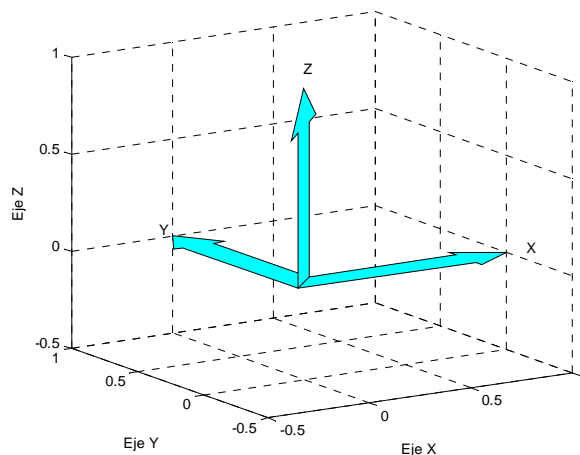


Figura 2.1 “Sistema de referencia en el origen”

Para describir la posición de un cuerpo con respecto a su propia base (referencia) se utiliza la expresión matemática definida por un vector de posición de tamaño 3x1, como se muestra a continuación:

$$p = \begin{pmatrix} px \\ py \\ pz \end{pmatrix} = p_x + p_y + p_z \quad (1)$$

Donde p_i representa la posición sobre el eje i .

Con lo anterior tenemos que los valores del vector de posición dependen de la ubicación (la posición y la orientación) respecto a la referencia asignada.

En términos generales, la posición de un elemento con respecto a su sistema de referencia es:

$${}^N p = \begin{pmatrix} px \\ py \\ pz \end{pmatrix} = {}^N p_x + {}^N p_y + {}^N p_z \quad (2)$$

En donde N representa el sistema de referencia del cuerpo.

2.1.2 Rotación en el sistema cartesiano

Para describir la rotación de una unión en un sistema con movimiento respecto a una referencia en un sistema fijo N . Se tiene que el sistema de referencia N de un cuerpo, esta relacionada al sistema móvil M , y esta dada por la matriz unitaria que se presenta a continuación:

$$R = {}^M R_N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

Cualquier rotación de este cuerpo alrededor de los ejes del sistema de referencia con un ángulo de rotación α (respecto al eje “x”), β (respecto al eje “y”) y θ (respecto al eje “z”) se puede representar con las siguientes matrices de rotación:

La rotación alrededor del eje “x” es:

$$R(X, \alpha) = {}^M R_N = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \text{sen} \alpha & -\text{sen} \alpha \\ 0 & \text{sen} \alpha & \cos \alpha \end{pmatrix} \quad (4)$$

la rotación alrededor del eje “y” es:

$$R(Y, \beta) = {}^M R_N = \begin{pmatrix} \cos \beta & 0 & \text{sen} \beta \\ 0 & 1 & 0 \\ -\text{sen} \beta & 0 & \cos \beta \end{pmatrix} \quad (5)$$

la rotación alrededor del eje “z” es:

$$R(Z, \gamma) = {}^M R_N = \begin{pmatrix} \cos \gamma & -\text{sen} \gamma & 0 \\ \text{sen} \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

Una característica muy importante de las matrices de rotación es que pueden multiplicarse entre ellas obteniendo como resultado una serie de transformaciones del tipo de rotación alrededor de los ejes del marco de referencia. Por ejemplo, si quisiéramos obtener una rotación alrededor del eje “z” y después una rotación respecto al eje “x”, la operación es la siguiente:

$$R(ZX, \theta) = R(Z, \gamma)R(X, \alpha) = {}^M R_N = \begin{pmatrix} \cos \gamma & -\text{sen} \gamma & 0 \\ \text{sen} \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \text{sen} \alpha & -\text{sen} \alpha \\ 0 & \text{sen} \alpha & \cos \alpha \end{pmatrix} \quad (7)$$

2.1.3 Transformaciones homogéneas

En este tipo de transformaciones se involucran tanto desplazamientos como rotaciones con respecto al sistema de referencia asignado. Para representar a este tipo de transformaciones que contiene los movimientos de cada unión, a este tipo de matriz se le llama matriz de transformación homogénea "A". La expresión $X_1 = AX_0$ permite obtener la posición y orientación de un cuerpo ubicado en un sistema "0" con respecto a un sistema "1", a esta se le denomina transformación homogénea. A continuación se muestra su forma general:

$$A = \begin{pmatrix} R_{3 \times 3} & d_{3 \times 1} \\ f_{1 \times 3} & S_{1 \times 3} \end{pmatrix} \quad (8)$$

Donde $R_{3 \times 3}$ representa la rotación del cuerpo rígido, $d_{3 \times 1}$ el desplazamiento o traslación, $f_{1 \times 3}$ la perspectiva, que generalmente tiene un valor de cero y $S_{1 \times 3}$ el factor de escala que tiene un valor unitario.

Para realizar la transformación de n uniones, el procedimiento es simple y solo deben multiplicarse las transformaciones individuales de cada articulación para obtener el total, como se indica en la ecuación siguiente:

$$T = \prod_{i=1}^n A_i \quad (9)$$

Esta expresión nos permite obtener la posición y orientación de un manipulador, mejor conocido como cadena cinemática.

2.1.4 Representación de Denavit – Hartenberg

Con este método se obtiene la formulación sistemática de la relación cinemática de n eslabones sucesivos en una cadena de cuerpos en el espacio [SM1989]. La *figura 2.2* muestra como tres eslabones se relacionan utilizando un mínimo de parámetros:

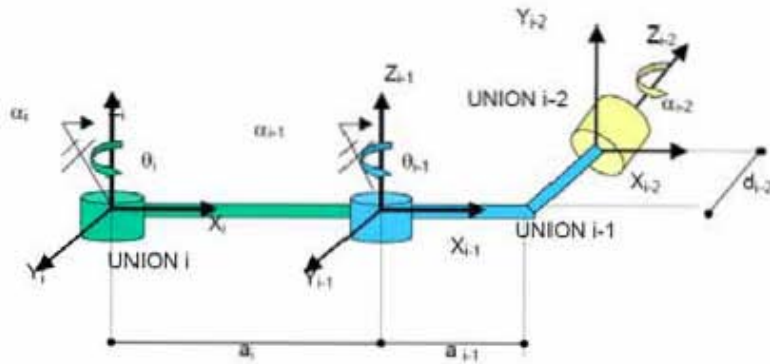


Figura 2.2 “Formulación de los parámetros de Denavit – Hartenberg”

Tomando como base la figura, los parámetros empleados son los siguientes:

a_i : distancia entre los orígenes de los sistemas de referencia.

d_i : distancia entre el origen del eslabón anterior en el sentido del eje “y”.

α_i : ángulo entre el eje z_{i-1} y el eje z_i medido en sentido antihorario alrededor del eje x_i

θ_i : ángulo entre el eje x_{i-1} y el eje x_i medido en sentido antihorario alrededor del eje z_{i-1}

Los parámetros antes mencionados se emplean para la representación Denavit –Hartenberg. El tipo de articulación puede ser rotacional o prismático que respectivamente se indica con los valores θ_i y d_i . El vector a_i es denominado longitud del eslabón $i-1$, su magnitud determina la longitud efectiva del eslabón. Así como el vector α_i es llamado torsión del eslabón.

Para la representación DH, en forma matricial el procedimiento se puede resumir de acuerdo al algoritmo planteado por Spong y Vidyasagar [SM1989].

1. Ubicar y etiquetar los ejes de z_0, \dots, z_{i-1} .
2. Establecer el sistema de referencia base. Colocar el origen en el eje z_0 . Los ejes x_0 y y_0 se eligen convenientemente para formar un sistema de referencia tipo mano derecha.
3. Para $i = 1, \dots, n-1$ efectuar los pasos 3 a 5.
4. Localizar el origen o_i donde la normal común a z_i y z_{i-1} tiene su punto de intersección con z_i . Si z_i cruza a z_{i-1} , ubicar a o_i en esta intersección. Si z_i y z_{i-1} son paralelo, localizar o_i en esta unión.
5. Establecer x_i a lo largo de la normal entre z_i y z_{i-1} hasta o_i , o en la dirección normal al plano $z_i - z_{i-1}$ y en la intersección z_i .

6. Establecer y_i para complementar al sistema de referencia tipo mano derecha.
7. Establecer el sistema de referencia del efector final $o_n x_n y_n z_n$. Si la n -ésima unión es de revolución asignar $kn=a$ a lo largo de la dirección z_i . Establecer el origen o_n convenientemente a lo largo de z_n , preferir al centro de la herramienta al final del brazo.
8. Crear una tabla conforma a los parámetros de unión a_i, d_i, α_i y θ_i .
9. Formar las matrices de transformación homogénea A , sustituyendo los parámetros antes indicados.
10. Formar la matriz

$$T_n^0 = A_1 \dots A_n \quad (10)$$

Este será el resultado de la posición y la orientación del efector final expresado en coordenadas cartesianas.

La representación de DH comúnmente usada en aplicaciones de robots se debe al producto de cuatro transformaciones básicas, como puede observarse:

$$A_i = Rot_{z, \theta_i} Tras_{z, d_i} Tras_{x, a_i} Rot_{x, \alpha_i} \quad (11)$$

A continuación se muestra el producto de estas matrices:

$$= \begin{pmatrix} C_{\theta_i} & -S_{\theta_i} & 0 & 0 \\ S_{\theta_i} & C_{\theta_i} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & C_{\alpha_i} & -S_{\alpha_i} & 0 \\ 0 & S_{\alpha_i} & C_{\alpha_i} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (12)$$

$$= \begin{pmatrix} C_{\theta_i} & -S_{\theta_i} C_{\alpha_i} & S_{\theta_i} S_{\alpha_i} & a_i C_{\theta_i} \\ S_{\theta_i} & C_{\theta_i} C_{\alpha_i} - C_{\theta_i} S_{\alpha_i} & a_i S_{\theta_i} \\ 0 & S_{\alpha_i} & 0 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (13)$$

2.1.5 Parámetros Denavit – Hartenberg para el sistema

En el capítulo pasado se planteo el uso de tres grados de libertad y la utilización de una configuración tipo antropomórfica como estructura base del manipulador, pero para poder representar de manera correcta los parámetros DH se tomaron como articulaciones rígidas los desfases de los diferentes elementos, así como el tipo de movimiento que efectúan. También se considero un sistema de referencia adecuado para fijar correctamente los parámetros y su ubicación. Para simplificar su obtención se fijaron los ejes de movimiento como perpendiculares.

A continuación, en la *figura 2.3* se muestran los ejes de cada uno de los sistemas de referencia en cada unión del manipulador de acuerdo a la representación de los parámetros DH en forma matricial:

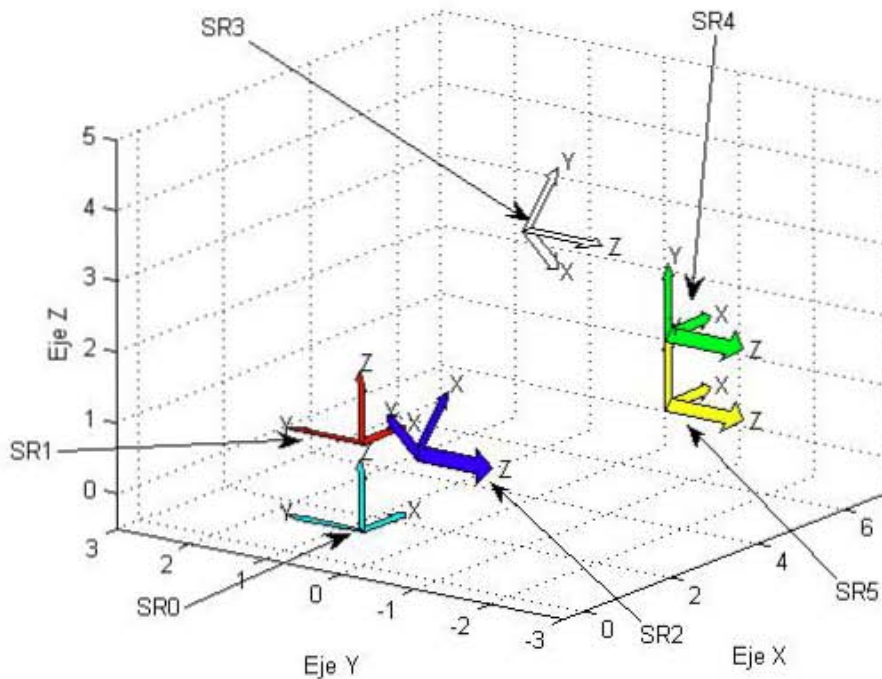


Figura 2.3 “Configuración y sistemas de referencia del manipulador”

Donde:

SR0: Sistema de referencia 0, fijado en la base del robot (eslabón cero).

SR1: Sistema de referencia 1. La distancia mas corta, medida en el eje “z” del SR0 (eje rotacional del primer grado de libertad), entre el origen del SR0 y el origen del SR1, es la longitud efectiva del primer eslabón.

SR2: Sistema de referencia 2. La distancia mas corta, medida en el eje “y” del SR1 (parte negativa), entre el origen del SR1 y el origen del SR2, es la longitud efectiva del segundo eslabón.

SR3: Sistema de referencia 3. La distancia mas corta, medida en el eje “x” del SR2, entre el origen del SR2 y el origen del SR3, es la longitud efectiva del tercer eslabón.

SR4: Sistema de referencia 4. La distancia mas corta, medida en el eje “x” del SR3, entre el origen del SR3 y el origen del SR4, es la longitud efectiva del cuarto eslabón.

SR5: Sistema de referencia 5. La distancia mas corta, medida en el eje “y” del SR4 (parte negativa), entre el origen del SR4 y el origen del SR5, es la longitud efectiva del quinto eslabón (órgano terminal).

En base al tipo de configuración antropomórfica y sistemas de referencia fijados anteriormente se determinaron los parámetros DH del sistema:

| Eslabón | α_i [rad] | a_i [cm] | θ_i [rad] | d_i [cm] | R/P |
|---------|------------------|------------|------------------|------------|-----|
| 0 | 0 | a_0 | θ_0 | 0 | 0 |
| 1 | 0 | 0 | 0 | d_1 | 0 |
| 2 | $\pi/2$ | 0 | θ_2 | d_2 | 0 |
| 3 | 0 | a_3 | θ_3 | 0 | 0 |
| 4 | 0 | a_4 | 0 | 0 | 0 |
| 5 | $-\pi/2$ | 0 | 0 | $-d_5$ | 0 |

Tabla 2.1 “Parámetros Denavit - Hartenberg para el modelo cinemático”

2.1.6 Modelo cinemático del sistema

Para el modelo se obtienen las matrices de transformación homogénea a partir de los parámetros DH, las cuales se muestran a continuación:

$$A = \begin{pmatrix} C_{\theta_0} & -S_{\theta_0} & 0 & a_0 \\ S_{\theta_0} & C_{\theta_0} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(14)

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(15)

$$C = \begin{pmatrix} C_{\theta_2} & -S_{\theta_2} & 0 & 0 \\ 0 & 0 & -1 & -d_2 \\ S_{\theta_2} & C_{\theta_2} & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(16)

$$D = \begin{pmatrix} C_{\theta_3} & -S_{\theta_3} & 0 & a_3 \\ S_{\theta_3} & C_{\theta_3} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(17)

$$E = \begin{pmatrix} 1 & 0 & 0 & a_4 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(18)

$$F = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -d_5 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(19)

La matriz de transformación total para el modelo, tomando como base las matrices homogéneas anteriores, es la siguiente (la transformación va desde la articulación 0 hasta la 5):

$$E = \begin{pmatrix} C_{\theta_0} C_{\theta_2} C_{\theta_3} - C_{\theta_0} S_{\theta_2} S_{\theta_3} & -S_{\theta_0} & -C_{\theta_0} C_{\theta_2} S_{\theta_3} - C_{\theta_0} S_{\theta_2} C_{\theta_3} & 10C_{\theta_0} C_{\theta_2} S_{\theta_3} + 10C_{\theta_0} S_{\theta_2} C_{\theta_3} + 46C_{\theta_0} C_{\theta_2} C_{\theta_3} - 46C_{\theta_0} S_{\theta_2} S_{\theta_3} + 36C_{\theta_0} C_{\theta_2} + 15/2 S_{\theta_0} \\ S_{\theta_0} C_{\theta_2} C_{\theta_3} - S_{\theta_0} S_{\theta_2} S_{\theta_3} & C_{\theta_0} & -S_{\theta_0} C_{\theta_2} S_{\theta_3} - S_{\theta_0} S_{\theta_2} C_{\theta_3} & 10S_{\theta_0} C_{\theta_2} S_{\theta_3} + 10S_{\theta_0} S_{\theta_2} C_{\theta_3} + 46S_{\theta_0} C_{\theta_2} C_{\theta_3} - 46S_{\theta_0} S_{\theta_2} S_{\theta_3} + 36S_{\theta_0} C_{\theta_2} - 15/2 C_{\theta_0} \\ S_{\theta_2} C_{\theta_3} + C_{\theta_2} S_{\theta_3} & 0 & -S_{\theta_2} S_{\theta_3} + C_{\theta_2} C_{\theta_3} & 25/2 + 10S_{\theta_2} S_{\theta_3} - 10C_{\theta_2} C_{\theta_3} + 46S_{\theta_2} C_{\theta_3} + 46C_{\theta_2} S_{\theta_3} + 36S_{\theta_2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

(20)

2.2 Simulación del modelo

A continuación se muestra la simulación, con ayuda de la herramienta de software HEMERO (apéndice B “Hemero Toolbox Robotics” para Matlab), conforme a la representación de los parámetros DH (ver *tabla 2.1*) del manipulador y sus movimientos.

Con ayuda de las matrices del modelo se puede verificar la representación gráfica del brazo en ambientes de 2D (*figura 2.4*) y 3D (*figura 2.5* y *figura 2.6*). También se puede visualizar los eslabonamientos de cada unión de una forma simplificada.

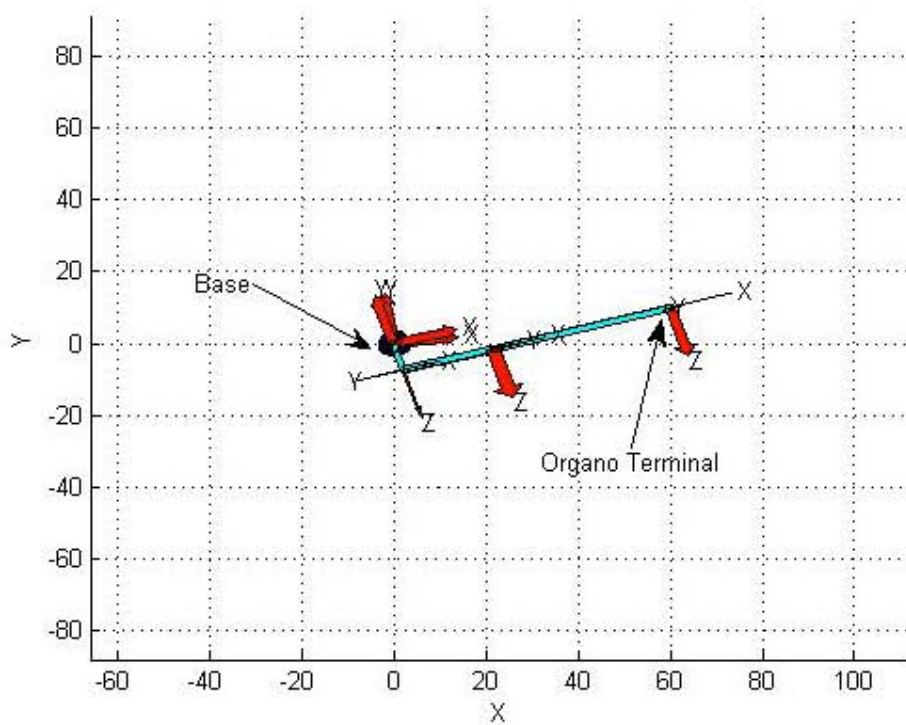


Figura 2.4 “Modelo del sistema de tres grados de libertad en 2D”

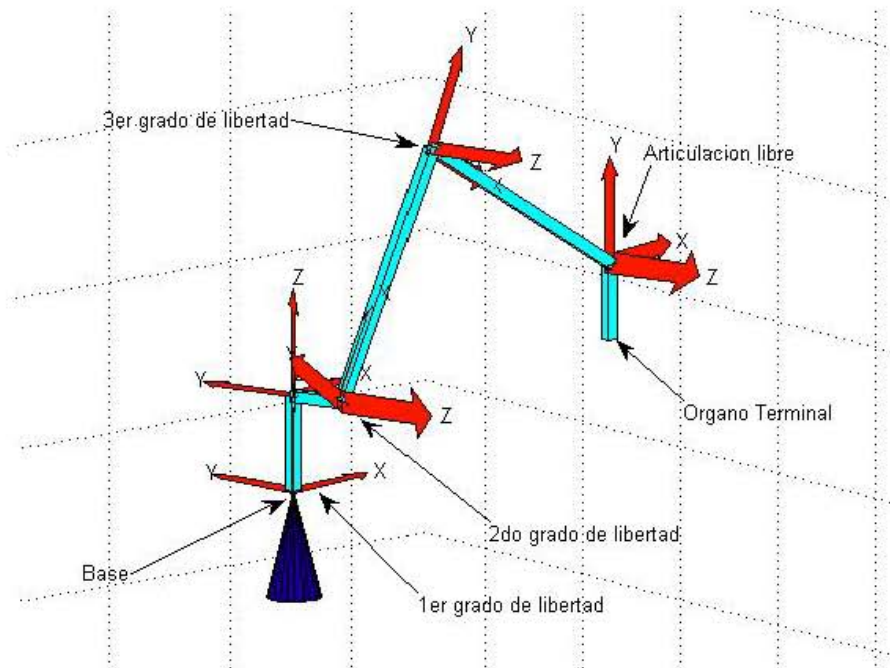


Figura 2.5 “Modelo del sistema en 3D (vista lateral derecha)”

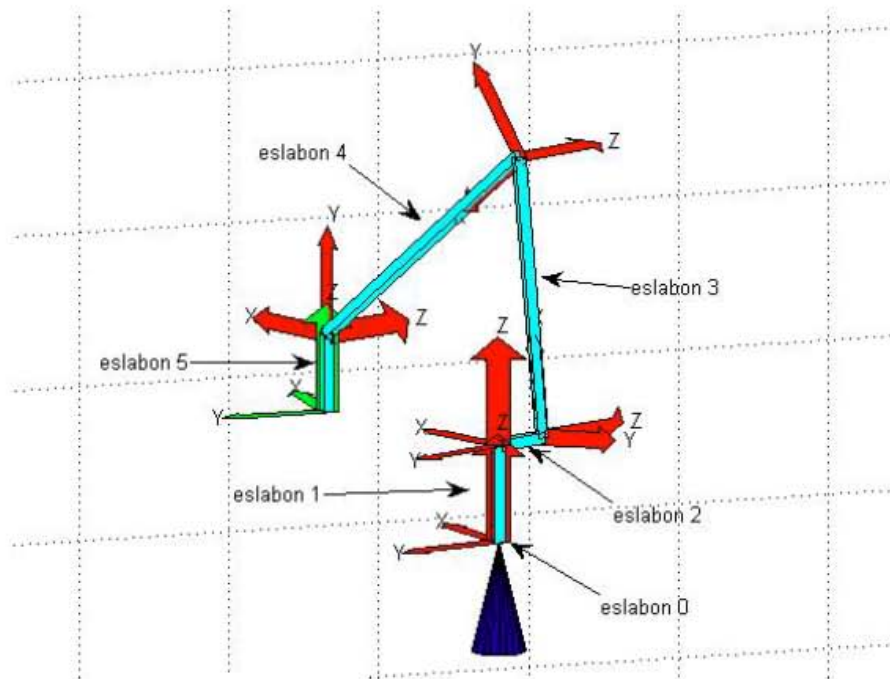


Figura 2.6 “Modelo del sistema en 3D (vista lateral izquierda)”

En las siguientes simulaciones (*figuras 2.7 y 2.8*) se puede observar los movimientos tridimensionales que realiza el robot a lo largo de una trayectoria (recta con orientación de

cero grados) de prueba. Para ello se asignaron datos numéricos a la matriz homogénea total para después realizar el cálculo de la cinemática inversa. El procedimiento anterior nos arrojó los valores angulares para cada uno de los puntos en la trayectoria (ver sección “pruebas y resultados”).

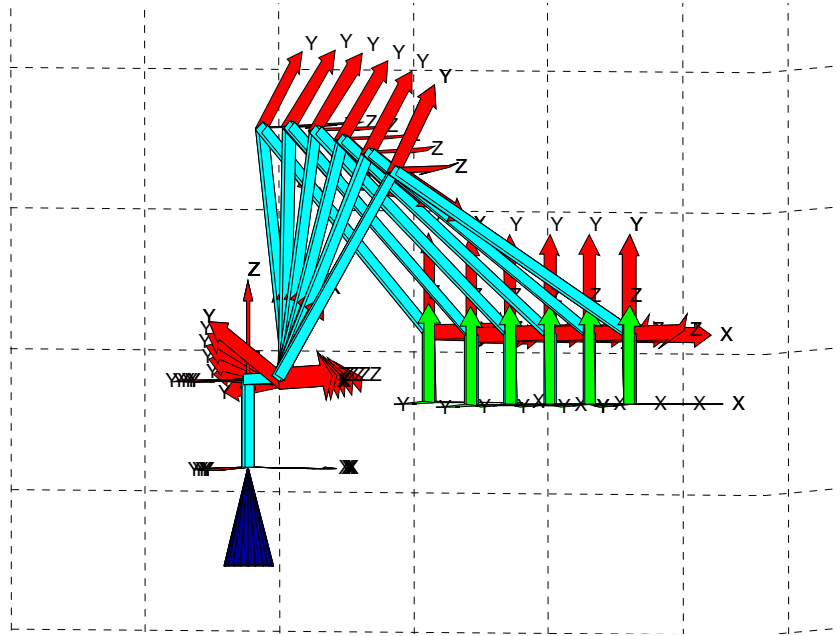


Figura 2.7 “Movimientos tridimensionales del sistema (vista lateral derecha)”

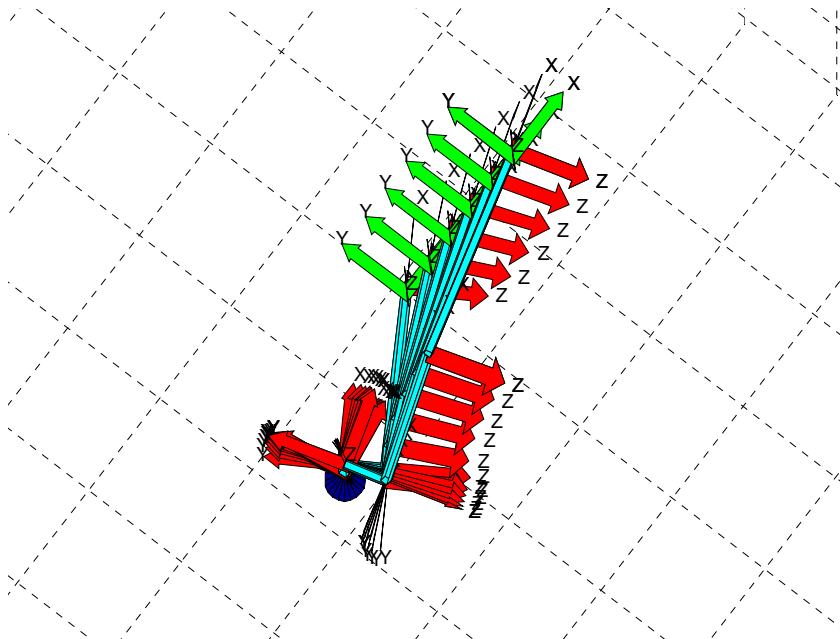


Figura 2.8 “Movimientos tridimensionales del sistema (vista superior)”

CAPÍTULO 3

Capítulo 3 Diseño electrónico

3.1 Etapa de sensado

3.1.1 Sensor de posición del sistema

De manera similar a los seres vivos, los sensores facilitan la información necesaria para que los robots interpreten el mundo real. Todo robot debe tener al menos un sensor para interactuar con su ambiente. Para nuestro sistema en particular, nos interesa medir el ángulo que existe en cada una de las articulaciones del brazo, interpretar esta información mediante el sistema de control, para después generar las señales necesarias para posicionar al brazo en alguna coordenada deseada en el espacio de trabajo.

Existen distintos dispositivos para efectuar dicha tarea, destacado principalmente el uso de codificadores ópticos (“encoders” por sus siglas ingles) y potenciómetros.

Los encoders son dispositivos digitales, pueden ser absolutos (se tiene una señal binaria para cada posición angular) o incrementales (a partir de una posición de referencia, el encoder cuenta el número de pulsos emitidos, cada pulso se tiene una equivalencia en grados). Este tipo de dispositivos no son relativamente económicos, aunque si permiten tener mucha precisión en la medición angular. Otra característica de este sensor, es que su señal de salida es digital, evitándose el uso de un convertidor analógico – digital.

Los potenciómetros son dispositivos analógicos capaces de medir la posición angular y pequeños desplazamientos de posición lineal. Constan de una resistencia a través de la cual hay una determinada diferencia de potencial. Además hay un contacto unido a la resistencia que se puede deslizar a su alrededor; este elemento es conocido como perilla. La perilla se acopla físicamente al elemento cuyo movimiento vamos a medir.

Cuando este elemento se mueva, en dicha perilla se irá variando la resistencia, y por consecuencia la tensión de salida en él. Si medimos esta tensión de salida, podremos determinar cuanto se ha desplazado la perilla, y por lo tanto cuanto se ha desplazado el elemento que pretendemos controlar.

Debido a las características mecánicas de nuestro sistema, particularmente hablando del espacio físico donde se incorporarán los sensores en las articulaciones del brazo (ver capítulo 5 “Implementación física del manipulador”), y a los recursos como convertidor analógico – digital del dispositivo electrónico que fue utilizado para tomar las lecturas de los sensores, la decisión fue incorporar potenciómetros de alta precisión (observar *figura 3.1*), que son capaces de medir la posición angular en las articulaciones del brazo. Además, en cuanto a costo, son considerablemente más económicos que los encoders.

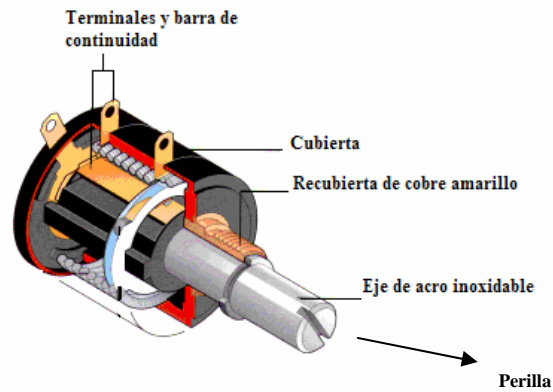


Figura 3.1 “Potenciómetro de alta precisión”

3.1.2 Sensor de color de objetos

La tarea principal del robot es ser capaz de seleccionar piezas blancas y negras, esto es, tomar y colocar estos objetos en su respectivo contenedor dependiendo el color.

Para poder detectar el color de la pieza, el sensor debe presentar las siguientes características:

- Inmunidad a cambios de iluminación del ambiente.
- Capacidad para identificar superficies de alto contraste (blanco-negro).
- Bajo consumo de corriente.
- Estabilidad en valores de salida.
- Tiempo de respuesta mínimo.

Para cubrir los requerimientos anteriores se utilizaron sensores ópticos, que son muy utilizados para aplicaciones de robótica. En éstos, las señales que se transmiten y detectan son infrarrojas. En los sensores ópticos el emisor y el receptor suelen ser elementos separados. El primero suele ser un diodo emisor de luz y el receptor un fototransistor. En la *figura 3.2* se puede observar como la base del fototransistor recibe la señal infrarroja proveniente del diodo emisor.

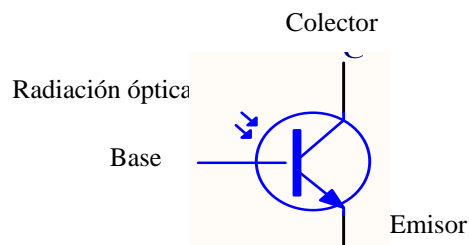


Figura 3.2 “Fototransistor”

El funcionamiento de un fototransistor es el siguiente:

Al exponer el fototransistor a la luz, los fotones entran en contacto con la base del mismo, generando una corriente de base, que hace que el transistor entre en la región activa (saturación), y se presente una corriente de colector a emisor.

Es por este motivo, la terminal correspondiente a la base está ausente del fototransistor. La característica más sobresaliente de un fototransistor es que permite detectar luz y amplificar mediante el uso de un sólo dispositivo.

Otras características de los fototransistores son las siguientes:

- Bajo costo
- Fácil adquisición
- Larga vida útil
- Tiempo de respuesta en microsegundos
- Bajo consumo de potencia

Por todas estas razones utilizamos el fototransistor modelo PT1302B/C2, cuyas características son: 5 [mm] de diámetro, 5 [us] de recuperación, y su rango de detección es de 700 – 1200 [nm], además tiene un filtro que solo permite el paso de luz infrarroja emitida por el diodo emisor, que en este caso utilizamos el tipo led infrarrojo azul transparente de 5 [mm] de diámetro, con longitud de onda de 940[nm].

El arreglo utilizado se muestra en la siguiente figura:

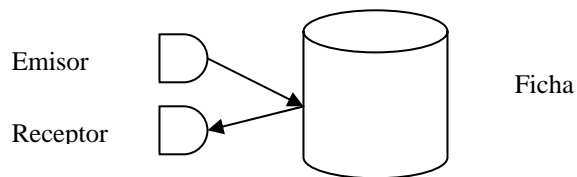


Figura 3.3 “Arreglo utilizado en el sensor infrarrojo”

En este arreglo el haz de luz infrarroja sale del emisor hacia la ficha (blanca o negra), parte de la energía emitida es reflejada por la superficie hacia el receptor, la cantidad de luz infrarroja reflejada al receptor depende del color de la ficha, el color blanco refleja más que los colores oscuros. En la *figura 3.4* se muestra la configuración final utilizada para el par emisor-receptor:

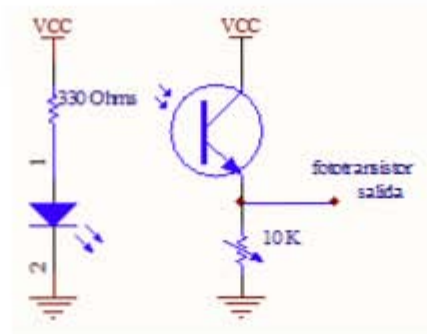


Figura 3.4 “Configuración del par emisor - receptor”

3.2 Etapa del microcontrolador

Antes de definir el tipo de microcontrolador a utilizar se plantean las acciones que el dispositivo electrónico debe realizar a partir que se energiza el circuito:

1. Inicialización de variables y configuración de los recursos necesarios para leer los sensores, manejar la comunicación con la computadora y controlar los actuadores (modulo convertidor analógico – digital, temporizadores, modulo de comunicación serial, modulo generador de señales PWM).
2. Indicar a la computadora que se encuentra listo para recibir información.
3. Recepción de las coordenadas angulares para una trayectoria dada.
4. Dirigir a brazo por la trayectoria dada.
5. Regresar al punto 2.

El microcontrolador a utilizar debe tener las características necesarias para la implementación del algoritmo antes planteado, además de contar con recursos necesarios para controlar los actuadores y leer los transductores implementados en este proyecto.

Actualmente en el mercado existen una gran variedad de microcontroladores que se utilizan en este tipo de aplicaciones, destacando principalmente los ya conocidos PIC de la empresa Microchip, dispositivos fáciles de adquirir y relativamente económicos (ver apéndice D “Hojas de especificación técnica”).

La elección del tipo de dispositivo depende mucho de las prestaciones del mismo, este tipo de características suelen ser: capacidad de memoria para alojar el programa, memoria de datos, número de canales de conversión de señales analógicas a digitales, número de temporizadores disponibles, etc. La siguiente tabla muestra las características de microcontrolador utilizado para este sistema.

| Modelo | PIC 16F877 |
|----------------------------------------------------|------------------------|
| Frecuencia de operación | 20 MHz |
| Memoria flash de programa (palabras de 14 bits) | 8K |
| Memoria de datos (bytes) | 368 |
| Memoria de datos EEPROM | 256 |
| Fuentes de Interrupciones | 14 |
| Puertos I/O | Puertos A,B,C,D,E |
| Temporizadores | 3(1-16 bits, 2-8 bits) |
| Módulos de Captura/Comparación /PWM | 2 |
| Comunicación serie | Si |
| Comunicación en paralelo | Si |
| Canales de conversión A/D | 8(10 bits) |

Tabla 3.1 “Características del microcontrolador del sistema”

A continuación se muestra, en la *figura 3.5*, el esquema de conexión de dicho microcontrolador, en donde podemos destacar el circuito de reset conectado a la terminal numero uno del PIC, el oscilador de 20 [MHz] y las líneas de recepción y transmisión para la comunicación serial.

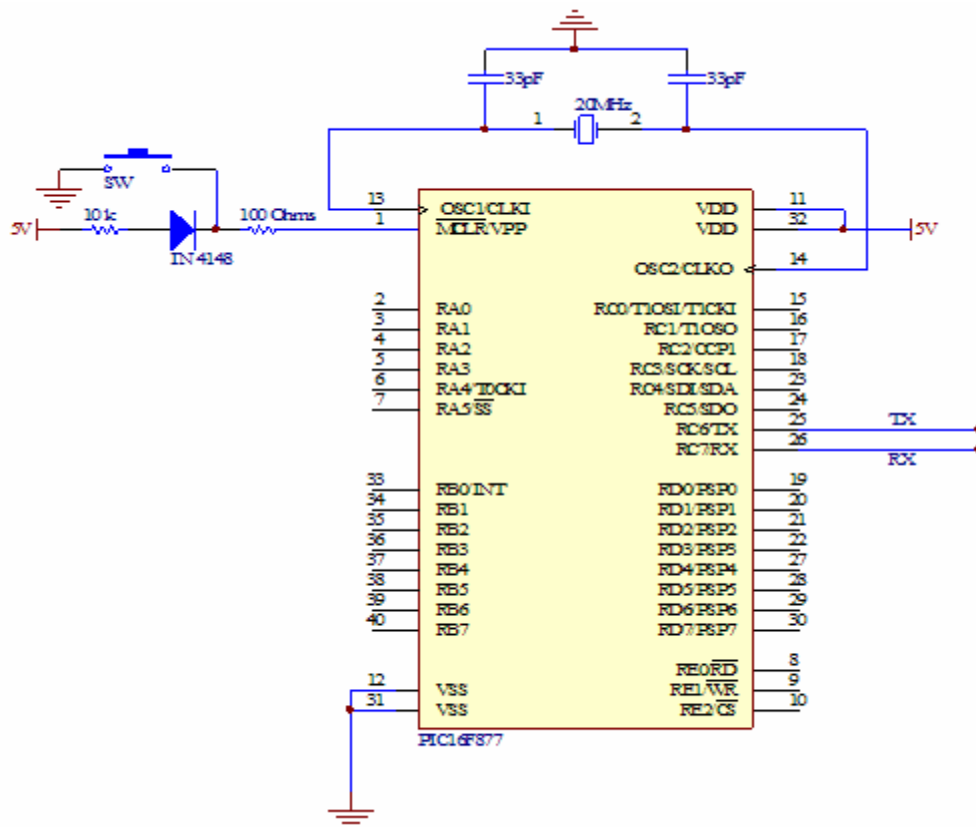


Figura 3.5 “Esquema eléctrico del sistema minino de operación del pic”

Como podemos observar, el PIC requiere de muy pocos elementos para operar. Estos son: un reset para reiniciar el programa en ejecución, un oscilador externo de 20 [MHz] para la frecuencia de operación. También se puede observar en el esquema las líneas de RX y TX, mismas que son utilizadas para la recepción y transmisión de información con la computadora. Ésta comunicación entre dispositivos es del tipo serie y esta basada en la interfaz RS-232. La implementación de ésta interfaz se observa en la *figura 3.6*, y para ello se utilizó el dispositivo max232 que convierte los niveles de voltaje TTL (0-5 volts), del microcontrolador, a niveles de ± 12 volts utilizados por la computadora.

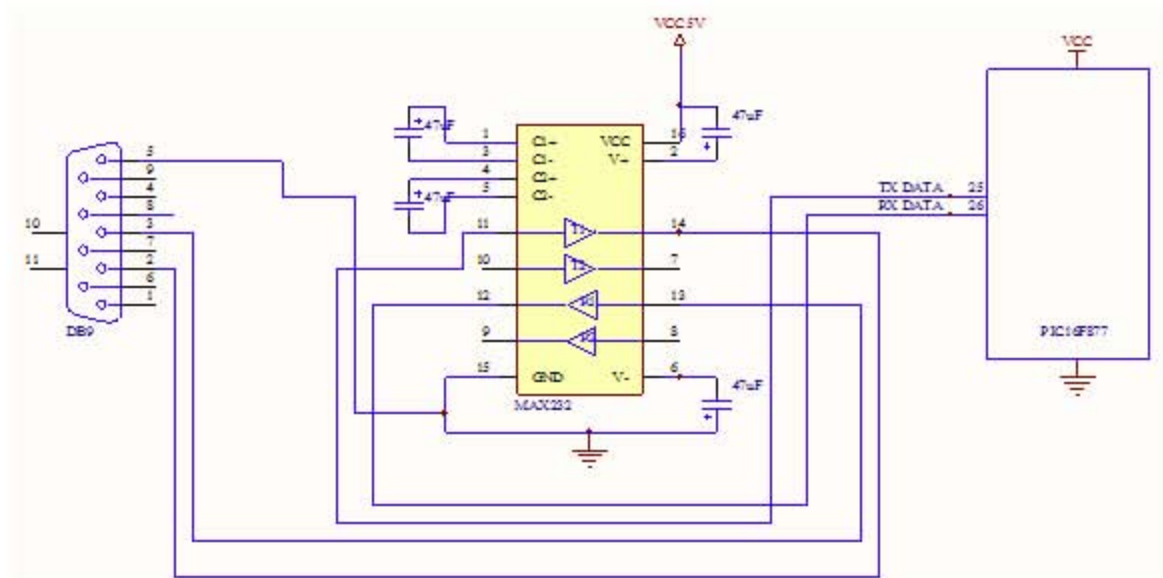


Figura 3.6 “Esquema eléctrico de la Interfaz RS-232”

La programación del microcontrolador se realizó mediante “programación serie en circuito” (ICSP, por sus siglas en inglés “In Circuit Serial Programming”), para ello previamente debe ser grabado, en la memoria flash del PIC, un pequeño programa de 200 bytes, conocido como bootloader, que permite auto programar la memoria del microcontrolador con ayuda de un programa llamado “pic downloader”, ejecutado desde la computadora, cuyas características se detallan en el apéndice B “Pic downloader”.

3.3 Circuito electrónico para el electroimán

El elemento encargado de tomar las fichas de la prueba es un electroimán, dispositivo que representa una carga inductiva para el sistema, esto genera, al igual de los actuadores, otra fuente de ruido para el microcontrolador. Por lo anterior se tuvo que diseñar un circuito basado en un triac capaz de manejar dicha carga. Este circuito consta de dos partes, unidas por un opto – acoplador modelo MOC 3011, la primera de ellas representa el señal de control proveniente del PIC, y la segunda el triac conectado en serie con la carga inductiva

y la fuente de alimentación de 120 [V AC] y 60 [Hz]. El esquema de este circuito puede observarse en la *figura 3.7*

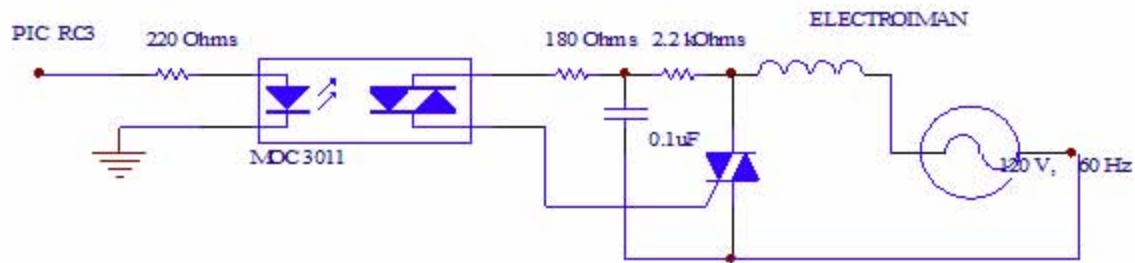


Figura 3.7 “Esquema eléctrico del circuito que controla al electroimán”

3.4 Etapa de potencia

Un dispositivo de control, como lo es un microcontrolador, por si solo no puede manejar directamente a un motor, ya que no puede suministrar suficiente corriente de manera directa. Para lo anterior existen circuitos integrados especialmente diseñados para entregar la corriente demandada por una carga inductiva, de manera que únicamente el control de las señales derive de un microcontrolador. Existen diversos circuitos que pueden ser implementados con una variedad de tecnologías, como pueden ser transistores bipolares, mosfets, relevadores y circuitos especialmente dedicados al control de motores de corriente directa, como los conocidos L293B, L293D y L298, variando, uno del otro, el voltaje de carga y la corriente que son capaces de proporcionar.

Tomando en cuenta las características de los motores de CD, que moverán la estructura mecánica del manipulador, propuestos en el primer capítulo, se optó por utilizar el circuito L298, el cual puede controlar hasta dos cargas de manera independientemente, además de estar diseñado para aceptar, como entradas, señales con niveles de voltaje TTL. Cuenta con dos entradas habilitadoras capaces de activar o desactivar el dispositivo, independientemente de las señales de entrada. Éste circuito soporta cargas de hasta 2 [A] por canal. Los motores utilizados para este proyecto, en especial, el que controla el segundo grado de libertad, en puntos singulares ⁽¹⁾ llega a demandar hasta 2.8 [A], por lo que se aumentó la capacidad de corriente del dispositivo conectando en paralelo los dos canales del circuito, con esto se logró aumentar la entrega de corriente de 2[A] a 3.5[A]. El esquema de la etapa de potencia se muestra en la *figura 3.8*.

(1). Físicamente los puntos singulares son aquellos en donde los torques de los motores aumentan drásticamente, lo cual demanda mayor corriente a los circuitos de potencia.

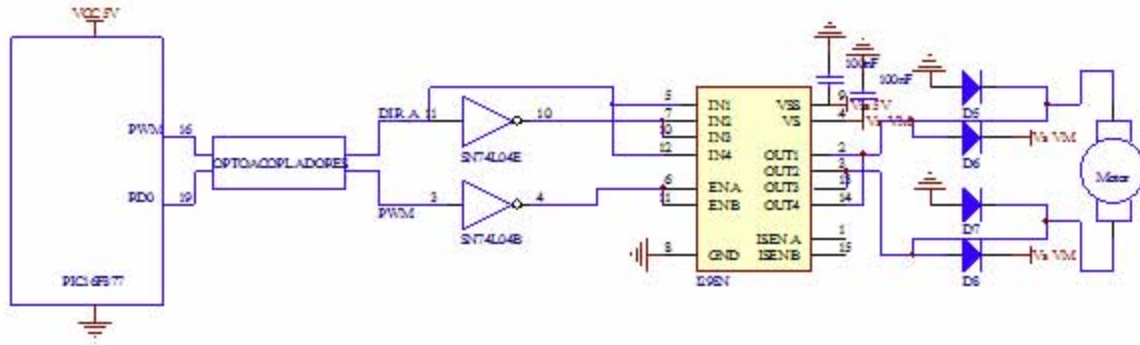


Figura 3.8 “Esquema eléctrico de la etapa de potencia de uno de los motores”

Para evitar el ruido electromagnético, comúnmente generado por los motores, en el microcontrolador, y por consecuencia el mal funcionamiento del brazo, se separaron las fuentes de alimentación de la etapa de control y la etapa de potencia. Para llevar a cabo lo anterior se implementaron circuitos opto - acopladores modelo H11L1 que garantizan un tiempo de switcheo menor a 4 [us], esto es importante, ya que además de opto acoplar las señales que controlan la dirección de los motores, también acoplan las salidas del microcontrolador que generan señales tipo PWM, con una frecuencia de 1.2 [KHz], que se encargan de regular la velocidad de actuadores. El esquema eléctrico de este circuito se muestra a continuación:

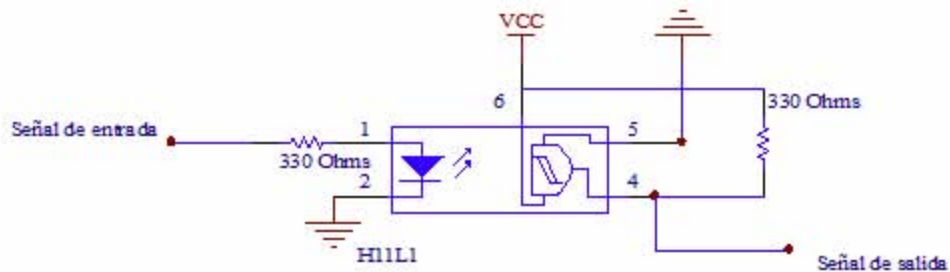


Figura 3.9 “Esquema eléctrico del opto - acoplador”

Las hojas de especificación técnica de los circuitos utilizados, para cada una de las etapas vistas anteriormente, se pueden observar en el apéndice D “Hojas de especificación técnica”.

CAPÍTULO 4

Capítulo 4 Diseño del software de control

El presente capítulo describe el diseño de software de control del brazo manipulador, en donde destaca el desarrollo del algoritmo de control de posición, modulo necesario para que el brazo logre obtener un desempeño óptimo en la tarea que debe realizar, definida en el capítulo 1.

Partiendo de que las coordenadas de las fichas situadas en el tablero son conocidas, es necesario realizar los cálculos mediante el modelo cinemático inverso para obtener los ángulos necesarios, en las articulaciones del brazo, para llevar la herramienta final a la posición de dichas piezas, y su traslado a su respectivo contenedor.

Para el diseño del software de control, se planteo el desarrollo de dos módulos principales: interfaz gráfica de usuario, en donde se debe capturar la información necesaria para la tarea a realizar, como las coordenadas de las fichas, parámetros de configuración para el puerto serie, etc; y como segundo modulo, control de posición, donde a partir de dichas coordenadas en el tablero, se resuelve el modelo cinemático inverso para encontrar los ángulos necesarios en cada articulación, así como la generación de trayectorias.

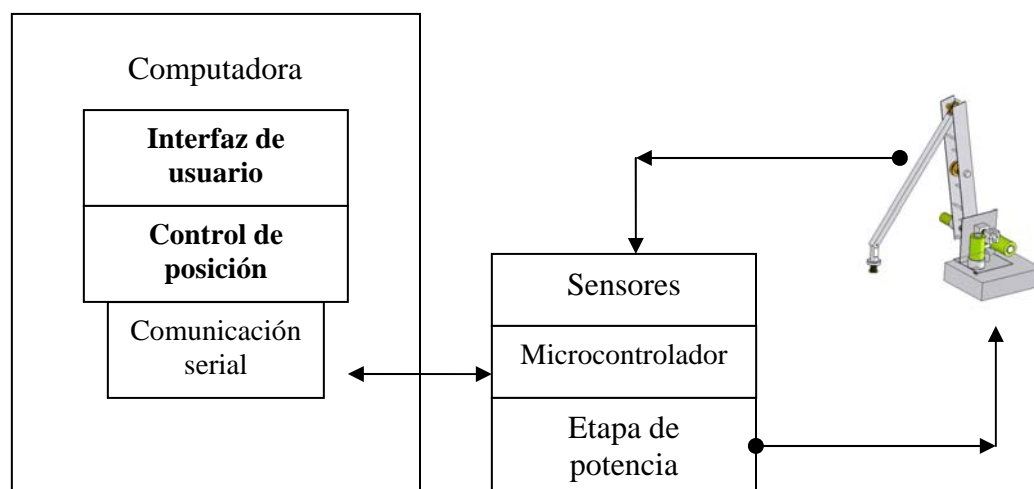


Figura 4.1 “Módulos del software de control”

Para el desarrollo de la interfaz gráfica y el modulo de comunicación serial fue utilizado el ambiente de desarrollo integrado Microsoft Visual Basic (IDE) en su versión 6.0. Se implemento en este lenguaje debido a que posee una gran cantidad de componentes gráficos para desarrollar interfaces de usuario de una manera muy rápida y eficiente, además posee de un componente controlador de puerto serie, elemento utilizado para implementar la comunicación con el microcontrolador.

4.1 Interfaz de usuario

La figura 4.2 muestra la interfaz gráfica para el sistema:

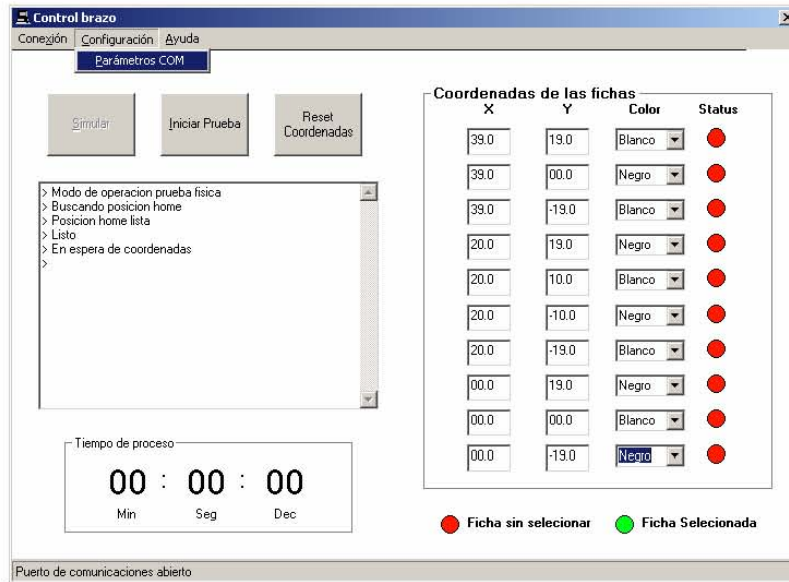


Figura 4.2 “Interfaz de usuario”

La tarea principal de la interfaz gráfica en este proyecto es recibir del usuario las coordenadas (x, y) del tablero donde se localizan las fichas (blancas o negras), que el brazo debe colocar en su respectivo contenedor. Mediante esta pantalla también es posible ingresar el color de la ficha y verificar el status en el que se encuentra, esto es, si ya fue seleccionada y dirigida a su respectivo contenedor, o si aun se localiza en el tablero (ver figura 4.3). Una vez recibida esta información es procesada al modulo de control de posición.

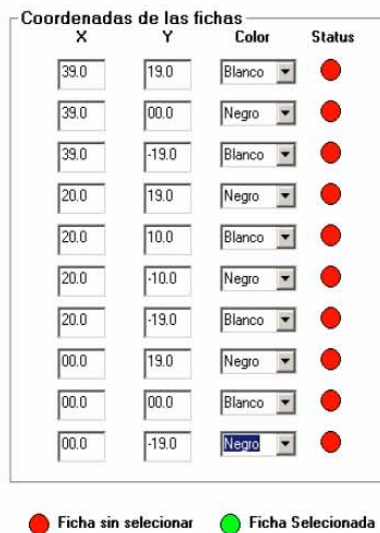


Figura 4.3 “Área de captura de coordenadas por parte del usuario”

La disposición de las fichas en el tablero de 400 x 400 [mm], con resolución de 5 [mm] es como se muestra en el *figura 4.4*, de esta manera es como deben ser ingresadas las coordenadas en la interfaz de usuario.

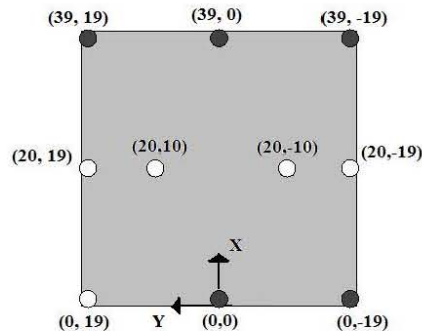


Figura 4.4 “Disposición de la fichas con respecto al sistema de referencia”

Esta interfaz cuenta también con un menú desplegable en el que pueden ser configurados los parámetros de comunicación serial (ver *figura 4.5*). Este recurso es importante debido a que se tiene la libertad de modificar la velocidad de transmisión en baudios (bits por segundo), con el microcontrolador que gobierna físicamente al manipulador, así como otros parámetros como paridad, número de puerto, bits de parada, control de flujo y modo de lectura.

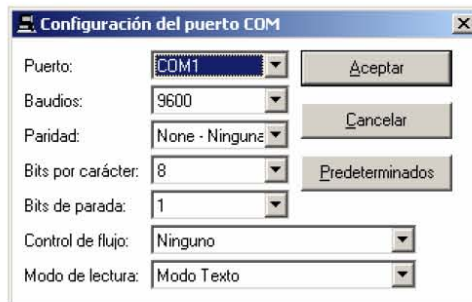


Figura 4.5 “Configuración de los parámetros de comunicación serial”

Para iniciar la prueba es posible hacerlo en dos formas de operación; simulación y prueba física (ver *figura 4.6*). Para la simulación no es necesario activar o abrir el puerto serie, ya que no es requerida la transferencia de información hacia el microcontrolador, por el contrario, se despliega una ventana en la que se muestra una animación en 3D, y en ella se puede observar las trayectorias seguidas por el manipulador (*figura 4.7*). Para iniciar la simulación es necesario introducir las coordenadas de las fichas, el color de las mismas, y presionar el botón correspondiente para comenzar. Para el segundo modo de operación o prueba física, es necesario introducir las posiciones de las fichas, abrir el puerto serie y configurar adecuadamente sus parámetros, una vez hecho esto solo hay que presionar el botón de inicial prueba.

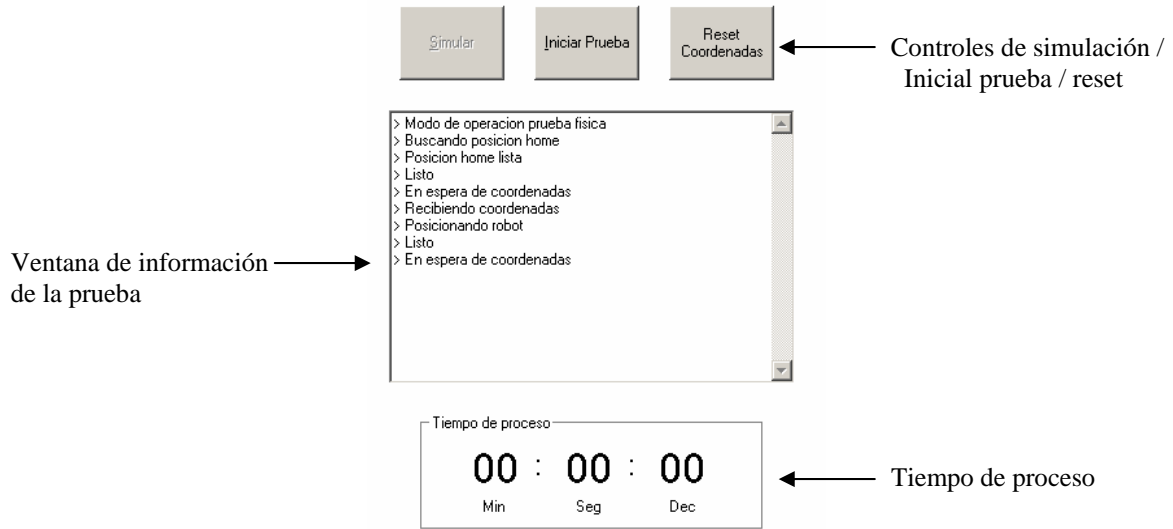


Figura 4.6 “Controles de simulación / prueba, tiempo de proceso y ventana de información de la prueba”

Para monitorear las diferentes fases de la prueba se tiene una ventana de información, así como también de un cronometro en el que se registra el tiempo que tardo el brazo en completar toda su rutina de diez fichas (ver *figura 4.6*).

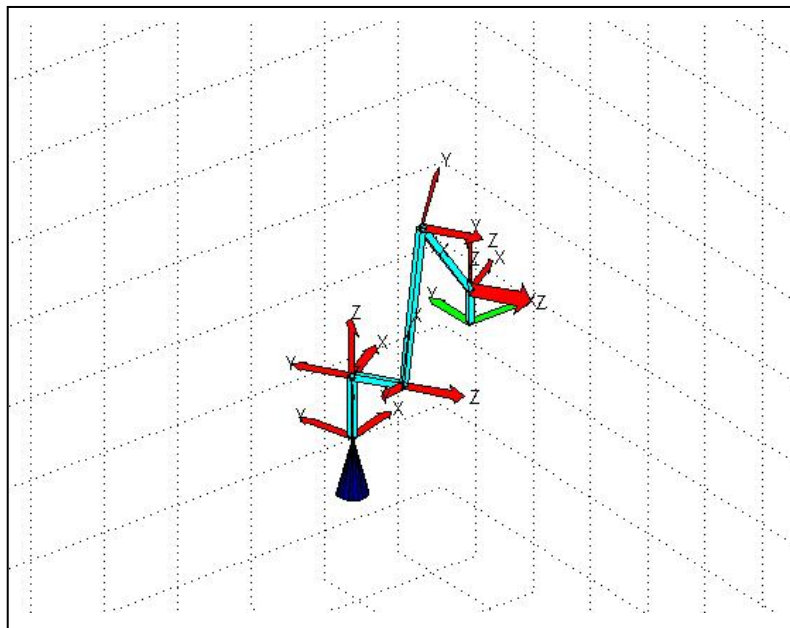


Figura 4.7 “Animación de la prueba en 3D”

4.2 Control de posición

4.2.1 Algoritmo de control de posición

Para poder diseñar el algoritmo que controla la posición de las articulaciones del brazo fue necesario definir el tipo de movimientos con los que se debe mover la herramienta final por encima del tablero de coordenadas. Tomando en cuenta el área de trabajo (ver *figura 1.3*) en la que se mueve el brazo y la ventaja de mantener libre la rotación del último eslabón (*figura 1.6*), en donde se acopla el electroimán que toma las fichas de manera vertical al tablero, se optó por que el seguimiento de trayectorias del elemento terminal fuera en línea recta; esto es, entre la posición actual y la posición deseada diseñar un algoritmo que calcule la línea recta entre esos dos puntos. Una vez obtenida esa función de correspondencia se obtienen todos los puntos $(x(m), y(m), z(m))$ en dicha recta con incrementos de 0.5 [cm] que representa la resolución del tablero. Con cada uno de los puntos de esa trayectoria se obtiene una matriz de transformaciones lineales con la siguiente estructura:

$$T(:, :, m) = \begin{pmatrix} 1 & 0 & 0 & x(m) \\ 0 & 1 & 0 & y(m) \\ 0 & 0 & 1 & z(m) \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

donde “ m ” representa el número de puntos en la trayectoria lineal.

Con la expresión anterior se obtienen “ m ” matrices de transformación, por cada una de las matrices se obtendrán los ángulos, mediante la solución del modelo cinemático del brazo, que deben tenerse en cada articulación para posicionarse en todos los puntos de dicha trayectoria lineal.

Todas las trayectorias horizontales al tablero, que trazará el brazo, serán con una cota constante de 14 [cm], (como sistema de referencia, ver la *figura 4.3*). Las trayectorias verticales serán para bajar y subir por las fichas, en este caso, se definirá una serie de vectores con “ x ” y “ y ” constantes, variando solamente “ z ” a partir de $z = 14$ [cm] y hasta $z = 2.5$ [cm] para bajar, y de $z = 2.5$ [cm] a $z = 14$ [cm] para subir.

La manera en que el brazo selecciona la ficha a tomar será calculando la distancia entre la coordenada actual, que al inicio de la rutina será una posición predeterminada, y la distancia entre cada una de las piezas disponibles en el tablero, ésta longitud siempre será medida en el plano x - y con $z = 14$ [cm].

Un ejemplo de lo anterior se muestra en la *figura 4.8*, en donde la herramienta final se encuentra posicionada en la coordenada $(20, 0, 14)$ o coordenada inicial, en el tablero se localizan tres fichas, donde la ficha 1 se encuentra en $(15, 7, 0)$, la ficha 2 en $(15, 19, 0)$ y la tres en $(15, -19, 0)$, las coordenadas $(20, 30, 14)$ y $(20, -30, 14)$ pertenecen al contenedor de las fichas blancas y al contenedor de las fichas negras, respectivamente. La distancia más cercana al elemento terminal es la de la ficha 1, por lo que el brazo seguirá la siguientes trayectorias lineales entre coordenadas (la ficha 1 es blanca por lo que su respectivo contenedor es el de la izquierda).

$$(20,0,14) \Rightarrow (15,7,14) \Rightarrow (15,7,2.5) \Rightarrow (15,7,14) \Rightarrow (20,30,14) \quad (21)$$

Trayectoria lineal

Con esta trayectoria la ficha es depositada en el contenedor izquierdo, siendo la coordenada $(20, 30, 14)$ la coordenada actual. El proceso se repite “*n*” veces, donde es *n* es el numero de fichas.

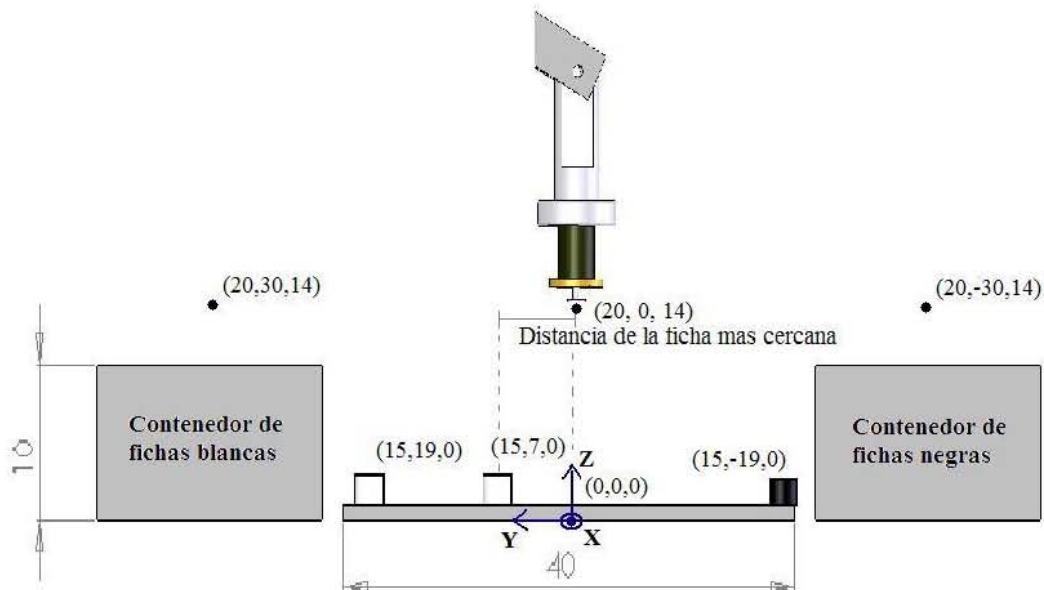


Figura 4.8 “Selección de la ficha más cercana a la coordenada actual de electroimán”

Con la información anterior podemos trazar el algoritmo que debe automatizar el control de posición como se muestra en el siguiente diagrama de flujo:

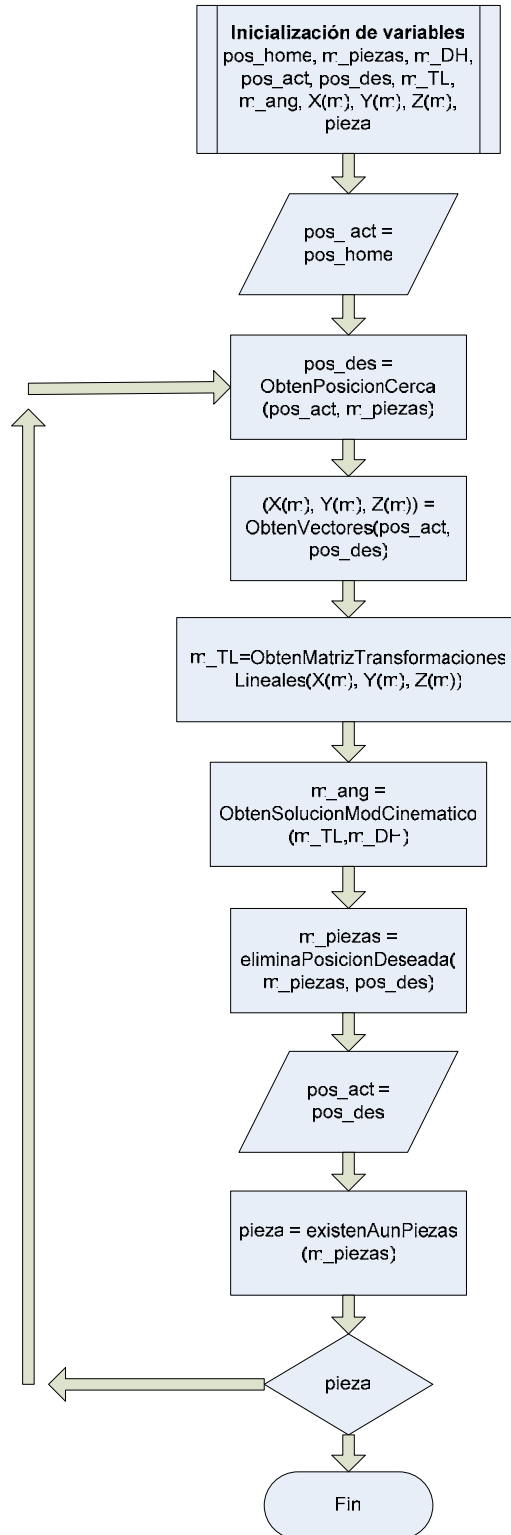


Figura 4.9 “Diagrama de flujo del algoritmo de control de posición del brazo”

Donde:

pos_home = coordenada inicial donde se posicionara el brazo (20, 0, 14).

m_piezas = matriz de coordenadas de orden $m \times n$, donde “ m ” es el número de renglones, que es igual a 10, “ n ” es el número de columnas y es igual a 3.

m_DH = matriz de parámetros de Denavit – Hartenberg (Tabla 2.1).

pos_act = coordenada actual donde se encuentra posicionado el elemento terminal.

pos_des = es el último punto de la trayectoria lineal a donde se dirigirá la herramienta final.

m_TL = matriz de transformaciones lineales para una trayectoria dada.

$X(m)$ = vector de orden $1 \times m$ que contiene el valor de las abscisas (x 's) para cada punto en la trayectoria lineal, donde “ m ” es el número de puntos.

$Y(m)$ = vector de orden $1 \times m$ que contiene el valor de las ordenadas (y 's) para cada punto en la trayectoria lineal, donde “ m ” es el número de puntos.

$Z(m)$ = vector de orden $1 \times m$ que contiene el valor de las cotas (z 's) para cada punto en la trayectoria lineal, donde “ m ” es el número de puntos.

m_ang = matriz de posiciones angulares de orden $m \times n$, donde “ m ” es el número de puntos que existen en la trayectoria lineal y “ n ” es el número de eslabones del manipulador. Ésta matriz es el resultado de resolver el modelo inverso para todos los puntos de la trayectoria deseada.

Éste algoritmo es ejecutado, desde la interfaz de usuario, hasta haber completado la selección de las diez fichas que se encuentran en el tablero.

Una vez obtenidos los ángulos requeridos en cada articulación para una trayectoria de transformación lineal, es necesario realizar la conversión de dichos ángulos, cuyas unidades son radianes [rad], a grados [$^{\circ}$], para después hacer una segunda conversión de cada uno de los grados, a un valor de lectura del convertidor analógico – digital, éstos valores se encuentran dentro del rango de 0 a 2^{10} , resolución del convertidor. A estos últimos valores obtenidos, para fines prácticos, las llamamos coordenadas máquina.

Para poder realizar estas conversiones, fue necesario colocar cada una de las articulaciones de brazo, en tres ángulos diferentes y realizar las mediciones correspondientes en cada sensor de posición. Cada una de las señales de salida, de cada sensor, fue dirigida a un canal de conversión analógico – digital del microcontrolador, arrojando las siguientes lecturas mostradas en la tabla 4.1.

| | Ángulo | Lectura digital del Sensor de posición (10 bits de precisión) Coordenada Máquina |
|----------------|-------------------------|----------------------------------------------------------------------------------|
| Articulación 1 | 90° (1.5707 radianes) | 510 |
| | 0° (0 radianes) | 256 |
| | -90° (-1.5707 radianes) | 2 |
| Articulación 2 | 180° (3.1415 radianes) | 2 |
| | 90° (1.5707 radianes) | 255 |
| | 0° (0 radianes) | 512 |
| Articulación 3 | -45° (-0.7853radianes) | 5 |
| | -90° (-1.5707 radianes) | 180 |
| | -170° (-2.9670radianes) | 355 |

Tabla 4.1 “Datos experimentales”

Para poder calcular la coordenada máquina de cada coordenada angular, fue necesario utilizar los datos obtenidos experimentalmente, para así poder realizar las interpolaciones necesarias.

El procedimiento de interpolación se realizó mediante el método de Diferencias Divididas interpolantes de Newton, cuya fórmula se muestra a continuación:

$$P(x) = \sum_{i=0}^n F(i, i) \prod_{j=0}^{i-1} (x - x_j) \quad (22)$$

Para implementar la fórmula anterior, fue necesario plantear la tabla como la que se muestra a continuación, donde las “x” son los grados y las $F(x)$ son las lecturas digitales del sensor de posición (coordenadas máquina).

| x | F(x) | 1ª Diferencia | 2ª Diferencia |
|-------|----------|-------------------------------------------------|----------------------------------------------------------------|
| x_0 | $F(x_0)$ | $F(x_0, x_1) = (f(x_1) - f(x_0)) / (x_1 - x_0)$ | |
| x_1 | $F(x_1)$ | $F(x_1, x_2) = (f(x_2) - f(x_1)) / (x_2 - x_1)$ | |
| x_2 | $F(x_2)$ | | $F(x_0, x_1, x_2) = (f(x_1, x_2) - f(x_0, x_1)) / (x_2 - x_0)$ |

Tabla 4.2 “1ª y 2ª diferencia del método de interpolación”

De la tabla anterior podemos sustituir en la *formula (21)*, quedando de la siguiente manera:

$$P(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) \quad (23)$$

Una vez sustituidos los valores de la *tabla 4.2* en la *formula (22)*, siendo la “*x*” la variable, se obtiene la coordenada maquina requerida. Para cada articulación se obtienen los datos de la *tabla 4.2* y un polinomio $P(x)$.

A continuación se muestra un ejemplo de la transformación de una coordenada angular ($45^\circ, 70^\circ, -100^\circ$), a su correspondiente coordenada maquina:

| | Ángulo | Coordenada Máquina |
|-----------|--------|--------------------|
| Eslabón 1 | 45° | 383 |
| Eslabón 2 | 70° | 313 |
| Eslabón 3 | -100° | 211 |

Tabla 4.3 “Ejemplo de conversión de coordenada angular a coordenada maquina”

4.2.2 Implementación del algoritmo de control de posición

El modulo fue implementado con la ayuda del toolbox HEMERO (apéndice B “Hemero Toolbox Robotics”) para Matlab (Matriz Laboratory), plataforma interactiva y a la vez muy poderosa para el calculo técnico y científico, en especial el calculo matricial, que es de gran ayuda en la resolución del modelo inverso del manipulador.

El medio de conexión para la interfaz de usuario y el control de posición, se llevo a cabo con la ayuda de otra herramienta de Matlab, llamada “Matlab Automation Server”, utilidad que permite a una aplicación o componente (el controlador) controlar otra aplicación o componente (el servidor). Así, el servidor de Matlab puede ser controlado por cualquier programa en Windows que actúe como controlador. Para nuestro caso, el controlador es la interfaz de usuario escrita en Visual Basic.

El código fuente desarrollado para este modulo, así como la información de cómo conectar el Automation Server de Matlab desde Visual Basic pueden ser observadas en el apéndice C y apéndice B respectivamente.

CAPÍTULO 5

Capítulo 5 Implementación física del manipulador

En este capítulo se muestra cada uno de los elementos que conforman físicamente la configuración del brazo manipulador. Cada una de las piezas que conforman la estructura mecánica del brazo (bandas, poleas, rodamientos, ejes, transductores y actuadores, etc), fueron diseñadas con ayuda del programa SolidWorks (apéndice B “SolidWorks”), paquete utilizado para el diseño de sólidos en 3D.

Una vez realizado el diseño de cada uno de los elementos, se prosiguió a fabricar (maquinar) cada una de las piezas en el centro de manufactura avanzada (CDM) de la facultad de ingeniería de la UNAM, haciendo uso principalmente de equipo de maquinas y herramientas, como son fresadora, taladro de banco y torno principalmente, de tal manera que se permitiera hacer un ensamble con el menor grado de error posible. Los planos de fabricación se pueden observar en el apéndice E.

5.1 Eslabón 1

Para la construcción de éste primer eslabón, se hizo uso de fresadora y torno, la primera se utilizó para maquinar las placas de soporte (sostén del eslabón 1 y 2) y la base del robot (eslabón 0). En el caso del bloque estructural A, el cople de nylamid, y el eje de rotación A se fabricaron con ayuda del torno (ver *figura 5.1*).

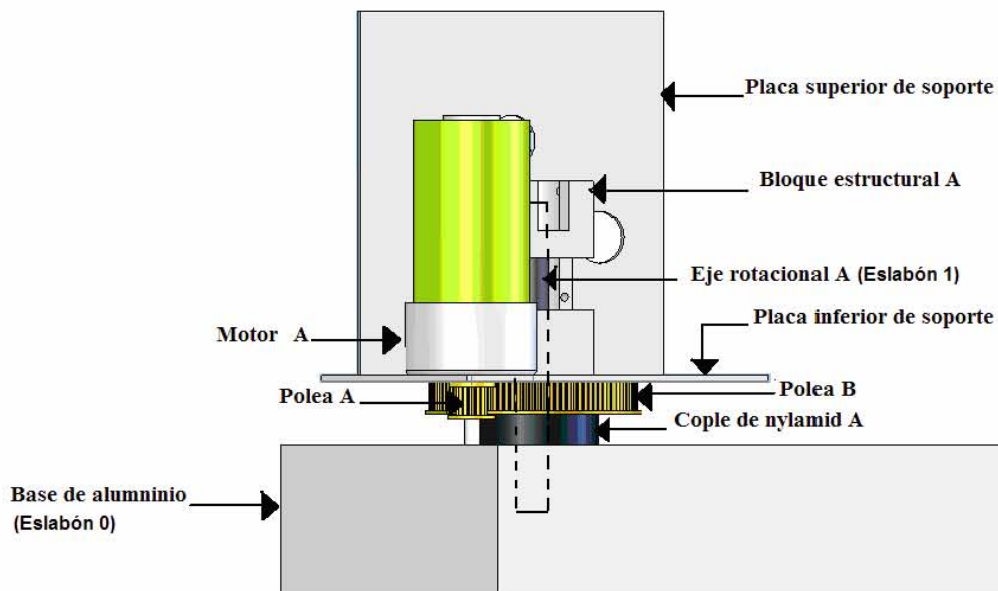


Figura 5.1 “Configuración del eslabón 1 (vista 1)”

En el caso de la base o eslabón 0, elemento que soporta todo el peso del brazo, se eligió utilizar aluminio con un espesor de 6 [mm], con esto garantizamos solidez y fuerza en ésta pieza fundamental del robot. Para las placas inferior y superior de soporte, también se ocupó aluminio, con un espesor menor (2 [mm]). La placa inferior soporta al bloque estructural A (en aluminio), a su vez, el bloque soporta al eje de rotación A de la primera articulación, en la parte superior de ésta pieza, acoplado al eje A, se localiza el sensor de posición A que realiza las mediciones angulares del primer grado de libertad (ver *figura 5.2*).

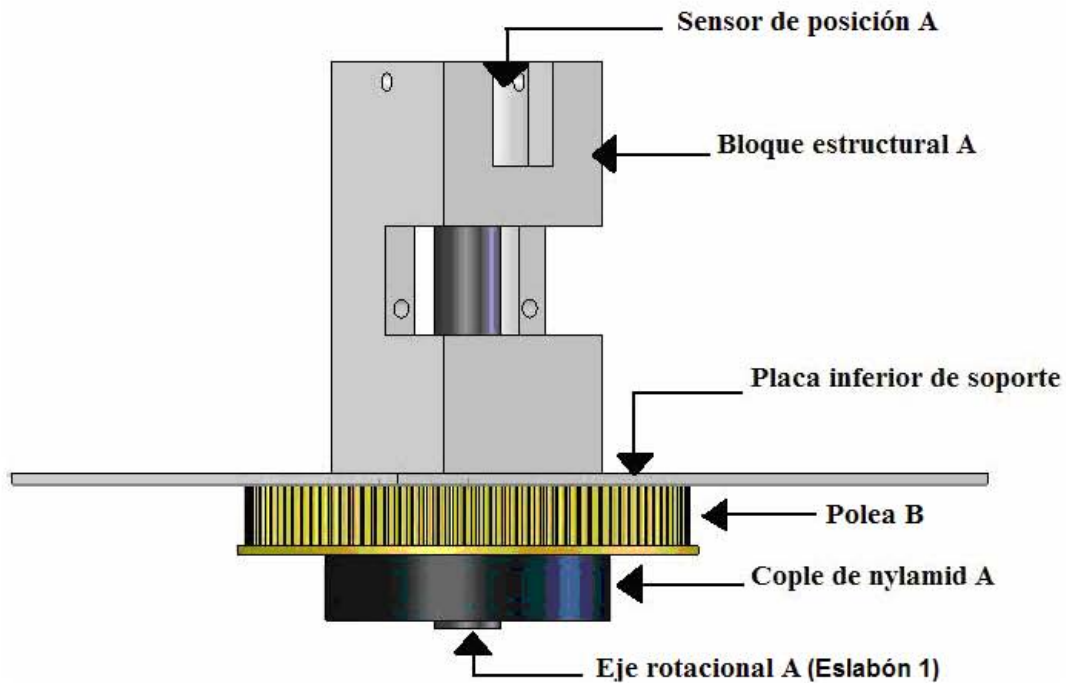


Figura 5.2 “Configuración del eslabón 1 (vista 2)”

El cople de nylamid A es negro, ésta característica significa que es súper lubricado, y con ello se reduce el coeficiente de fricción, con lo cual garantizamos que el juego entre acoplamientos, y los errores en las mediciones angulares, sean mínimos (ver *figura 5.3*). Para construir los ejes de rotación se utilizó acero, con este material se obtiene alta dureza y resistencia en los movimientos rotacionales.

El eje de rotación A, como se muestra en la *figura 5.3*, está sujeto al centro de masa de la polea B y el cople de nylamid A, de esta manera cuando existe transmisión de movimiento de la polea A a la polea B (*figura 5.1*), el eje de rotación A desplaza la perilla del potenciómetro (sensor de posición A), generando así una variación en el ángulo.

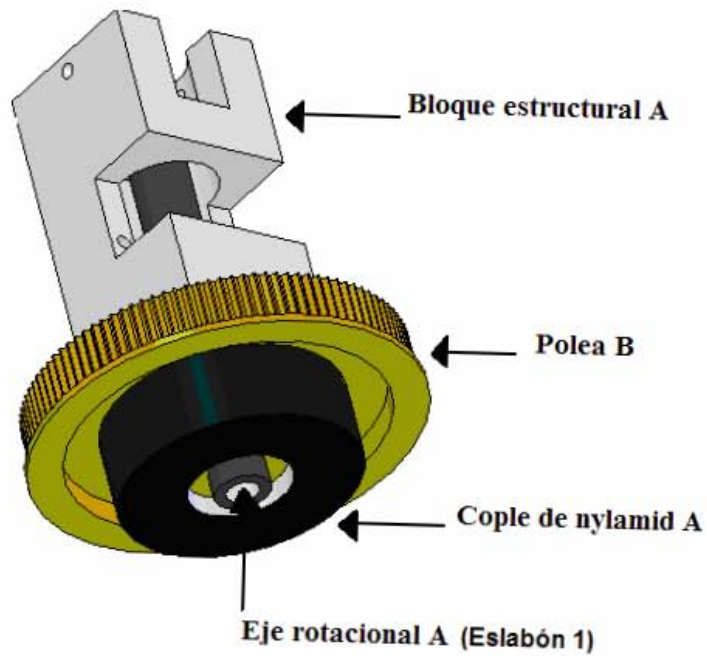


Figura 5.3 “Bloque estructural A y elementos acoplados”

En la *figura 5.4* se muestra el motor A, montado en la placa inferior de soporte, éste actuador es quien acciona a la primera articulación por medio de la banda acoplada entre las poleas A y B.

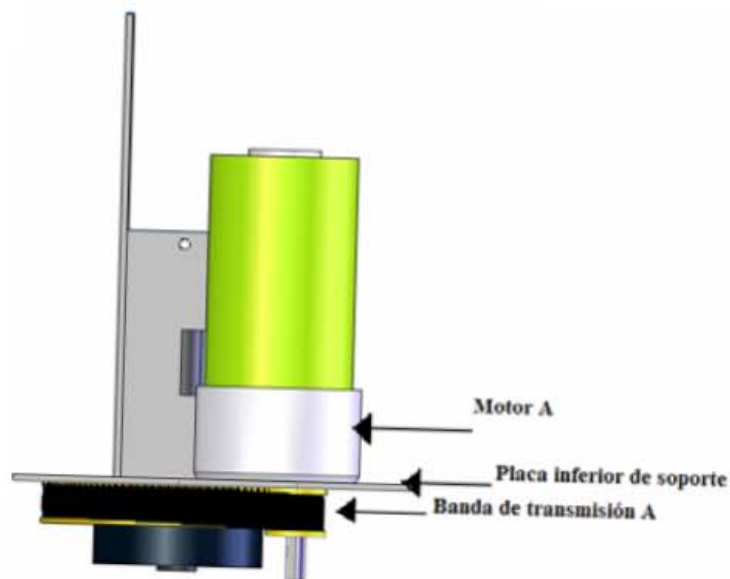


Figura 5.4 “Motor A y transmisión del eslabón 1”

5.2 Eslabón 2

A continuación se muestran los elementos que conforman el eslabón 2, sobre el cual se acopla la transmisión que mueve al eslabón 3. En la *figura 5.5* se puede observar el motor *B*, bloque estructural *B*, eje rotación *B*, polea *C* y *D*, cople de nylamid *B*, todas éstas piezas, ensambladas, permiten que el eslabón 2 pueda rotar sobre su propio eje, y a su vez accionan al eslabón 3. El material utilizado, para éste eslabón, básicamente es el mismo que el eslabón 1.

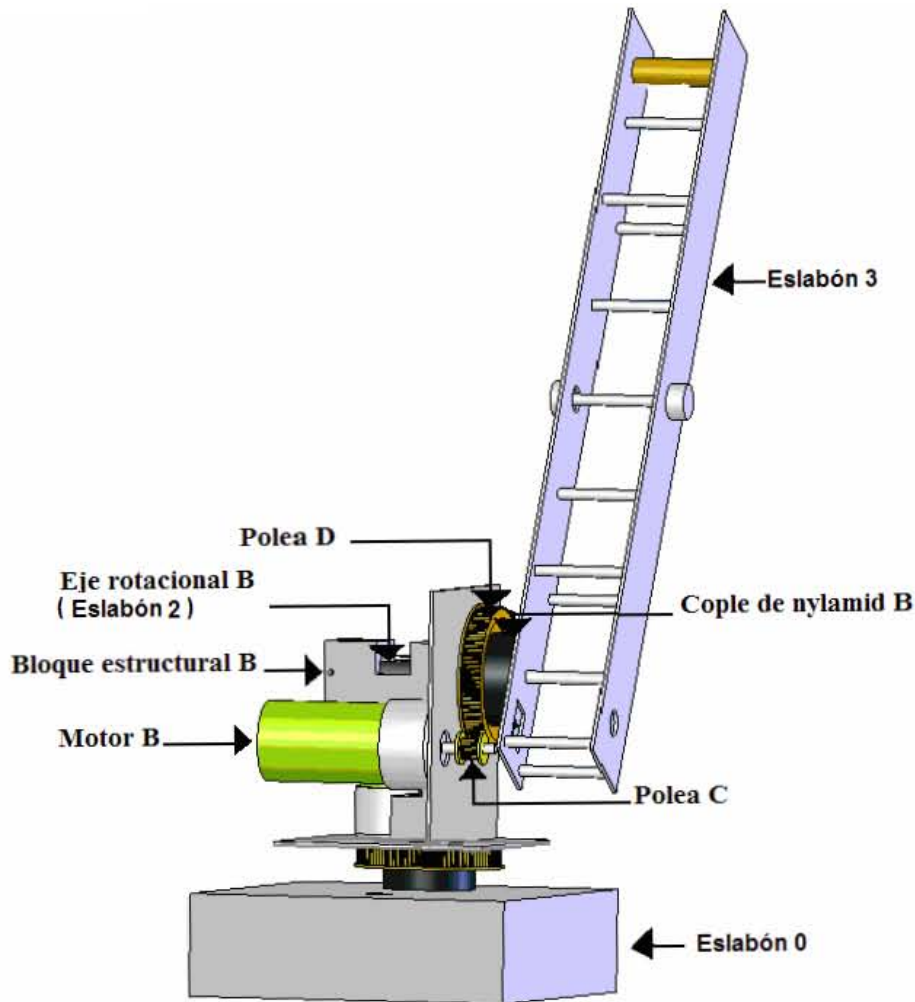


Figura 5.5 “Elementos de transmisión del eslabón 2”

El bloque estructural *B* se localiza acoplado verticalmente a la placa de soporte superior y al bloque estructural *A*. Al igual que el bloque *A*, el bloque *B* soporta al eje rotacional *B*, y éste a su vez al sensor de posición *B*, para cuantificar el ángulo de la segunda articulación (ver *figura 5.6*).

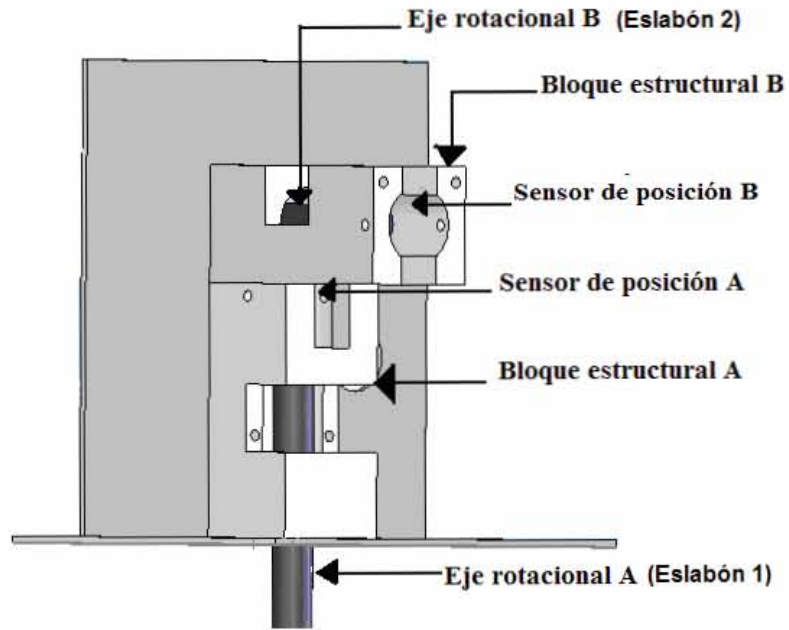


Figura 5.6 “Bloque estructural B y elementos acoplados (vista 1)”

En la *figura 5.7* se puede observar claramente como el eje rotacional *B* se encuentra sujeto al centro de la polea *C* y cople de nylamid *B*.

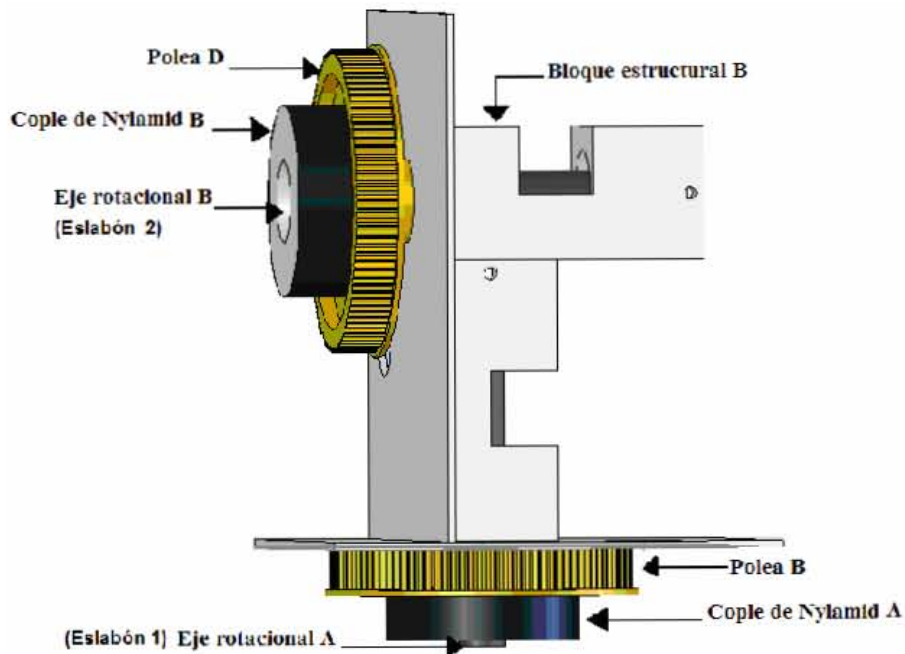


Figura 5.7 “Bloque estructural B y elementos acoplados (vista 2)”

El motor *B* se sujetó a la placa superior de soporte. En el rotor del actuador se tiene la polea *C*, que por medio de una banda dentada transmite el par a la polea *D*, quien está sujeta al cople *B*. El eslabón 2 está sujeto por medio del cople *B* á la transmisión por banda, quien mueve la segunda articulación del brazo (ver *figura 5.5*).

5.3 Eslabón 3 y 4

El par necesario para mover al cuarto eslabón es generado por la transmisión de doble banda que se encuentra montada sobre el eslabón 3. Los elementos que componen ésta transmisión son: polea *E*, *F* *G* y *H*, banda de transmisión *C* y *D*, eje rotacional *C* y *D*, y sensor de posición *C*. Todos estos elementos pueden ser observados en la siguiente figura:

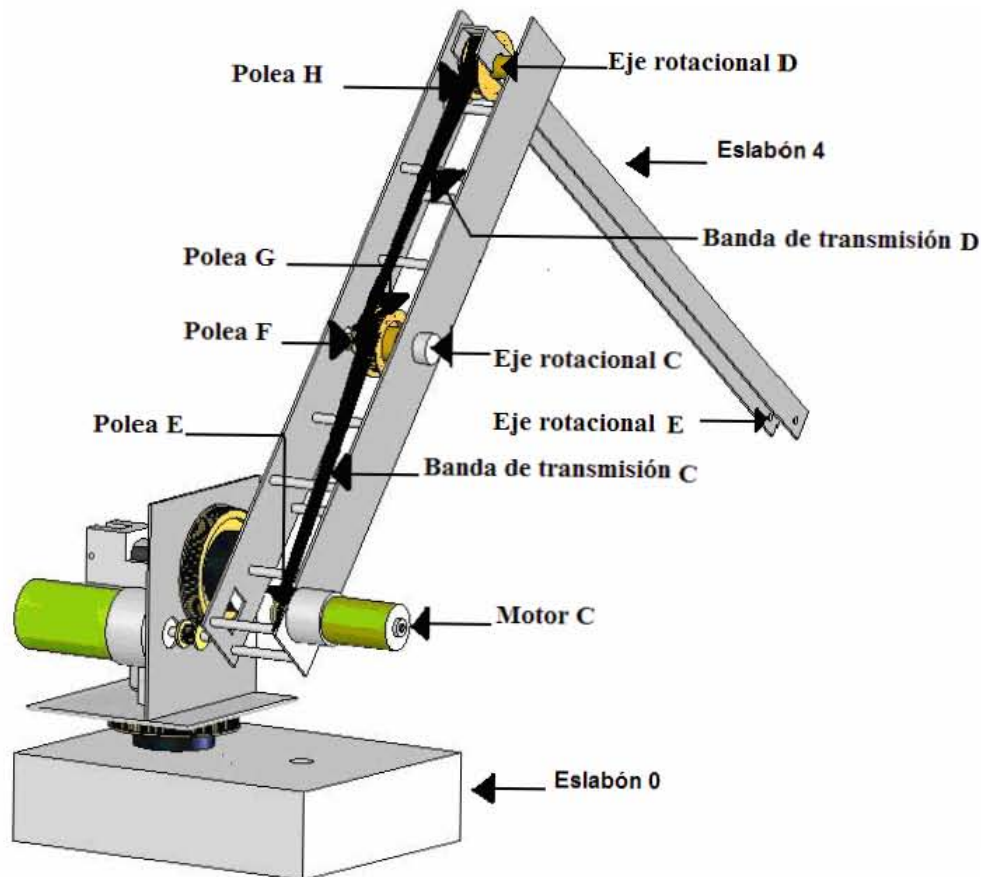


Figura 5.8 “Transmisión de doble banda, eslabones 3 y 4”

Para poder transmitir el par generado por el actuador *C*, fue necesario diseñar una transmisión de doble banda debido a que no se contaba con una banda que tuviera una longitud mayor. El motor se encuentra sujeto al tercer eslabón en su parte inferior (ver *figura 5.8*), de esta manera se evita concentrar todo el peso en la parte superior del eslabón 3.

En la parte central del eslabón 3 se localiza el eje rotacional *C*, sujetas a él se localizan las poleas *F* y *G*. En la parte superior se encuentra el eje rotacional *D*, en donde se acopla el eslabón 3 y el sensor de posición que mide el ángulo de la tercera articulación.

5.4 Eslabón 5

El eslabón 5, quien sujeta a la herramienta final, en todo momento mantendrá la vertical por efecto gravitatorio, para lograr lo anterior, en el extremo del eslabón 4 se acoplo el eje rotacional *E*, al que se sujeta el eslabón 5 (*Figura 5.9*).

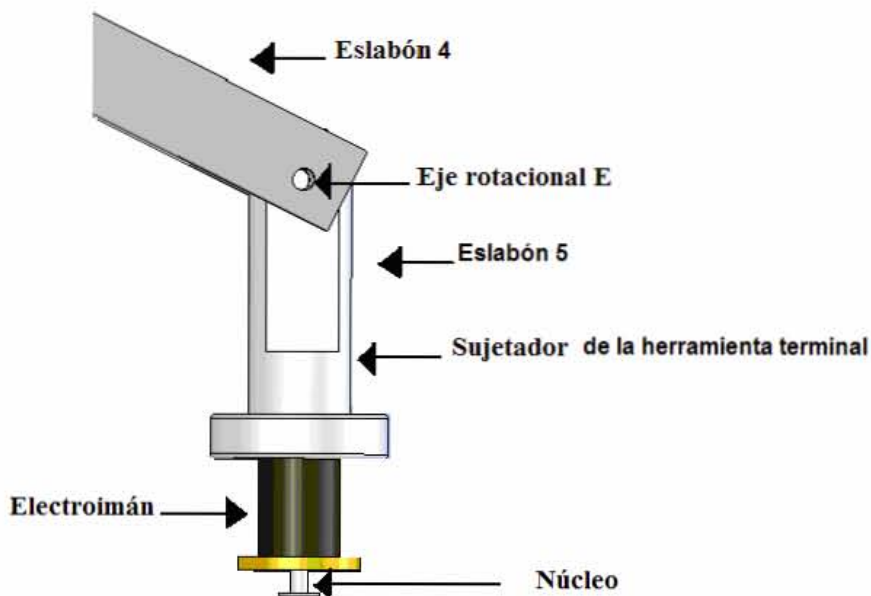


Figura 5.9 “Eslabón 5”

Para el diseño de la herramienta final, se tomo en cuenta la forma y peso de la pieza a seleccionar así como el tipo de movimiento deseado. Una de las características que tienen los objetos a manipular es que en la parte superior tienen hierro, de 2[mm] de espesor, siendo ésta la razón principal por la cual se eligió colocar en el extremo a un electroimán, de esta manera se puede asegurar una buena sujeción.

En la *figura 5.10* se muestra el acoplamiento del sensor, que detectará el color de la ficha, al eslabón 5.

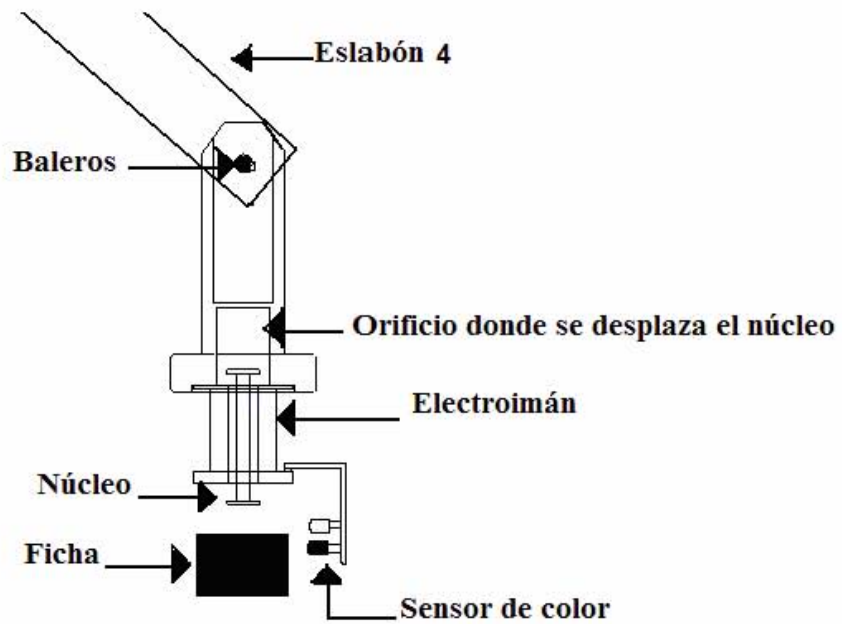


Figura 5.10 “Órgano terminal y sensor del color de la ficha”

5.5 Ensamble final

Una vez definida la configuración de cada uno de los elementos, en la *figura 5.11* se muestra el ensamble final que caracteriza físicamente el brazo manipulador de tres grados de libertad:

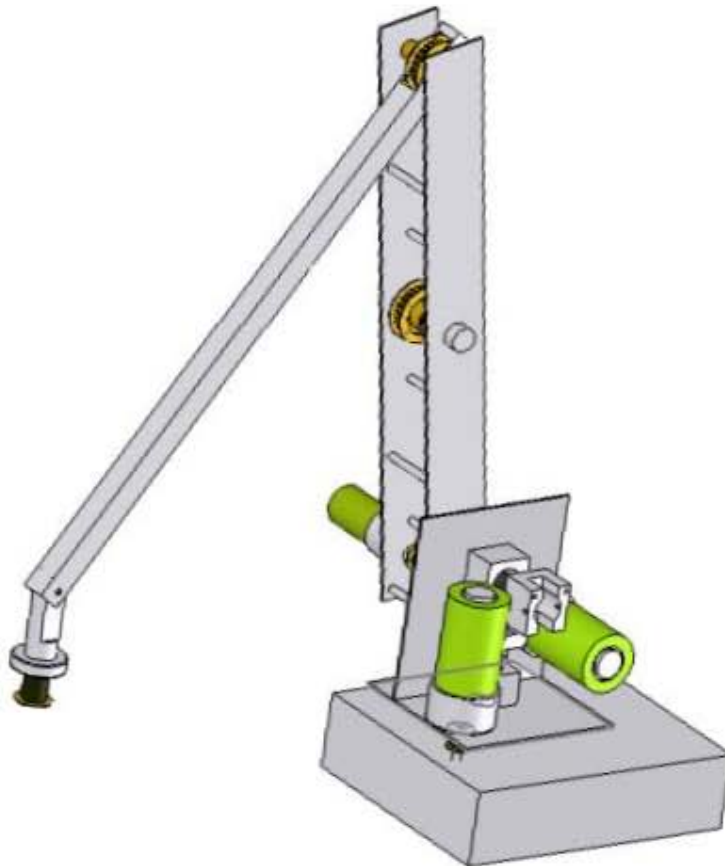


Figura 5.11 “Ensamble final del brazo manipulador”

CAPÍTULO 6

Capítulo 6 Resultados y conclusiones

La fase más importante del proyecto es acoplar cada una de las etapas realizadas a lo largo del proyecto, y comprobar que los diseños planteados funcionan adecuadamente. En el presente capítulo se mencionan las pruebas realizadas, los resultados obtenidos y las conclusiones de este trabajo.

6.1 Prueba del control de posición

Inicialmente se probó la implementación del algoritmo de control de posición, de manera que el manipulador trazará trayectorias lineales dentro de su espacio de trabajo. El algoritmo involucra la localización de la coordenada más cercana a la coordenada actual, la generación de las matrices de transformación que contienen las coordenadas de la trayectoria lineal entre dos puntos, el cálculo de la transformación lineal en forma dinámica mediante la resolución del modelo cinemático inverso para cada punto, y la conversión dichas coordenadas angulares, en radianes, a coordenadas máquina, mediante el método de diferencias divididas interpolantes de Newton.

6.1.1 Trayectoria lineal

La trayectoria lineal que se probó, es la planteada en el ejemplo del capítulo tres (ec. 21):

$$\begin{matrix} (20,0,14) & \Rightarrow & (15,7,14) & \Rightarrow & (15,7,2.5) & \Rightarrow & (15,7,14) & \Rightarrow & (20,30,14) \\ 1 & & 2 & & 3 & & 4 & & 5 \end{matrix}$$

A continuación se muestran los resultados arrojados por el algoritmo:

Los ángulos y coordenadas máquina necesarios en las uniones del brazo, para moverse de la posición 1 (coordenada inicial), a la posición 2 (coordenada más cercana) son:

| Trayectoria lineal [cm] | Ángulo en eje rotacional A [°] | Coordenada máquina [1] | Ángulo en eje rotacional B [°] | Coordenada máquina [1] | Ángulo en eje rotacional D [°] | Coordenada máquina [1] |
|-------------------------|--------------------------------|------------------------|--------------------------------|------------------------|--------------------------------|------------------------|
| (20,0,14) | 9.4793 | 283 | 126.8742 | 151 | -155.1731 | 336 |
| (19.5,0.7,14) | 10.7325 | 286 | 128.6088 | 146 | -155.9699 | 337 |
| (19,1.4,14) | 12.0569 | 290 | 130.3484 | 141 | -156.7387 | 338 |
| (18.5,2.1,14) | 13.4311 | 294 | 132.0513 | 136 | -157.4647 | 339 |
| (18,2.8,14) | 14.8517 | 298 | 133.7014 | 132 | -158.1446 | 340 |
| (17.5,3.5,14) | 16.3152 | 302 | 135.2802 | 127 | -158.7745 | 341 |
| (17,4.2,14) | 17.8167 | 306 | 136.7672 | 123 | -159.3501 | 342 |
| (16.5,4.9,14) | 19.3505 | 311 | 138.1401 | 119 | -159.8668 | 342 |
| (16,5.6,14) | 20.9093 | 315 | 139.3758 | 116 | -160.3204 | 343 |
| (15.5,6.3,14) | 22.4851 | 319 | 140.4511 | 113 | -160.7065 | 344 |
| (15,7.0,14) | 24.0690 | 324 | 141.3438 | 110 | -161.0211 | 344 |

Tabla 6.1 “Ángulos requeridos para trasladarse de la coordenada 1 a la 2”

La señal generada en el sensor de posición (potenciómetro) es retroalimentada al microcontrolador, dicha señal varía analógicamente entre 0 y 5 volts, después mediante el convertidor A/D es expresada digitalmente (coordenada máquina). Los rangos de lectura de los sensores de posición, colocados en las tres articulaciones del brazo, pueden verse en la *tabla 4.1*. Con las coordenadas máquina de la *tabla 6.1*, el brazo se coloca en la coordenada (15, 7, 14), listo para bajar por la ficha.

Los ángulos y coordenadas máquina necesarios en las articulaciones, para desplazarse de la posición 2 a la posición 3 (físicamente el elemento terminal baja por la ficha) son:

| Trayectoria lineal [cm] | Ángulo en eje rotacional A [°] | Coordenada máquina [1] | Ángulo en eje rotacional B [°] | Coordenada máquina [1] | Ángulo en eje rotacional D [°] | Coordenada máquina [1] |
|-------------------------|--------------------------------|------------------------|--------------------------------|------------------------|--------------------------------|------------------------|
| (15,7,14) | 24.0690 | 324 | 141.2381 | 110 | -160.9838 | 344 |
| (15,7,13.5) | 24.0691 | 324 | 140.4196 | 113 | -161.3552 | 344 |
| (15,7,13) | 24.0691 | 324 | 139.4981 | 115 | -161.6971 | 345 |
| (15,7,12.5) | 24.0691 | 324 | 138.5065 | 118 | -162.0170 | 345 |
| (15,7,12) | 24.0691 | 324 | 137.4428 | 121 | -162.3139 | 346 |
| (15,7,11.5) | 24.0691 | 324 | 136.3042 | 124 | -162.5868 | 346 |
| (15,7,11) | 24.0691 | 324 | 135.0888 | 128 | -162.8345 | 346 |
| (15,7,10.5) | 24.0691 | 324 | 133.7945 | 131 | -163.0561 | 347 |
| (15,7,10) | 24.0691 | 324 | 132.4202 | 135 | -163.2504 | 347 |
| (15,7,9.5) | 24.0691 | 324 | 130.9648 | 139 | -163.4166 | 347 |
| (15,7,9) | 24.0691 | 324 | 129.4284 | 144 | -163.5537 | 347 |
| (15,7,8.5) | 24.0691 | 324 | 127.8114 | 148 | -163.6612 | 347 |
| (15,7,8) | 24.0691 | 324 | 126.1151 | 153 | -163.7384 | 348 |
| (15,7,7.5) | 24.0691 | 324 | 124.3416 | 158 | -163.7849 | 348 |
| (15,7,7) | 24.0691 | 324 | 122.4938 | 163 | -163.8005 | 348 |
| (15,7,6.5) | 24.0691 | 324 | 120.5754 | 169 | -163.7850 | 348 |
| (15,7,6) | 24.0691 | 324 | 118.5908 | 174 | -163.7385 | 348 |
| (15,7,5.5) | 24.0691 | 324 | 116.5452 | 180 | -163.6613 | 347 |
| (15,7,5) | 24.0691 | 324 | 114.4444 | 186 | -163.5539 | 347 |
| (15,7,4.5) | 24.0691 | 324 | 112.2944 | 192 | -163.4167 | 347 |
| (15,7,4) | 24.0691 | 324 | 110.1020 | 198 | -163.2505 | 347 |
| (15,7,3.5) | 24.0691 | 324 | 107.8740 | 204 | -163.0562 | 347 |
| (15,7,3) | 24.0691 | 324 | 105.6174 | 211 | -162.8347 | 346 |
| (15,7,2.5) | 24.0691 | 324 | 103.3390 | 217 | -162.5869 | 346 |

Tabla 6.2 “Ángulos requeridos para desplazarse de la coordenada 2 a la 3”

Con la tabla anterior podemos verificar la coherencia de los datos, con lo que ocurre físicamente, ya que para esta trayectoria, en la que el brazo baja por la ficha, el vector “x” y el vector “y” se mantienen constantes, variando solamente el vector “z” de 14 [cm] a 2.5 [cm], por consecuencia el ángulo en el eje rotacional A se mantiene constante, variando solamente el ángulo de las articulaciones 1 y 2.

Los ángulos y coordenadas máquina necesarios en las articulaciones, para trasladarse de la posición 3 a la posición 4 (físicamente la herramienta final sube la ficha) son:

| Trayectoria lineal [cm] | Ángulo en eje rotacional A [°] | Coordenada máquina [1] | Ángulo en eje rotacional B [°] | Coordenada máquina [1] | Ángulo en eje rotacional D [°] | Coordenada máquina [1] |
|-------------------------|--------------------------------|------------------------|--------------------------------|------------------------|--------------------------------|------------------------|
| (15,7,2.5) | 24.0690 | 324 | 103.3347 | 217 | -162.6068 | 346 |
| (15,7,3) | 24.0691 | 324 | 105.6194 | 211 | -162.8355 | 346 |
| (15,7,3.5) | 24.0691 | 324 | 107.8729 | 204 | -163.0562 | 347 |
| (15,7,4) | 24.0691 | 324 | 110.1008 | 198 | -163.2505 | 347 |
| (15,7,4.5) | 24.0691 | 324 | 112.2932 | 192 | -163.4166 | 347 |
| (15,7,5) | 24.0691 | 324 | 114.4432 | 186 | -163.5538 | 347 |
| (15,7,5.5) | 24.0691 | 324 | 116.5442 | 180 | -163.6613 | 347 |
| (15,7,6) | 24.0691 | 324 | 118.5899 | 174 | -163.7385 | 348 |
| (15,7,6.5) | 24.0691 | 324 | 120.5746 | 169 | -163.7849 | 348 |
| (15,7,7) | 24.0691 | 324 | 122.4931 | 163 | -163.8005 | 348 |
| (15,7,7.5) | 24.0691 | 324 | 124.3411 | 158 | -163.7850 | 348 |
| (15,7,8) | 24.0691 | 324 | 126.1148 | 153 | -163.7385 | 348 |
| (15,7,8.5) | 24.0691 | 324 | 127.8113 | 148 | -163.6613 | 347 |
| (15,7,9) | 24.0691 | 324 | 129.4285 | 144 | -163.5539 | 347 |
| (15,7,9.5) | 24.0691 | 324 | 130.9651 | 139 | -163.4167 | 347 |
| (15,7,10) | 24.0691 | 324 | 132.4206 | 135 | -163.2506 | 347 |
| (15,7,10.5) | 24.0691 | 324 | 133.7951 | 131 | -163.0563 | 347 |
| (15,7,11) | 24.0691 | 324 | 135.0895 | 128 | -162.8347 | 346 |
| (15,7,11.5) | 24.0691 | 324 | 136.3051 | 124 | -162.5870 | 346 |
| (15,7,12) | 24.0691 | 324 | 137.4437 | 121 | -162.3141 | 346 |
| (15,7,12.5) | 24.0691 | 324 | 138.5075 | 118 | -162.0172 | 345 |
| (15,7,13) | 24.0691 | 324 | 139.4992 | 115 | -161.6974 | 345 |
| (15,7,13.5) | 24.0691 | 324 | 140.4213 | 113 | -161.3558 | 344 |
| (15,7,14) | 24.0691 | 324 | 141.2769 | 110 | -160.9936 | 344 |

Tabla 6.3 “Ángulos requeridos para moverse de la coordenada 3 a la 4”

Al igual que en la trayectoria anterior, el vector “x” y el vector “y” se mantienen constantes. Físicamente el brazo sube la ficha a la coordenada (15, 7, 14).

Los ángulos y coordenadas máquina necesarios en los eslabones, para ir de la posición 4 a la posición 5 (físicamente el brazo lleva la ficha a su respectivo contenedor) son:

| Trayectoria lineal [cm] | Ángulo en eje rotacional A [°] | Coordenada máquina [1] | Ángulo en eje rotacional B [°] | Coordenada máquina [1] | Ángulo en eje rotacional D [°] | Coordenada máquina [1] |
|-------------------------|--------------------------------|------------------------|--------------------------------|------------------------|--------------------------------|------------------------|
| (15,7,14) | 24.0690 | 324 | 141.2381 | 110 | -160.9838 | 344 |
| (15,5,9.3,14) | 26.0196 | 329 | 134.2596 | 130 | -158.3587 | 340 |
| (16,11,6,14) | 27.5724 | 334 | 127.6920 | 149 | -155.5388 | 336 |
| (16,5,13,9,14) | 28.8232 | 337 | 121.6375 | 166 | -152.5765 | 332 |
| (17,16,2,14) | 29.8445 | 340 | 116.0895 | 181 | -149.5040 | 327 |
| (17,5,18,5,14) | 30.6897 | 343 | 110.9998 | 196 | -146.3417 | 321 |
| (18,20,8,14) | 31.3980 | 345 | 106.3073 | 209 | -143.1020 | 316 |
| (18,5,23,1,14) | 31.9987 | 346 | 101.9514 | 221 | -139.7922 | 309 |
| (19,25,4,14) | 32.5533 | 348 | 97.7444 | 233 | -136.4487 | 303 |
| (19,5,27,7,14) | 32.9884 | 349 | 93.9309 | 244 | -133.0034 | 296 |
| (20,30,14) | 33.3724 | 350 | 90.3010 | 254 | -129.4981 | 288 |

Tabla 6.4 “Ángulos requeridos para ir de la coordenada 4 a la 5”

Con los datos arrojados anteriormente, podemos verificar que el algoritmo programado en Matlab (apéndice C) trabaja de manera correcta. La trayectoria anterior consistió en la selección de una sola ficha, para una rutina de n piezas, el ciclo de control se ejecuta desde la interfaz desarrollada en Visual Basic.

6.2 Funcionamiento

Una vez probado el algoritmo de control de posición y habiendo analizado la coherencia de los resultados obtenidos, mismos que son trasferidos al microcontrolador, quien con dicha información desplaza al robot, se prosiguió a conjuntar: la interfaz de usuario, el control de posición, la comunicación serial y la tarjeta electrónica, quien a su vez se conecta directamente a los actuadores y transductores del manipulador.

Para medir la eficiencia del brazo, así como el tiempo de proceso, se realizaron diferentes pruebas, donde se cuantificaron dos aspectos principalmente: cuantas fichas tomo y en que tiempo realizo la prueba. Se llevaron a cabo 5 rutinas en donde el brazo debió seleccionar 5 fichas por cada una. Los resultados se muestran a continuación:

La disposición de las fichas se muestra en la *figura 4.3*.

| Coordenada | Color | ¿Tomo la ficha? | ¿La coloco en el contenedor correcto? |
|----------------------------|-------|-----------------|---------------------------------------|
| (20,0) | B | Si | Si |
| (10,5) | B | Si | Si |
| (8,-12) | B | Si | No |
| (37,-7) | N | Si | Si |
| (35,13) | N | Si | Si |
| Tiempo de proceso: 37 seg. | | | |

Tabla 6.5 “Rutina 1”

| Coordenada | Color | ¿Tomo la ficha? | ¿La coloco en el contenedor correcto? |
|----------------------------|-------|-----------------|---------------------------------------|
| (25,0) | N | Si | Si |
| (10,-5) | B | Si | Si |
| (8,12) | B | Si | Si |
| (40,-10) | N | Si | Si |
| (35,20) | N | Si | Si |
| Tiempo de proceso: 38 seg. | | | |

Tabla 6.6 “Rutina 2”

| Coordenada | Color | ¿Tomo la ficha? | ¿La coloco en el contenedor correcto? |
|----------------------------|-------|-----------------|---------------------------------------|
| (0,20) | B | Si | Si |
| (10,10) | B | Si | Si |
| (15,-10) | B | Si | Si |
| (37,-18) | B | Si | Si |
| (35,19) | B | Si | No |
| Tiempo de proceso: 36 seg. | | | |

Tabla 6.7 “Rutina 3”

| Coordenada | Color | ¿Tomo la ficha? | ¿La coloco en el contenedor correcto? |
|---------------------------|-------|-----------------|---------------------------------------|
| (0,0) | N | Si | Si |
| (40,0) | N | Si | Si |
| (20,20) | N | Si | Si |
| (20,10) | N | Si | Si |
| (20,-10) | N | SI | Si |
| Tiempo de proceso: 36seg. | | | |

Tabla 6.8 “Rutina 1”

| Coordenada | Color | ¿Tomo la ficha? | ¿La coloco en el contenedor correcto? |
|----------------------------|-------|-----------------|---------------------------------------|
| (10,10) | N | Si | Si |
| (20,9) | N | Si | Si |
| (8,15) | B | Si | No |
| (37,-7) | B | Si | Si |
| (35,-13) | B | No | Si |
| Tiempo de proceso: 38 seg. | | | |

Tabla 6.9 “Rutina 1”

Con los resultados obtenidos en las tablas anteriores, se logró cuantificar una eficiencia del 88 %, ya que de 25 fichas, 23 de ellas fueron colocadas en su contenedor correcto.

6.3 Repetibilidad

La repetibilidad que tiene un brazo manipulador, se refiere a la capacidad de regresar a un punto, dentro del espacio de trabajo, las veces que sean necesarias. Esta magnitud establece el grado de exactitud en la repetición de los movimientos de un manipulador al realizar una tarea programada.



Figura 6.1 “Repetibilidad del manipulador”

Para calcular este parámetro, se programó al manipulador de manera que regresara durante 10 ocasiones al punto *PI* (un punto programado). Durante el transcurso de esta rutina el brazo se colocó en el punto *PI* y en diferentes puntos *PR* (puntos de regreso). En cada repetición se midió, con ayuda de un calibrador, la distancia entre el punto *PI* y los distintos puntos *PR*. Para cuantificar la repetibilidad del brazo se tomó la máxima distancia, encontrando que dicha característica fue del ± 6 [mm] (ver *figura 6.1*).

Como se observa en la *figura 6.1*, el diámetro de la ficha es de 25 [mm] y la repetibilidad del brazo es de ± 6 [mm] (sin ajustes) esta distancia podría considerarse exagerada, sin embargo, dependiendo del trabajo que se vaya a realizar, la precisión en dicha característica es mayor o menor.

Cuando se hace referencia a los ajustes es en relación con el sensor de posición (potenciómetro), por medio del cual es posible recalibrar el sistema y ajustarlo a la coordenada solicitada, produciendo como consecuencia disminución del error en la posición.

6.4 Conclusiones

Con el fin de evaluar el trabajo realizado, a continuación se presentan las conclusiones del mismo, divididas de la siguiente manera:

6.4.1 En base al algoritmo de control por posición

El modelo cinemático representa al sistema sin considerar perturbaciones debidas a fricciones, gravedad, inercia y peso, (parámetros comúnmente considerados en un modelo dinámico). Se verificó su desempeño mediante los parámetros Denavit – Hartenberg, que contempla distancias (longitud de los eslabones) y ángulos (entre eslabones), y su simulación permitió comprobar el movimiento realizado por cada eje de rotación del manipulador en 2D y 3D.

El algoritmo para el control por posición mediante la resolución del modelo cinemático nos permitió calcular, de manera acertada, a partir de la posición de los objetos en el tablero, los ángulos necesarios en cada una de las articulaciones del brazo para que el elemento terminal se posicionara en dicho objeto.

Un aspecto importante, dentro del algoritmo, fue la generación de trayectorias lineales, ya que, ayudo en gran medida a evitar colisiones con otros objetos (fichas) en el tablero, así como con los contenedores.

6.4.2 En base al diseño mecánico

Durante el planteamiento del problema se propusieron las dimensiones necesarias de los eslabones, con la finalidad que el manipulador pudiera cubrir el área de trabajo sin problemas, sin embargo, en el diseño de configuración se puede observar dimensiones mayores a las calculadas, eso fue debido a que ya se contaba con algunos elementos que fueron incorporados como bandas y poleas. Por lo anterior la estructura dinámica del brazo al estar en movimiento sufre de ligeras oscilaciones en el eslabón 4 y 5, afectando la eficiencia del mismo.

Se determinó que para mejorar el control del robot era necesario involucrar la variable de velocidad, por consecuencia realizar el calculo del modelo dinámico, lo que implicaría tener una mejor caracterización matemática del brazo. Lo anterior eliminaría considerablemente las oscilaciones mencionadas.

6.4.3 En base al diseño electrónico

El desempeño del sensor que identifica el color de los objetos, no fue el óptimo debido a que es afectado por la vibración que sufre el tercer eslabón, así como de frecuencias luminosas generadas por la luz natural o del mismo entorno (condiciones de iluminación).

En base a las pruebas hechas durante el funcionamiento del brazo, podemos decir que para la tarea (selección de fichas), para la que fue diseñado el brazo, así como la repetibilidad de ± 6 [mm] y eficiencia del 88 %, son características aceptables para éste sistema de manipulación de objetos.

6.4.4 En base al sistema en general

Cuando se pensó en diseñar un brazo mecánico, que realizará una tarea muy definida, como lo es la manipulación de objetos planteada en el capítulo 1, se pensó en modularizar cada una de sus etapas, con la finalidad de poder tener la libertad de acoplarlo a diferentes tareas, modificando ligeramente su software de control. Lo anterior es una característica muy importante del sistema, debido a que lo hace muy flexible a cambios en su operación.

En conclusión, podemos decir que la metodología empleada para este proyecto fue muy viable, logrando con ello, brindar una propuesta de solución que será de utilidad para futuros trabajos mecatrónicos, los cuales integren un sistema de manipulación, como el mostrado en esta tesis.

6.4.5 Trabajo futuro

Para lograr un mejor desempeño del sistema, se plantea desarrollar el modelo dinámico, citado con anterioridad, con el cual se logre obtener una mejor caracterización del brazo, así como la incorporación de un controlador de velocidad, que pueda ayudar a estabilizar, de una mejor manera, las trayectorias seguidas por el manipulador.

Para el funcionamiento de este sistema es indispensable que el algoritmo de control sea ejecutado en una computadora. Debido a esto se propone encapsular dicho algoritmo dentro de un microcontrolador con mayores recursos y prestaciones, que el utilizado en este proyecto. Como lo puede ser un procesador de señales digitales (“DSP” por sus siglas en ingles).

APENDICE A
**“Principales configuraciones de un brazo
manipulador”**

La estructura del un manipulador y la relación entre sus elementos, proporcionan una configuración mecánica, que da origen al establecimiento de los parámetros que hay que conocer para definir la posición y orientación del elemento terminal. Fundamentalmente, existen cuatro estructuras clásicas en los manipuladores, que se relacionan con los correspondientes modelos de coordenadas en el espacio, y que se citan a continuación: cartesianas, cilíndricas, esféricas, angulares. Así, el brazo del manipulador puede presentar cuatro configuraciones clásicas:

- cartesiana
- cilíndrica
- esférica
- de brazo articulado

El empleo de diferentes combinaciones de articulaciones en un robot, da lugar a diferentes configuraciones, con características a tener en cuenta tanto en el diseño y construcción del robot, como en su aplicación. Las combinaciones más frecuentes son con tres articulaciones (3 grados de libertad), que son las más importantes a la hora de posicionar su extremo en un punto en el espacio de trabajo. A continuación se presentan las características principales de las configuraciones del brazo manipulador.

Cartesiana. El posicionando en el espacio de trabajo, se hace utilizando articulaciones prismáticas. Esta configuración se utiliza comúnmente cuando un espacio de trabajo es grande y debe cubrirse en su totalidad. Posee tres movimientos lineales, es decir, tiene tres grados de libertad, los cuales corresponden a los movimientos localizados en los ejes X , Y y Z .

Los movimientos que realiza este robot, entre un punto y otro, son con base en interpolaciones lineales. Interpolación, en este caso, significa el tipo de trayectoria que realiza el manipulador cuando se desplaza entre un punto y otro. A la trayectoria realizada en línea recta se le conoce como interpolación lineal y a la trayectoria hecha de acuerdo con el tipo de movimientos que tienen sus articulaciones se le llama interpolación por articulación.

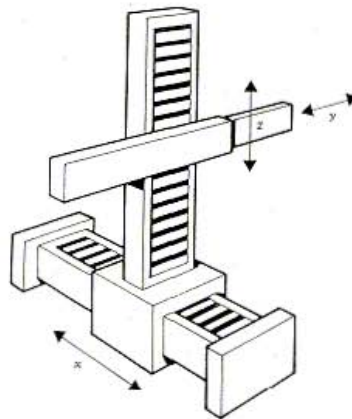


Figura A.1 “Configuración cartesiana”

Cilíndrica. El robot tiene un movimiento de rotación sobre una base, una articulación prismática para la altura, y una prismática para el radio. Este robot se ajusta muy bien para espacios de trabajo redondos. Puede realizar dos movimientos lineales y uno rotacional, de manera que presenta tres grados de libertad.

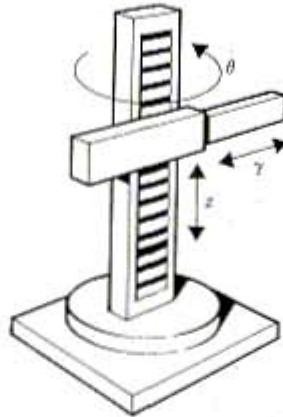


Figura A.2 “Configuración cilíndrica”

Esférica. Dos uniones de rotación y una prismática permiten al robot apuntar en muchas direcciones, y extender la herramienta final a una distancia radial dentro del espacio de trabajo. Los movimientos son: rotacional, angular y lineal. Este robot utiliza la interpolación por articulación para moverse en sus dos primeras articulaciones y la interpolación lineal para la extensión y retracción.

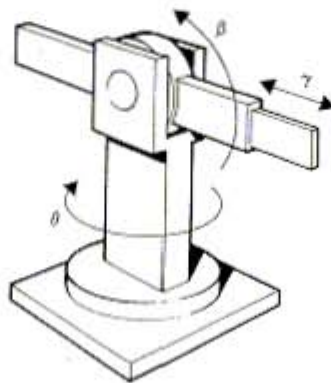


Figura A.3 “Configuración esférica”

De brazo articulado ó antropomórfico. El robot usa tres articulaciones de rotación para posicionarse. Generalmente, el volumen de trabajo es esférico. Estos tipos de robot se parecen al brazo humano, con una cintura, el hombro, el codo, y la muñeca. Presenta una articulación con movimiento rotacional y dos angulares. Aunque el brazo articulado puede realizar el movimiento llamado interpolación lineal (para lo cual requiere mover simultáneamente dos o tres de sus articulaciones), el movimiento natural es el de interpolación por articulación, tanto rotacional como angular.

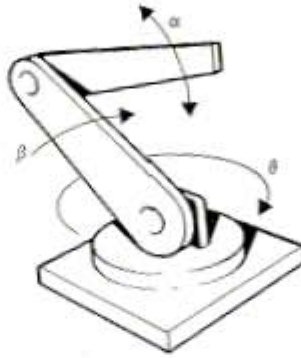


Figura A.4 “Brazo articulado”

APENDICE B
“Software utilizado”

COMPILADOR PCW DE CCS



Este compilador es del tipo PCW (microcontroladores PIC de 14 bits) de la casa CCS Inc. A su vez, el compilador lo integraremos en un *entorno de desarrollo integrado* (IDE) que nos va a permitir desarrollar todas y cada una de las fases que se compone un proyecto, desde la edición hasta la compilación pasando por la depuración de errores.

La última fase, a excepción de la depuración y retoques de hardware finales, será programar el PIC. Al igual que el compilador de Turbo C, éste "*traduce*" el código C del archivo fuente (.C) a lenguaje máquina para los microcontroladores PIC, generando así un archivo en formato hexadecimal (.HEX).

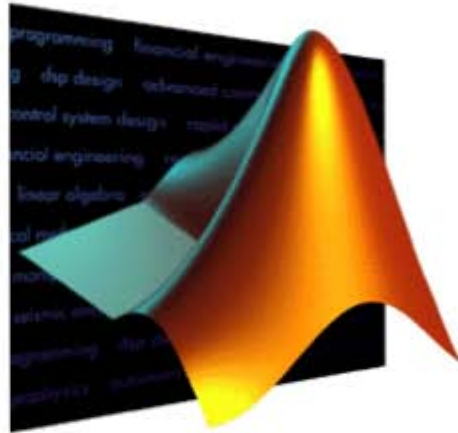
PIC DOWNLOADER

Este software se utiliza para descargar un nuevo archivo hexadecimal, generado por el compilador de C, en el microcontrolador. Este pequeño programa trabaja en sincronía con el archivo Bootloader, previamente programado en el PIC. Es necesario conectar los pines del puerto serie (TX y RX) del micro al los correspondientes del puerto COM de la computadora.

Para descargar el programa solo se debe configurar el puerto utilizado, la velocidad de transmisión, que debe ser la misma a la que opera el bootloader en el micro, seleccionar la ruta del archivo *.hex. y oprimir la tecla WRITE (F4). Debido a que la memoria del PIC utilizado es del tipo Flash esta operación consume tal solo unos cuantos segundos.



MATLAB 7.0



MATLAB es el nombre abreviado de “MATrix LABoratory”. MATLAB es un programa para realizar cálculos numéricos con *vectores* y *matrices*. Como caso particular puede también trabajar con números escalares, tanto reales como complejos. Una de las capacidades más atractivas es la de realizar una amplia variedad de *gráficos* en dos y tres dimensiones. MATLAB tiene también un lenguaje de programación propio.

MATLAB es una toda una plataforma para el cálculo técnico y científico. Para ciertas operaciones es muy rápido, cuando puede ejecutar sus funciones en código nativo con los tamaños más adecuados para aprovechar sus capacidades de vectorización.

En otras aplicaciones resulta bastante más lento que el código equivalente desarrollado en C/C++ o Fortran. Sin embargo, siempre es una magnífica herramienta de alto nivel para desarrollar aplicaciones técnicas, fácil de utilizar y que aumenta la productividad de los programadores respecto a otros entornos de desarrollo. Dispone de un código básico y de varias librerías especializadas (*toolboxes*).

Una de sus características es que con grandes matrices o grandes sistemas de ecuaciones MATLAB obtiene toda la potencia del ordenador. Por ejemplo, las siguientes instrucciones permiten calcular la potencia de cálculo del ordenador en Megaflops (millones de operaciones aritméticas por segundo).

HEMERO (Toolbox Robotics)



Herramienta de MATLAB para el estudio de manipuladores y robots móviles

La herramienta HEMERO fue concebida como un apoyo al aprendizaje de diversos aspectos de la robótica. Permite resolver los problemas más usuales en robótica. La herramienta se denomina HEMERO (Herramienta Matlab-simulink para el Estudio de manipuladores y Robots móviles).

Su característica más significativa es que permite integrar análisis cinemático, dinámico, diseño de sistemas de control de robots y simulación, pudiendo aplicarse tanto a robots manipuladores como móviles. Asimismo se han previsto interfases con programas de representación gráfica en 3D que facilitan la visualización del robot.

HEMERO dispone de un conjunto de funciones MATLAB para la definición de robot manipuladores y para la visualización de esquemas simplificados del manipulador y los sistemas de referencia involucrados. Las posiciones y orientaciones se definen utilizando vectores de posición y matrices de transformación. Es posible también emplear los ángulos de Euler para definir la orientación.

Desde el punto de vista de la cinemática, el manipulador se caracteriza por su matriz de parámetros de Denavit-Hartenberg. Una vez definido el manipulador existen funciones que permiten obtener su modelo cinemático directo expresado simbólicamente. Alternativamente, existen también una función para resolver numéricamente el modelo cinemático inverso.

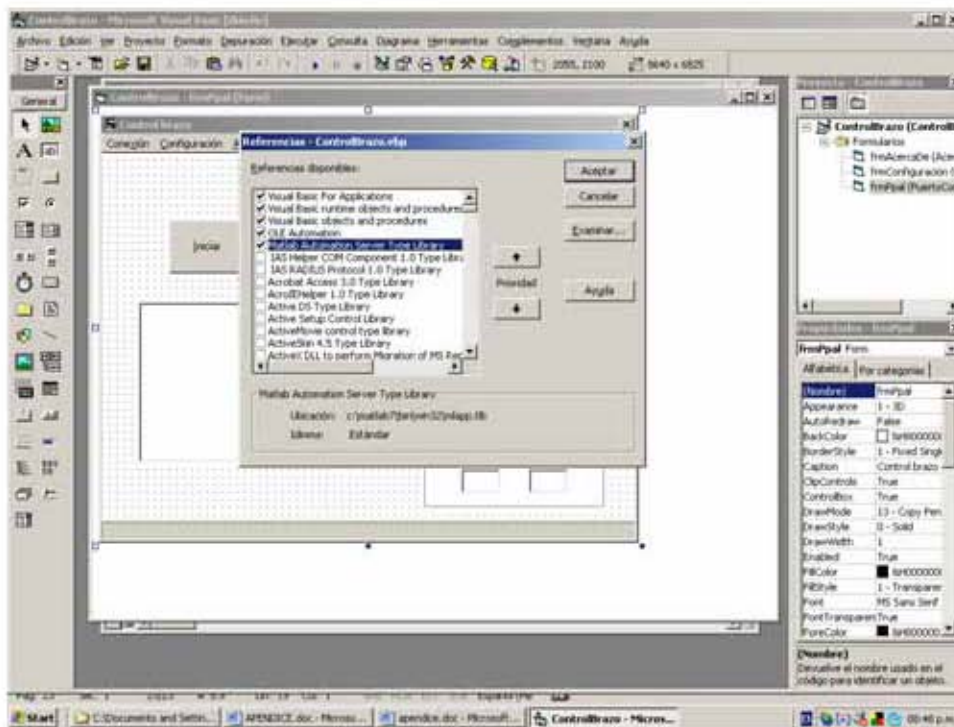
Para la utilización de esta herramienta HEMERO fue instalado en el toolbox de MATLAB versión 7.0.1

MATLAB AUTOMATION SERVER

Matlab Automation Server es una interfaz entre la plataforma de Matlab (servidor) y aplicaciones escritas en otros lenguajes de programación (cliente o controlador). Con esta interfaz se puede tener acceso a la potencia que brinda Matlab, en cuanto al cálculo de vectores y matrices. Para nuestro caso en particular mediante esta herramienta se pudo tener control desde la aplicación escrita en Visual Basic hacia el algoritmo de control de posición escrita en Matlab (HEMERO Toolbox Robotics). La forma de configurar el automation Server es la siguiente:

Previamente debe tenerse instalada la versión 7.0 de Matlab en al computadora.

Dentro del IDE de Visual Basic, aplicación cliente, debe incorporarse al proyecto actual la referencia de la librería de Matlab Automation Server. Para realizar lo anterior, en la barra de menú principal debe desplegarse la etiqueta de proyecto y seleccionar la opción de referencia, esto nos abrirá una ventana emergente en donde debemos de seleccionar la caja de que habilita a “Matlab Automation Server Type Library” como se muestra en la siguiente figura:



Con el procedimiento anterior quedo configurado el Automation Server, Ahora para crear una instancia de servidor de Matlab, dentro del código de nuestra aplicación es necesario definir un objeto como se muestra a continuación:

Dim Matlab As Object

Una vez declarado el objeto se debe de crear con la siguiente instrucción:

```
Set Matlab = CreateObject("Matlab.Application")
```

En estos momentos ya se encuentra creada una instancia del servidor de Matlab y esta totalmente disponible para ejecutar alguna de las siguientes funciones permitidas desde Visual Basic:

Ejecución de comandos al servidor Matlab

El programa cliente puede ejecutar comandos en el servidor de Matlab usando las siguientes funciones:

| Función | Descripción |
|----------------|-------------------------------------------|
| <u>Execute</u> | Ejecuta comandos de MATLAB en el servidor |
| <u>Eval</u> | Evalúa comandos de MATLAB en el servidor |

Intercambiando datos con el servidor

El servidor de Matlab provee de algunas funciones para leer y escribir datos en algún espacio de trabajo de Matlab (workspace), en cada uno de estos comandos pasas el nombre de la variable a leer o escribir y el nombre del espacio de trabajo donde se almacenara el dato.

| Función | Descripción |
|-------------------------|------------------------------------------------|
| <u>GetCharArray</u> | Devuelve un arreglo de caracteres del servidor |
| <u>GetFullMatrix</u> | Devuelve una matriz del servidor |
| <u>GetWorkspaceData</u> | Devuelve cualquier tipo de dato del servidor |
| <u>PutCharArray</u> | Coloca un arreglo de caracteres del servidor |
| <u>PutFullMatrix</u> | Coloca una matriz del servidor |
| <u>PutWorkspaceData</u> | Coloca cualquier tipo de dato del servidor |

Controlando la ventana del servidor

Estas funciones permiten hacer visible o no la ventana del servidor de Matlab

| Función | Descripción |
|-----------------------|---------------------------------------------------------------|
| MaximizeCommandWindow | Hace visible la ventana de Matlab en el escritorio de Windows |
| MinimizeCommandWindow | Minimiza la ventana de Matlab |

Terminando el servidor

La siguiente función termina la instancia creada del servidor de Matlab

| Función | Descripción |
|---------|-------------------------------|
| Quit | Termina el servidor de MATLAB |

SOLID WORKS



SolidWorks es un software de diseño mecánico en 3D y Sólidos, siendo sus características más relevantes:

- Totalmente nativo para Windows, pero también existen versiones para otras plataformas.
- Fácil de aprender y usar.
- Liderazgo en Innovación, funcionalidades únicas y exclusivas.
- Compatibilidad inigualada, con software de CAD existentes.
- Potencia para realizar cualquier diseño, sin limitaciones técnicas.
- Totalmente abierto, uso de estándares Windows.
- Aplicaciones verticales integradas, amplían la capacidad de SolidWorks a análisis, simulación, gestión, ...

Con solidworks se puede hacer lo siguiente:

Modelar cualquier objeto que se pueda imaginar, incluyendo piezas de chapa y objetos con complejas superficies esculpidas, crear piezas y ensamblajes según especificaciones, probar alternativas de diseño a increíble velocidad y generar formas complejas con esfuerzo mínimo entre otras cosas. Por ello este es este programa de diseño cada una de las partes del manipulador mostradas en el capítulo 4 “Diseño de configuración”.

APENDICE C
“Código fuente”

Control de posición (Matlab)

| ControlBrazo.m | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> %CONTROL DE POSICION MEDIANTE LA RESOLUCION DEL MODELO INVERSO %MANIPULADOR DE TRES GRADOS DE LIBERTAD % se limpian las variables clear x % vector de abscisas de la trayectoria clear y % vector de ordenadas de la trayectoria clear z % vector de cotas de la trayectoria clear phi % vector de ángulos de la trayectoria clear TG % matriz de transformaciones lineales clear q % matriz de ángulos % se calculan los vectores x, y, z para la línea recta entre dos puntos % (coordenada actual y coordenada mas cercana) % que sequira el manipulador % Si band = 0 es un trayectoria en el plano un XY con Z constante % Si band = 1 es un trayectoria en el plano un Z con X y Y constantes % (Si band = 1 físicamente baja o sube el órgano terminal por las fichas) if band == 0 % En este bloque se calcula la coordenada a donde se dirigirá el brazo % ya sea un contenedor o coordenada de alguna ficha if contenedor == 0 % se obtiene el renglón de la coordenada mas cercana a la actual renglon = renglonCoordCercana(coordActual,B); A = B(renglon,:); B(renglon,:)= []; % Se borra coordenada de la pieza seleccionada else % si contenedor = 1 la próxima posición es la de un contenedor if contPieza == 0 A = coordConteBlanco; else A = coordConteNegro; end end if coordActual(1,1) == A(1,1) if coordActual(1,2)>A(1,2) y=[coordActual(1,2):-0.5:A(1,2)]; else y=[coordActual(1,2):0.5:A(1,2)]; end x= A(1,1)*ones(1,length(y)); z = 14 * ones(1,length(y)); % orientación del órgano terminal con cero grados phi=zeros(1,length(y)); else if coordActual(1,1)>A(1,1) x = [coordActual(1,1):-0.5:A(1,1)]; else x = [coordActual(1,1):0.5:A(1,1)]; end end % generación del vector y a partir de la ecuación de la recta % entre dos puntos for i = 1:length(x) y(i)= ((A(1,2)-coordActual(1,2))/(A(1,1)- coordActual(1,1)))*(x(i)-coordActual(1,1)) + coordActual(1,2); z = 14 * ones(1,length(x)); end phi=zeros(1,length(x)); end </pre> | <pre> % se guarda la próxima coordenada actual coordActual = A(1,:); else % si band = 1 es una trayectoria para bajar o subir la ficha if bajaOrgano == 1 % baja el órgano terminal z=[Zmax:-0.5:Zmin]; else % sube el órgano terminal z=[Zmin:0.5:Zmax]; end % vectores x y y constantes x=coordActual(1,1)*ones(1,length(z)); y=coordActual(1,2)*ones(1,length(z)); % orientacion cero phi=zeros(1,length(z)); end % Creamos la trayectoria de transformaciones lineales de manera dinámica for k=1:length(x) TG(:,k) = [cos(phi(k)) -sin(phi(k)) 0 x(k); sin(phi(k)) cos(phi(k)) 0 y(k); 0 0 1 z(k); 0 0 0 1]; end % Calculamos la cinemática inversa para cada uno de los puntos de la % trayectoria de transformación lineal q = ikine(dhparam,tolerancia,ilimite,TG); % Representamos gráficamente el manipulador mientras recorre la trayectoria figure(1) plotbot(dhparam,q,'f') % elimino las columnas no necesarias de la matriz q de ángulos q(:,6)= [] q(:,5)= [] q(:,2)= [] %Transformación de radianes a grados q(:,2)= (q(:,2))*180/pi; % Conversión de ángulos a coordenadas maquina for j=1:3 for i=1:size(q,1) q(i,j)= CoordMaqEslabones(q(i,j),j); end end % En NumReg se guarda el numero de renglones que hay en la matriz q % variable necesaria para el almacenamiento la matriz y transmisión al % microcontrolador NumReg = size(q,1) % En fichas se guarda el numero de fichas que aun faltan por seleccionar, % variable necesaria para el ciclo de ejecución desde la interfaz de Visual % Basic fichas= size(B,1) % FIN CONTROL BRAZO </pre> |

| constantes.m | CoordMaqEslabones.m |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> % CONSTANTES PARA EL CONTROL DE POSICION DEL % MANIPULADOR DE TRES GRADOS DE LIBERTAD % Longitud efectiva de los eslabones de la cadena cinemática en cm a0=0 d1=17 d2=6.5 a3=36 a4=46 d5=10 contador = 0 ; % posiciones de home, contenedor de fichas blancas y negras coordhome = [33 0] coordConteBlanco = [33 30] coordConteNegro = [33 -30] % cota maxima y minima que alcanzara el manipulador Zmax= 14 Zmin= 2.5 % la coordenada inicial es la posicion home coordActual = coordhome % Matriz de parametros Denavit - Hartenberg dhparam= [0 a0 0 0 0; % eslabon 0 0 0 0 d1 0; % eslabon 1 pi/2 0 0 d2 0; % eslabon 2 0 a3 0 0 0; % eslabon 3 0 a4 0 0 0; % eslabon 4 -pi/2 0 0 -d5 0]; % eslabon 5 % Establecemos la tolerancia y el número máximo de iteraciones tolerancia=1e-1 ilimite=1000 % Sumamos 13 a todas las abscisas de las coordenadas de las piezas % ya que el brazo se encuentra 13 cm separado del tablero sobre el % eje x % se suma 13 a todas las abscisas debido al sistema inercial del brazo % localizado en el centro de rotación del eslabón 1 for i = 1:length(B) B(i,1) = B(i,1) + 13 end </pre> | <pre> % si Y=1 para el eslabon1 % si Y=2 para el eslabon2 % si Y=3 para el eslabon3 function CoordMaq = CoordMaqEslabones(X,Y) % asignacion de variables para el eslabon 1 if Y == 1 %Ángulos experimentales X0 = 90; X1 = 0; X2 = -90; %Coordenadas maquina experimentales FX0 = 510; FX1 = 256; FX2 = 2; end % asignación de variables para el eslabón 2 if Y == 2 %Ángulos experimentales X0 = 180; X1 = 90; X2 = 0; %Coordenadas maquina experimentales FX0 = 2; FX1 = 255; FX2 = 512; end % asignación de variables para el eslabón 3 if Y == 3 %Ángulos experimentales X0 = -45; X1 = -90; X2 = -170; %Coordenadas maquina experimentales FX0 = 5; FX1 = 180; FX2 = 355; end %Diferencias FX01 = (FX1-FX0)/(X1-X0); FX12 = (FX2-FX1)/(X2-X1); FX012 = (FX12-FX01)/(X2-X0); % Formula de diferencias divididas CoordMaq = round(FX0 + FX01*(X-X0) + FX012*(X-X0)*(X- X1)); </pre> |
| renglonCoordCercana.m | |
| <pre> % Función que recibe la coordenada actual y la matriz de coordenadas % de las piezas. % regresa el renglón de la coordenada mas cercana a la actual function renglon = renglonCoordCercana(home,coord) for i = 1:size(coord,1) vecDist(i)= sqrt(((coord(i,1)- home(1,1))^2) + ((coord(i,2)- home(1,2))^2)); end </pre> | <pre> for i = 1:size(coord,1) if vecDist(i) == min(vecDist) renglon = i; break end end </pre> |

Código del microcontrolador (CCS)

| control_automatico1.c | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> /**ESTE PROGRAMA CONTROLA LA POSICION DE LOS**** ****ESLABONES DE UN BRAZO MANIPULADORCON DE **** *****TRES GRADOS DE LIBERTAD***** ****PARA EL CONCURSO NACIONAL DE MINIROBOTICA**** *****AUTORES: AGUILAR M. JEANNETE**** ***** TREJO S. HERIBERTO**** *****VERSION : 0.1***** *****DEFINICION DE LAS LIBRERIAS NECESARIAS**** #include <16f877.h> #include <stdlib.h> #include HS,NOWDT,NOPROTECT,NOLVP #include delay(clock=2000000) #include rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7) #include fast_io(A) #include fast_io(B) #include fast_io(D) #include fast_io(C) #include <input.c> #include <posiciona_motores.c> /**PROTECCION DEL ESPACIO DE MEMORIA DONDE RESIDE*/ *****EL BOOTLOADER***** #include ORG 0x1F00,0x1FFF void loader() { } /**NUMERO DE COORDENAS MAXIMA DE UNA TRAYECTORIA DE*/ /** DE TRANSFORMACION LINEAL*/ #define SIZE 100 /**ESTRUCTURA DE DATOS QUE ALMACENA UNA COORDENADA*/ struct coord { signed int m1; // valor para el angulo del eslabon 1 int m2; // valor para el angulo del eslabon 2 int m3; // valor para el angulo del eslabon 3 }; /**DECLARACION DE LAS ESTRUCTURAS TIPO coord UTILIZADAS*/ /** COORDENADA HOME*/ struct coord home = { 272,239,299}; /**TRAYECTORIA DE TRANSFORMACION LINEAL*/ struct coord coordenadas[SIZE]; </pre> | <pre> /*FUNCION DE QUE CONFIGURA LOS RECURSOS NECESARIOS*/ *****PARA EL CONTROL FISICO***** void configura (void){ // CINTURA: sensor posicion ANA5, bit de control RD0 , // velocidad CCP1 (PWM) // HOMBRO: sensor posicion ANA6, bit de control RD1 , // velocidad CCP2 (PWM) // CODO: sensor posicion ANA7, bit de control RD2 , // velocidad RD3 (PWM) // CONTROL ELECTROIMAN : RC3 // CONTROL SENSOR COLOR : RC0 set_tris_e(0xff); set_tris_c(0x81); set_tris_d(0x00); setup_adc_ports(ALL_ANALOG); setup_adc(ADC_CLOCK_INTERNAL); setup_CCP1(CCP_PWM); setup_CCP2(CCP_PWM); setup_timer_2(T2_DIV_BY_1,255,1); //1.2kHz set_pwm1_duty(0); set_pwm2_duty(0); output_low(PIN_D3); } // end configure /*FUNCION QUE HABILITA EL ELECTROIMAN PARA TOMAR LAS FICHAS*/ int toma_ficha(){ int color; output_high(PIN_C3); delay_ms(300); color = input(PIN_C0); printf("\n\r color=%u",color); return (color); } // end toma_ficha /*FUNCION QUE DESHABILITA EL ELECTROIMAN PARA DEJAR LAS FICHAS*/ void deja_ficha(){ output_low(PIN_C3); } // end deja_ficha /*FUNCION QUE RECIBE UNA TRAYECTORIA DE TRANSFORMACION LINEAL*/ void introduce_rutina(int numReg){ int i; for(i=0; i<numReg; i++) { coordenadas[i].m1= get_int(); coordenadas[i].m2= get_int(); coordenadas[i].m3= get_int(); } // end for } // end introduce_rutina </pre> |

control_automatico1.c

```

/*FUNCION QUE POSICIONA UNA TRATECTORIA DE TRANSFORMACION LINEAL*/
void empieza_trayectoria(int numReg){
    int i;

    for(i=0;i<numReg;i++){
        posiciona_m1(coordenadas[i].m1);
        delay_ms(100);
        posiciona_m3(coordenadas[i].m3);
        delay_ms(100);
        posiciona_m2(coordenadas[i].m2);
    }
} // Fin empieza_trayectoria

/*FUNCION QUE POSICIONA EN COORDENADA HOME*/
void home1(){
    posiciona_m1(home.m1);
    delay_ms(100);
    posiciona_m2(home.m2);
    delay_ms(100);
    posiciona_m3(home.m3);
} // fin home

void ejecuta() {
    int numReg;
/******RECIBE EL NUMERO DE COORDENADAS DE LA TRAYECTORIA***/
    numReg = get_int();
/******RECIBE UNA TRAYECTORIA DE TRANSFORMACIONES***/
    introduce_rutina(numReg);
/******EMPIEZA LA TRAYECTORIA DE TRANSFORMACIONES***/
    empieza_trayectoria(numReg);
} // fin ejecuta

/******FUNCION PRINCIPAL******/
main() {
    int color;
    configura();
    delay_ms(100);

/******INICIALMENTE EL BRAZO SE POSICIONA EN HOME***/
    printf("Buscando posicion home");
    home1();
    printf("Posicion home lista");
    printf("listo");

    while(TRUE){
        delay_ms(100);
        printf("listo");
        printf("En espera de coordenadas");
        ejecuta();
    }
}

/******MODIFICO BANDERA DE LISTO EN LA INTERFAZ DE VB ***/
    printf("listo");
/* EL BRAZO FICICAMENTE SE ENCUENTRA ARRIBA DE LA POSICION*/
/* MAS CERCANA*/
    printf("En espera de coordenadas");
    ejecuta();
    color = toma_ficha();
/******MODIFICO BANDERA DE COLOR EN LA INTERFAZ DE VB ***/
    if ( color == 0){
        printf("negro");
    }
    else{
        printf("blanco");
    }
/* EL BRAZO FICICAMENTE BAJO POR LA FICHA*/
/******MODIFICO BANDERA DE LISTO EN LA INTERFAZ DE VB ***/
    printf("listo");
    printf("En espera de coordenadas");
    ejecuta();
/******MODIFICO BANDERA DE LISTO EN LA INTERFAZ DE VB ***/
    printf("listo");
/* EL BRAZO FICICAMENTE SUBIO LA FICHA*/
    printf("En espera de coordenadas");
    ejecuta();
    deja_ficha();
/* EL BRAZO DEJA LA FICHA EN EL CONTENEDOR CORRESPONDIENTE*/

} //fin while

} // fin main

/*FIN*/

```

posiciona_motores.c

```
int valor;
int coor_actual_m1;
int band;

void posiciona_m1(int posicion_grado1){
set_adc_channel(5);
delay_ms(10);
valor = read_adc();
if ( valor < posicion_grado1)
{
output_low(PIN_D0); //giro horario
band=1;
set_pwm1_duty(230);
do {
if (valor >= (posicion_grado1-30))
{
set_pwm1_duty(190);
}
else if( (valor>=(posicion_grado1-80)) &&
(valor<(posicion_grado1-30)) )
{
set_pwm1_duty(210);
}
valor= read_adc();
} while(!(valor== posicion_grado1));
set_pwm1_duty(0);
} // end if
else if ( valor > posicion_grado1)
{
output_high(PIN_D0); //giro ant
band=0;
set_pwm1_duty(230);
do {
if (valor <= (posicion_grado1+30))
{
set_pwm1_duty(190);
}
else if( (valor<=(posicion_grado1+80)) &&
(valor>(posicion_grado1+30)) )
{
set_pwm1_duty(210);
}
valor= read_adc();
} while(!(valor== posicion_grado1));
set_pwm1_duty(0);
} //end if
coor_actual_m1=posicion_grado1;
} // end posiciona_m1

void posiciona_m2(int posicion_grado2){
set_adc_channel(6);
delay_ms(10);
valor = read_adc();
if ( valor < posicion_grado2)
{
output_low(PIN_D1); //giro horario
band=1;
set_pwm2_duty(170);
do {
if (valor >= (posicion_grado2-30))
{
set_pwm2_duty(135);
}
else if( (valor>=(posicion_grado2-80)) &&
(valor<(posicion_grado2-30)) )
{
set_pwm2_duty(140);
}
valor= read_adc();
} while(!(valor== posicion_grado2));
set_pwm2_duty(0);
} // end if
else if ( valor > posicion_grado2)
{
output_high(PIN_D1); //giro ant
band=0;
set_pwm2_duty(220);
do {
if (valor <= (posicion_grado2+30))
{
set_pwm2_duty(180);
}
else if( (valor<=(posicion_grado2+80)) &&
(valor>(posicion_grado2+30)) )
{
set_pwm2_duty(200);
}
valor= read_adc();
} while(!(valor== posicion_grado2));
set_pwm2_duty(0);
} //end if
} // end posiciona_m2

void posiciona_m3(int posicion_grado3){
set_adc_channel(7);
delay_ms(10);
valor = read_adc();
if ( valor < posicion_grado3)
{
output_low(PIN_D2); //giro horario
band=1;
output_high(PIN_D3);
do {
valor= read_adc();
} while(!(valor== posicion_grado3));
output_low(PIN_D3);
} // end if
else if ( valor > posicion_grado3)
{
output_high(PIN_D2); //giro ant
band=0;
output_high(PIN_D3);
do {
valor= read_adc();
} while(!(valor== posicion_grado3));
output_low(PIN_D3);
} //end if
} // end posiciona_m3
```

APENDICE D
“Hojas de especificación técnica”



PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

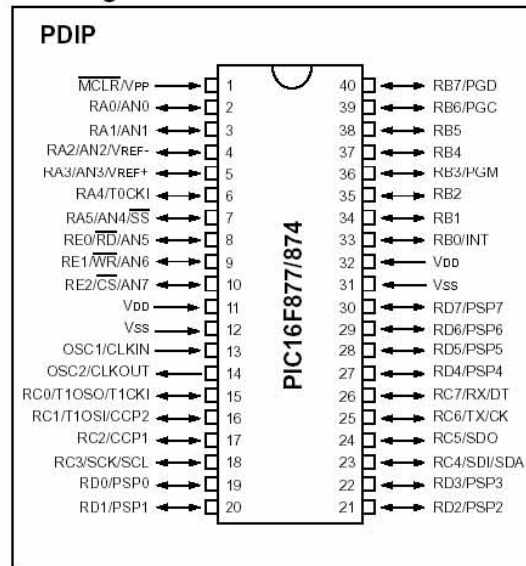
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F876
- PIC16F874
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
Up to 368 x 8 bytes of Data Memory (RAM)
Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC
oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM
technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two
pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature
ranges
- Low-power consumption:
 - < 0.6 mA typical @ 3V, 4 MHz
 - 20 µA typical @ 3V, 32 kHz
 - < 1 µA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler,
can be incremented during SLEEP via external
crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period
register, prescaler and postscaler
- Two Capture, Compare, PWM modules
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master
mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver
Transmitter (USART/SCI) with 9-bit address
detection
- Parallel Slave Port (PSP) 8-bits wide, with
external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for
Brown-out Reset (BOR)

LM139/LM239/LM339/LM2901/LM3302 Low Power Low Offset Voltage Quad Comparators

General Description

The LM139 series consists of four independent precision voltage comparators with an offset voltage specification as low as 2 mV max for all four comparators. These were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. These comparators also have a unique characteristic in that the input common-mode voltage range includes ground, even though operated from a single power supply voltage.

Application areas include limit comparators, simple analog to digital converters; pulse, squarewave and time delay generators; wide range VCO; MOS clock timers; multivibrators and high voltage digital logic gates. The LM139 series was designed to directly interface with TTL and CMOS. When operated from both plus and minus power supplies, they will directly interface with MOS logic— where the low power drain of the LM339 is a distinct advantage over standard comparators.

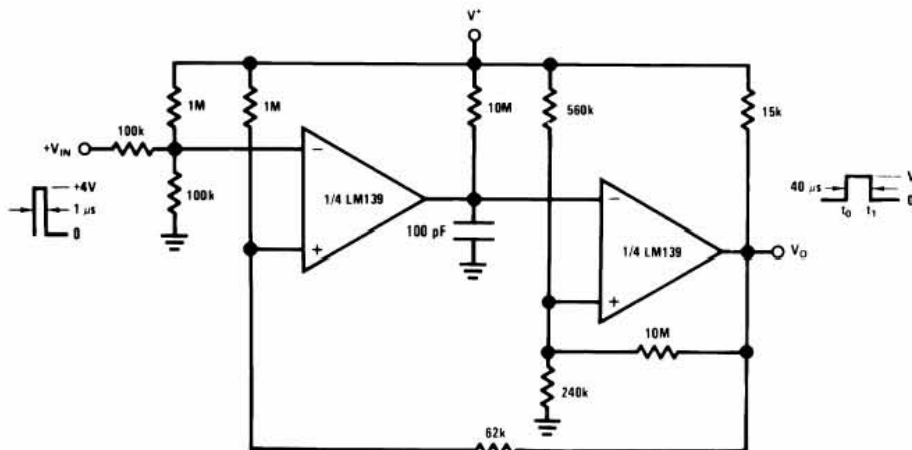
Features

- Wide supply voltage range
- LM139/139A Series: 2 to 36 V_{DC} or ± 1 to $\pm 18 V_{DC}$
- LM2901: 2 to 36 V_{DC} or ± 1 to $\pm 18 V_{DC}$
- LM3302: 2 to 28 V_{DC} or ± 1 to $\pm 14 V_{DC}$
- Very low supply current drain (0.8 mA) — independent of supply voltage
- Low input biasing current: 25 nA
- Low input offset current: ± 5 nA
- Offset voltage: ± 3 mV
- Input common-mode voltage range includes GND
- Differential input voltage range equal to the power supply voltage
- Low output saturation voltage: 250 mV at 4 mA
- Output voltage compatible with TTL, DTL, ECL, MOS and CMOS logic systems

Advantages

- High precision comparators
- Reduced V_{OS} drift over temperature
- Eliminates need for dual supplies
- Allows sensing near GND
- Compatible with all forms of logic
- Power drain suitable for battery operation

One-Shot Multivibrator with Input Lock Out



00570612

LM78XX

Series Voltage Regulators

General Description

The LM78XX series of three terminal regulators is available with several fixed output voltages making them useful in a wide range of applications. One of these is local on card regulation, eliminating the distribution problems associated with single point regulation. The voltages available allow these regulators to be used in logic systems, instrumentation, HiFi, and other solid state electronic equipment. Although designed primarily as fixed voltage regulators these devices can be used with external components to obtain adjustable voltages and currents.

The LM78XX series is available in an aluminum TO-3 package which will allow over 1.0A load current if adequate heat sinking is provided. Current limiting is included to limit the peak output current to a safe value. Safe area protection for the output transistor is provided to limit internal power dissipation. If internal power dissipation becomes too high for the heat sinking provided, the thermal shutdown circuit takes over preventing the IC from overheating.

Considerable effort was expended to make the LM78XX series of regulators easy to use and minimize the number of external components. It is not necessary to bypass the out-

put, although this does improve transient response. Input bypassing is needed only if the regulator is located far from the filter capacitor of the power supply.

For output voltage other than 5V, 12V and 15V the LM117 series provides an output voltage range from 1.2V to 57V.

Features

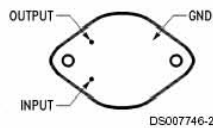
- Output current in excess of 1A
- Internal thermal overload protection
- No external components required
- Output transistor safe area protection
- Internal short circuit current limit
- Available in the aluminum TO-3 package

Voltage Range

| | |
|---------|-----|
| LM7805C | 5V |
| LM7812C | 12V |
| LM7815C | 15V |

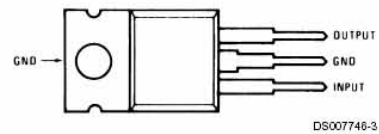
Connection Diagrams

**Metal Can Package
TO-3 (K)
Aluminum**



Bottom View
Order Number LM7805CK,
LM7812CK or LM7815CK
See NS Package Number KC02A

**Plastic Package
TO-220 (T)**



Top View
Order Number LM7805CT,
LM7812CT or LM7815CT
See NS Package Number T03B

H11L1M

H11L2M

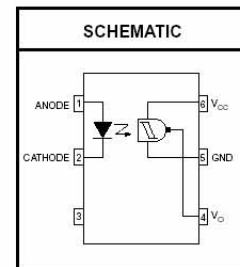
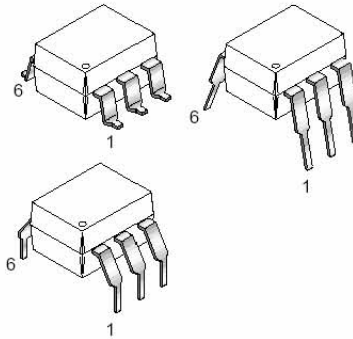
H11L3M

DESCRIPTION

The H11LX series has a high speed integrated circuit detector optically coupled to a gallium-arsenide infrared emitting diode. The output incorporates a Schmitt trigger, which provides hysteresis for noise immunity and pulse shaping. The detector circuit is optimized for simplicity of operation and utilizes an open collector output for maximum application flexibility.

FEATURES

- High data rate, 1 MHz typical (NRZ)
- Free from latch up and oscillation throughout voltage and temperature ranges.
- Microprocessor compatible drive
- Logic compatible output sinks 16 mA at 0.4 V maximum
- Guaranteed on/off threshold hysteresis
- Wide supply voltage capability, compatible with all popular logic systems
- Underwriter Laboratory (UL) recognized—file #E90700



APPLICATIONS

- Logic to logic isolator
- Programmable current level sensor
- Line receiver—eliminate noise and transient problems
- A.C. to TTL conversion—square wave shaping
- Digital programming of power supplies
- Interfaces computers with peripherals

| ABSOLUTE MAXIMUM RATINGS | | | | |
|-----------------------------------------------------------------|--------------------|--------|----------------|-------|
| Parameters | Symbol | Device | Value | Units |
| TOTAL DEVICE | | | | |
| Storage Temperature | T _{STG} | All | -55 to +150 | °C |
| Operating Temperature | T _{OPR} | All | -40 to +85 | °C |
| Lead Solder Temperature | T _{SOL} | All | 260 for 10 sec | °C |
| Total Device Power Dissipation @ 25°C Derate Above 25°C | P _D | All | 250 | mW |
| | | | 2.94 | mW/°C |
| EMITTER | | | | |
| Continuous Forward Current | I _F | All | 60 | mA |
| Reverse Voltage | V _R | All | 6 | V |
| Forward Current - Peak (1 μs pulse, 300 pps) | I _{F(pk)} | All | 3.0 | A |
| LED Power Dissipation 25°C Ambient Derate Linearly From 25°C | P _D | All | 120 | mW |
| | | | 1.41 | mW/°C |
| DETECTOR | | | | |
| Detector Power Dissipation @ 25°C Derate Linearly from 25°C | P _D | All | 150 | mW |
| | | | 2.0 | mW/°C |
| V ₄₅ Allowed Range | V _O | All | 0 to 16 | V |
| V ₆₅ Allowed Range | V _{CC} | All | 3 to 16 | V |
| I ₄ Output Current | I _O | All | 50 | mA |

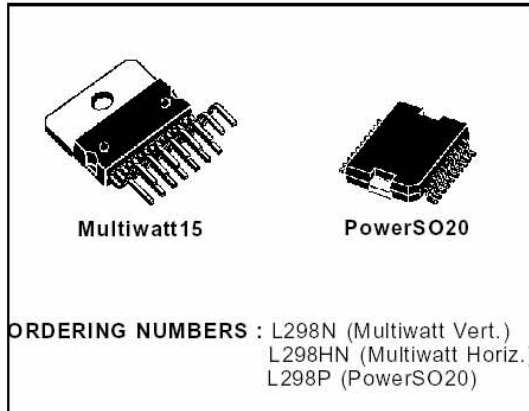


DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

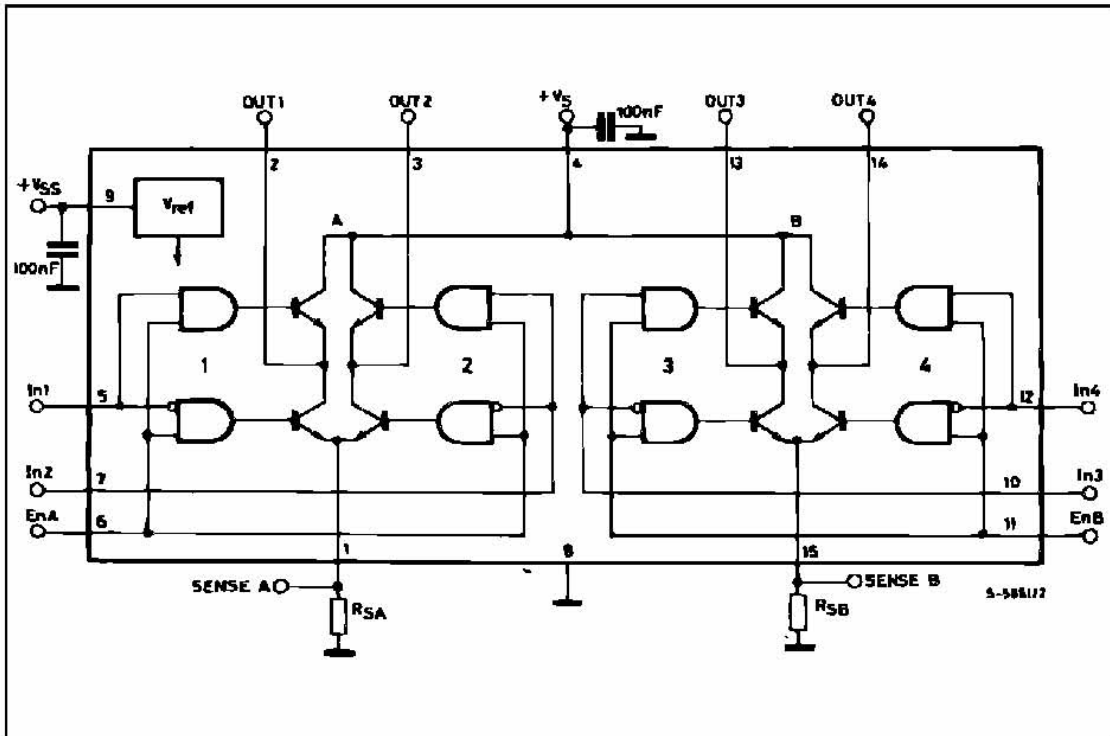
DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the con-



nection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM

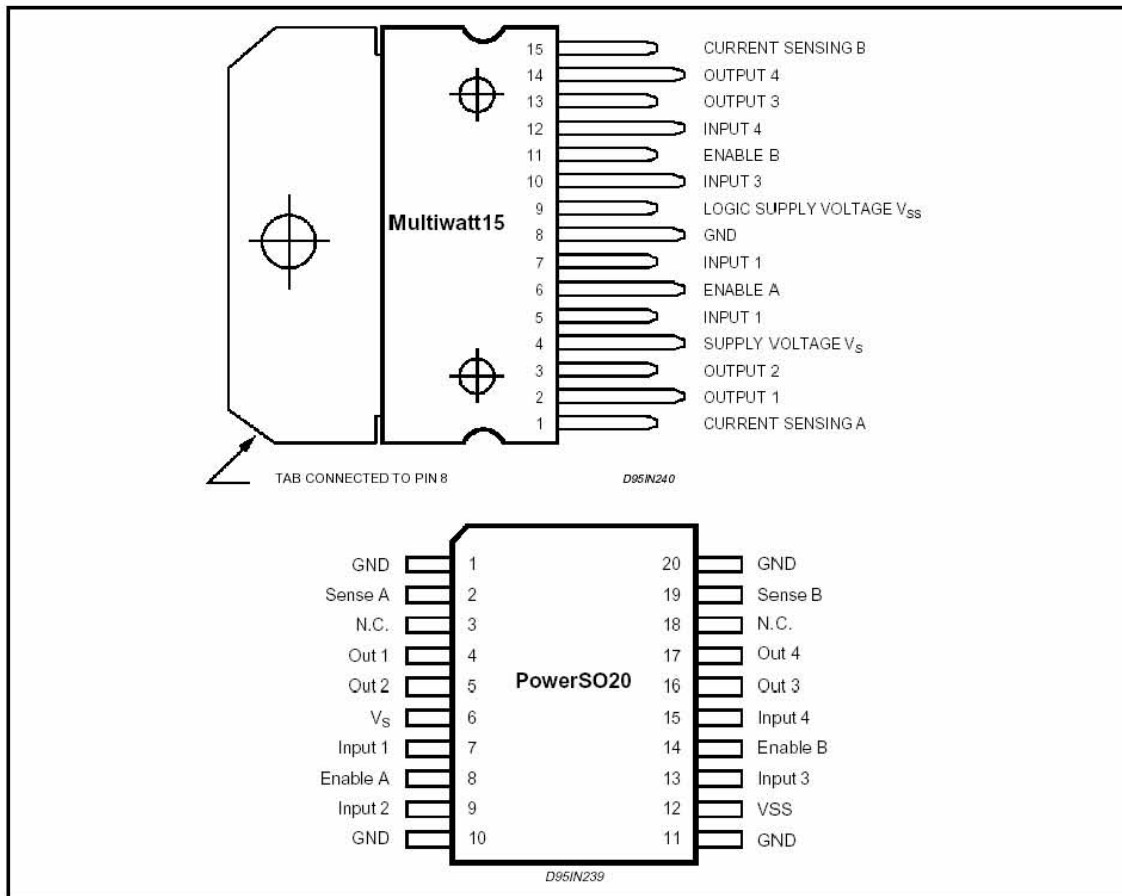


L298

ABSOLUTE MAXIMUM RATINGS

| Symbol | Parameter | Value | Unit |
|----------------|-----------------------------------------------------|------------|------------|
| V_S | Power Supply | 50 | V |
| V_{SS} | Logic Supply Voltage | 7 | V |
| V_I, V_{en} | Input and Enable Voltage | -0.3 to 7 | V |
| I_O | Peak Output Current (each Channel) | | |
| | – Non Repetitive ($t = 100\mu s$) | 3 | A |
| | – Repetitive (80% on –20% off; $t_{on} = 10ms$) | 2.5 | A |
| | –DC Operation | 2 | A |
| V_{sens} | Sensing Voltage | -1 to 2.3 | V |
| P_{tot} | Total Power Dissipation ($T_{case} = 75^\circ C$) | 25 | W |
| T_{op} | Junction Operating Temperature | -25 to 130 | $^\circ C$ |
| T_{stg}, T_j | Storage and Junction Temperature | -40 to 150 | $^\circ C$ |

PIN CONNECTIONS (top view)

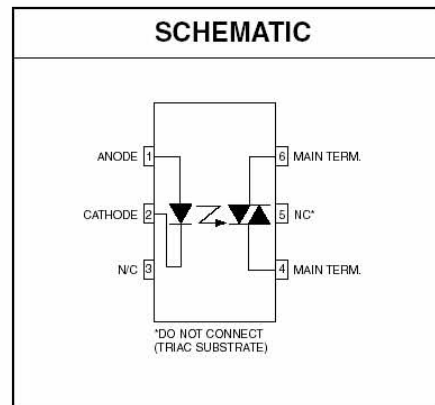
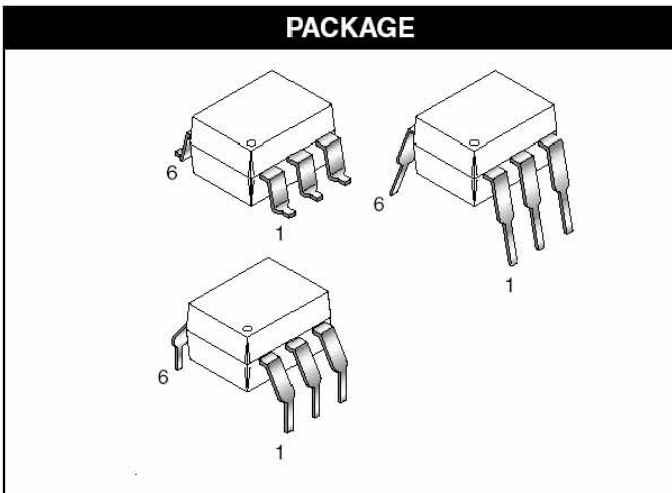


THERMAL DATA

| Symbol | Parameter | PowerSO20 | Multiwatt15 | Unit |
|------------------|-------------------------------------|-----------|-------------|--------------|
| $R_{th\ j-case}$ | Thermal Resistance Junction-case | Max. | 3 | $^\circ C/W$ |
| $R_{th\ j-amb}$ | Thermal Resistance Junction-ambient | Max. | 35 | $^\circ C/W$ |

(*) Mounted on aluminum substrate

MOC3010M MOC3011M MOC3012M MOC3020M MOC3021M MOC3022M MOC3023M



DESCRIPTION

The MOC301XM and MOC302XM series are optically isolated triac driver devices. These devices contain a GaAs infrared emitting diode and a light activated silicon bilateral switch, which functions like a triac. They are designed for interfacing between electronic controls and power triacs to control resistive and inductive loads for 115 VAC operations.

FEATURES

- Excellent I_{FT} stability—IR emitting diode has low degradation
- High isolation voltage—minimum 5300 VAC RMS
- Underwriters Laboratory (UL) recognized—File #E90700
- Peak blocking voltage
 - 250V-MOC301XM
 - 400V-MOC302XM
- VDE recognized (File #94766)
 - Ordering option V (e.g. MOC3023VM)

APPLICATIONS

- | | |
|-----------------------|-----------------------------|
| • Industrial controls | • Solenoid/valve controls |
| • Traffic lights | • Static AC power switch |
| • Vending machines | • Incandescent lamp dimmers |
| • Solid state relay | • Motor control |
| • Lamp ballasts | |

Sensitive Gate Triacs

Silicon Bidirectional Thyristors

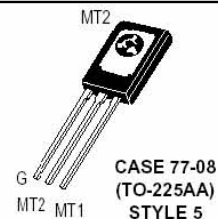
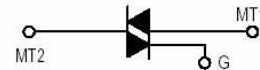
... designed primarily for full-wave ac control applications, such as light dimmers, motor controls, heating controls and power supplies; or wherever full-wave silicon gate controlled solid-state devices are needed. Triac type thyristors switch from a blocking to a conducting state for either polarity of applied anode voltage with positive or negative gate triggering.

- Sensitive Gate Triggering Uniquely Compatible for Direct Coupling to TTL, HTL, CMOS and Operational Amplifier Integrated Circuit Logic Functions
- Gate Triggering 4 Mode — 2N6071A,B, 2N6073A,B, 2N6075A,B
- Blocking Voltages to 600 Volts
- All Diffused and Glass Passivated Junctions for Greater Parameter Uniformity and Stability
- Small, Rugged, Thermopad Construction for Low Thermal Resistance, High Heat Dissipation and Durability

2N6071A,B*
2N6073A,B*
2N6075A,B*

*Motorola preferred devices

TRIACS
4 AMPERES RMS
200 thru 600 VOLTS



MAXIMUM RATINGS ($T_J = 25^\circ\text{C}$ unless otherwise noted.)

| Rating | Symbol | Value | Unit |
|------------------------------------------------------------------------------------------------------|---------------------|-------------------|----------------------|
| *Peak Repetitive Off-State Voltage ⁽¹⁾ (Gate Open, $T_J = 25$ to 110°C) | V_{DRM} | 200 400 600 | Volts |
| *On-State Current RMS ($T_C = 85^\circ\text{C}$) | $I_{\text{T(RMS)}}$ | 4 | Amps |
| *Peak Surge Current (One Full cycle, 60 Hz, $T_J = -40$ to $+110^\circ\text{C}$) | I_{TSM} | 30 | Amps |
| Circuit Fusing Considerations ($t = 8.3$ ms) | I^2t | 3.7 | A^2s |
| *Peak Gate Power | P_{GM} | 10 | Watts |
| *Average Gate Power | $P_{\text{G(AV)}}$ | 0.5 | Watt |
| *Peak Gate Voltage | V_{GM} | 5 | Volts |

*Indicates JEDEC Registered Data.

1. V_{DRM} for all types can be applied on a continuous basis. Blocking voltages shall not be tested with a constant current source such that the voltage ratings of the devices are exceeded.

Preferred devices are Motorola recommended choices for future use and best overall value.

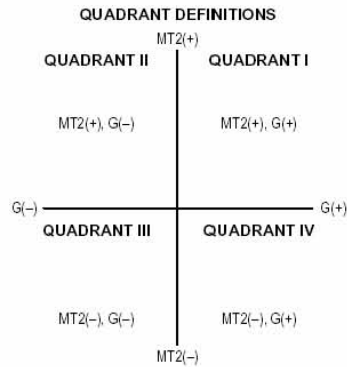
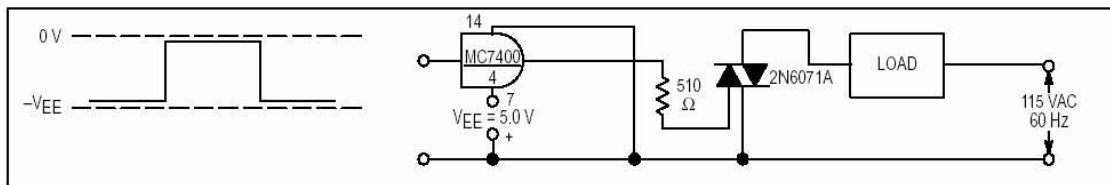
REV 1

2N6071A,B 2N6073A,B 2N6075A,B

| Gate Trigger Current (Continuous dc) (Main Terminal Voltage = 12 Vdc, R _L = 100 ohms) Maximum Value | Type | I _{GT} @ T _J | QUADRANT (See Definition Below) | | | |
|----------------------------------------------------------------------------------------------------------------------|-------------------------------|-------------------------------------|------------------------------------|----------|-----------|----------|
| | | | I mA | II mA | III mA | IV mA |
| | 2N6071A 2N6073A 2N6075A | +25°C | 5 | 5 | 5 | 10 |
| | | -40°C | 20 | 20 | 20 | 30 |
| | 2N6071B 2N6073B 2N6075B | +25°C | 3 | 3 | 3 | 5 |
| | | -40°C | 15 | 15 | 15 | 20 |

*Indicates JEDEC Registered Data.

**SAMPLE APPLICATION:
TTL-SENSITIVE GATE 4 AMPERE TRIAC
TRIGGERS IN MODES II AND III**

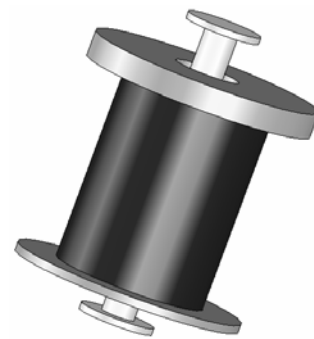
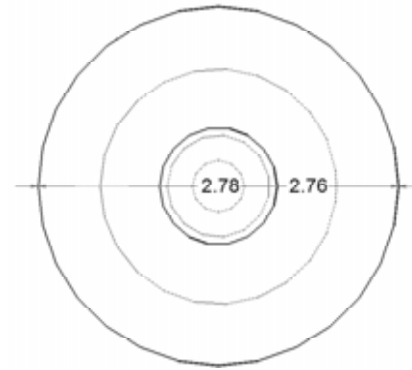
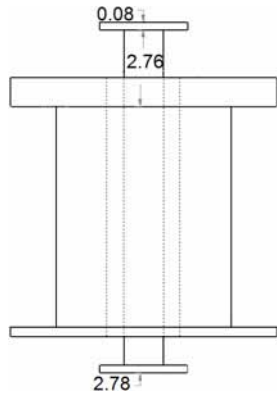


- Trigger devices are recommended for gating on Triacs. They provide:
1. Consistent predictable turn-on points.
 2. Simplified circuitry.
 3. Fast turn-on time for cooler, more efficient and reliable operation.

SENSITIVE GATE LOGIC REFERENCE

| IC Logic Functions | Firing Quadrant | | | |
|-----------------------|-----------------|----------------|----------------|----------------|
| | I | II | III | IV |
| TTL | | 2N6071A Series | 2N6071A Series | |
| HTL | | 2N6071A Series | 2N6071A Series | |
| CMOS (NAND) | 2N6071B Series | | | 2N6071B Series |
| CMOS (Buffer) | | 2N6071B Series | 2N6071B Series | |
| Operational Amplifier | 2N6071A Series | | | 2N6071A Series |
| Zero Voltage Switch | | 2N6071A Series | 2N6071A Series | |

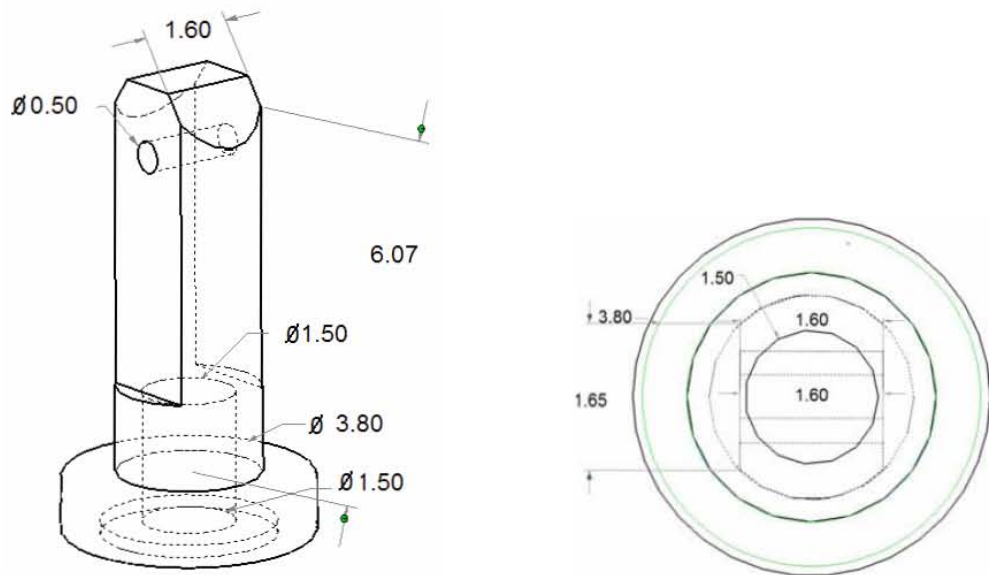
APENDICE E
“Planos de fabricación”



Dimensiones en [mm]

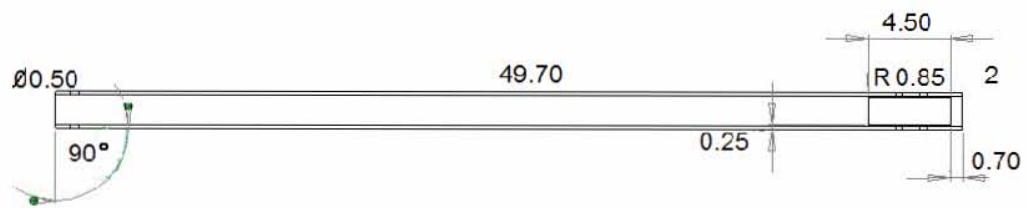
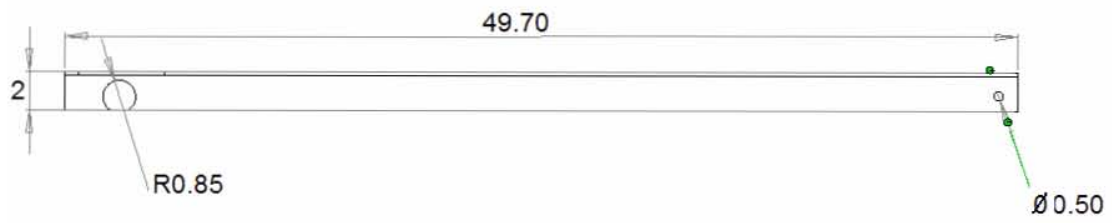
Pieza: Electroimán

Material: Acero al silicio y polietileno



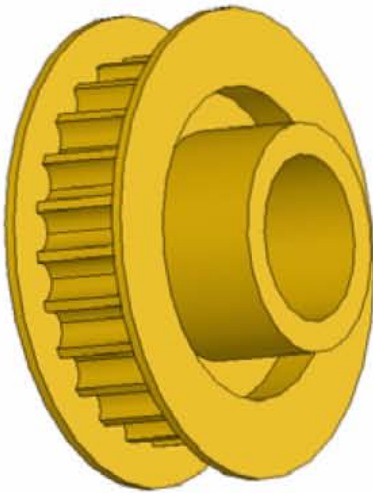
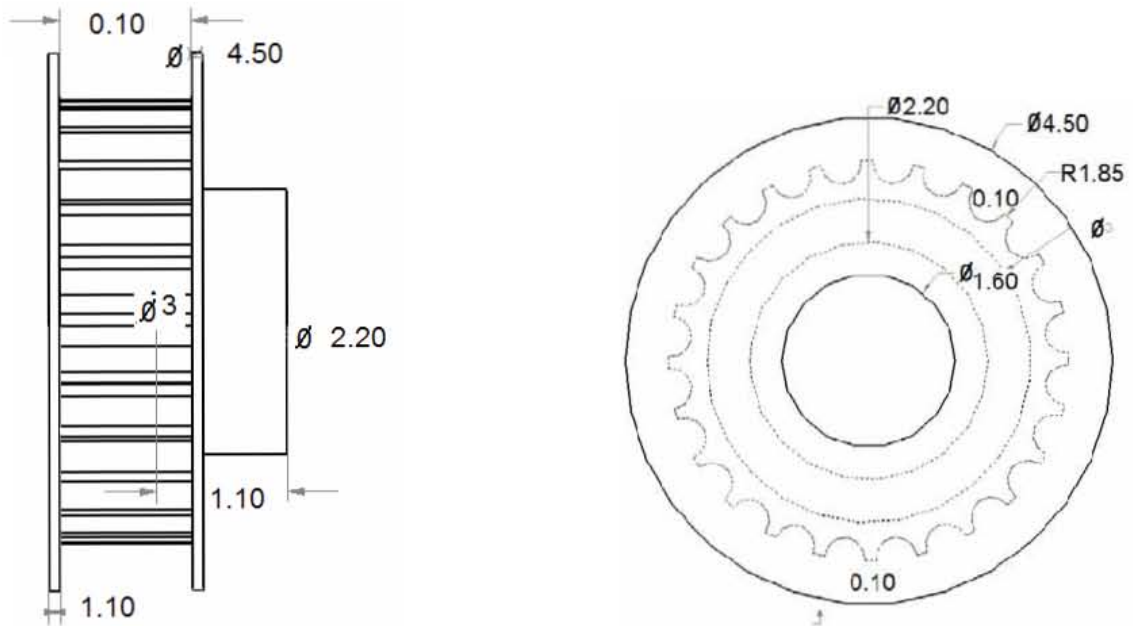
Dimensiones en [mm]

Pieza: Órgano terminal
Material: Polietileno



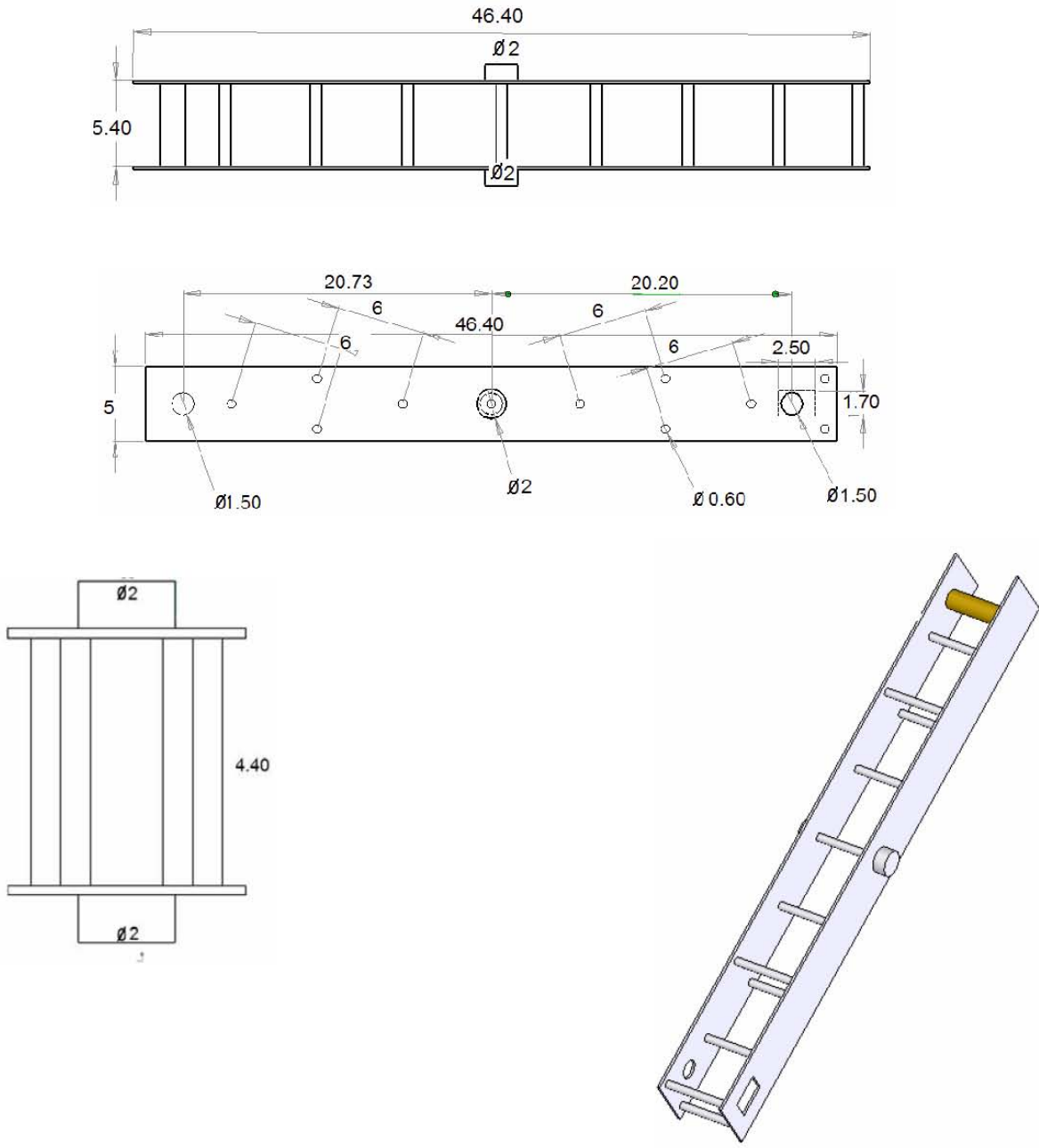
Dimensiones en [mm]

Pieza: Eslabón 3
Material: Aluminio



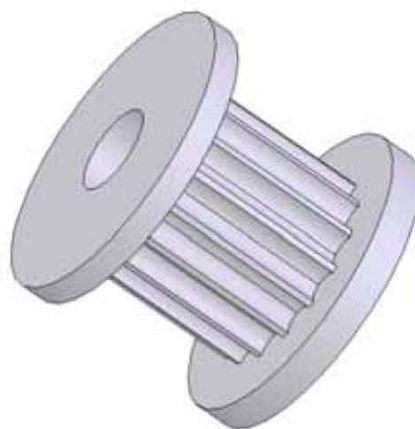
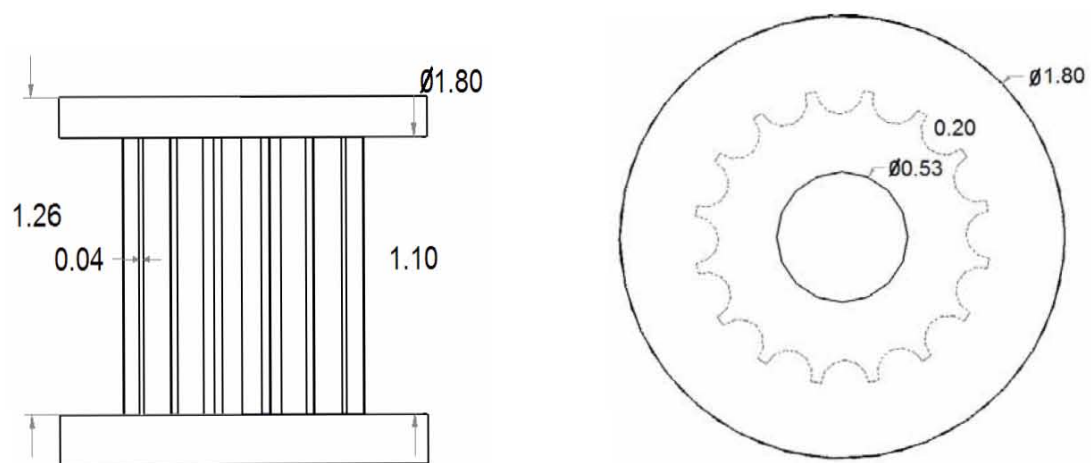
Dimensiones en [mm]

Pieza: Polea
Material: Nylamid



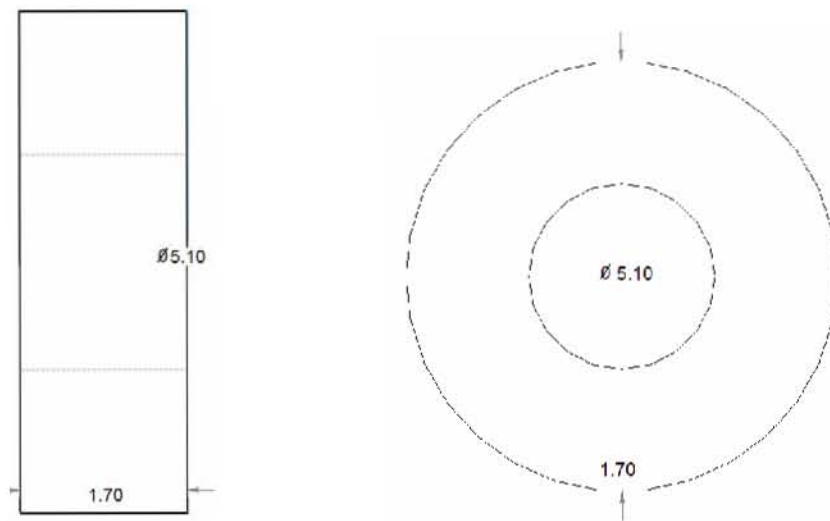
Dimensiones en [mm]

Pieza: Eslabón 2
Material: Aluminio



Dimensiones en [mm]

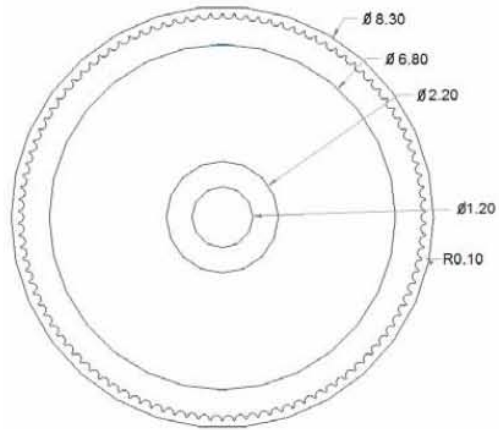
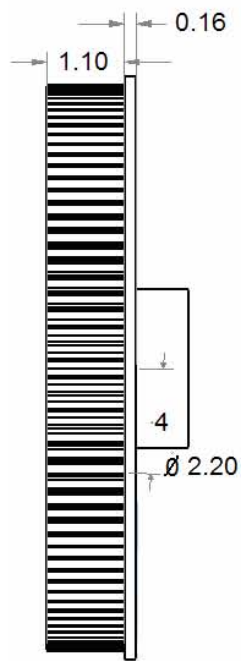
Pieza: Polea
Material: Nylamid



Dimensiones en [mm]

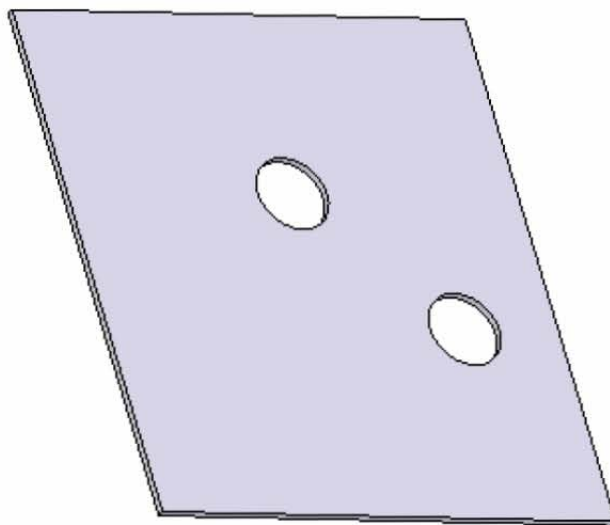
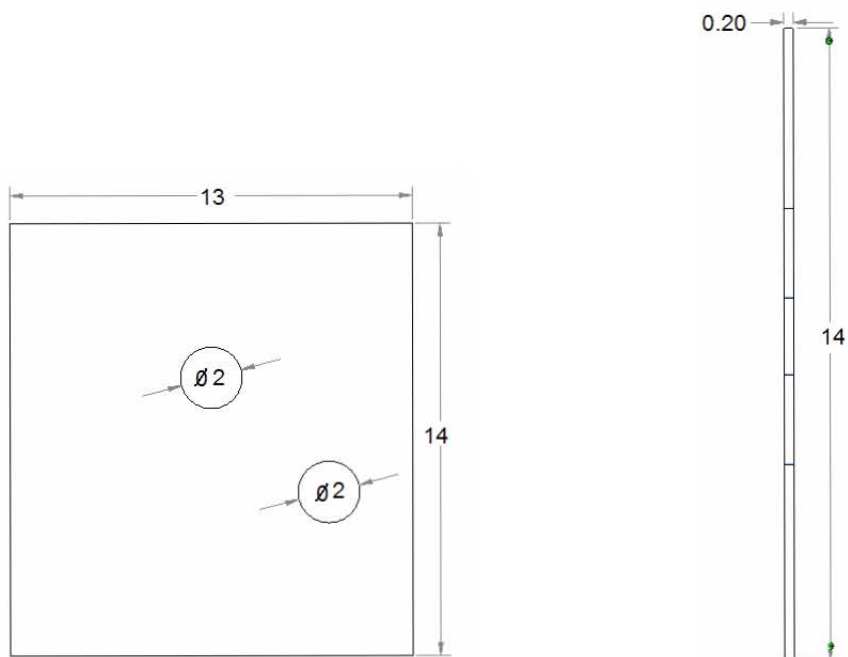
Pieza: Cople

Material: Polietileno negro



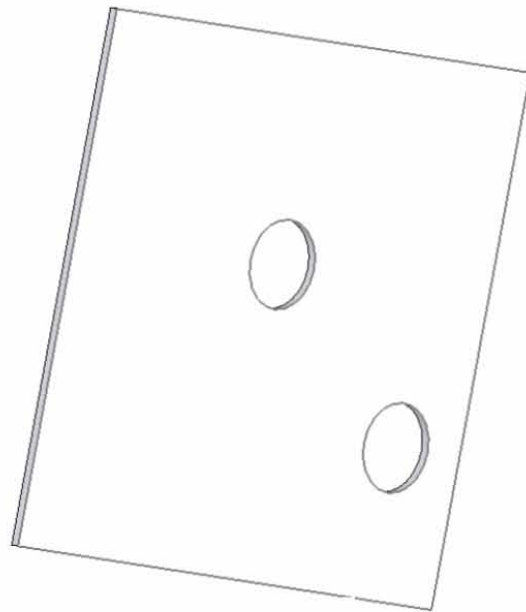
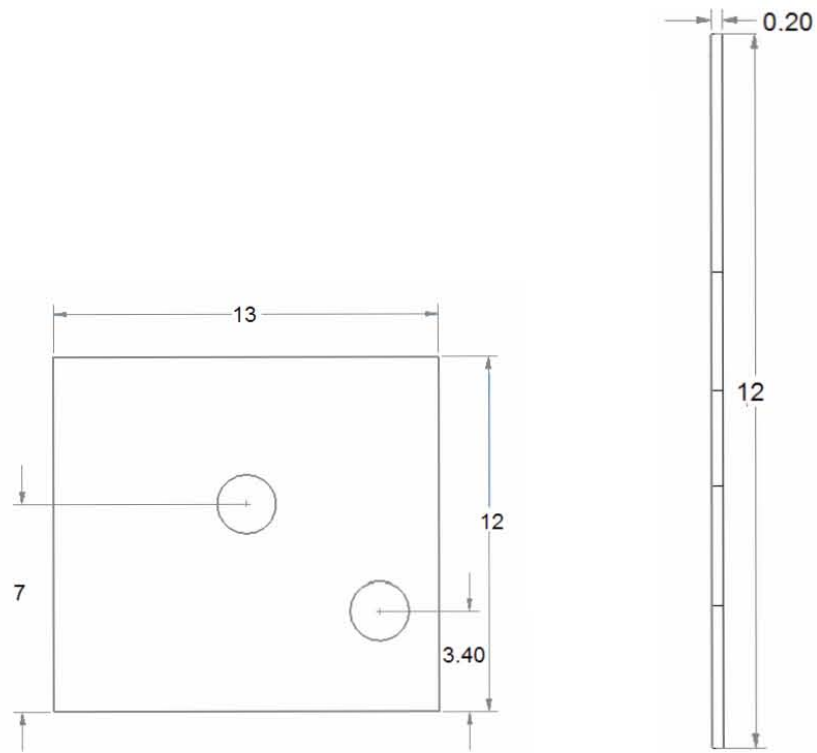
Dimensiones en [mm]

Pieza: Polea
Material: Polietileno



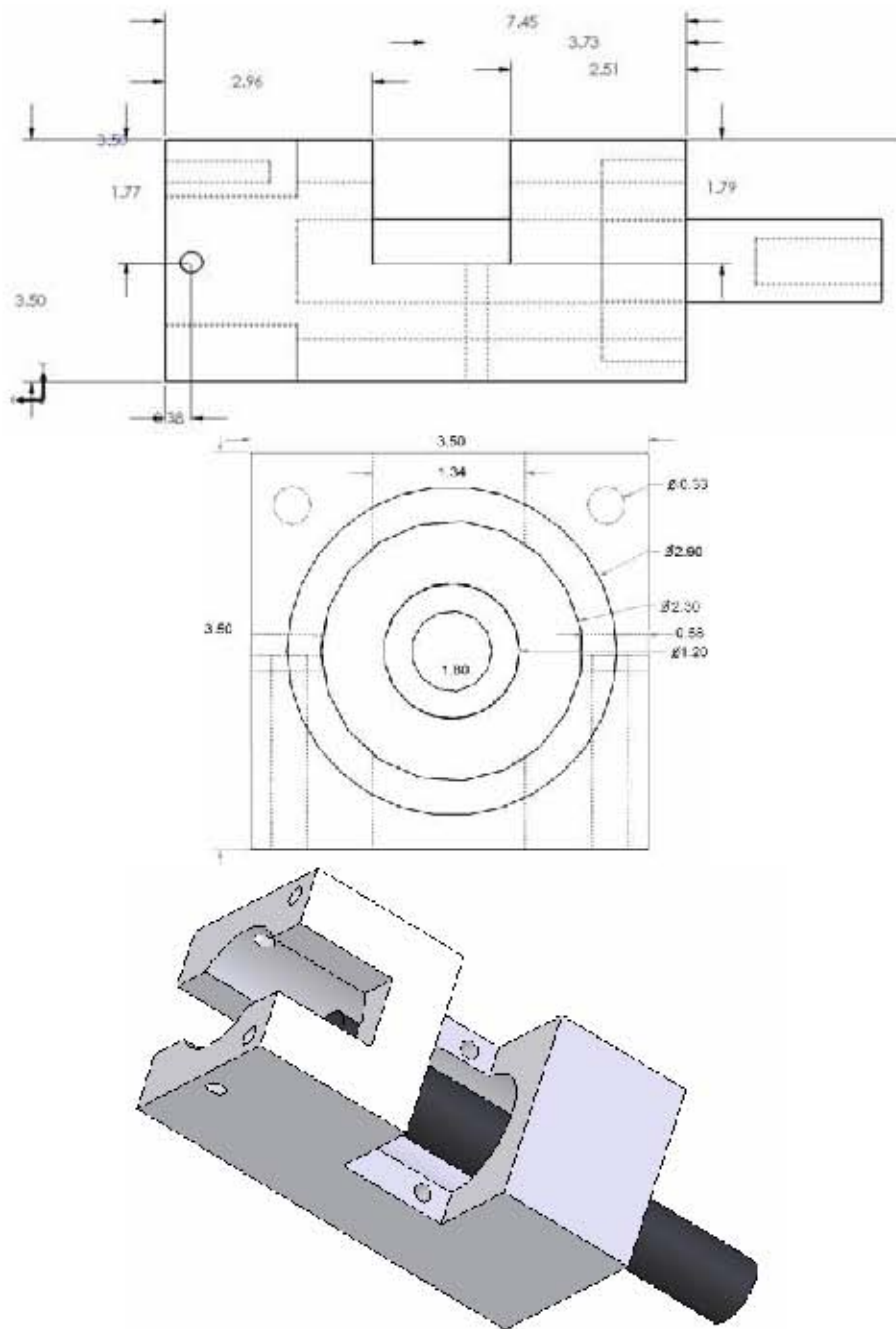
Dimensiones en [mm]

Pieza: Placa superior de soporte
Material: Aluminio



Dimensiones en [mm]

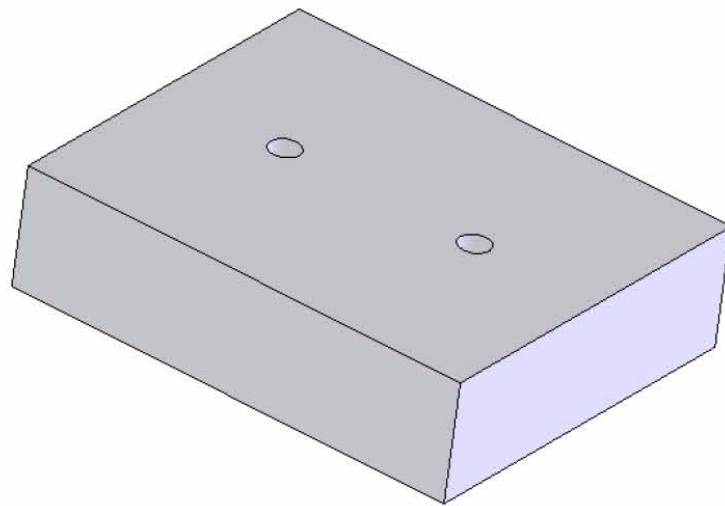
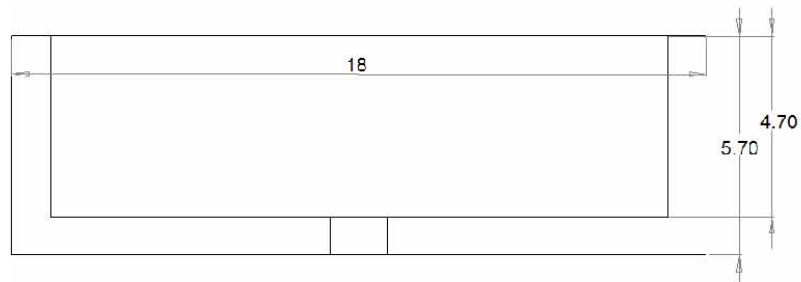
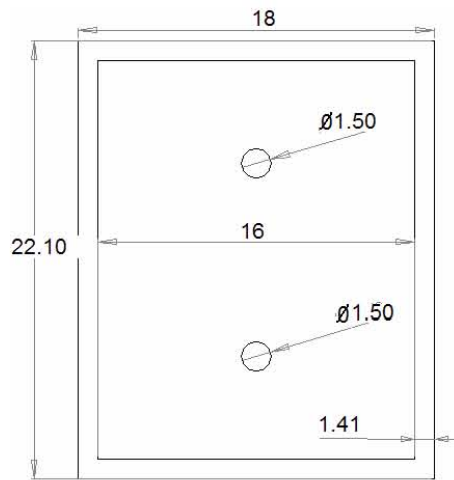
Pieza: Placa inferior de soporte
Material: Aluminio



Dimensiones en [mm]

Pieza: Bloque estructural

Material: Aluminio



Dimensiones en [mm]

Pieza: Base
Material: Aluminio

Lista de figuras

Lista de figuras

Capítulo 1

| | |
|-------------------------------------------------------------------------------|----|
| Figura 1.1. Robot y su interacción con el entorno..... | 2 |
| Figura 1.2. Disposición de las fichas en el tablero y sus contenedores..... | 5 |
| Figura 1.3. Escenario principal..... | 5 |
| Figura 1.4. Eslabones del sistema brazo manipulador..... | 7 |
| Figura 1.5. Dimensiones del escenario..... | 5 |
| Figura 1.6. Eslabón 5 unido al eslabón 4 mediante una articulación libre..... | 10 |
| Figura 1.7. Electroimán..... | 10 |
| Figura 1.8. Quinto eslabón o elemento terminal..... | 11 |

Capítulo 2

| | |
|-----------------------------------------------------------------------------------|----|
| Figura 2.1. Sistema de referencia en el origen..... | 14 |
| Figura 2.2. Formulación de los parámetros Denavit - Hartenberg..... | 18 |
| Figura 2.3. Configuración y sistemas de referencia del manipulador..... | 20 |
| Figura 2.4. Modelo del sistema de tres grados de libertad en 2D..... | 23 |
| Figura 2.5. Modelo del sistema en 3D (vista lateral derecha)..... | 24 |
| Figura 2.6. Modelo del sistema en 3D (vista lateral izquierda)..... | 24 |
| Figura 2.7. Movimientos tridimensionales del sistema (vista lateral derecha).... | 25 |
| Figura 2.8. Movimientos tridimensionales del sistema (vista lateral superior).... | 25 |

Capítulo 3

| | |
|---------------------------------------------------------------------------------|----|
| Figura 3.1. Potenciómetro de alta precisión..... | 28 |
| Figura 3.2. Fototransistor..... | 28 |
| Figura 3.3. Arreglo utilizado en el sensor infrarrojo..... | 29 |
| Figura 3.4. Configuración del par emisor – receptor..... | 30 |
| Figura 3.5. Esquema eléctrico del mínimo de operación del pic..... | 31 |
| Figura 3.6. Sistema eléctrico de la interfaz RS-232..... | 32 |
| Figura 3.7. Esquema eléctrico del circuito que controla al electroimán..... | 33 |
| Figura 3.8. Esquema eléctrico de la etapa de potencia de uno de los motores.... | 34 |
| Figura 3.9. Esquema eléctrico del opto-acoplador..... | 34 |

Capítulo 4

| | |
|------------------------------------------------------------------------------------------------------------|----|
| Figura 4.1. Módulos del software de control..... | 36 |
| Figura 4.2. Interfaz de usuario..... | 37 |
| Figura 4.3. Área de captura de coordenadas por parte del usuario..... | 37 |
| Figura 4.4. Disposición de la fichas con respecto al sistema de referencia..... | 38 |
| Figura 4.5. Configuración de los parámetros de comunicación serial..... | 38 |
| Figura 4.6. Controles de simulación / prueba, tiempo de proceso y ventana de información de la prueba..... | 39 |
| Figura 4.7. Animación de la prueba en 3D..... | 39 |
| Figura 4.8. Selección de la ficha más cercana, de la coordenada actual, al electroimán..... | 41 |
| Figura 4.9. Diagrama de flujo del algoritmo de control de posición del brazo... | 42 |

Capítulo 5

| | |
|-----------------------------------------------------------------------|----|
| Figura 5.1. Configuración del eslabón 1 (vista 1)..... | 47 |
| Figura 5.2. Configuración del eslabón 1 (vista 2)..... | 48 |
| Figura 5.3. Bloque estructural A y elementos acoplados..... | 49 |
| Figura 5.4. Motor A y transmisión del eslabón 1..... | 49 |
| Figura 5.5. Elementos de la transmisión del eslabón 2..... | 50 |
| Figura 5.6. Bloque estructural B y elementos acoplados (vista 1)..... | 51 |
| Figura 5.7. Bloque estructural B y elementos acoplados (vista 2)..... | 51 |
| Figura 5.8. Transmisión de doble banda, eslabones 3 y 4..... | 52 |
| Figura 5.9. Eslabón 5..... | 53 |
| Figura 5.10. Órgano terminal y sensor del color de la ficha..... | 54 |
| Figura 5.11. Ensamble final del brazo manipulador..... | 55 |

Capítulo 6

| | |
|------------------------------------------------|----|
| Figura 6.1. Repetibilidad del manipulador..... | 62 |
|------------------------------------------------|----|

Lista de tablas

Lista de tablas

| | |
|----------------------------------------------------------------------------------|----|
| Tabla 1.1. Materiales utilizados para la construcción del robot..... | 9 |
| Tabla 2.1. Parámetros Denavit – Hartenberg para el modelo cinemático..... | 21 |
| Tabla 3.1. Características del microcontrolador..... | 31 |
| Tabla 4.1. Datos experimentales (lecturas del sensor de posición)..... | 44 |
| Tabla 4.2. 1ª y 2ª diferencia del método de interpolación..... | 44 |
| Tabla 4.3. Ejemplo de conversión de coordenada angular a coordenada máquina..... | 45 |
| Tabla 6.1. Ángulos requeridos para trasladarse de la coordenada 1 a la 2..... | 57 |
| Tabla 6.2. Ángulos requeridos para trasladarse de la coordenada 2 a la 3..... | 58 |
| Tabla 6.3. Ángulos requeridos para trasladarse de la coordenada 3 a la 4..... | 58 |
| Tabla 6.4. Ángulos requeridos para trasladarse de la coordenada 4 a la 5..... | 58 |
| Tabla 6.5. Rutina 1..... | 60 |
| Tabla 6.6. Rutina 2..... | 60 |
| Tabla 6.7. Rutina 3..... | 61 |
| Tabla 6.8. Rutina 4..... | 61 |
| Tabla 6.9. Rutina 5..... | 61 |

Referencia bibliográfica

Referencia bibliográfica

- [ES2004] Esteves Rebollo Sergio F., “Diseño mecánico de un manipulador”, Ed. Facultad de Ingeniería UNAM 2004.
- [LF1993] Lewis F., Abdallah C., Dawson D., Control of Robot Manipulators. Ed. Macmillan 1993.
- [AJ2000] Jose M. Angulo Usategui, Microcontroladores PIC. Ed. M
- [MP1986] Mikell P., Groover Mitchell Weiss, Roger N., Nagel Nicholas G., Industrial Robotics Technology, Programming, and Applications, Ed. McGraw-Hill 1986.
- [OA2001] Ollero A., Robótica, Manipuladores y Robots móviles. Ed. Alfa Omega Marcombo 2001.
- [SM1989] Spong Mark W, Vidyasagar M., Robot Dynamic and Control, Ed. John Wiley and Sons EUA, 1989.