



**TECNOLÓGICO
DE MONTERREY.®**

Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Ciudad de México

División de Ingeniería y Arquitectura

Ingeniería en Mecatrónica

Departamento de Ingeniería Eléctrica y Electrónica

Diseño y construcción de robot SCARA con procesamiento de imágenes e inteligencia artificial

Autores:

Benjamín Celis García

Ricardo Ruiz Torres

José Antonio Miranda Thewissen



Asesor:

Dr. Pedro Ponce

México D.F. Mayo, 2006

ÍNDICE

1. OBJETIVOS.....	3
1.1 OBJETIVOS GENERALES	3
2. MOTIVACIÓN	4
3. INTRODUCCIÓN	6
3.1 HISTORIA DE LOS SISTEMAS SCARA.....	6
3.2 MATERIALES MODERNOS. Nylamid®.....	7
3.3 PROCESAMIENTO DIGITAL DE IMÁGENES	10
3.4 SISTEMAS INTELIGENTES	11
Funciones de Membresía.....	11
Defusificación	12
Sistemas Tipo Mamdani[16].....	13
Sistemas Tipo Takagi-Sugeno-Kang[16]	13
4. CONCEPCIÓN DEL PROYECTO	15
5. MECÁNICA	18
5.1 MATERIALES	18
5.2 DISEÑO MECÁNICO	19
5.3 CARACTERÍSTICAS MECÁNICAS DE LOS SERVOMOTORES.....	19
Principio de Funcionamiento de los Servos	20
6. ETAPA DE CONTROL.....	22
6.1 DESCRIPCIÓN GENERAL	22
6.2 DIAGRAMA DE BLOQUES	22
6.3 ENTRADA Y SALIDA DE DATOS.....	24
6.4 PROCESAMIENTO DE IMAGENES	25
6.5 PROCESAMIENTO DE INFORMACIÓN	27
6.6 CONTROL DE MOTORES.....	29
7. ETAPA DE POTENCIA.....	32
8. RESULTADOS.....	33
8.1 DISEÑO MECÁNICO DEL SISTEMA	33
8.2 DISEÑO Y CONSTRUCCIÓN DEL GRIPPER.....	41
8.3 ESTRUCTURA DEL SISTEMA Y WEBCAM	43
8.4 PROGRAMAS EN MATLAB®.....	47
8.5 SISTEMAS DE CALIBRACIÓN DEL SISTEMA.....	49
Programa de calibración manual (creado en LabVIEW®).....	49
8.6 OPCIONES SOBRE EL TIPO DE CONTROL DEL SISTEMA	52
8.7 RESULTADOS DEL PROCESAMIENTO DE IMÁGENES	56
8.8 RESULTADOS DEL SISTEMA ROBÓTICO (APLICACIÓN DE CONCURSO DE MINI-ROBÓTICA IEEE)	58
8.9 RESULTADOS DEL SISTEMA ROBÓTICO (APLICACIÓN DE JUEGO DE GATO).....	59
8.10 RETROALIMENTACIÓN DEL SISTEMA	60
8.11 RESULTADOS DEL CONCURSO IEEE DE MINIROBÓTICA	61
9. OBSERVACIONES	62
10. CONCLUSIONES	64
11. TRABAJO A FUTURO	66
12. REFERENCIAS	67
13. ANEXOS	69
ANEXO 1 Bases del 10º Concurso de Minirobótica IEEE. Categoría de “Brazo Manipulador”	69
ANEXO 2 Codigos de Matlab®	73

1. OBJETIVOS

1.1 OBJETIVOS GENERALES

- Desarrollar un sistema robótico tipo SCARA capaz de competir cumpliendo las bases del concurso de minirobótica de la IEEE (modalidad de Brazo Manipulador). El sistema deberá tener cualidades de ser transportable y ensamblable en un tiempo menor a 10 minutos.
- Adquirir conocimientos sobre Inteligencia Artificial.
 - Aplicar la Teoría de Lógica Difusa y aplicarla en el control de velocidad de los brazos del robot.
 - Aplicar Inteligencia Artificial al sistema para lograr un jugador de "GATO" invencible.
- Desarrollar un sistema funcional de percepción mediante el procesamiento de imágenes para reconocimiento de objetos. Este sistema deberá estar bajo las bases del concurso de la IEEE y ser de bajo costo.

2. MOTIVACIÓN

El presente proyecto satisface dos grandes intereses del equipo:

1. Participar en el concurso de Mini robótica en la categoría de Brazo Manipulador organizado anualmente por la rama estudiantil de la IEEE. Este evento es uno de los concursos nacionales en robótica de mayor prestigio tanto por el renombre de la IEEE como por la calidad en los proyectos presentados por partes de las universidades participantes año con año. Se desea participar en el concurso para medir los conocimientos y habilidades del equipo en comparación con los de los alumnos de otras universidades del país.
2. Se tiene un amplio interés en adquirir mayores conocimientos sobre la teoría de Lógica Difusa y observar las ventajas y desventajas al aplicar dicha teoría en sistemas robóticos tipo SCARA (Selective Compliant Articulated Robot Arm).

Para lograr lo anterior, se construirá un sistema mecatrónico integral tipo SCARA que consta de las siguientes de tres grandes etapas: Mecánica, de Potencia y de Control, cada una de las cuales deberá de cumplir las bases del 11° Concurso de Minirobótica IEEE, las cuales se muestran a continuación:

El objetivo es diseñar, fabricar y controlar un robot manipulador, capaz de seleccionar piezas blancas y negras. Las piezas se encontrarán distribuidas aleatoriamente sobre un tablero de 40 cm. x 40 cm. y el robot se encargará de tomar y colocar las piezas en su respectivo contenedor, de manera autónoma y en el menor tiempo posible. El robot manipulador podrá seleccionar las piezas en cualquier orden, y colocarlas en su contenedor correspondiente conforme al color de la misma (Blanco y Negro). Se dispondrá de un total de 10 piezas de las cuales 5 serán de color negro y 5 de color blanco. Las piezas a seleccionar se colocarán dentro del tablero con una distribución geométrica predeterminada, la disposición de las piezas se dará a conocer 5 horas antes de la primera evaluación del robot, y la disposición será la misma en todas las pruebas;

siendo la separación mínima entre una pieza y otra de 2.5 cm. en todas las direcciones. La asignación de la configuración correspondiente a cada concursante se realizará mediante un sorteo el día del evento.

Especificaciones

1. Tablero de 40 cm 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.
2. Contenedores con medidas interiores de 20 cm. de cada lado y 10 cm. de altura, los cuales se colocarán como lo indica la figura, 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 2 cm. Del tablero. La posición de los contenedores, a cada lado del tablero, será fija y sin posibilidad de movimiento.
3. El robot se deberá colocar a un lado del tablero, equidistante a ambos contenedores, en el lugar indicado en la fotografía del final. La distancia del centro del robot al tablero, medida sobre el eje de simetría del arreglo, será libre. El concursante fijará su prototipo y no podrá moverlo una vez iniciada su rutina. El robot podrá 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.
4. El robot puede tener uno o más puntos de apoyo, y podrá realizar movimientos prismáticos o rotacionales en cualquiera de sus articulaciones.
5. Las piezas a mover serán cilíndricas, con medidas de 2.5 cm. de diámetro y altura variable entre 1.5 y 2.5 cm., con hierro de al menos 0.5 mm de espesor en la parte superior y sin perforaciones. Cada una de las piezas tendrá un peso máximo de 15 gramos.
6. El acomodo de las fichas se dará a conocer antes de las pruebas, pero básicamente se partirá como si el tablero fuera una mesa de coordenadas con una resolución de 0.5cm entre posiciones de centros de fichas. Los colores de las fichas y sus alturas no serán dados a conocer y corresponderá al robot identificarlas.

Los criterios de calificación y otros datos importantes sobre el concurso se encuentran en el Anexo I.

3. INTRODUCCIÓN

3.1 HISTORIA DE LOS SISTEMAS SCARA

SCARA son las siglas de "Selective Compliant Articulated Robot Arm" o "Brazo Robótica Articulado, Selectivo y Flexible".

Estos sistemas tienen sus inicios en 1979 cuando las empresas Sankyo y IBM lanzan al mercado el primer robot con estas características. Su desarrollo se llevó a cabo en la Universidad de Yamanashi en Japón [2][3].

Estos sistemas han tenido gran aceptación en la industria debido a la gran simplicidad en la cinemática que los rige: tienen tres juntas de revolución y una junta vertical prismática en el efector final. Son utilizados principalmente para operaciones de "recoger y colocar" (pick-and-place), en las cuales se requiere que el robot sea rígido en la posición vertical (debido a que debe empujar objetos hacia otros objetos) y que permita variaciones en el plano horizontal debido a las relativas imperfecciones en el posicionamiento entre los objetos manipulados y su contraparte en el ensamble. Estas características son intrínsecas de los sistemas SCARA [4].

En la Figura 3.1 se observa un sistema SCARA (a) y un manipulador serial (b) (Cincinnati Milacron T³). Al comparar ambos sistemas, el SCARA tiene una cinemática más sencilla por tener 4 grados de libertad (algunos tienen sólo 3 al carecer de la rotación en el efector final) contra los 6 del sistema de la Figura 3.1(b).

Para tareas espacialmente complejas (como el ensamble de los componentes de un auto), prefiere un sistema similar al de la Figura 3.1(b).

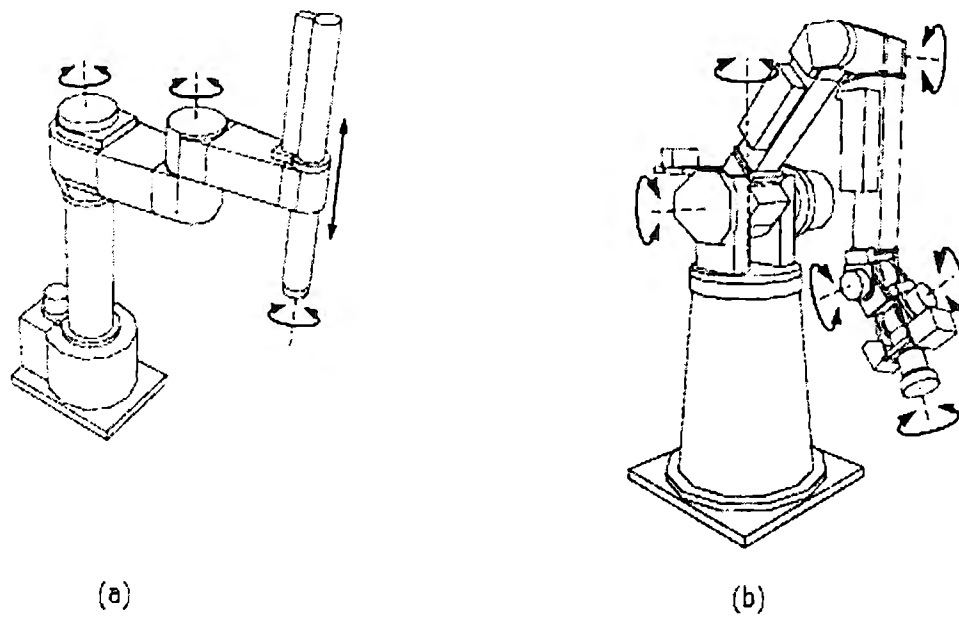


Figura 3.1a. Sistema tipo SCARA (a) y Manipulador Serial Cincinnati Milacron T³ (b) [4]

3.2 MATERIALES MODERNOS. Nylamid®.

Con los rápidos avances tecnológicos de nuestra época, los productos necesitan características cada vez más específicas. En cuanto a las propiedades mecánicas la mayor parte de las ocasiones lo que se busca es mejorarlas pero disminuyendo el peso de las componentes de los productos.

La disminución del peso de los objetos representa ahorros de energía especialmente durante los procesos industriales.

Hasta hace unos años, los materiales con un factor alto de resistencia-densidad eran demasiado caros, siendo el aluminio uno de los más accesibles y el titanio de los menos.

Sin embargo, en la última década se ha observado un decremento considerable de los precios en este tipo de materiales, haciéndolos viables para cada vez más aplicaciones. Los procesos que han influido en este hecho son principalmente las nuevas aleaciones, materiales cerámicos, materiales compuestos y nuevas formas de síntesis de polímeros.

El material que nos interesa en el presente proyecto es el Nylamid SL®, un polímero de fabricación mexicana que ofrece características muy deseables en materiales de construcción de componentes a nivel industrial [5]

Las ventajas más destacables de este material son las siguientes:

- **Buena relación Costo-Beneficio.** El rendimiento de la piezas fabricadas con **Nylamid®**, en combinación con el precio, es el principal motivo para su aplicación, ya que aunque en algunas ocasiones existan materiales mas baratos, su limitada durabilidad generan gastos extras derivados de la mayor frecuencia de recambio de las refacciones, los paros para mantenimiento, montaje y desmontaje.
- **Facilidad de maquinado.** Por su suavidad, la mayoría de estos productos se maquinan más rápido que los metales, dando como resultado, ahorros hasta del 50% en el tiempo de maquinado (ver Tabla 3.2a).
- **Ligereza.** Al pesar 2.7 veces menos que el Aluminio y hasta 8 veces menos que el Acero 1045 (ver Tabla 3.2a), los polímeros estructurales exigen menor esfuerzo de los motores, contribuyendo así a consumir menos energía.
- **Menor consumo de lubricantes.** Gracias a su bajo coeficiente de fricción el **Nylamid®** ayuda a reducir el consumo de lubricantes y en algunos casos lo elimina.
- **Reduce el nivel de ruido.** La capacidad de absorción de impactos tan característica de los plásticos ayudan a disminuir la emisión de ruido provocado por las partes que trabajan en movimiento y en contacto directo entre si.

El sufijo “SL” del Nylamid SL® significa “Súper Lubricado”. Está cargado con disulfuro de molibdeno para reducir su coeficiente de fricción. Esta variante se produce por polimerización directa de monómeros para formar polímeros de Nylon 6/12, al ser vaciados a presión atmosférica. Así se logran piezas de mayor volumen que con la inyección y extrusión.

Un comparativo de las características físicas del Nylamid SL® con otros materiales se muestra a continuación en la Tabla 3.2a.

Propiedad	Aluminio	Acero AISI 1045	Nylamid SL®
Gravedad Específica	2.7	7.86	1.02
Esfuerzo último a la tensión (MPa)	455	585	410
Módulo de elasticidad a la tensión (Kg/cm ²)	70,000	211,000	28,124
Relación esfuerzo último a la tensión/gravedad específica (para comparación)	168.5	74.43	400
Módulo de elasticidad a la compresión (Kg/m ²)	70,000	211,000	28,124
Dureza escala Shore D	- (sólo para polímeros)	- (sólo para polímeros)	D 85
Coefficiente de fricción dinámico	0.5	0.8	0.2 (vs acero)
Coefficiente de expansión térmica lineal (m/m K)	23.1	11.8	90 x 10 ⁻⁶
Punto de fusión (°C)	660.32	1538	420
Conductividad térmica W/(K m)	237	80.4	0.30
Resistividad volumétrica (Ohm-cm)	2.65 x 10 ⁻⁶	96.1 x 10 ⁻⁶	> 10 ¹³

Tabla 3.2.a Comparativo entre propiedades físicas del aluminio, acero 1045 y Nylamid. [5],[6],[7],[8]

En la Tabla 3.2a se puede observar que la relación resistencia-densidad del Nylamid SL® es más de 4 veces la del aluminio y casi 10 veces la del acero.

En cuanto al precio, en la Tabla 3.2b se muestra un comparativo

	Aluminio	Acero 1045	Nylamid SL®
Precio (usd/Kg)	5.09	1.80	20.00

Tabla 3.2b. Comparativo entre los precios del aluminio, acero 1045 y Nylamid

3.3 PROCESAMIENTO DIGITAL DE IMÁGENES

El hombre ha tenido el deseo de procesar digitalmente las imágenes debido a la gran cantidad de aplicaciones a las que se puede dirigir el proceso. Además, la gran versatilidad y velocidad que ofrecen los paquetes computacionales para llevarlo a cabo hacen del procesamiento una tarea muy interesante, buscándose cada vez más nuevas y mejores formas de análisis de imágenes.

El procesamiento de imágenes tiene aplicaciones muy variadas: identificación de objetivos militares, rostros, huellas digitales, vehículos, animales, tumores en el cuerpo humano, obstrucción de arterias, etc.

En muchos de los sistemas robóticos de hoy en día, el desempeño total depende por completo de la etapa de visión. Como es el caso de nuestro proyecto.

Para el procesamiento visual de una escena, es necesario tener una imagen digital, que es un conjunto de valores acomodados en una matriz que indican la intensidad de luz en cada punto descrito por la misma. Para analizar la imagen se pasa por 3 niveles de procesamiento: bajo, medio y alto. En el nivel bajo se filtra el ruido de la imagen, dándole más nitidez. El nivel medio consta del mejoramiento de la imagen, para darle mayor contraste, y la segmentación de la imagen en las partes a reconocer [12]. El procesamiento de alto nivel tiene tres pasos: el reconocimiento de los objetos segmentados, el etiquetado de la imagen y la interpretación final de la escena. [11]

En la actualidad existen varios ejemplos de reconocimiento de imágenes especialmente en el área de medicina, en donde se practica el reconocimiento de imágenes en estudios de resonancia magnética (MRI), Rayos-X, mamografías, medicina nuclear, entre otros [13]. La detección de enfermedades y malformaciones ha mejorado notablemente desde la aplicación de métodos computacionales para el reconocimiento de imágenes.

3.4 SISTEMAS INTELIGENTES

El "Soft Computing" según el Dr. Lotfi Asker Zadeh, pionero en el desarrollo de la lógica difusa, es "una nueva forma de cómputo, que simula la notable habilidad de la mente humana de razonar y aprender en un ambiente de incertidumbre e imprecisión" [11]. Es posible por tanto referirnos a estos sistemas como "inteligentes", aunque la magnitud de su capacidad de "razonamiento" es aún muy limitada en comparación con la del ser humano.

Existen varias herramientas que conforman a esta disciplina surgida en la segunda mitad del siglo XX; algunas de ellas son la lógica difusa, las redes neuronales artificiales y los algoritmos genéticos. En el siguiente capítulo se explica la forma en que opera la lógica difusa, herramienta utilizada para el desarrollo de este proyecto.

La lógica clásica consta de proposiciones que adquieren valores de 0 o 1 para indicar si son verdaderas o falsas. En contraste, la lógica difusa permite asignar cualquier valor entre 0 y 1, estos incluidos, a una proposición. Se puede decir entonces que la veracidad de una proposición es parcial.

Utilizando reglas lingüísticas, del tipo "grande", "frío", etc., se pueden crear condiciones para la pertenencia de un valor a cada una de las variables en relación a cada característica que se desea evaluar.

Funciones de Membresía

Una función de membresía es la representación matemática de una regla lingüística. La forma de la función de membresía depende del concepto que se quiere modelar y del caso particular de la aplicación que se le está dando. Las formas más comunes son:

Triangular

Son las funciones más comunes. Son lineales y fáciles de implementar y programar.

$$A(x) = \begin{cases} \alpha \left(\frac{x-a}{c-a} \right) & \text{si } (a \leq x \leq c) \\ \alpha \left(\frac{x-b}{c-b} \right) & \text{si } (c \leq x \leq b) \\ 0 & \text{en cualquier otro caso} \end{cases}$$

Trapezoidal

Esta función también es comúnmente utilizada. Es lineal y fácil de implementar y programar. Casos particulares de esto son el hombro izquierdo y derecho.

$$A(x) = \begin{cases} \alpha \left(\frac{x-a}{c-a} \right) & \text{si } (a \leq x \leq c) \\ \alpha & \text{si } (c \leq x \leq d) \\ \alpha \left(\frac{x-b}{c-b} \right) & \text{si } (d \leq x \leq b) \\ 0 & \text{en cualquier otro caso} \end{cases}$$

Gauss

$$A(x) = e^{-\frac{(x-c)^2}{2\sigma^2}}$$

Campana

$$A(x) = \frac{1}{\left(1 + \left| \frac{x-c}{a} \right|^{2b} \right)}$$

Sigmoidal

$$A(x) = \frac{1}{1 + e^{-(x-m)\sigma}}$$

Defusificación

A partir de los resultados de las reglas, muchas veces se necesita obtener un valor nítido. Para hacer esto se utilizan diversos métodos de defusificación. Los más comunes son los siguientes[16]:

Método del Centro de Área

Se obtiene el centro del área de las funciones de salida.

$$z_0 = \frac{\int_a^b zC(z)dz}{\int_a^b C(z)dz} = \frac{\sum_{j=1}^n z_j C(z_j)}{\sum_{j=1}^n C(z_j)}$$

Método del centro de área con altura

Funciona igual que el de centro de área pero el área que evalúa esta restringida por la altura. Solo toma el área que tiene un valor de pertenencia mayor al escogido.

Método del Máximo

Toma cualquier valor dentro del segmento máximo. Este método da un rango

Método del principio del máximo

Toma el primer valor del máximo

Método de la mitad del máximo

Toma el centro del máximo.

Sistemas Tipo Mamdani[16]

Mamdani el primer modelo usado para realizar control con lógica difusa.

Las reglas que utiliza son del siguiente tipo:

"If x is A_i then y is b_i ," $i=1, \dots, n$ en donde $x=(x_1, x_2, \dots, x_k)$

Estas se combinan como

$$R(x,y) = \bigvee_{i=1}^n (A_i(x) \wedge B_i(y))$$

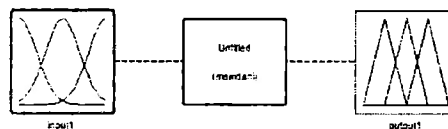


Figura 3.4m Sistema tipo Mamdani

Una característica distintiva es que tiene funciones de membresía tanto a la entrada como a la salida.

La salida de este sistema se realiza mediante métodos de defusificación. Generalmente se utiliza el método de centro del área

Sistemas Tipo Takagi-Sugeno-Kang[16]

Este tipo de sistemas fue introducido por Takagi y Sugeno (1985) y Sugeno y Kang(1988)

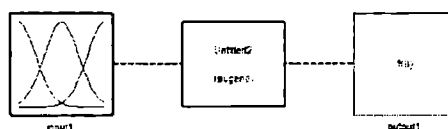


Figura 3.4n Sistema tipo Mamdani

Las reglas se dan de la siguiente manera:

Ri: If xi is Ai then fi(x), i=1,2,...,n

Estos se combinan para general un valor real de la siguiente forma:

$$R(x) = \frac{A_1(x)f_1 + A_2(x)f_2 + \dots + A_n(x)f_n}{A_1(x) + A_2(x) + \dots + A_n(x)}$$

Dado que este modelo produce un valor nítido, no es necesario defusificar.

Otros modelos que se pueden utilizar son los de Tsukamoto y el de Larsen [16]

4. CONCEPCIÓN DEL PROYECTO

El sistema mecatrónico propuesto para lograr los objetivos planteados se visualiza de la siguiente forma:

Un sistema mecatrónico integral compuesto por tres grandes etapas: Mecánica, de Potencia y de Control, cada una de las cuales se detalla más adelante.

La figura 4.1a muestra una imagen de los fundamentos del sistema SCARA planteado y de sus partes básicas.

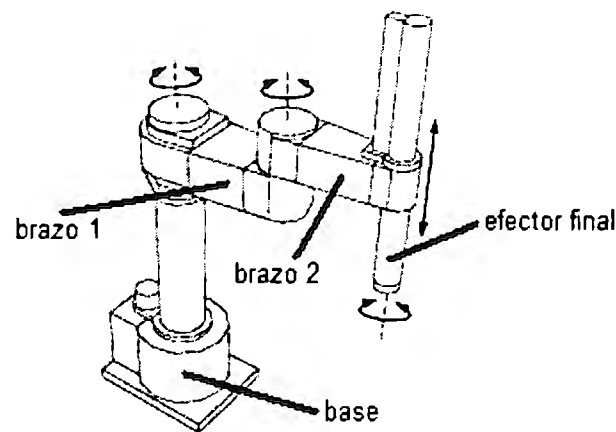


Figura 4.1a. Imagen de la concepción de un sistema SCARA genérico. Partes básicas. [4]

Ya que el presente trabajo contiene información tanto para el Concurso de Mini-robótica de la IEEE como para el juego de Gato, la información se denotará como "CONCURSO IEEE" y "GATO" para su identificación. Cuando la información no sea catalogada, significará que es aplicable a ambos rubros.

Para lograr el movimiento del sistema y la sujeción de las piezas durante el CONCURSO IEEE (ver Anexo I) se construyen dos brazos y un efector final diseñados y

manufacturados por el equipo y dentro de las instalaciones del Tecnológico de Monterrey, Campus Ciudad de México.

Estos dos brazos adquieren movimiento gracias a los actuadores servomotores (un servomotor por cada brazo), los cuales se controlan por señales de control de pulso positivo (PPC).

Estas señales son producidas por el Controlador Serial de Servos (SSC), el cual las genera a partir de la información recibida en forma serial.

La posición de los servomotores es definida por medio de algoritmos computacionales calculados en una computadora, la cual envía la posición deseada del servomotor codificada en un número binario. Esta señal binaria se envía a través del puerto Serial hacia el SSC para ser transformada en PPC y controlar así el servomotor.

La definición de la posición de los servos depende de la ubicación de las piezas en el tablero del concurso IEEE, la cual se conoce gracias al sistema de visión compuesto por una WebCam colocada en la parte superior del sistema y capturando imágenes del tablero y las piezas. El proceso de ubicación de las piezas lleva consigo un procesamiento de las imágenes obtenidas.

Salvo la computadora y la WebCam, todos los componentes restantes serán alimentados por medio de una fuente de corriente directa.

El diagrama de bloques de una fuente de corriente directa presenta las siguientes etapas:

- **Etapas de transformación de la señal.** Por medio de un transformador, se reduce la señal de entrada de típicamente 110 VAC o 220 VAC a un valor aproximado al nivel que se desea obtener a la salida en DC. En este caso, se tienen etapas de transformación a 30, 12 y 5 VAC.
- **Etapas de Rectificación.** Por medio de componentes electrónicos (usualmente un puente de diodos), se corta la parte negativa de la señal, permaneciendo únicamente la sección positiva.
- **Etapas de Filtrado.** Por medio de capacitores se reducen las cimas de la señal. Esto sucede cargando los capacitores al inicio del ciclo, y justo después de que empiezan a descargarse, se cargan de nuevo con el inicio del nuevo ciclo.
- **Etapas de Regulación.** Se elimina el rizo remanente con reguladores electrónicos, los cuales cortan los pequeños rizados y arrojan a la salida una señal continua.

5. MECÁNICA

5.1 MATERIALES

En un inicio, el material seleccionado para los brazos del robot fue el Nylamid SL®. Esta decisión se tomó debido a la enorme relación resistencia-peso de este polímero en comparación con metales tradicionales. Igualmente, se decidió utilizar dicho material con el fin de experimentar con plásticos, que normalmente no son tomados en cuenta para la fabricación de las partes principales de maquinaria. Fueron adquiridas dos barras de cara cuadrada de 1.25in de lado por 60cm de largo.

A pesar de la gran diferencia entre los precios por Kilogramo del Nylamid SL®, al comparar los precios de una barra cuadrada de las mismas dimensiones (1.25 in de lado y 60cm de largo) tenemos que:

Material	Precio (usd/barra)
Aluminio	8.31
Acero 1045	8.60
Nylamid SL®	14.55

Tabla 5.1a. Comparativo entre los precios de distintos materiales de una barra de dimensiones 1.25 in de lado por 60cm de largo

Como se observa en la tabla 5.1a, aún cuando el precio de las barras adquiridas de Nylamid SL® es mayor, se considera que los beneficios que ofrece el Nylamid SL® como mejor maquinabilidad y autolubricación son difíciles de medir y comparar contra los otros materiales y ofrecerían un factor más grande de Costo/Beneficio.

Por ello y por el deseo del equipo de experimentar con materiales modernos, se eligió el Nylamid SL® como material estructural de los brazos del sistema.

Sin embargo, luego de varias sesiones de trabajo con los brazos fabricados con Nylamid SL®, el material presentó fallas severas, las cuales no pueden ser toleradas en el presente proyecto, aun cuando el novedoso polímero cuente con otras grandes características. Dichas fallas se presentan en la sección 8.RESULTADOS.

Por lo anterior, el material elegido para la construcción de los brazos pasó de ser Nylamid SL® a Aluminio. A pesar del cambio de material, el Nylamid SL® no se desechó por completo, ya que se aprovechó su característica de autolubricación y se fabricaron bases de este material para ser utilizadas en todas aquellas áreas en las que se presentaría contacto entre dos o más piezas.

En cuanto a la caja base del robot, el material utilizado para ello lámina de aluminio de 1.4mm de espesor; se seleccionó este material por su bajo peso y por sus características mecánicas que hicieron su manufactura relativamente sencilla y el producto final lo suficientemente rígido y estructuralmente confiable.

5.2 DISEÑO MECÁNICO

Para el diseño mecánico, se utiliza el software de Pro Engineer® para la estructura del robot y el software postprocesador Mastercam® para generar el código G-M (geometry-miscellaneous) necesario para maquinar las piezas en la Fresadora CNC.

Para fines del presente proyecto, la información proporcionada por los 2 paquetes computacionales antes mencionados fue suficiente para llevar a cabo todas las tareas necesarias. Cálculos fuera de estos dos paquetes computacionales no fueron necesarios.

La funcionalidad del sistema tiene una prioridad mayor a la estética del mismo. Por lo tanto, algunos sistemas, mecanismos o adaptaciones pueden sacrificar la estética del sistema para cumplir con la funcionalidad del mismo.

En la sección 8. RESULTADOS se detalla el proceso de diseño mecánico.

5.3 CARACTERÍSTICAS MECÁNICAS DE LOS SERVOMOTORES

Los servomotores utilizados para el movimiento de los brazos son originalmente destinados a vehículos de radiocontrol. El primer eje de rotación, al ser el que debe de vencer el mayor momento de inercia, es controlado por un servo marca Hobbico Command modelo CS-80, de tipo "Giant Scale". Este actuador es capaz de dar un par de 24.7Kg-cm, que comparado con los servos estándar, que dan un par de alrededor de 3kg-

cm, es muy poderoso. El equipo contaba con este servo, que era utilizado para dar movimiento rotacional vertical a un robot de tipo articulado, donde el par requerido era comparativamente muy superior al que requiere el robot actual. Se decidió no cambiar de actuador por cuestiones económicas y porque un motor tan poderoso puede dar al brazo velocidades mayores a las que se puedan requerir.

El segundo servo es de menor poder, dado que únicamente tiene que vencer el momento de inercia del segundo brazo y el efector final. Dicho servo es marca Futaba modelo S3305. Se seleccionó este modelo por ser de tamaño estándar y alto par. El tamaño le permite estar colocado casi completamente dentro del primer brazo, protuberando únicamente la parte inferior del actuador (1.2 cm). El par que es capaz de entregar este servo es de 8.9kg-cm, que es mucho mayor al necesario para mover el brazo, pero que nuevamente puede ser útil para dar altas velocidades al segundo segmento del robot.

Principio de Funcionamiento de los Servos

Los servomotores de radio-control reciben una señal que consiste en pulsos positivos, con ciclo de trabajo variable. El ancho de estos pulsos (t_{on}) varía entre .3 y 2.3ms, que representa un rango de 200°. Se necesitan enviar 60 pulsos por segundo para lograr mantener la posición del servo en la posición deseada.

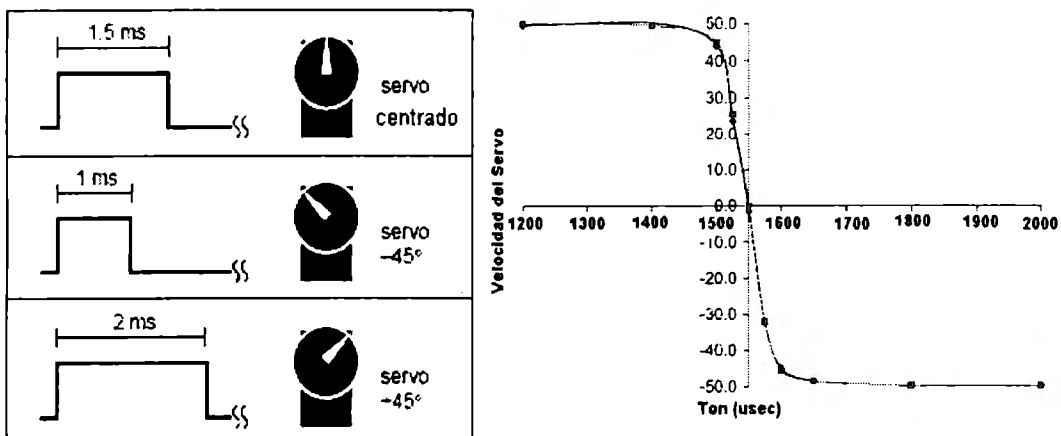


Figura 5.3a. Diagrama de posición del servo según el ancho de pulso y gráfica de respuesta de velocidad contra ancho de pulso [18],[19]

Los servomotores contienen un sistema interno de retroalimentación el cual compara la posición teórica a la que debe de ir (mediante los pulsos recibidos) con la posición real en la que se encuentra mediante la posición de un potenciómetro interno. Gracias a esta funcionalidad, podemos confiar en que el servomotor realmente llegará a la posición a la que se mande, salvo en excepciones en donde exista algún impedimento mecánico (en dichos casos se debe de tener sumo cuidado).

La figura 5.3b muestra la imagen de un servomotor típico

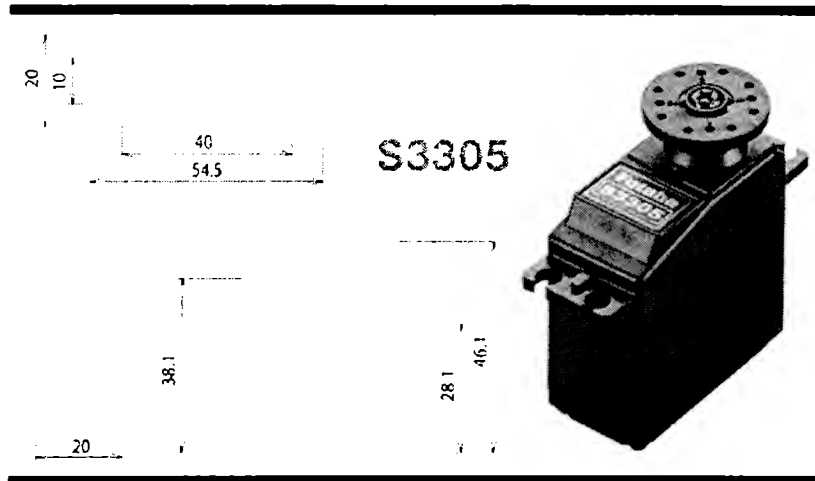


Figura 5.3b. Imagen de un servomotor típico [14]

6. ETAPA DE CONTROL

6.1 DESCRIPCIÓN GENERAL

El sistema general de control se encuentra en la computadora, esta recibe y procesa toda la información, toma las decisiones de movimiento y envía las instrucciones necesarias a los servos.

La paquetería computacional utilizada en la etapa de control se divide en dos grandes secciones:

- Calibración de los servomotores. Se utiliza el software LabVIEW®
 - o Se aprovechan las características gráficas de este sistema para la creación de interfases útiles en la calibración de forma más sencilla y rápida que en otros ambientes no gráficos.
- Control del sistema. Se utiliza el software MATLAB®
 - o Dentro de esta sección se incluye el procesamiento de imágenes, algoritmos de lógica difusa para el movimiento de los servomotores y la toma de decisiones sobre el orden de las piezas a recolectar.

6.2 DIAGRAMA DE BLOQUES

El diagrama de bloques del control del sistema consta de 3 grandes etapas comunicadas a Lazo Abierto como se muestra a continuación en la Figura 6.2a:

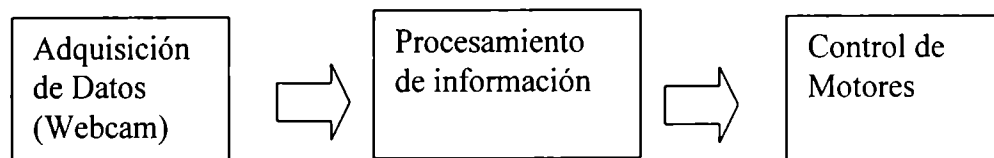


Figura 6.2a. Diagrama de bloques del control de los motores

Para realizar el control del movimiento de los brazos del sistema, se propusieron 4 posibilidades iniciales:

- Controlar la posición en formato X,Y mediante Control P
- Controlar el ángulo de cada brazo mediante Control P
- Controlar la posición en formato X,Y mediante Control Difuso
- Controlar el ángulo de cada brazo mediante Control Difuso

Para el Control Difuso, se utilizan reglas que relacionan la distancia entre el punto inicial y el destino del gripper con la velocidad a la que debe moverse el sistema. La figura 6.2b muestra dichas relaciones:

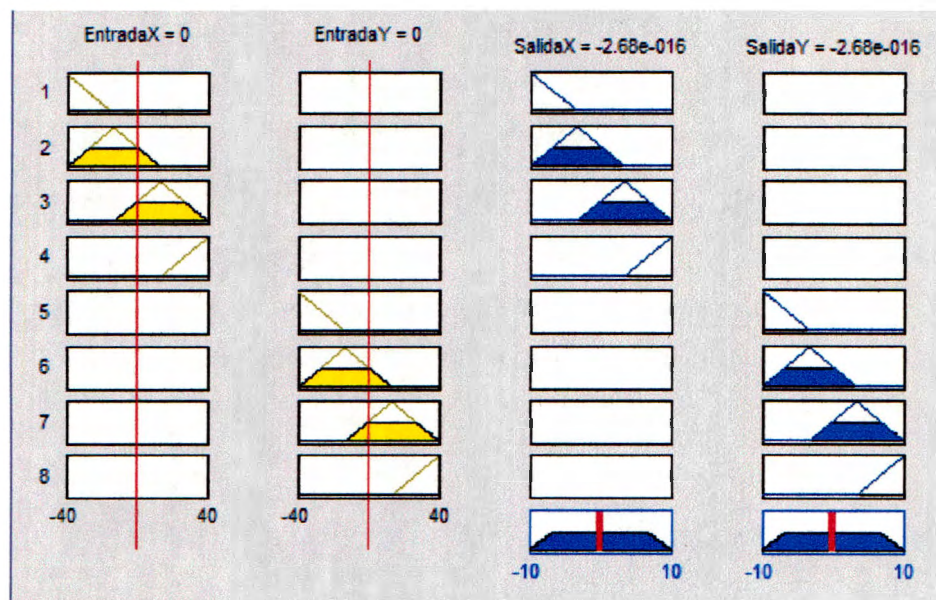


Figura 6.2b. Reglas del Control Difuso. A mayor distancia entre el punto inicial y el destino, el sistema se mueve a mayor velocidad. De forma análoga, la velocidad será menor conforme la distancia vaya disminuyendo.

Aunque el esfuerzo computacional requerido por el Control Difuso es considerablemente mayor que el requerido por el Control P, la decisión final sobre la alternativa a usar dependerá también del desempeño del sistema.

La opción seleccionada toma en cuenta las dos variables anteriores y se muestra en la sección 8. RESULTADOS

La elección del tipo de control utilizado para el movimiento de los brazos del sistema es independiente de la concepción del proyecto global. Es decir, que el control utilizado para el movimiento de los brazos presentará retroalimentación, sin que esto signifique que la posición final (salida) del sistema esté retroalimentada, tal y como se explica en el siguiente apartado.

6.3 ENTRADA Y SALIDA DE DATOS

La precisión requerida en la posición final de gripper no requiere la implementación de Lazo Cerrado. Esto significa que se pueden tolerar pequeñas imprecisiones sin afectar el resultado final del sistema. Adicionalmente, la implementación del Lazo Cerrado requeriría de mayores inversiones de recursos por parte del equipo. Por ello, La posición final del sistema está concebida como Lazo Abierto.

Debido a ello, la única entrada de datos es la imagen capturada por la WebCam conectada a través de un puerto USB a la computadora. Por ello, no es necesaria una tarjeta de adquisición de datos externa.

En cuanto a la salida de datos de la computadora hacia el sistema, se utiliza el puerto serial. Sin embargo, es necesario un circuito convertidor de la señal RS232 (formato de comunicación del puerto Serial de la computadora) hacia un formato TTL capaz de ser manipulado por el dispositivo elegido para el control de los tres servomotores del sistema (ver Figura 6.3a).

Dicho dispositivo es un Controlador Serial de Servos (SSC) con el que el equipo contaba y había trabajado previamente. Al comenzar el presente proyecto, ya se contaba con conocimiento y práctica sobre el funcionamiento del dispositivo.

Por lo anterior, en lo que respecta a la salida de datos desde la computadora, tampoco se considera necesaria la compra de una tarjeta de adquisición de datos externa.

Las características del SSC utilizado se discute en la sección 6.6 Control de Motores

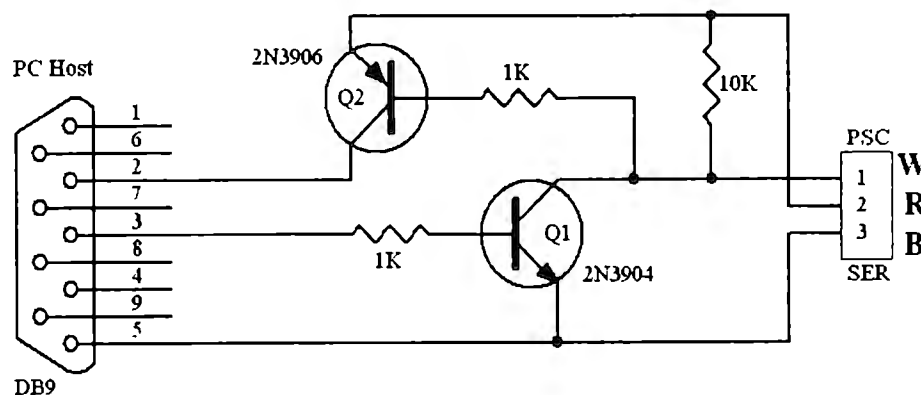


Figura 6.3a. Circuito convertidor de señal RS232 - TTL

Con el circuito anterior, se logra la comunicación entre MATLAB® y el controlador de servomotores. Realizando la conectividad entre MATLAB® y el controlador de servos

mediante el puerto serial no se presentaron problemas y se trabajó de forma adecuada, incluso al utilizar un convertidor USB-Serial debido a la ausencia de puerto serial en las computadoras utilizadas.

6.4 PROCESAMIENTO DE IMAGENES

El procesamiento de imágenes es realizado mediante una WebCam de la marca Creative®. Se elige dicha cámara por su capacidad de enfoque de imagen mecánico (mediante una perilla), lo cual facilita la calibración del sistema dependiendo de las variables del entorno como iluminación, distancia de la cámara al su objetivo, brillo del entorno, entre otros.

El diagrama de bloques sobre el proceso utilizado por el momento para la identificación de objetos se muestra en la siguiente Figura 6.4a:

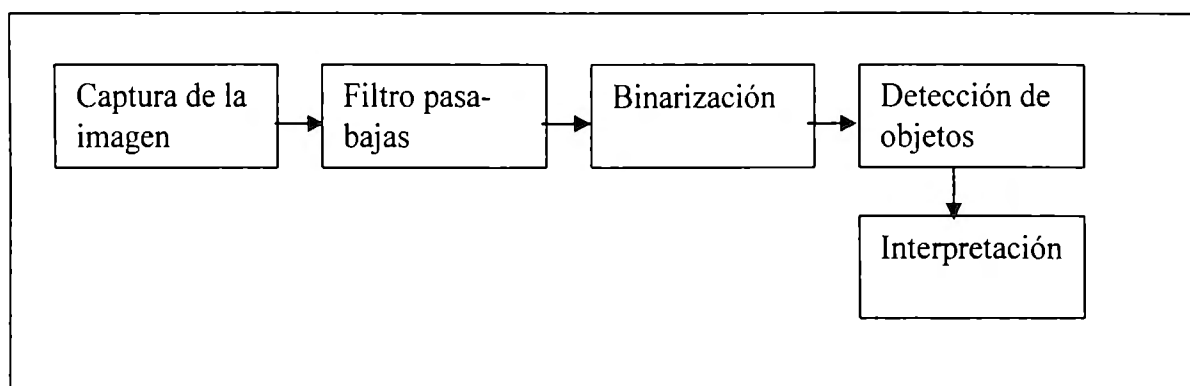


Figura 6.4a. Diagrama de bloques del procesamiento de imágenes

A continuación explicamos cada uno de los bloques de figura 6.4a. Los resultados obtenidos se presentan en la sección 8.

Captura de la imagen: Se crea un objeto de entrada de video, se captura un cuadro con la figura que es pretende analizar con la WebCam y se convierte en un objeto de imagen de Matlab®. En la modalidad de JUEGO DE GATO, la imagen muestra el tablero y los símbolos que han sido dibujados en él. En la modalidad de CONCURSO IEEE, la imagen muestra el tablero y las piezas que aún permanecen sobre él. En la sección de 8. RESULTADOS se muestra gráficamente la secuencia en el procesamiento de las imágenes.

Filtrado pasa-bajas: El objetivo de esta etapa es eliminar el ruido de la imagen (elementos como brillos, sombras, etc pueden causar ruido en la imagen). Para hacer esto, se desenfoca la imagen y se eliminan las secciones de alta frecuencia, que son las que se consideran como ruido.

Para lograr lo anterior, se utiliza un filtro tipo disco con un radio de 3 píxeles que combina la información del píxel actual con la de los demás píxeles dentro del círculo descrito por el filtro, resultando una imagen matizada con un color ponderado homogéneo. Un filtro tipo disco no es representado por una ecuación matemática, sino por el radio de trabajo.

Binarización: La imagen seleccionada se convierte a una imagen en escala de grises. Es decir, que todos los objetos dentro de la imagen se convierten a sólo uno de los siguientes: blanco, gris o negro. Para hacer esto, se toma el valor del color negro (0) y se le suma un valor dinámico "thr1". De forma análoga, se toma el color blanco (255) y se le resta otro valor dinámico "thr2". Una vez hecho esto, todos los puntos dentro de la imagen que estén dentro del rango de $(0+thr1)$ se convertirán a "negros", los que estén dentro del rango $(255-thr2)$ se convertirán a "blancos". Todos los puntos restantes se convertirán a "gris" con un valor en escala de grises igual al promedio de los puntos que han sido convertidos a "blancos" y "negros".

Cabe señalar que el dinamismo de los valores "thr1" y "thr2" varía en función del brillo de la imagen tomada, la cual también está en función de la luz presente en la habitación. Con lo anterior, a mayor luz presente en la habitación, los colores de la imagen tenderán a ser "más blancos", por lo "thr1" será mayor que "thr2" y viceversa en el caso de poca luz en la habitación. Esto procura que al momento de tomar la imagen, los puntos más parecidos al color blanco sean detectados como blancos y de la misma forma con los más parecidos al color gris y al negro.

Detección de objetos: Una vez binarizada la imagen, ésta se recorre de izquierda a derecha en todas las líneas horizontales de la imagen. El programa detecta todos los cambios en el valor de un píxel, resaltando el primer píxel que presenta el nuevo valor. Así, cada vez que la imagen presenta un cambio de color entre negro, gris y blanco, se conoce que nos encontramos en un borde de algún objeto.

Interpretación: Una vez que se han detectado los bordes de los objetos, se guarda en memoria la posición y el color de los objetos (el color es importante en la modalidad de CONCURSO IEEE). Ahora el sistema está preparado para enviar a los brazos a la posición requerida para alcanzar el punto deseado para recoger alguna pieza (IEEE) o

para dibujar un símbolo (GATO). Por último, la información requerida para ser enviada al controlador de servos es calculada y enviada (ver sección 6.6 CONTROL DE MOTORES).

6.5 PROCESAMIENTO DE INFORMACIÓN

Todo el procesamiento de información y toma de decisiones será realizado por una laptop (Pentium 4 HT a 3.06Ghz ,512MB en Ram). El software utilizado para el desarrollo del software es Matlab® v 14, dado que es la herramienta sobre la cual se tienen los conocimientos necesarios para desarrollar los sistemas involucrados en el proyecto.

Una vez que la imagen ha sido procesada y se conocen las posiciones de las piezas sobre el tablero, se requiere mover a los servos a posiciones determinadas para recolectarlas.

Se utiliza la lógica difusa para modular la velocidad de los brazos. Gracias al programa generado, se puede optimizar la velocidad de éstos. Así, la velocidad será mayor mientras más lejano esté el efecto final de la pieza objetivo; y se irá reduciendo conforme se vaya acercando a ésta. De este modo, se reducen los efectos de la inercia mecánica del sistema. Los detalles sobre la implementación de la lógica difusa se muestran en la sección 8.2.

Para la modalidad de CONCURSO IEEE, la elección del orden de las piezas a tomar en cada momento se realiza por medio de las siguientes leyes euclidianas:

- Una vez que se toma la imagen del tablero y con las piezas sobre él y se ha procesado la imagen, se guarda en memoria el color y posición de las 10 piezas sobre el tablero.
- La primera pieza a tomar es siempre la más cercana a la posición de "HOME", la cual es el centro geométrico del tablero. Esa pieza se deposita en el contenedor que esté más cercano al promedio de las distancias entre ese contenedor y las demás piezas del mismo color que la primera pieza tomada. En ese momento, los contenedores se clasifican como "Blanco" y "Negro" dependiendo del color de la primera pieza y en dónde ha sido colocada ésta.
- La pieza siguiente a tomar será la más cercana al gripper en ese momento. Y así subsecuentemente con todas las piezas del tablero.

Para la modalidad de GATO, el sistema cumple con los siguientes puntos:

- Mediante el cambio de una bandera en el software principal del sistema, se determina quién inicia el juego.

- Se toma una imagen del tablero antes de cada turno para así detectar el cambio con respecto al último tiro.
- Luego de cada tiro del jugador, éste debe presionar "ENTER" en la computadora para continuar con el turno del robot.
- El jugador debe siempre esperar a la instrucción auditiva de "TU TURNO" por parte de la computadora para realizar su turno. De lo contrario, provocará interferencia si obstruye la visibilidad de la Webcam.
- El robot toma las decisiones sobre en dónde realizar el tiro mediante las siguientes reglas:
 - o Se asigna una ponderación a cada casilla vacía (ver Figura 6.5a) dependiendo de la ocupación de las casillas con las que forma una líneas de 3 casillas
 - Una casilla ocupada por el jugador tiene un valor de "-1"
 - Una casilla ocupada por el robot tienen un valor de "1".
 - Una línea con la que potencialmente el robot puede ganar el juego, tiene un valor "a favor" entre 0 y 10 dependiendo de la ocupación de las casillas de dicha línea (una línea con 1 casilla ocupada por el robot y las otras 2 casillas vacías tiene un valor mayor a una línea vacía pero menor a una línea con 2 casillas ocupadas por el robot y la tercera casilla vacía).
 - De forma análoga, una línea con la que potencialmente el jugador puede ganar el juego, tiene un valor "en contra" que también va de 0 a 10 de pendiendo de la ocupación de las casillas (una línea tiene un valor "en contra" mayor mientras más peligro exista de que gane el jugador).

Mediante las reglas anteriores, el robot decide en qué casilla tirar en cada uno de sus turnos.

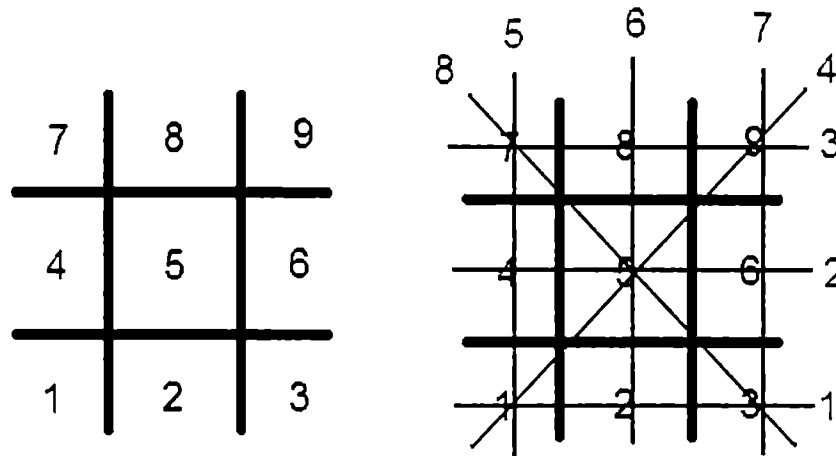


Figura 6.5a. Simbología de las casillas del tablero de GATO (izquierda). Simbología de líneas en el tablero de GATO (derecha).

La estructura jerárquica de los archivos utilizados tanto en la modalidad CONCURSO IEEE como en la modalidad de GATO se muestran en la sección 8.3 *Programas en Matlab®*.

6.6 CONTROL DE MOTORES

Todos los motores que utiliza nuestro sistema son servomotores comerciales.

Las partes principales de operación de un servomotor son:

- **Externas**
 - **Cables.** 2 cables de alimentación y 1 cable de control.
 - **Eje.** Transfiere la potencia del motor hacia el engrane
 - **Cuerpo.** Protege las partes internas.
- **Internas**
 - **Motor de DC.** Trabaja típicamente con voltajes DC entre 4.8 y 6 V
 - **Tren de engranaje.** Transfiere potencia y reduce la velocidad de giro.
 - **Circuito de Control.** Regula las señales de entrada tipo PPC para lograr y mantener la posición del eje del motor definida por la señal de control.
 - **Potenciómetro de retroalimentación.** Compara la señal de control con la posición real del servo.

Los servomotores utilizados para controlar el movimiento de los brazos se muestran en la tabla 6.6a:

Servo	Dimensiones físicas (mm)	Masa (g)	A 4.8V		a 6V	
			Velocidad (seg/60°)	Par (kg-cm)	Velocidad (seg/60°)	Par (kg-cm)
Futaba S3005	46.1 x 20 x 45	45.4	0.25	2.51	0.2	3
Hobbico Command CS-80	66 x 30 x 58	152	0.19	19.8	0.14	24.7

Tabla 6.6a. Especificaciones de los servomotores [10], [14]

Para comunicar la computadora con estos servomotores se utiliza el Controlador de Servos Serial "Mini SSC II" de la marca Parallax. Este dispositivo tiene la función principal de recibir las señales en formato Serial de la computadora y transformarlas a formato PPC requerido por los servomotores.

Sus características principales son las siguientes:

- Puede controlar hasta 16 servos y 32 si se conecta en red con otro dispositivo similar utilizando sólo una línea de entrada/salida.
- Presenta control de la velocidad de los servomotores mediante rampas. Contiene la posibilidad de elegir 63 velocidades distintas.
- Controla todos los pulsos de los servos. Por lo que no es necesario enviar señales para mantener la posición de los servos.
- Se reporta la posición de cualquier servo en cualquier tiempo, ya sea estacionario o dinámico.
- Se requieren seis bytes en formato Serial enviados desde la computadora:
 - El primer, segundo y tercer bytes son "ISC" y sirven para la identificación entre la computadora y el Mini SSC2
 - El cuarto byte se usa para establecer el servomotor a controlar (de 0 a 31).
 - El quinto byte establece la rampa (velocidad deseada del servo).
 - El sexto byte establece la posición a la que se desea llegar el servomotor.

Con estas características, se requiere únicamente enviar la señal Serial de la posición y la velocidad deseada de cada servomotor y el dispositivo Mini SSC 2 enviará las señales

PPC para lograr la posición de cada servomotor con las características deseadas. Lo anterior nos da una resolución suficiente para las tareas que tenemos planeadas.

Los diagramas de las dimensiones y conexiones del dispositivo se muestran a continuación en las Ilustraciones 6.6a y 6.6b respectivamente.

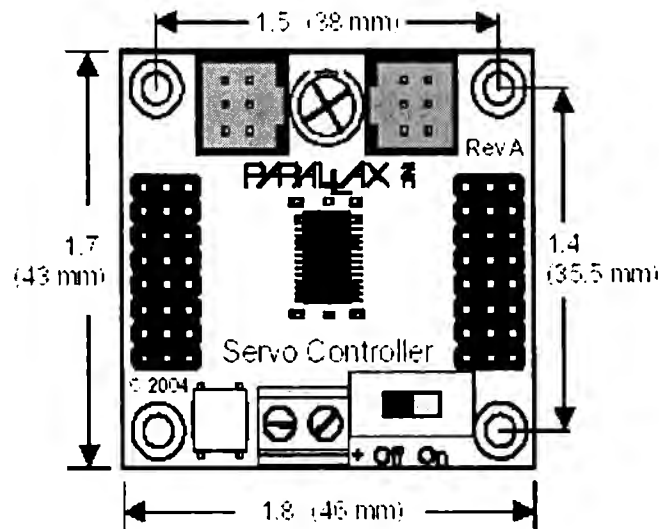


Figura 6.6a. Dimensiones del Controlador de Servos [24]

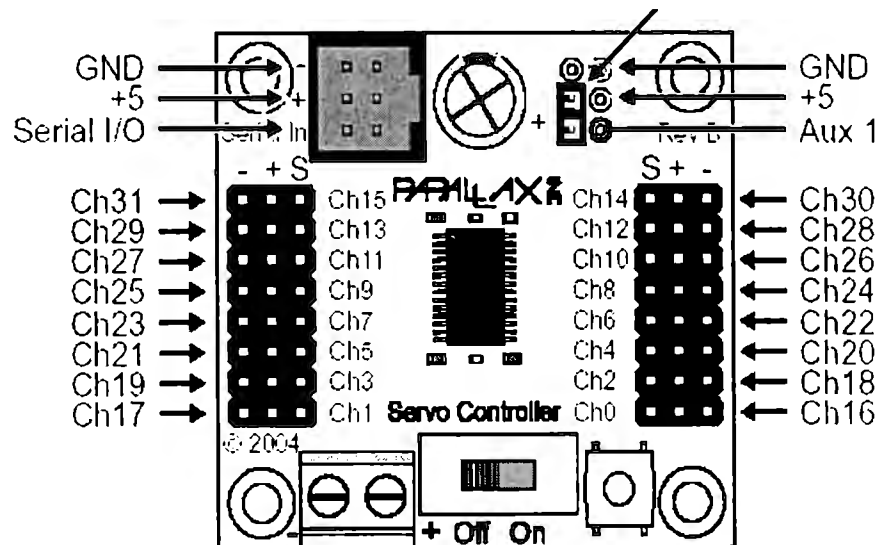


Figura 6.6b. Diagrama de conexiones del Controlador de Servos [24]

7. ETAPA DE POTENCIA

La fuente de energía utilizada para alimentar a los servomotores y su controlador es una diseñada originalmente para computadora.

Las computadoras tienen demandas de voltaje distintas en sus varios componentes, por ello, la fuente con la que contamos tiene salidas de 3v, 5v y 12v. El controlador de servos puede ser alimentado con un rango que va de 7v a 15v, por lo que la salida de 12v puede ser utilizada para él. El consumo de corriente del controlador es mínimo (del orden de 10mA), por lo que puede ser entregada sin problemas por la fuente [23].

Los servos pueden funcionar con voltajes de entre 5v y 6v, por lo que la salida de 5v de la fuente es ideal. El consumo de corriente del primer servo, que es el de mayor potencia, es de hasta 6A (cuando se requiere un par máximo de 24Kg-cm). En la configuración actual del robot no se requiere de tanto par, aunque por seguridad se asume que la mayor demanda de corriente es la anterior. El segundo servo tiene un consumo de corriente mucho menor, de hasta 3A. La fuente con la que se cuenta, que fue diseñada para computadoras de alto desempeño, es capaz de entregar hasta 30A en su salida de 5v, por lo que puede cubrir la demanda total de los servomotores sin problemas.

La computadora, que obtendrá las imágenes, procesará la información y dará las señales de control para los servomotores, es alimentada por su propia fuente de poder, por lo que no debe ser tomada en cuenta para el diseño del robot.

El diseño del sistema eléctrico es ideal puesto que la computadora está aislada de la fuente de alimentación de los servomotores por medio del controlador de los mismos. Al tener dicho controlador una entrada por puerto serial, una entrada de voltaje para su funcionamiento y una entrada independiente de voltaje para la alimentación de los servos, se asegura que la energía entregada por la fuente de poder del robot no será transmitida a la computadora, que es vulnerable a descargas de alta corriente.

8. RESULTADOS

Los paquetes de software que se han utilizado para el desarrollo de este proyecto son los siguientes:

Matlab® para todos los cálculos y software desarrollado

Pro Engineering 2000i ® para el diseño de las partes mecánicas

Master-Cam® para la generación del código de CNC para el maquinado de los brazos

8.1 DISEÑO MECÁNICO DEL SISTEMA

Ambos brazos fueron diseñados con el software Pro Engineer®, que permitió su visualización tridimensional y el cálculo del centro de masa y momento de inercia antes de maquinar el material.

En la figura 8.1a se muestra el diseño y dimensiones de ambos brazos, el servomotor acoplado al brazo # 1 y el ensamble # 1

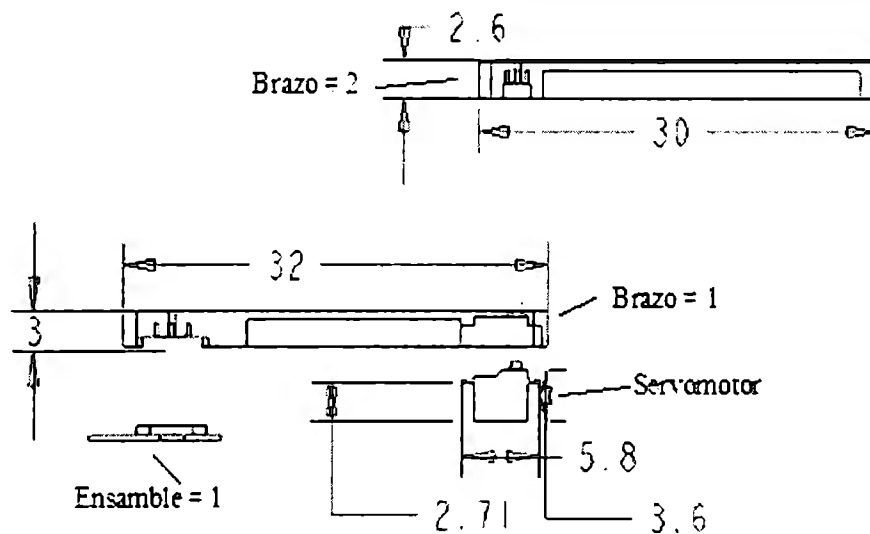


Figura 8.1a. Diseño de los brazos del sistema en ProE®. Las dimensiones están en cm.

En la figura anterior, el ensamble # 1 va unido a la caja del robot. El brazo # 1 desliza sobre el ensamble # 1 y también está unido al brazo # 2 por medio del servomotor mostrado. En la punta del brazo # va el efector final.

Una vez generado el diseño en ProE®, y con ayuda de Mastercam®, se generaron las instrucciones en código G-M para el maquinado de las piezas en la fresa CNC.

Posteriormente, se exportó el archivo del diseño en formato IGES (Initial Graphics Exchange Specification) para poder ser interpretado por el software Mastercam Mill®. Una vez que se importó el archivo en dicho software, se crearon las trayectorias para el maquinado de las piezas, especificando los diámetros de los cortadores, velocidades de corte, avance de las herramientas y profundidades de corte [5].

Con el postprocesador de Mastercam® se creó el programa de códigos G-M (geometry-miscellaneous), que se alimentó a la fresa de control numérico computarizado para finalmente maquinar los brazos. Tras este proceso, los brazos fueron lijados y estaban listos para ser ensamblados.

La base del robot consta de una caja fabricada por el equipo (ver Figura 8.1b). La lámina de aluminio se cortó con una guillotina de metal y se dobló con una dobladora de lámina y con una prensa de banco y un mazo de goma.

Por último, se taladraron varios hoyos para remachar la caja de forma que mantuviera su forma. La cara superior únicamente cubría la mitad de la caja, puesto que la otra mitad es una tapa que se levanta para dar acceso al interior de la base. Dos bisagras fueron colocadas para realizar dicha tarea.

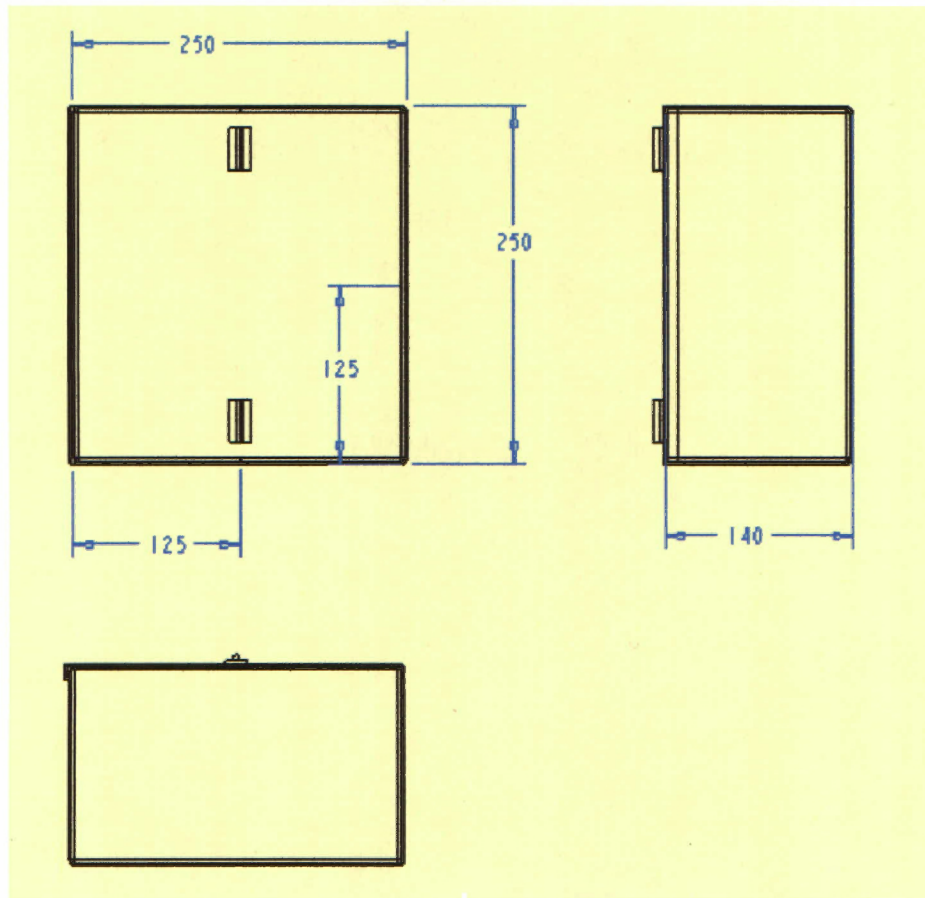


Figura 8.1b Dimensiones en cm de la base de aluminio. Diseño creado en ProE®

En las figuras 8.1c a la 8.1f se presenta el diseño realizado en Mastercam® y la simulación de maquinado tras la generación de las trayectorias de la herramienta:

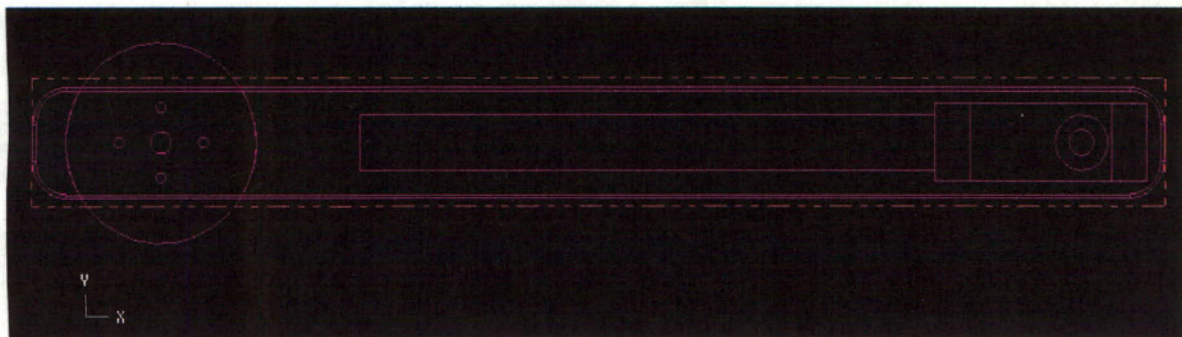


Figura 8.1c. Diseño en Mastercam®, vista superior del primer brazo.

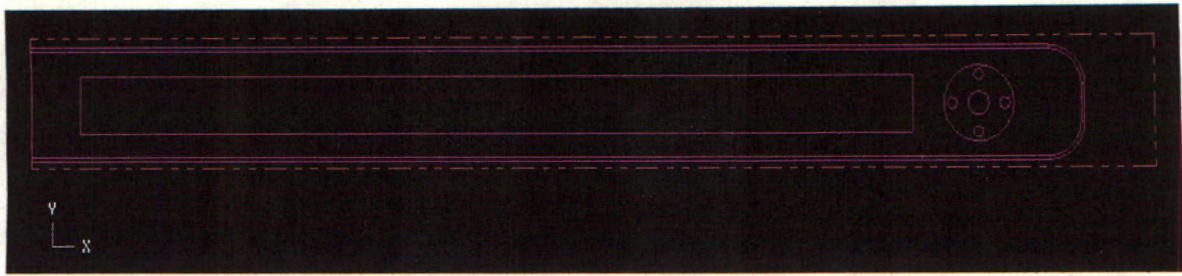


Figura 8.1d. Diseño en Mastercam®, vista superior del segundo brazo.

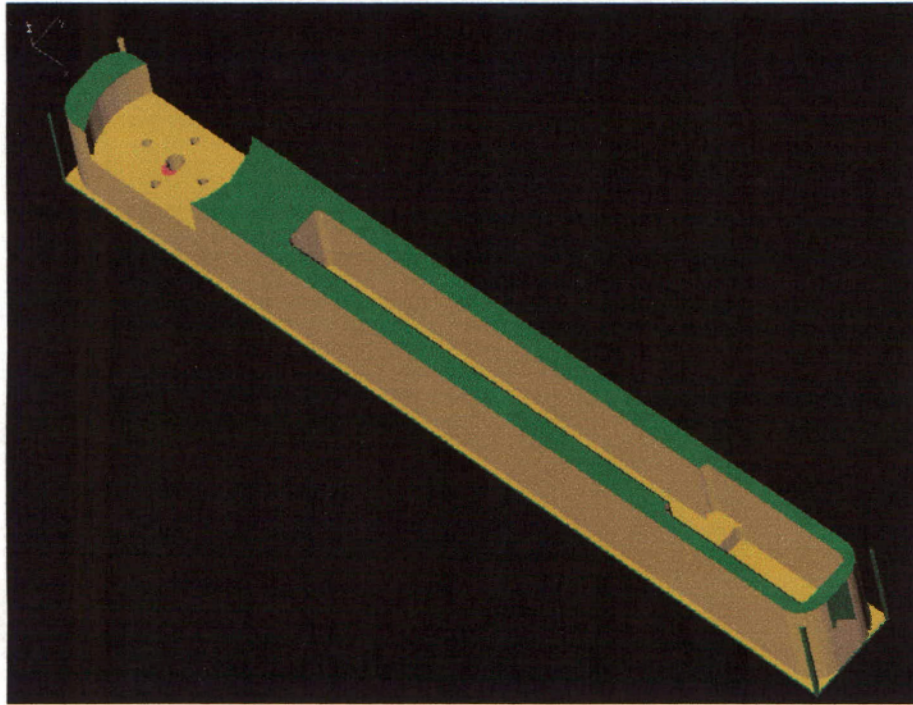


Figura 8.1e. Simulación en Mastercam®, vista isométrica del primer brazo.

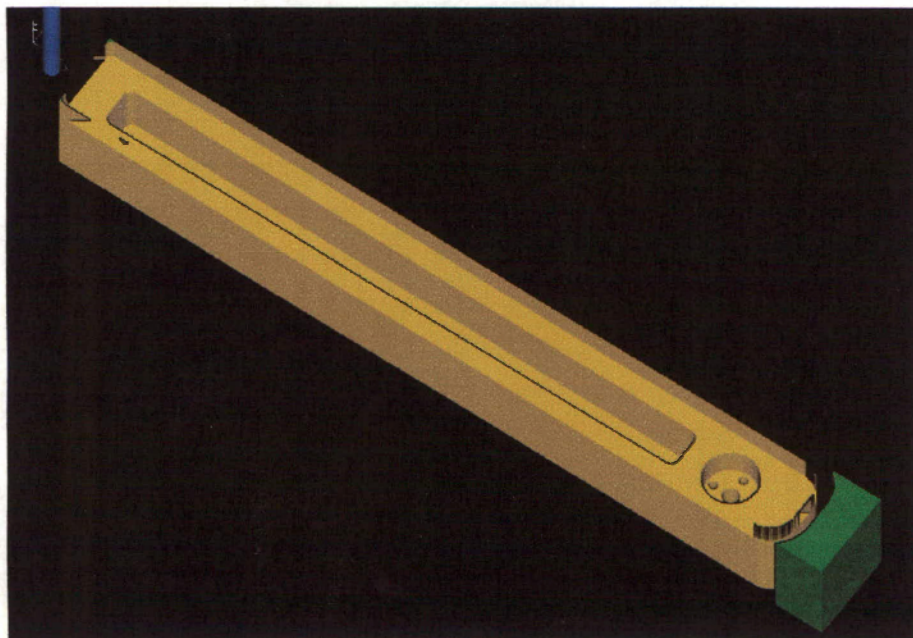


Figura 8.1f. Simulación en Mastercam®, vista isométrica del segundo brazo.

Gracias a la programación y obtención del código G-M generado, se logró el maquinado CNC de ambos brazos.

Las figuras 8.1g y 8.1h muestran la barra de Nylamid SL® antes y después del maquinado del brazo # 1.



Figura 8.1g. Barra de Nylamid SL® antes del maquinado



Figura 8.1h. Barra de Nylamid SL® después del maquinado. Se observa el brazo #1

FALLAS CON EL NYLAMID SL®

Luego de obtener la estructura mecánica completa y de realizar varias sesiones de trabajo, ambos brazos comenzaron a presentar deformaciones mecánicas considerables (ver imagen 8.1i y 8.1j). Los defectos eran tales, que el efector final se encontraba a una altura sobre el tablero de aproximadamente 11cm en lugar de los 15cm para los cuales se había diseñado el sistema. Este error de 27% no fue tolerable para las características del sistema y se decidió cambiar el material estructural de los brazos por Aluminio (ver imagen 8.1k).

La decisión de tomar el Aluminio como segunda opción se originó a partir de las necesidades de un material ligero y resistente. El material que mejor cumple con estas dos características (ver Tabla 3.2a) y estuvo al alcance del presupuesto del equipo (ver Tabla 5.1a) es el Aluminio (incluso el Aluminio tiene un precio menor al Nylamid SL®). Los cálculos sobre los parámetros necesarios en cuando a energía, y torque de los servomotores indicaron que el Aluminio era un material viable para construir los brazos del sistema. Una vez construidos, la se comprobó prácticamente que este material fue una buena elección. El proceso de manufactura fue exactamente el mismo que el descrito en la presente sección. El cambio de material de Nylamid SL® a Aluminio únicamente requirió el ajuste de los parámetros de corte adecuados en los códigos de Mastercam ®. Una investigación posterior sugirió que la deformación se debió al calor al cual se sometió a los brazos al permanecer dentro de un automóvil al aire libre y durante algunos días calurosos. Se tomaron algunas medidas correctivas del estado de los brazos mediante barras estructurales colocadas a los costados de los brazos con el fin de enderezarlos. El resultado fue una corrección considerable del defecto pero aún así sus condiciones no eran las adecuadas para trabajar con ellos.



Imagen 8.1i Brazos deformados fabricados con Nylamid SL®. Al colocarlos sobre una superficie plana, se observa una deformación en el extremo de la derecha tanto del brazo 1 (arriba) como del brazo 2 (abajo). La Imagen 8.1j muestra un aumento para observar la deformación.

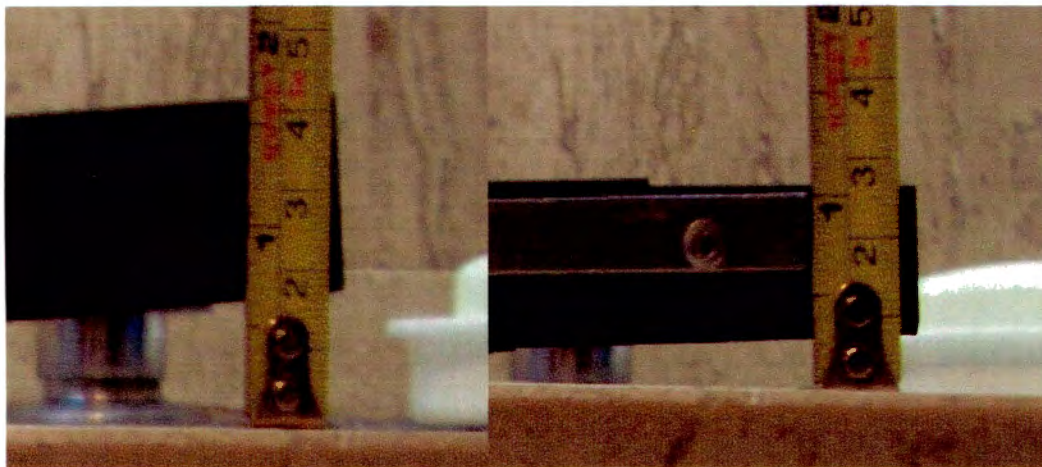


Imagen 8.1j. Deformación del brazo 1 (izquierda): Aproximadamente 1.8cm
Deformación del brazo 2 (derecha): Aproximadamente 0.8cm

Sin embargo, se decidió no desechar el Nylamid SL® por completo y aprovechar sus características de autolubricación mediante la fabricación de bases para ser colocadas en los extremos de los brazos en donde se presenta contacto con otra pieza del sistema (ver imagen 8.1i). Sin estas piezas, la fricción entre las piezas de aluminio provocaría esfuerzos elevados en los servomotores e incluso errores en la posición deseada del efector final.

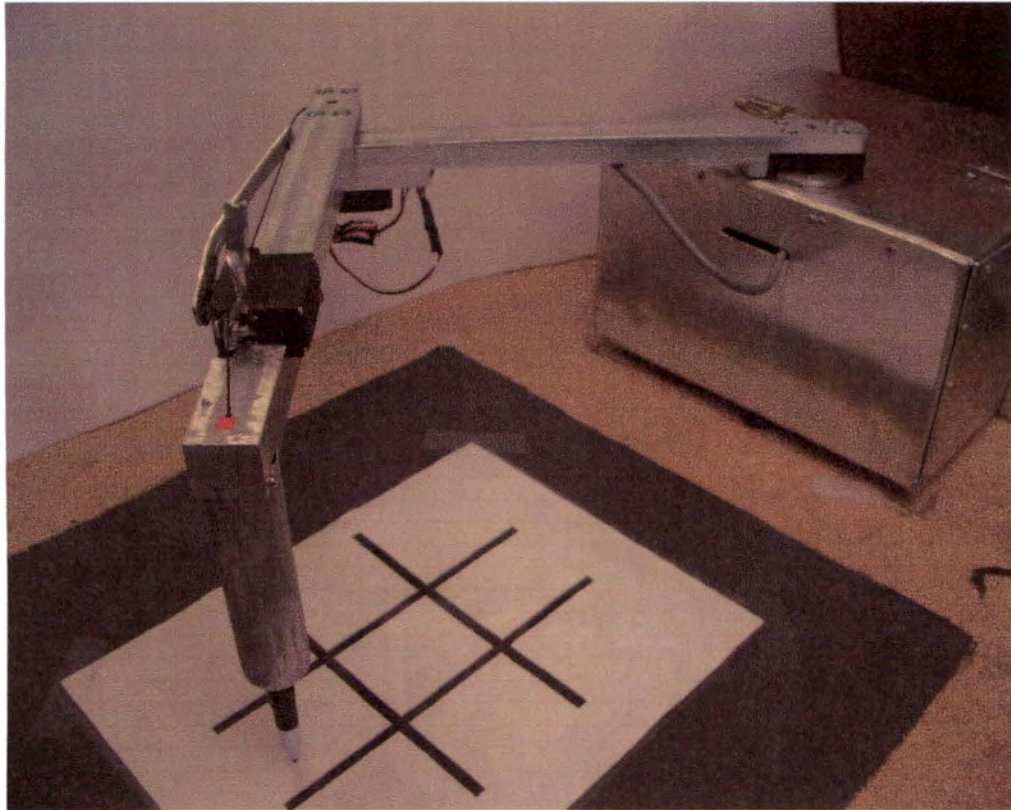


IMAGEN 8.1k Brazos de aluminio. Se observa el sistema en modalidad JUEGO DE GATO

En la imagen anterior se observa el sistema ensamblado correctamente para su funcionamiento. Los brazos ya se encuentran contruidos en Aluminio y las secciones de Nylamid SL® se observan en las articulaciones. Estas secciones reducen considerablemente la fricción entre los brazos, logrando así una disminución en la demanda mecánica de los servomotores, lo cual se traduce en menor esfuerzo físico y menor consumo de energía del sistema completo. Estas partes se observan en la figura 8.1l.

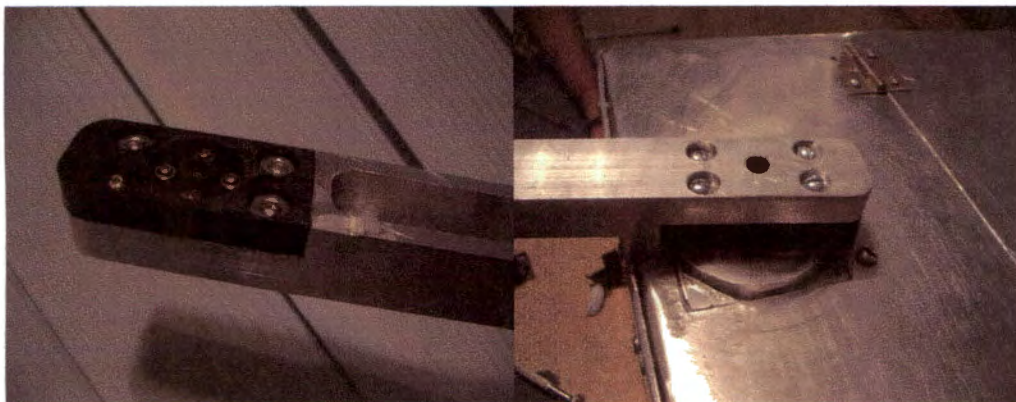


IMAGEN 8.1l Piezas de Nylamid SL® para reducir la fricción entre los componentes del sistema. A la izquierda se observa la pieza reductora de fricción empotrada en el brazo 2 para actuar en la articulación brazo 1 - brazo2. A la derecha se observa el brazo 1 con la pieza reductora de fricción empotrada en él; dicha pieza actúa en la articulación Base – brazo 1.

8.2 DISEÑO Y CONSTRUCCIÓN DEL GRIPPER

En cuanto al efector final (gripper) del sistema, su diseño se basó en una caña de pescar. Esto es, con un elemento sujetador que se mueve sólo en la dirección vertical para sujetar las piezas.

Para la modalidad de CONCURSO IEEE, el gripper consta de un imán permanente sujeto con un hilo que sube y baja gracias a la acción de un servomotor colocado en la parte superior del brazo 2, el cual tiene sujeto a su eje un brazo que al rotar, jala o libera el hilo que sujeta al imán permanente. La imagen 8.2a muestra una fotografía del gripper.

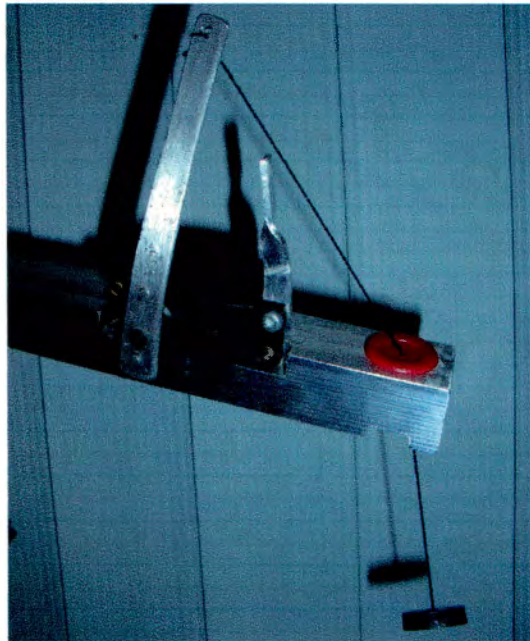


Figura 8.2a. Efector final del robot. Se observa el servo #3 montado sobre el brazo #2 con un pequeño brazo sujeto a su eje. Al girar dicho eje, el hilo con el imán atado sube y baja

Una vez que el imán ha sujetado la pieza (durante el CONCURSO IEEE), el imán sube hasta una posición en la que el imán queda sujeto al brazo #2 y un movimiento del sistema no provocará oscilaciones del imán causando una caída potencial de la pieza que está sujetando.

Cuando se desea liberar la pieza del gripper, se hace subir el imán un poco más, en donde entra a una cavidad circular cuyo diámetro es lo suficientemente grande como para dejar pasar al imán pero no a la pieza. Así, la pieza cae (ver Figura 8.2b)

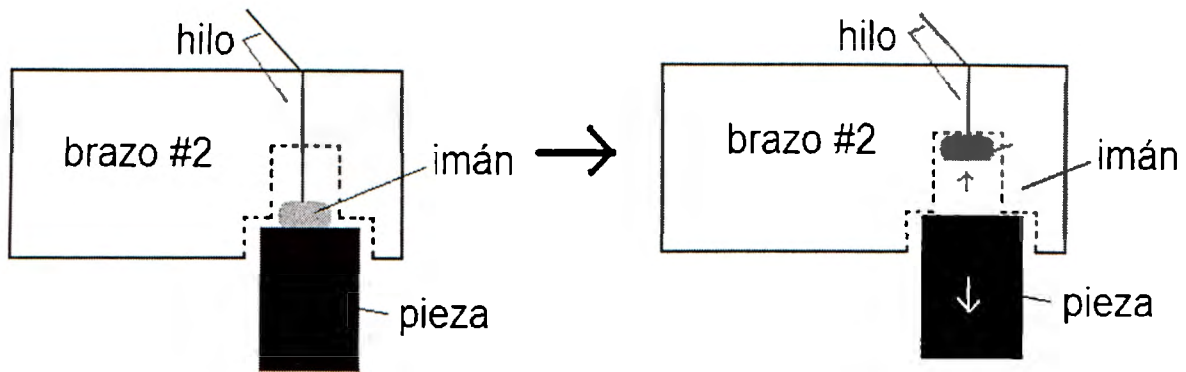


Figura 8.2b. Pieza sujeta al imán del gripper de modo que el imán no oscila con los movimientos del sistema (izquierda). El imán se levanta un poco dentro de la cavidad, provocando que la pieza caiga al ser más grande que la cavidad (derecha).

Para la modalidad de JUEGO DE GATO, el gripper presenta algunas modificaciones (ver figura 8.2c):

- Debido a que se requiere sostener un plumón, se añade un cilindro a modo de sostén para el plumón.
- El plumón se sujeta al imán mediante otro imán colocado horizontalmente en la base del plumón
- Se colocó un resorte dentro de la cavidad del brazo #2 para lograr que el plumón ejerza cierta fuerza sobre el tablero y pinte adecuadamente.

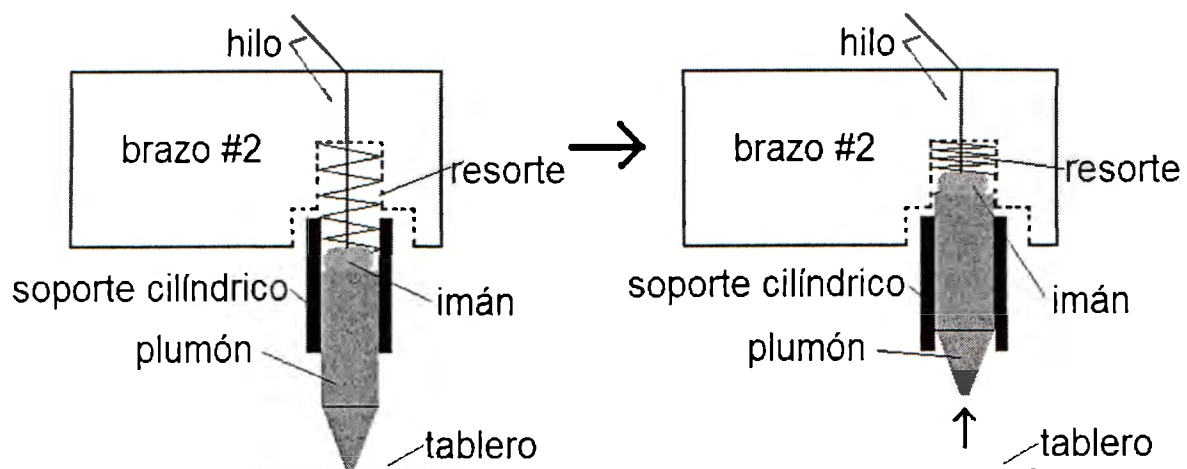


Figura 8.2c. Se observa el plumón en posición baja presionado por el resorte hacia el tablero (izquierda). El servo #3 gira, jala el hilo y hace que el plumón se levante del tablero (derecha).

8.3 ESTRUCTURA DEL SISTEMA Y WEBCAM

En cuanto a la estructura para la webcam, éste se realizó con perfiles estructurales de aluminio y el diseño se muestra en la Figura 8.3a. Cuenta con 4 grados de libertad (ver Figura 8.3b), los cuales son de gran ayuda para localizar el centro del tablero en la imagen. Adicionalmente, la Webcam puede ser enfocada para un mejor procesamiento de las imágenes.

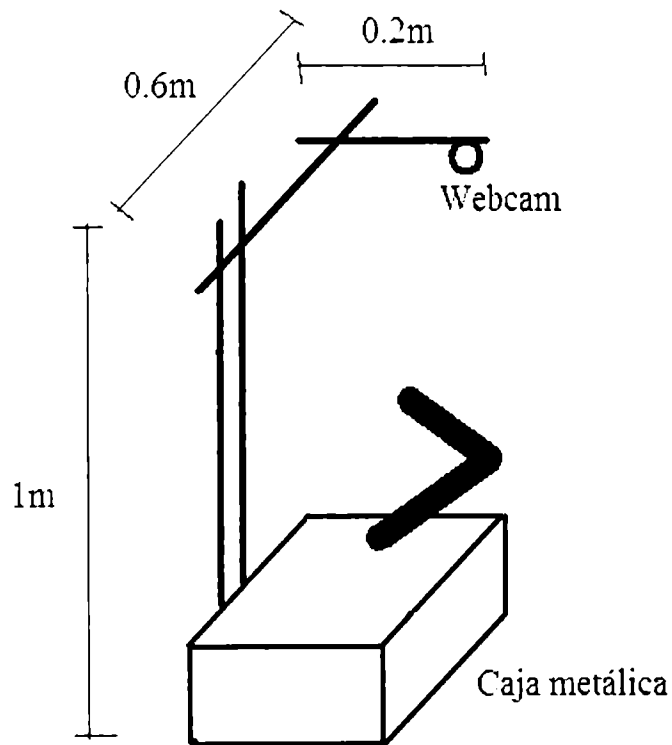


Figura 8.3a. Caja metálica con estructura para la Webcam
NOTA: El diagrama no está a escala

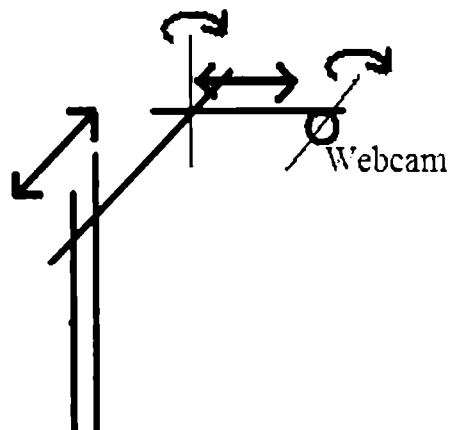


Figura. 8.3b. Estructura de la Webcam. Se observan los 4 grados de libertad con los que cuenta

La ubicación del sistema SCARA con respecto al tablero es la siguiente:

El "offset" del robot está a la izquierda, es decir, que el centro del robot se encuentra a 1cm a la izquierda del centro del tablero y a 10cm del borde del tablero. La posición del robot con respecto al tablero se muestra en la Figura 8.3c.

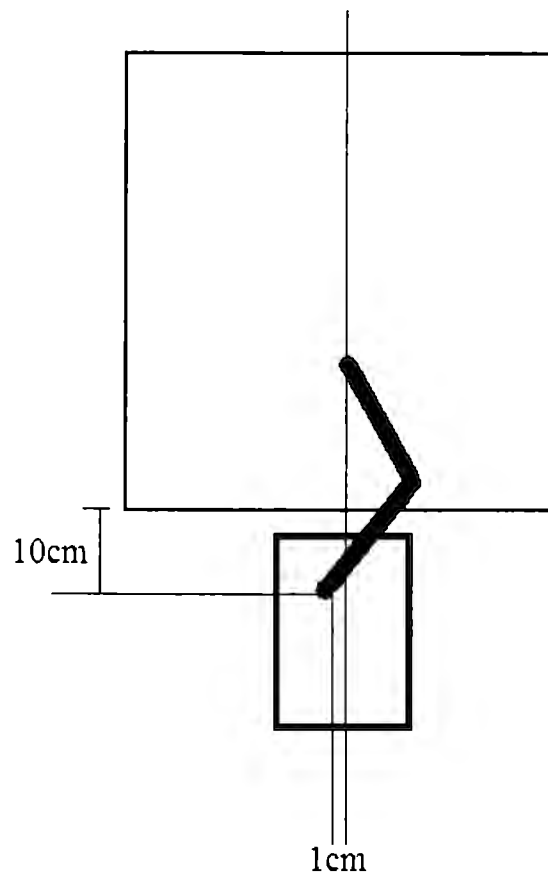


Figura. 8.3c. Nuevo offset e inclinación del robot
NOTA: El diagrama no está a escala

La configuración anterior permite alcanzar la totalidad de los puntos del tablero sin llegar a los límites físicos de los servomotores.

A la caja metálica se le añadieron tiras de Velcro® en la parte inferior con el fin de poder asegurarla a una superficie plana y mantener una posición segura y fija durante el trabajo con el robot.

Las dimensiones máximas finales del sistema completo se presentan en la Tabla 8.3

Elemento	Dimensiones	Peso
Brazo 1	35.0 x 3.0 x 3.0 cm	0.226 Kg
Brazo 2	32.0 x 3.0 x 2.0 cm	0.289 Kg
Caja Base	25.0 x 25.0 x 14.0 cm	2.500 Kg
Estructura de webcam	100 x 60 x 20 cm	0.800 Kg
Servo 1	6.0 x 3.0 x 4.0 cm	0.090 Kg
Servo 2	4.0 x 2.0 x 3.5 cm	0.045 Kg
Servo 3	4.0 x 2.0 x 3.5 cm	0.045 Kg
Webcam	6.0 x 6.0 x 6.0 cm	0.100 Kg
TOTAL	100 x 60 x 30 cm	5.700 Kg

Tabla 8.3. Dimensiones y peso de los principales componentes del sistema

NOTA: Las dimensiones totales contemplan las dimensiones máximas del sistema como un conjunto. El peso total contempla elementos secundarios del sistema

La figura 8.4d muestra una imagen del sistema completo. La figura 8.4e muestra la perspectiva desde la webcam.

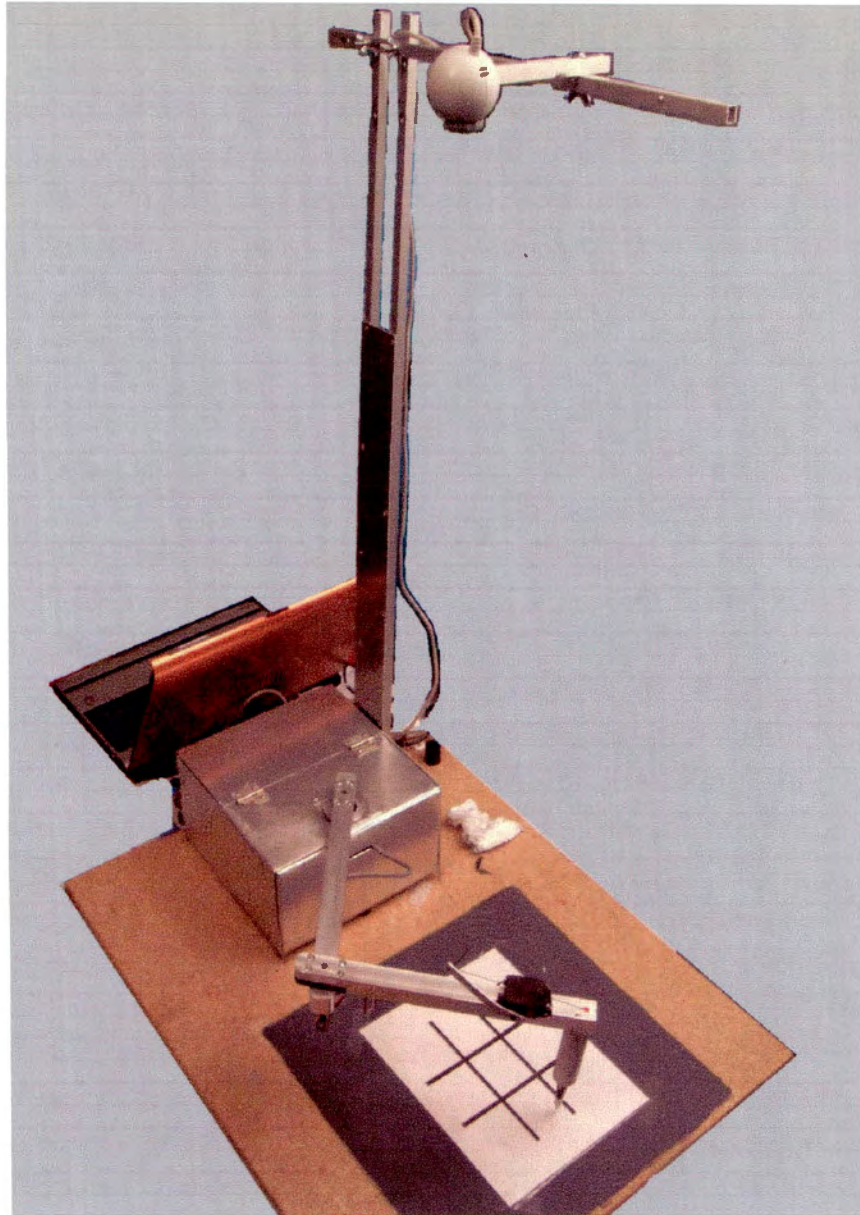


Figura 8.4d. Construcción completa del sistema SCARA

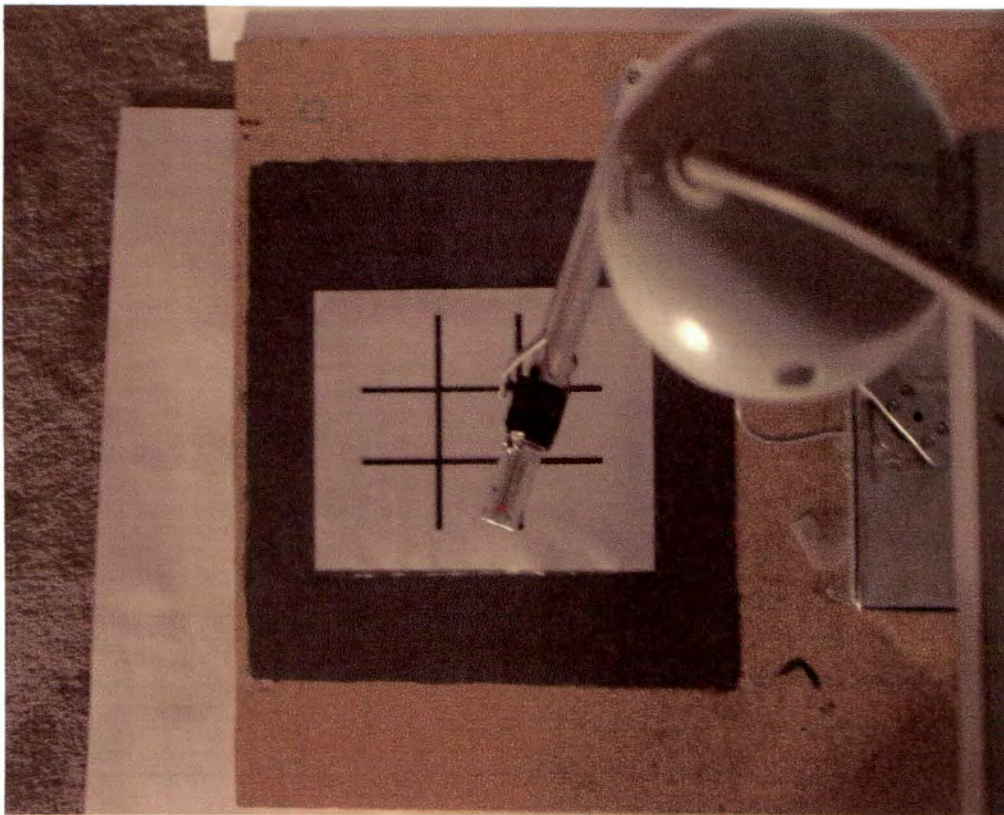


Figura 8.4e. Perspectiva desde la webcam. Se observa la webcam vista desde arriba, el tablero de GATO y los brazos del sistema

8.4 PROGRAMAS EN MATLAB®

Los sistemas de cómputo para el sistema fueron desarrollados con MATLAB®. A continuación se muestra la estructura jerárquica de los programas utilizados así como una pequeña explicación de cada uno de ellos. Los códigos fuente pueden consultarse en el Anexo 2.

Para la modalidad de juego de GATO, el programa de "Gatorobot" es el archivo principal. La estructura de programas y subprogramas es la siguiente:

Gatorobot

- **Quitar brazo**: manda los brazos del sistema a una posición que no interfiera en la visibilidad de la cámara hacia el tablero
- **Pintargato**: en la pantalla de la computadora, dibuja el símbolo correspondiente en el cuadrante correspondiente.
- **tomatablerogato**: Toma una foto del tablero e identifica el centro de cada cuadrante
- **tomatiro**: Toma una foto del tablero e identifica el último tiro del humano
- **pintarcirculo**: En la pantalla de la computadora, dibuja un círculo en el cuadrante seleccionado
- **pintarcuadrado**: En la pantalla de la computadora, dibuja un cuadrado en el cuadrante seleccionado
- **esquinacuadrante**: encuentra la esquina superior izquierda del cuadrante seleccionado
- **Aigato**: Utiliza un sistema de ponderación para cada casilla libre dependiendo de los tiros anteriores tanto del humano como del robot para así determinar el siguiente tiro del robot.
- **ganador**: Determina si existe algún ganador del juego
- **pintarcruzrobotcinematica**: En el tablero y durante el turno del robot, dibuja el símbolo "X" en la casilla seleccionada
 - **esquinacuadranterobot**: Determina los 4 puntos finales de una "X" para ser dibujada en el tablero

Para la modalidad de CONCURSO IEEE, la estructura jerárquica de códigos es la siguiente:

Correr completo

- **Abrir serial:** Ejecuta los comandos para habilitar el puerto serial de la computadora.
- **Quitar brazo:** Manda las instrucciones necesarias al controlador de servos para colocar el sistema fuera del campo visual de la webcam.
- **Tomar tablero ficha 2:** Toma una fotografía del tablero e identifica la posición de las piezas en formato 'x', 'y'.
- **Rutas:** Calcula la ruta completa que seguirá el sistema para recolectar la totalidad de las piezas
- **Mover pieastro:** Mueve los brazos para llegar a la posición deseada controlando la posición en formato 'XY' con un control P.

8.5 SISTEMAS DE CALIBRACIÓN DEL SISTEMA

Durante el desarrollo del presente proyecto se construyeron sistemas de calibración manual del robot, los cuales se describen a continuación:

Programa de calibración manual (creado en LabVIEW®)

El siguiente programa permite al usuario calibrar al robot en tres diferentes formas:

- Ingresando las posiciones en 'x' y 'y' en las cuales se desea que se posicione el gripper (ver Figura 8.5a);
- Ajustando los ángulos del brazo 1 y brazo 2 con respecto a la horizontal (ver Figura 8.5b); y,
- Seleccionando un punto en una simulación del tablero con el Mouse de la computadora (ver Figura 8.5c). El gripper irá al punto correspondiente del tablero real.

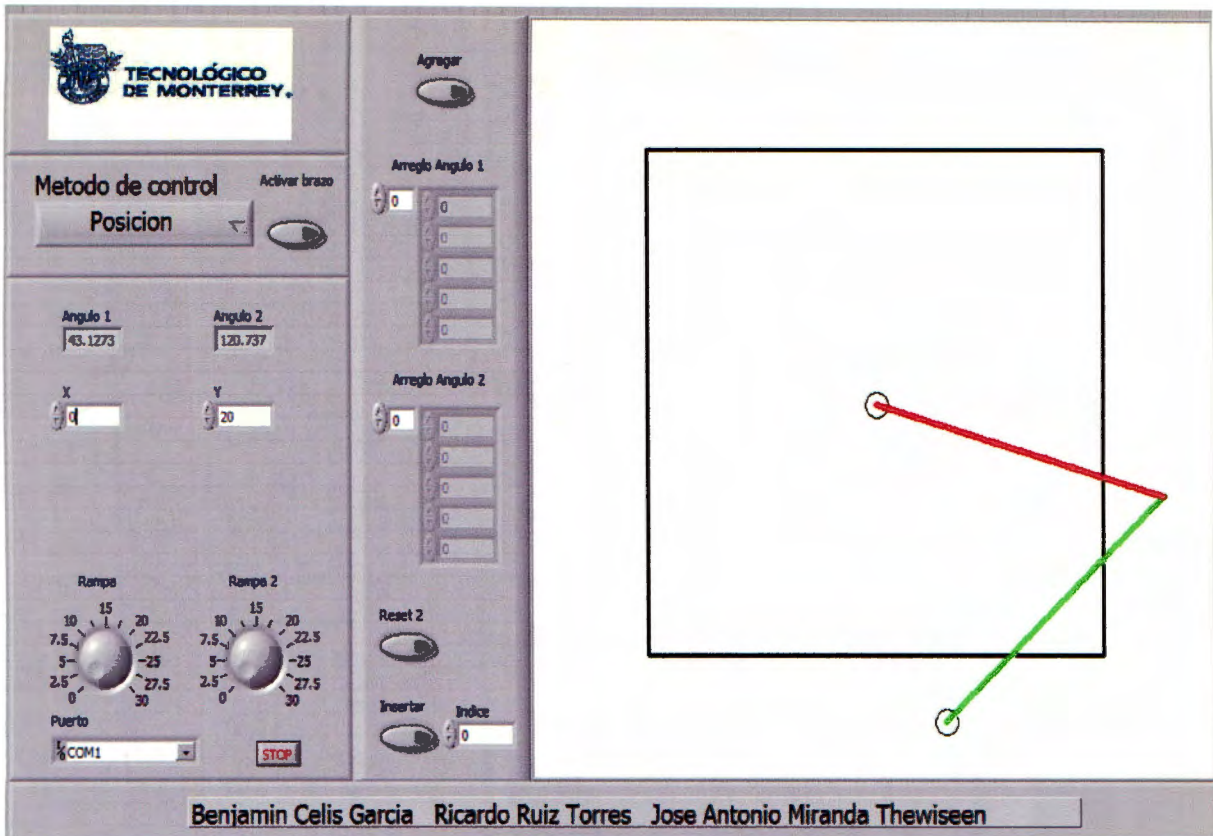


Fig. 8.5a. Interfaz de Calibración para seleccionar las coordenadas en 'x' y 'y' de un punto deseado. El sistema SCARA irá al punto seleccionado

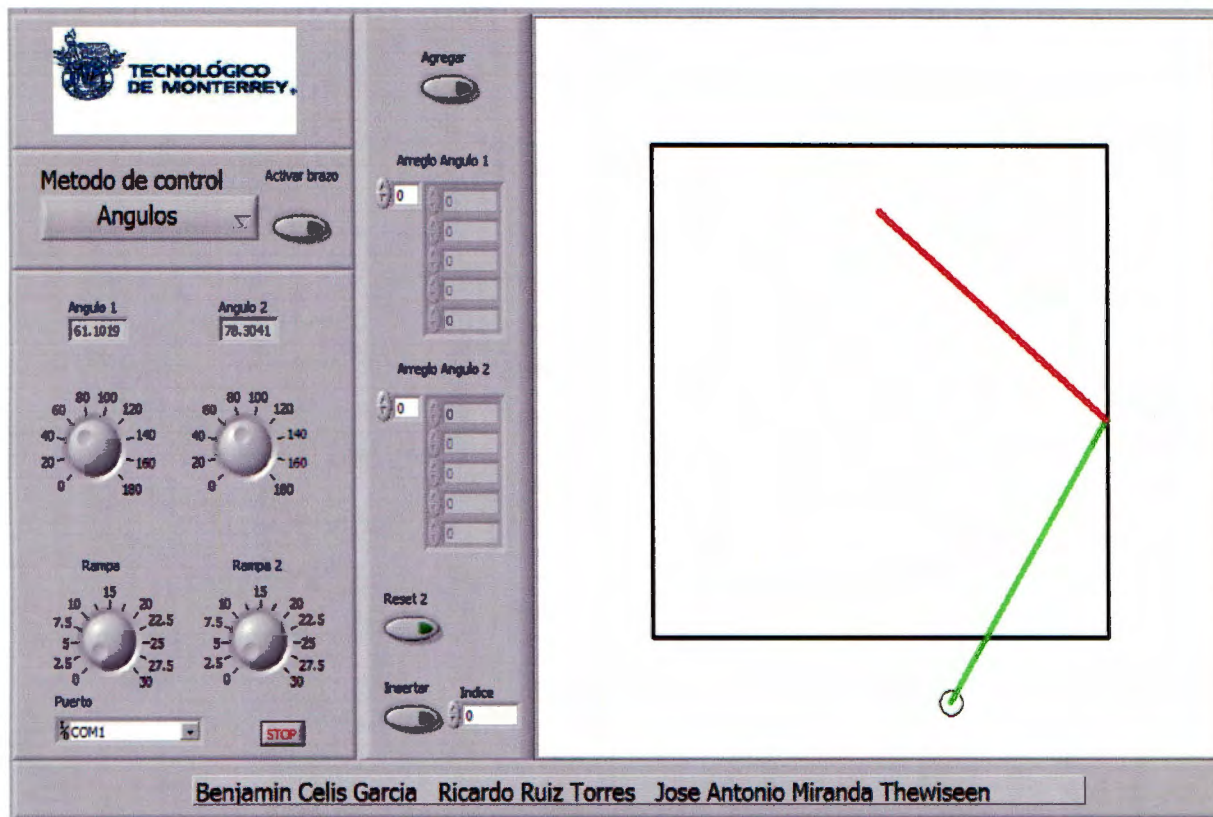


Fig. 8.5b. Interfaz de Calibración para seleccionar el ángulo con respecto a la horizontal de cada uno de los brazos (1 y 2). El sistema SCARA irá al punto seleccionado

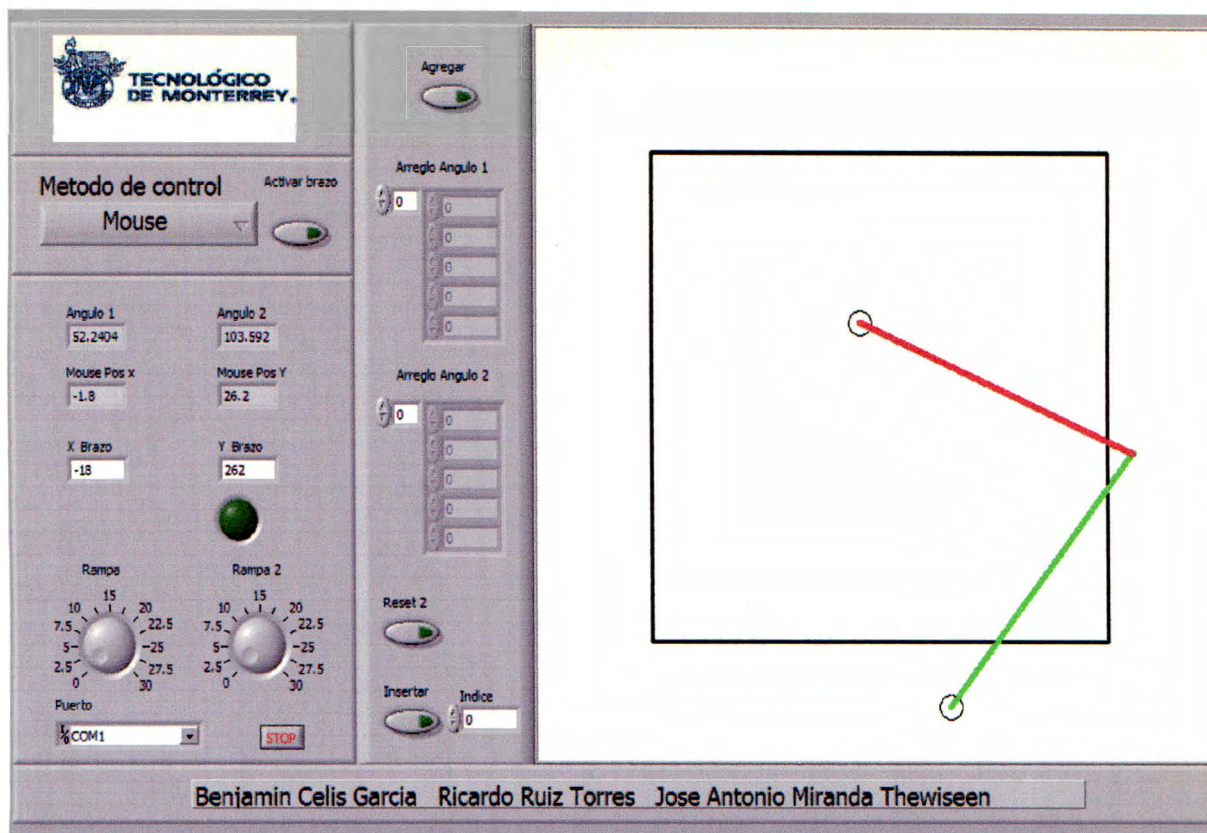


Fig. 8.5c. Interfaz de Calibración para seleccionar un punto con el "Mouse" de la computadora. El sistema SCARA irá al punto seleccionado

Adicionalmente, el programa presenta la modalidad de guardar en memoria cualquier número de puntos y posteriormente ir a cada uno de ellos en forma secuencial, como se muestra en la Figura 8.5d

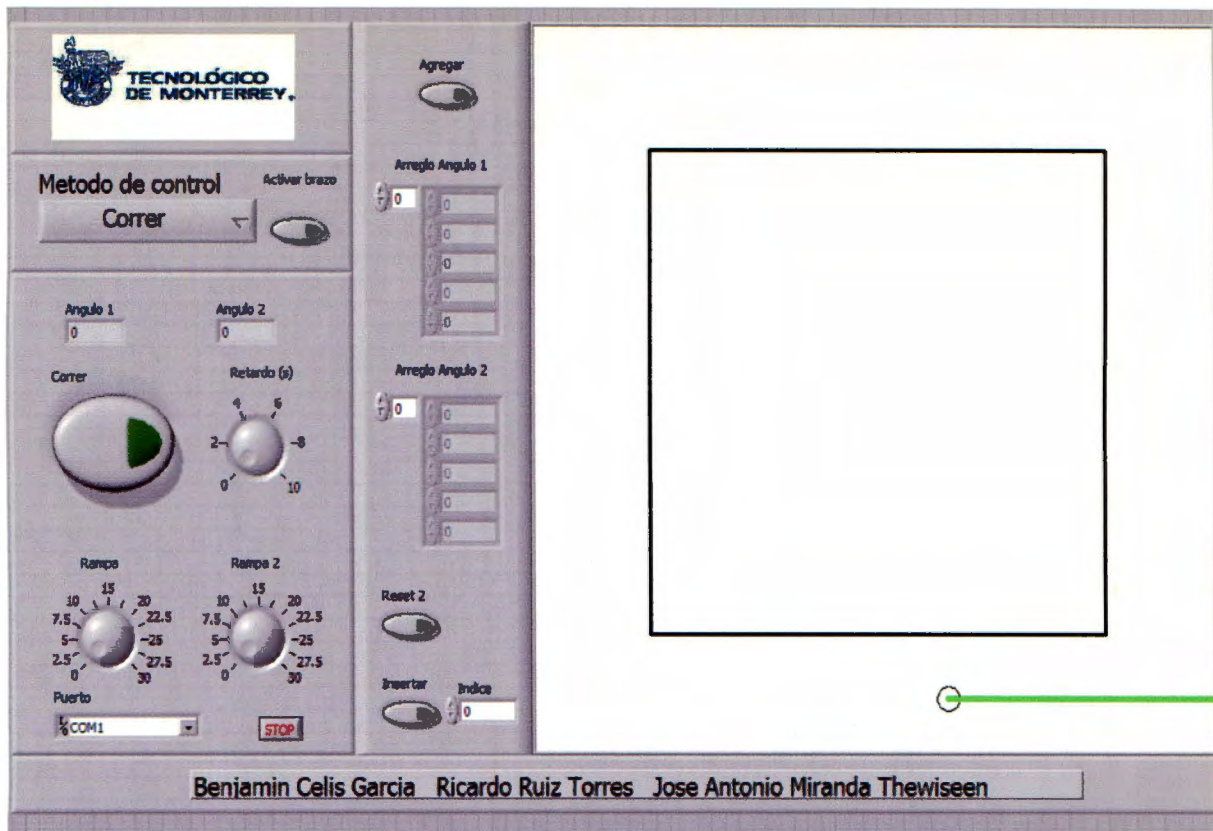


Fig. 8.5d interfaz de Calibración para guardar cualquier cantidad de números y posteriormente seguirlos secuencialmente.

8.6 OPCIONES SOBRE EL TIPO DE CONTROL DEL SISTEMA

La decisión sobre el tipo de control utilizado en el proyecto se basó en el desempeño de cada tipo de control. Al graficar las salidas de cada opción de control, se obtuvieron las imágenes siguientes:

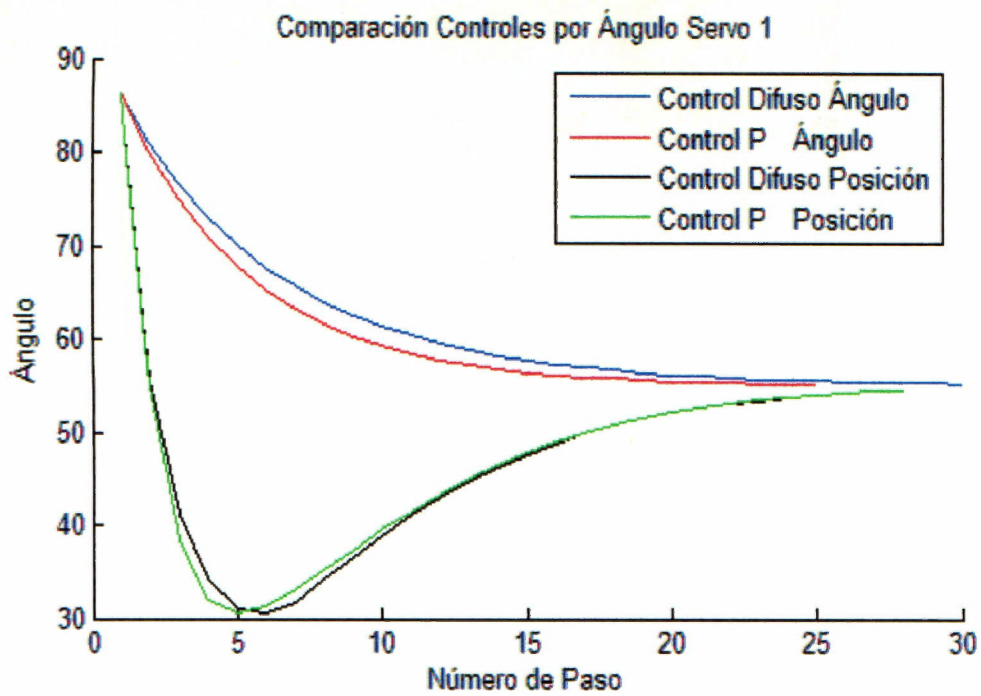


Figura 8.6a. Salida del ángulo del servo #1 en el tiempo (medido en número de pasos). Se observa que el control de posición (tanto Difuso como P) provocan un cambio más brusco en el servo.

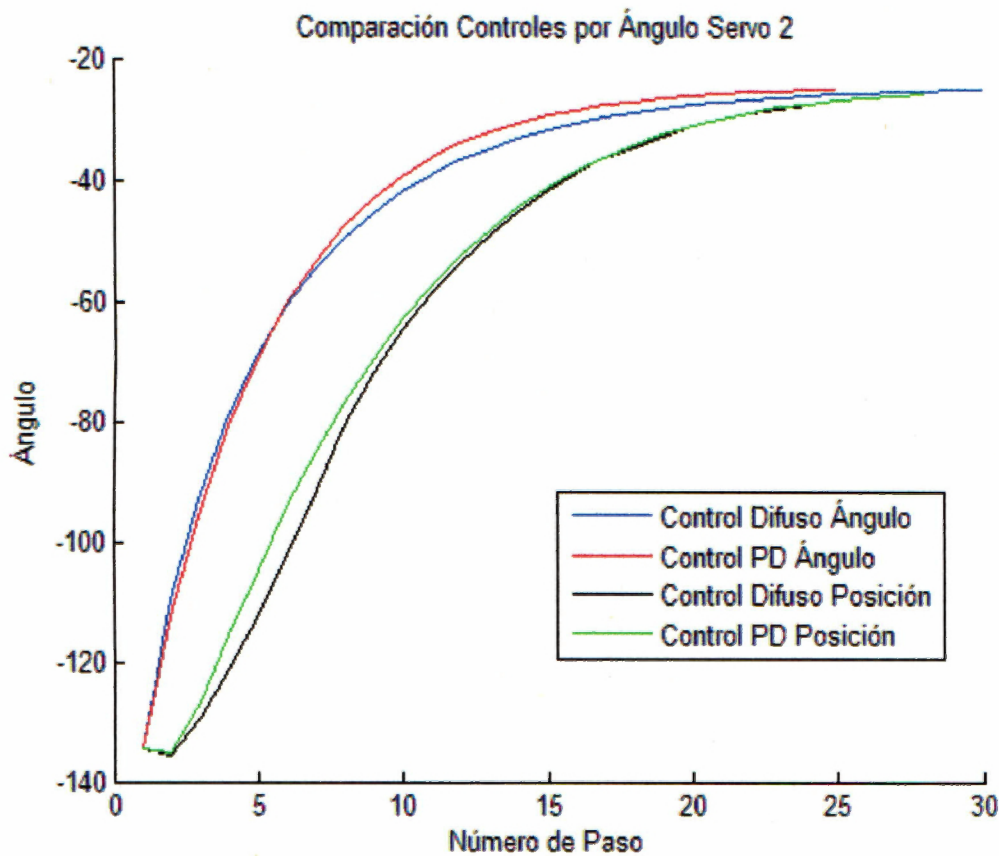


Figura 8.6b. Salida del ángulo del servo #2 en el tiempo (medido en número de pasos). Se observa que las salidas de los controles de ángulo (tanto Difuso como P) llegan más rápido a la referencia que las demás.

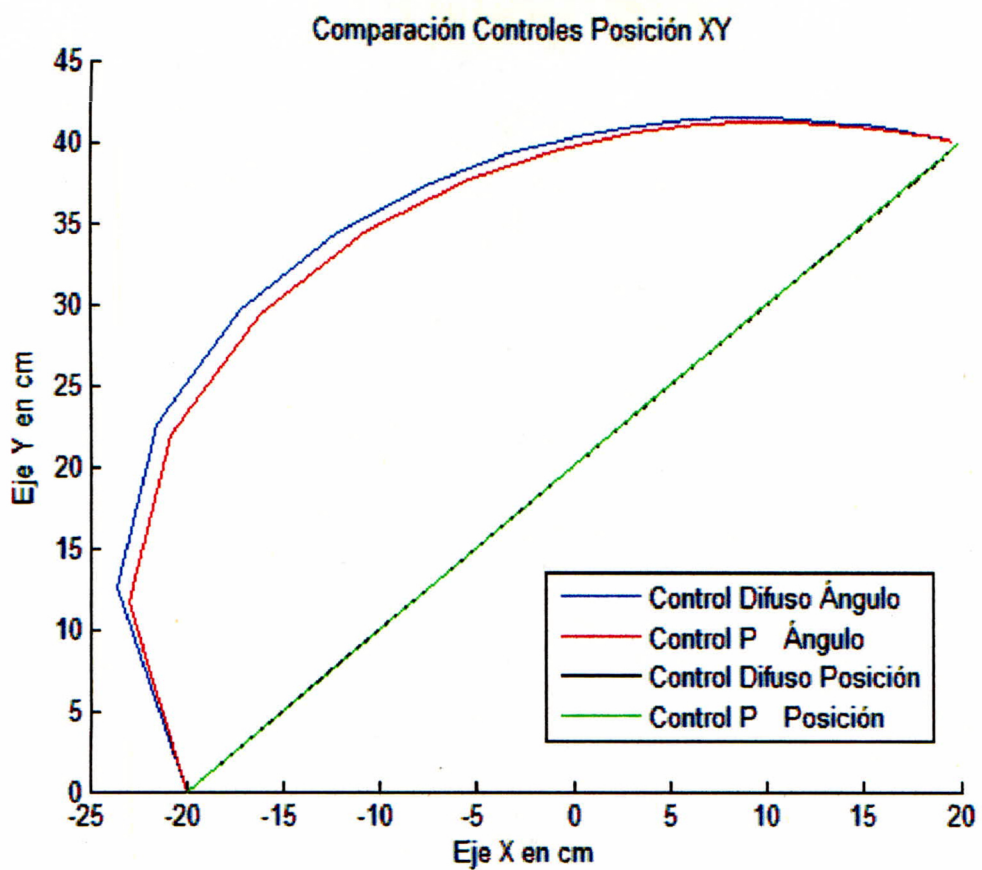


Figura 8.6c. Comparación de la posición de los 2 servos. Se observa que la relación con los 2 controles de posición es lineal, mientras que con los controles de ángulos la relación no es así.

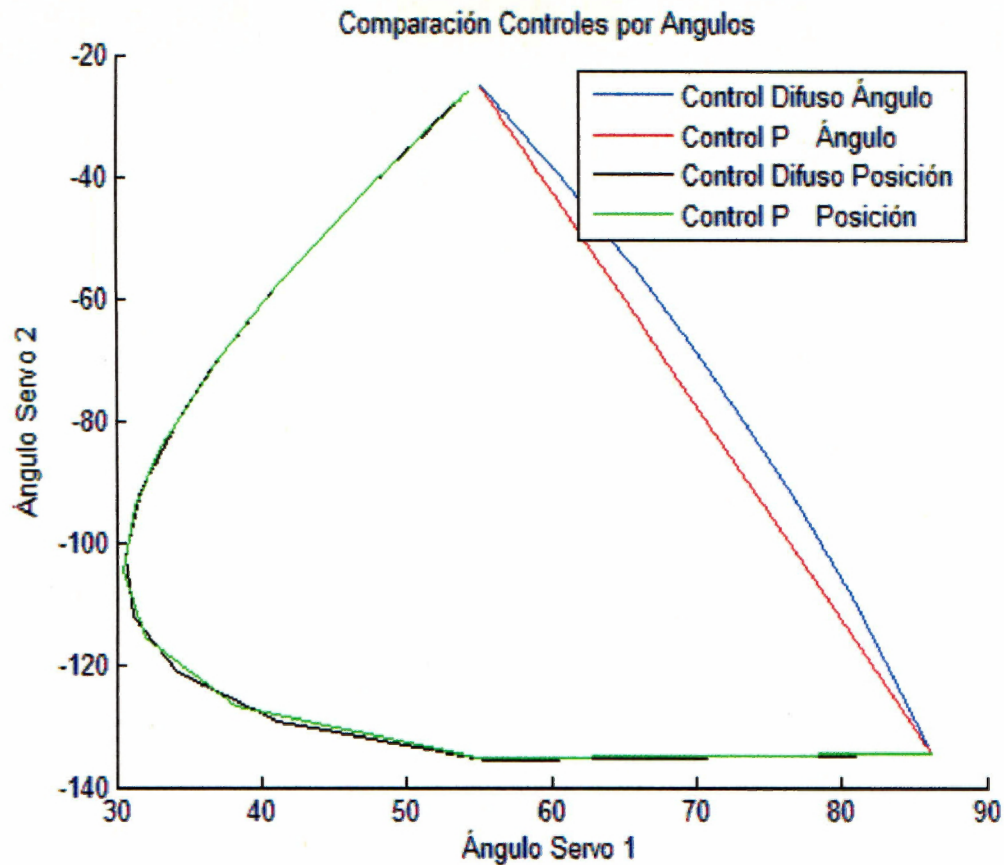


Figura 8.6d. Comparación de los ángulos de los 2 servos. Se observa que la relación con los 2 controles por ángulos es casi lineal, mientras que con los controles de posición, la relación no es así.

Al analizar las imágenes anteriores, se decidió utilizar el Control P por ángulos debido a los siguientes argumentos:

- Como se observa en la figura 8.6b, el control por ángulos es más rápido en el servo #2 que el control por posición. Esto es deseable especialmente en la modalidad de CONCURSO IEEE.
- Como se observa en la Figura 8.6a, el control por ángulos provoca cambios más suaves en el servo #1 que el control por posición.
- Aún cuando la relación entre las posiciones en X y Y sea lineal en el control por posición, no nos perjudica utilizar el control por ángulos, el cual no es lineal en la relación de posiciones en los ejes.
- Una vez tomada la decisión de utilizar el control por ángulos, la segunda decisión de utilizar el Control P en vez del Control Difuso se tomó debido a que las salidas mostradas en las figuras anteriores muestra que la diferencia entre el control P y el Difuso es pequeña y no justifica el consumo adicional de recursos de poder y tiempo de cómputo que representa el Control Difuso.

Una vez tomada la decisión sobre el Control P, la ganancia del sistema se determinó a prueba y error dependiendo de las necesidades del sistema. Esto es, que en la modalidad de juego de GATO, la velocidad de los brazos no es crítica y se puede utilizar una ganancia K de 0.5. Sin embargo, en la modalidad de IEEE, la velocidad con la que el gripper llega a las piezas tiene mayor importancia, por lo que la ganancia K puede tomar valores mayores (la experiencia mostró que con $K = 0.7$, el desempeño del sistema era adecuado en velocidad y precisión).

Es importante notar que debido a la naturaleza de los servomotores, mientras mayor sea la K, el sobretiro del sistema será mayor, y la posición final de los servomotores podría no ser la deseada. Para asegurar que la posición deseada es alcanzada, el sistema debe comportarse como un sistema de primer orden (en donde no hay sobretiro). Mediante la experimentación, sabemos que nuestro sistema presenta sobretiro cuando $K > 0.7$.

El código referente al control P del movimiento de los brazos se encuentra en el Anexo 2, código de "CONTROLPDIESTRO.M" (ver página 82).

8.7 RESULTADOS DEL PROCESAMIENTO DE IMÁGENES

A continuación se muestra la secuencia del procesamiento de las imágenes, tal y como se describe en la sección 6.4 PROCESAMIENTO DE IMÁGENES.

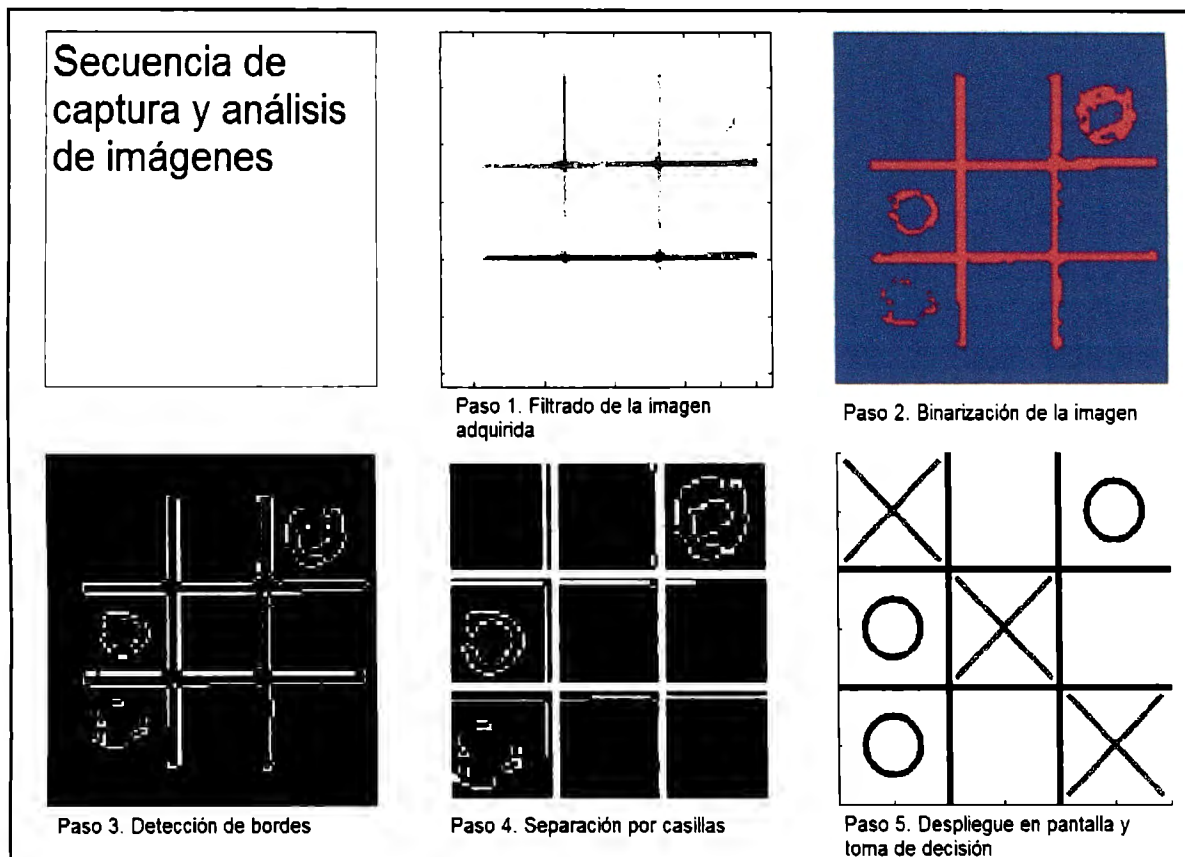


Figura 8.7a. Secuencia del procesamiento de la imagen en la modalidad de GATO. Se detectan los tiros del jugador (círculos). Los tiros del robot ('X') no se procesan como imagen y se manejan por software (el sistema guarda la información de dónde ha tirado).

Al realizar la secuencia anterior, se encuentran e identifican los símbolos de los tiros del jugador (en el GATO) y se localizan las piezas y se identifica su color (CONCURSO IEEE).

De forma similar, el procesamiento de imágenes para el CONCURSO IEEE sigue la misma secuencia como puede observarse en la figura 8.7b

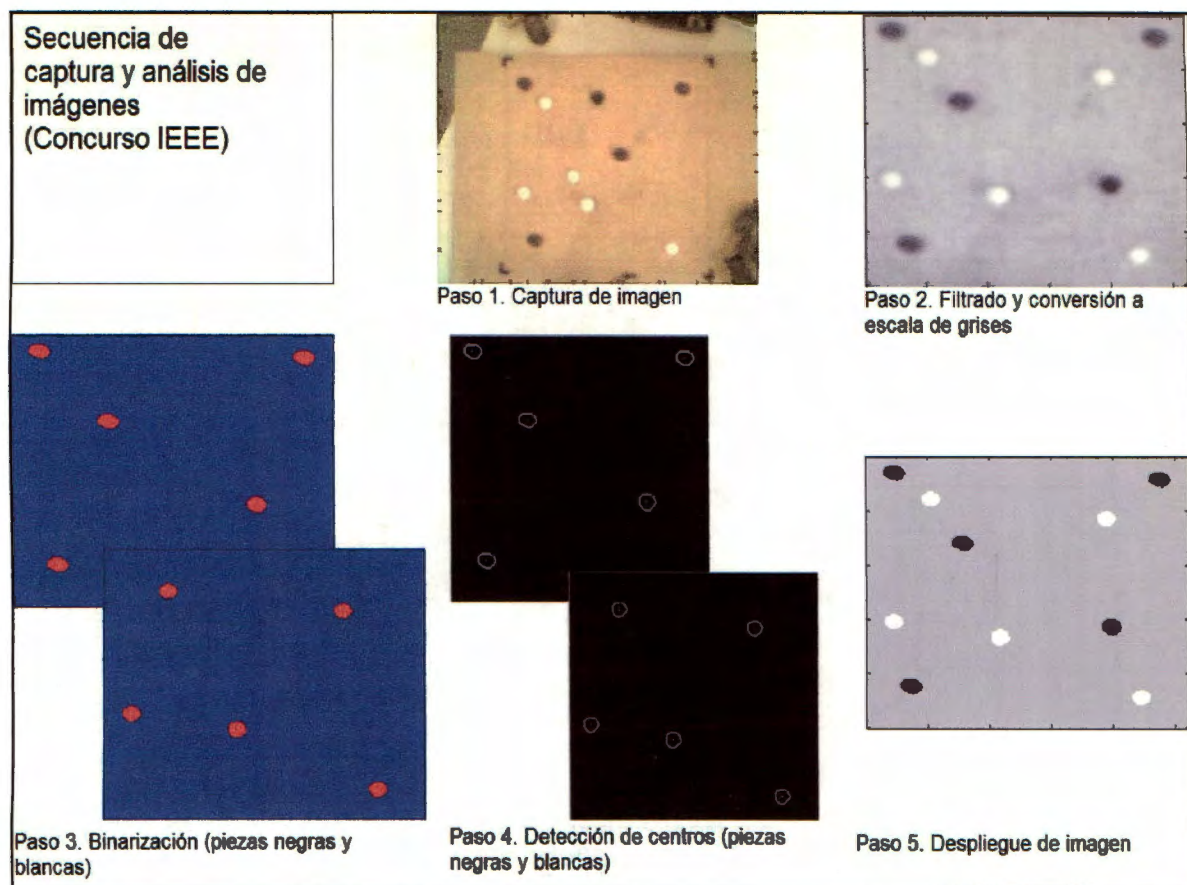


Figura 8.7b. Secuencia del procesamiento de la imagen en la modalidad de CONCURSO IEEE

8.8 RESULTADOS DEL SISTEMA ROBÓTICO (APLICACIÓN DE CONCURSO DE MINI-ROBÓTICA IEEE)

Luego de aplicar todos los programas descritos anteriormente, el sistema es capaz de resolver el siguiente escenario:

- ❖ Dadas las condiciones del concurso de mini-robótica IEEE descritas en la sección 2. MOTIVACIÓN, todas las piezas de un mismo color son separadas de las demás y colocadas en un mismo contenedor.
 - La selección de la primera pieza a recoger siempre es la más cercana al punto inicial "HOME" del efector final.
 - La selección del contenedor en el cual se deposita la primera pieza recolectada se determina como el más cercano a la ubicación de dicha pieza.
 - Una vez depositada una pieza en el contenedor correspondiente, la siguiente pieza a recolectar será la más cercana al contenedor en el cual se ha depositado la pieza anterior, sin importar el color de la siguiente pieza.

El cálculo de la ruta a utilizar se observa gráficamente en la Figura 8.8a.

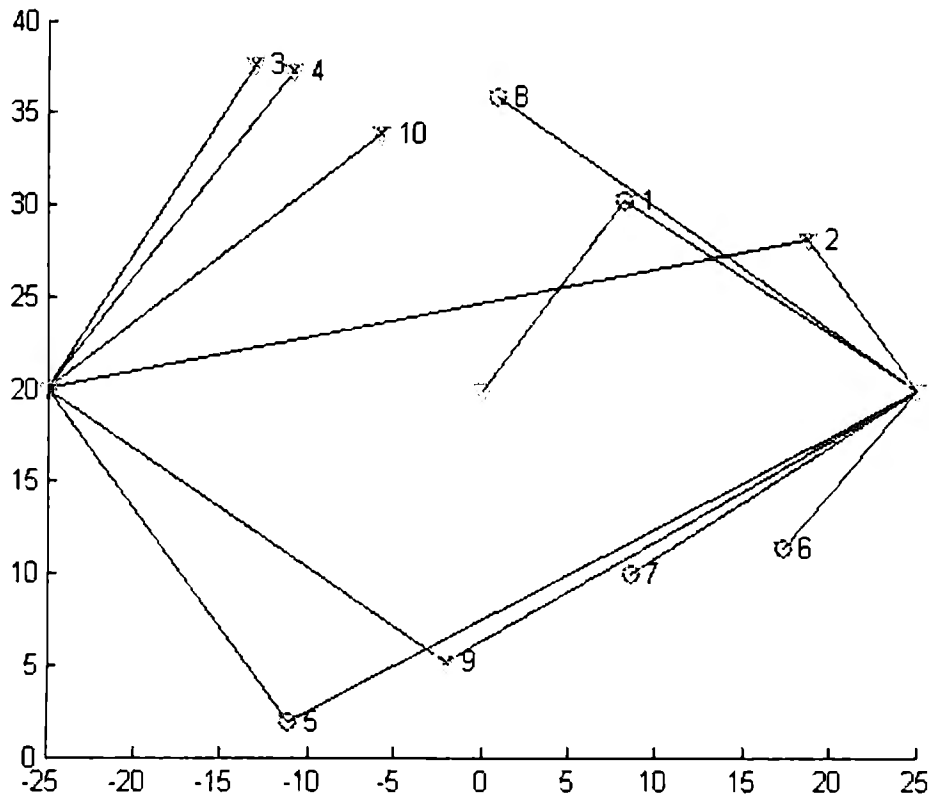


Figura 8.8a. Distancias totales desde el punto inicial del gripper ("HOME") en el centro del tablero y todo el recorrido al recoger y depositar todas las piezas.

- En cuanto a la precisión del sistema, ésta es variable dependiendo de la posición del gripper en el tablero. Esto se debe a que pequeñas variaciones dimensionales provocan desviaciones en la posición final del gripper que son variables en el tablero.

8.9 RESULTADOS DEL SISTEMA ROBÓTICO (APLICACIÓN DE JUEGO DE GATO)

Al aplicar las leyes mostradas en la sección 6.5 PROCESAMIENTO DE INFORMACIÓN, el juego de GATO obtenido fue exitoso y cumple con el objetivo de ser invencible. Sin importar quién inicie el juego, el robot no perderá ningún juego. En cambio, el triunfo del robot en algún juego depende de la habilidad del jugador al que se enfrente.

En cuanto a la precisión del sistema, el tiro del robot se realiza siempre dentro de la casilla correcta. Es decir que el robot no pinta fuera de la casilla, lo cual provocaría problemas en el procesamiento de las imágenes.

8.10 RETROALIMENTACIÓN DEL SISTEMA

Los resultados mostrados en el presente documento son satisfactorios con base en los objetivos planteados al inicio del proyecto y el desempeño en ambas modalidades (IEEE y GATO) es satisfactorio. Por ello, hacer el sistema a Lazo Cerrado no fue finalmente necesario aunque se intentó implementar para mejorar la precisión del sistema.

La idea de retroalimentación fue concebida como sigue:

- Los brazos serían pintados de colores distintos para ser identificados por el programa de interpretación de imágenes. Las caras superiores de los brazos serían pintadas completamente o incluso únicamente algunos puntos sobre las caras superiores.
 - o Sin embargo, existen posiciones de los brazos en las que se hallan fuera de la visibilidad de la webcam, como al intentar colocar el gripper en alguna de las esquinas inferiores. Esto provocaría que la posición del gripper no se pudiera retroalimentar cuando el sistema se encontrara en alguna de dichas esquinas.
- Como otra alternativa, se pensó colocar un punto rojo sobre el extremo del gripper el cual sería identificado por el software y el sistema rectificaría la posición.
 - o Sin embargo, debido a que el gripper se encuentra a una altura aproximada de 10cm sobre el tablero, el factor de perspectiva de la imagen provocaría que el sistema considerara que el gripper se encontrara en una posición incorrecta (ver Figura 8.10a). Un factor de corrección de este error podría ser utilizado, sin embargo es complicado obtener un solo factor de corrección debido a que la webcam no se encuentra exactamente sobre el centro geométrico del tablero; y obtener varios factores provocaría que en ocasiones, el tratar de retroalimentar la posición resultara contraproducente y empeorar la posición final del gripper.

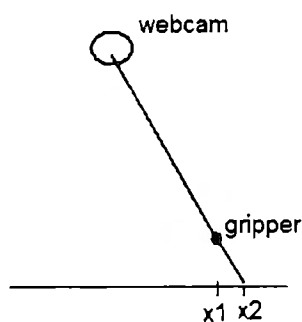


Figura 8.10a. Perspectiva de una posición errónea del gripper. El sistema contempla que el gripper se encuentra en la posición x_2 cuando en realidad se encuentra en x_1 .

8.11 RESULTADOS DEL CONCURSO IEEE DE MINIROBÓTICA (MAYO 4 y 5, 2006)

El concurso IEEE de Minirobótica tuvo lugar en la ciudad de Querétaro, México en las instalaciones del Centro Cultural 'Gomez Morín'. En la modalidad de 'Brazo Manipulador', se contó con una participación de 26 equipos y el proyecto expuesto en el presente trabajo logró el 3er lugar con un tiempo total de 31 segundos.

Los tiempos finales de los 26 equipos fueron de los 14 segundos a los 5:12 minutos. Los sistemas que lograron los lugares 1º y 2º alcanzaron tiempo de 14 y 20 segundos respectivamente y ambos contaban con sistemas neumáticos y electrónicos de tipo industrial.

El desempeño del presente diseño fue satisfactorio. El sistema trabajó al límite para lograr el tiempo de 14 segundos. Para lograr un tiempo menor, el sistema tendría que ser rediseñado con modificaciones considerables, las cuales requerirían de altas inversiones económicas.

En la figura 8.11 se muestra uno de los certificados de 3er lugar obtenidos.



Figura 8.11 Muestra de reconocimiento por haber obtenido el 3er lugar

9. OBSERVACIONES

Las observaciones con respecto al sistema SCARA construido son las siguientes:

- Se diseñó, construyó y ensambló el sistema SCARA previsto el cual tiene las siguientes características:
 - Cumple con todos los requisitos establecidos para el Concurso Nacional de Mini robótica (en la rama de Brazo Manipulador) de la IEEE.
 - Aunque estéticamente puede no ser un ejemplo a seguir para los sistemas SCARA, se le dio prioridad a la funcionalidad del sistema antes que a la estética.
 - Se logró relacionar el diseño en paquetería profesional como ProE® y Mastercam® para generar el código G para el maquinado en las máquinas CNC.
 - El diseño y manufactura del efector final es un elemento clave para el éxito del proyecto. Fallas en este rubro son más visibles en el desempeño general del sistema. Aunque nuestro efector final es funcional, se reconoce que debido a la complejidad del subsistema, se dejan áreas de oportunidad como trabajo a futuro.
- Los conocimientos previos por parte del equipo en cuanto a la electrónica de potencia simplificó el trabajo de un posible diseño y construcción de una fuente de alimentación. Así, la etapa de potencia fue solucionada con la reaplicación de una fuente de alimentación con la que ya se contaba.
- El estudio y aplicación de las teorías de Inteligencia Artificial dentro del presente proyecto fue una experiencia agradable para el equipo. El juego de Gato aplicado al sistema, logró cumplir con uno de los objetivos del proyecto.
- El desempeño del Nylamid SL® pareció cumplir con los requerimientos mecánicos del proyecto en un inicio. Al término del primer semestre de trabajo, el polímero había presentado características satisfactorias en cuanto a maquinabilidad y

rigidez. Sin embargo, al ser sometido accidentalmente a calor, presentó deformaciones inesperadas las cuales llegaron a un punto tal en el que se comprometió el desempeño adecuado del sistema. Finalmente, se tuvo que renunciar a la atractiva relación de resistencia-peso mostrada por el Nylamid SL® debido a la poca resistencia al calor. Sin embargo, la experiencia de haber trabajado con este novedoso material y de haber conocido sus capacidades y límites de forma práctica deja una vivencia agradable al equipo.

- Aunque el sistema a Lazo Abierto es funcional y cumple con todos los objetivos del proyecto, lograr implementar el Lazo Cerrado es altamente recomendable para lograr mejorar la precisión del sistema. Esto resalta especialmente cuando se utilizan servomotores de mediana precisión como es nuestro caso.

10. CONCLUSIONES

Luego de realizar el proyecto descrito en el presente documento se tienen las siguientes conclusiones:

- El equipo logró diseñar, manufacturar y ensamblar todas las partes mecánicas involucradas en el sistema. Los sistemas ingenieriles de paquetería computacional, y equipo de maquinado CNC y convencional fueron un pilar fundamental durante el desarrollo del proyecto. Las nociones previas de dichos sistemas permitieron la adquisición autónoma de nuevos y especializados conocimientos que dirigieron el curso del proyecto hacia su concepción exitosa.
- El estudio previo de las características eléctricas de los componentes electrónicos utilizados nos permitió la flexibilidad de tomar la decisión de utilizar una fuente de alimentación con la que ya se contaba en lugar de construir o adquirir una nueva. Esto dio como resultado un ahorro de recursos durante el proyecto.
- El procesamiento de imágenes es una parte fundamental del proyecto. Esta etapa es fundamental para el éxito del proyecto, por lo que pequeñas imprecisiones pueden llevar a que no se alcancen los objetivos. Por ello, el equipo hizo especial hincapié en su desarrollo y dedicó recursos importantes para su desarrollo. Su implementación se logró con ayuda de las herramientas teóricas y tecnológicas con las que el equipo contaba al inicio del proyecto y aquellas nuevas que fueron adquiridas durante el desarrollo.
- El trabajar a Lazo Abierto resultó con un buen desempeño del sistema. Los errores en la posición del brazo no representan problemas críticos para lograr un buen trabajo bajo las condiciones del proyecto de la IEEE y del juego de GATO. Sin embargo, la implantación del Lazo Cerrado eliminaría las impresiones que aunque no son críticas en nuestro rango de trabajo, podrían llegar a serlo bajo otras condiciones (como ampliar el campo de trabajo o aumentar la carga en los brazos, por ejemplo). Adicionalmente, al cambiar de material de Nylamid SL® a Aluminio

para los brazos del sistema, el peso de los mismos aumentó (debido a la mayor densidad del Aluminio). Esto provocó que la inercia de los brazos en movimiento aumentara y elevara la carga sobre los servomotores, disminuyendo así su precisión.

- Los conocimientos que la carrera de Ingeniería en Mecatrónica con la que cuenta el equipo de trabajo fueron fundamentales para concepción y desarrollo del proyecto. Las tres grandes áreas de conocimiento (Mecánica, Electrónica, Computación) con la que cuenta esta Ingeniería se aplicaron a la perfección para las tres grandes etapas del proyecto (Manufactura, Potencia y Control respectivamente).

11. TRABAJO A FUTURO

Aunque el proyecto expuesto en el presente documento presenta un desempeño adecuado conforme a lo esperado y planteado en los objetivos, ciertas mejoras al sistema harían que el sistema fuera más robusto. Por ello, se plantea el siguiente trabajo a futuro para ser terminado incluso después de terminar nuestros estudios en el Tecnológico de Monterrey:

- *Implementación del Sistema a Lazo Cerrado.* El retroalimentar la posición del gripper sobre el tablero es un factor clave para asegurar una buena precisión del sistema y elevar las posibilidades de triunfo en el concurso de la IEEE, en donde pequeñas variaciones en la posición del gripper pueden provocar que la pieza no sea recolectada o que se caiga debido a una mala sujeción con el gripper.
- *Reducir las variaciones dimensionales del sistema.* A lo largo del trabajo en el proyecto, nos dimos cuenta de que pequeñas imprecisiones en las dimensiones pueden llegar a ser críticas en la posición final del gripper. Por ejemplo, aún cuando las dimensiones de los brazos fueron muy precisas (gracias a un buen dimensionamiento en ProE® y un buen maquinado en CNC), las articulaciones llegaron a presentar pequeñas variaciones al atornillar los elementos a mano (como las estrellas de los servomotores o el empotramiento de los mismos servomotores).
- *Mejorar el rendimiento del sistema.* Luego del concurso de la IEEE el 4 y 5 de mayo 2006, el equipo sabrá si es necesario invertir mayor cantidad de recursos para mejorar el desempeño del sistema. Comprar servomotores más rápidos y precisos podría mejorar el tiempo en el que el robot completa una corrida en el concurso IEEE.

12. REFERENCIAS

- [1]. IEEE Concurso Nacional de Mini robótica. <http://www.minirobotica.org>

- [2]. Angeles, J.; Morozov, A.; Navarro, O. *A novel manipulator architecture for the production of SCARA motions*, 2000. Base de datos IEEEExplore

- [3]. Robot History. http://trueforce.com/Articles/Robot_History.htm

- [4]. Kinematics. <http://www.roble.info/robotics/html/SerialKinematicChains-1se1.html>

- [5]. Nylamid® <http://www.nylamid.com.mx/>

- [6]. Bernard J. Hamrock, Steven R. Schmid, Bo O Jacobson. *Fundamentals of Machine Elements*. McGraw-Hill Professional

- [7]. WebElements. <http://www.webelements.com>

- [8]. Engineering Fundamentals. <http://www.efunda.com>

- [9]. Tower Hobbies. <http://www.towerhobbies.com>

- [10]. Hobbico. <http://www.hobbico.com>

- [11]. Konar, A. *Artificial Intelligence and Soft Computing*. CRC Press. E.U.A., 2000

- [12]. Kondo, T.; Pandya, A.S. *Medical image recognition by using logistic GMDH-type neural networks*. 2001. Base de datos IEEEExplore

- [13]. Farroha, B.S.; Deshmukh, R.G. *An alternative method for designing parallel image recognition systems: a feasibility study*. 1995. Base de datos IEEEExplore
- [14]. Futaba. <http://www.futaba-rc.com>
- [15]. Singh, J.; *Computational intelligence with fuzzy logic for complex systems* Modern Communication Technologies, 2001. The IEEE-Siberian Workshop of Students and Young Researchers. Base de datos IEEEExplore
- [16]. Nguyen, Hung, et al. *A first Course in Fuzzy and Neural Control*. Chapman & Hall/CRC EUA, 2003
- [17]. Czogala, Ernest, et al. *Fuzzy and Neuro-Fuzzy Intelligent Systems*. Physica-Verlag. Polonia, 2000.
- [18]. Scott Edwards Electronics Inc. Mini SSC II Serial servo Controller Users Manual EUA, 2000.
- [19]. Angulo, J., Curso de minirobótica impartido en el Instituto Tecnológico y de Estudios Superiores de Monterrey CCM. México, 2001.
- [20] Paul-I-Hai Lin; Sentai Hwang; Chou, J.; *Comparison on fuzzy logic and PID controls for a DC motor position controller*. Industry Applications Society Annual Meeting, 1994. Base de datos IEEEExplore.
- [21] Martinez, J.; Bowles, J.; Mills, P.; *A fuzzy logic positioning system for an articulated robot arm*. Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference. Base de datos IEEEExplore.
- [22] Electrónica Unicrom. http://www.unicrom.com/Tut_fuentepoder.asp
- [23] Power Electronics. Rashid M. Prentice Hall. 2nd edition. 1993
- [24]. Parallax. Inc. <http://www.parallax.com>

13. ANEXOS

ANEXO 1 Bases del 10º Concurso de Minirobótica IEEE.

Categoría de “Brazo Manipulador”

Brazo Manipulador

Criterio de Calificación

El criterio de calificación es de tal manera que se minimizará la posible subjetividad de los jueces, y permitirá que si los mismos concursantes evalúan a sus compañeros obtengan un resultado equivalente. En esta categoría se evaluará el robot tomando en cuenta si el primer grado de libertad es de tipo lineal o rotacional, teniendo mayor puntuación los prototipos que utilicen una articulación rotacional. Para ello en la tabla de calificaciones se contará con una puntuación máxima de 20 para los que manejen el primer grado de libertad lineal y una puntuación máxima de 40 para quienes manejen el tipo rotacional. En todos los incisos a calificar, se sigue como criterio asignar la calificación más alta para el mejor y cero para el peor, repartiéndose los demás entre estos dos valores. Por ejemplo, si el mejor saca 9 de 10 en originalidad y el menor 6 de 10, entonces el de 9 tiene 100% de los puntos y el de 6 tiene 0 y todos los demás que se encuentran con calificación entre 6 y 9, tendrán puntos repartidos entre 0% y 100%. Se valen empates.

Este criterio permite evaluar al mejor equipo dando prioridad a ciertos aspectos. El mayor peso está en terminar de colocar las 10 piezas en sus respectivos contenedores. En segundo lugar pesa el tiempo en ejecutar la tarea. El siguiente punto de mayor peso es la originalidad y creatividad, que al igual que los otros puntos se evaluará de forma relativa dando 100% de los puntos al más original y creativo y 0 puntos al menos original y creativo, repartiendo los puntos proporcionalmente para los intermedios.

Véase al final de estas bases un ejemplo de evaluación.

Con estos criterios en mente, la puntuación que se asignará a cada participante se efectuará de acuerdo a los siguientes criterios de calificación:

1. Cada equipo deberá entregar al jurado, al momento de su primera participación, un documento en formato Microsoft Word y un máximo de 10 páginas con la descripción técnica del diseño y cálculos. (4 puntos al mejor y 0 al más bajo).

2. Exposición pública. Un representante de cada equipo hará una presentación oral sobre las características del manipulador robótico en cuanto a diseño, construcción y control del mismo. La presentación oral se efectuará en público y para la cual contará con 10 minutos.

(4 puntos al mejor y 0 al más bajo).

3. Originalidad y creatividad en piezas, mecanismos y desarrollo de la electrónica utilizada, así como la construcción versus integración de componentes. (14 puntos al mejor y 0 al más bajo).

4. Tiempo de realización de la prueba. (30 puntos al mejor y 0 al más bajo).

5. Número de aciertos del robot al manipular las piezas y colocarlas en el contenedor correcto. (20 puntos al mejor y 0 al más bajo en caso del primer grado de libertad lineal, 40 puntos como máximo en el manipulador de eje rotacional).

6. Calidad en la Ejecución. (5 puntos al mejor y 0 al más bajo).

• Vibraciones en el movimiento * estabilidad de los movimientos.

7. Presentación final, calidad en los acabados y estética (5 puntos al mejor y 0 al más bajo.) A continuación se da un ejemplo de la forma de calificar. Suponiendo que el equipo "e" utiliza un manipulador con el primer grado de libertad rotacional, mientras que el resto utilizan movimientos lineales.

En relación a la competencia

Habrá tableros iguales a los del concurso, disponibles para realizar ajustes y prácticas antes de la competencia para practicar antes de ser llamado. Cada concursante tendrá un área de trabajo con contacto eléctrico para hacer sus pruebas y ajustes desde el día de registro a la competencia. En cada paso eliminatorio cada concursante tendrá oportunidad de 3 ensayos no consecutivos, para dar oportunidad a que haga ajustes. Se tomará el ensayo que el concursante considere mejor de los 3.

3

Para ello hará su propio juicio tomando en cuenta el número de piezas acertadas y el tiempo utilizado.

El tiempo correrá cuando el concursante dé la señal de arranque, y retire manos y cualquier forma de interacción con el robot. El tiempo se detendrá cuando el concursante lo decida o después de un tiempo de 30 segundos sin que su robot ya no pueda recoger mas fichas, cuando denote avería que no le permita continuar o cuando la última ficha en el tablero sea liberada sobre un supuesto contenedor.

Habrà un juez tomando el tiempo de la prueba.

Iluminación

La iluminación será de luz indirecta colocada a no menos de 3 metros del tablero, de preferencia blanca tipo luz Neón, entre 500 y 850 luxes, que equivale a una luz interior normal.

Ejemplo de datos de pruebas

Ficha Posición

Ficha	Posición
1	(5.5 , 3.0)
2	(4.0 , 5.5)
3	(38.5 , 21.0)
4	(25.0 , 25.0)
5	(18.0 , 21.5)
6	(7.5 , 35.5)
7	(12.5 , 19.5)
8	(22.5 , 21.5)
9	(16.5 , 32.0)
10	(35.5 , 4.0)

Figura A.1ª. Ejemplo de la posición de las fichas

Nota: no se dará a conocer el color de la ficha ni la altura.

Ejemplo de Evaluación

DATOS DEL JURADO

	Concursantes	a	b	c	d	e	f	g	h	i
Reporte (0-10)		10	8	7	5	6	9	4	8	8
Exposición (0-10)		8	10	9	9	6	7	9	8	7
Originalidad (0-10)		6	9	8	6	8	7	7	9	4
Tiempo (min.)		8	11	3	9	8	6	5	7	2
Aciertos (piezas)		8	5	9	8	8	9	8	10	8
Calidad (0-10)		9	8	6	8	9	9	8	7	8
Presentación (0-10)		8	9	9	8	10	9	7	8	7

Figura A.1b Ejemplo de evaluación

PUNTUACIÓN POR COMPUTADORA SEGÚN LAS REGLAS

	Ptos.	Factor	a	b	c	d	e	f	g	h	i
Reporte	4	1.67	4	2.7	2	0.7	1.3	3.3	--	2.7	2.7
Excesión	4	2.5	2	4	3	3	--	1	3	2	1
Originalidad	14	2	5.6	14	11.2	5.6	11.2	8.4	8.4	14	--
Tiempo (min.)	30	3.33	10	--	26.7	6.7	10	16.7	20	13.3	30
Acertos (lineal)	20	2.5	12	--	15	12		16	12	20	12
Acertos (rotacional)	40	2.5					24				
Calidad	4	2.5	3	2	--	2	3	3	2	1	2
Presentación	4	2.5	2	3	3	2	4	3	1	2	1
Total	100	17.0	38.6	25.7	61.9	32	53.5	61.4	46.4	45	48.7

Figura A.1c Puntuación por computadora según las reglas

Nota: El factor se determina automáticamente al conocer la calificación máxima y mínima otorgada y permite abrir el espectro cuando la competencia está muy pareja, evitando discusiones de punto fino.

Con este método, y por dar un ejemplo, si todos los tiempos están entre 3 y 5 minutos, obtienen muchos puntos los que estuvieron cerca de los 3 minutos y muy pocos los que estuvieron en 5.

También si todos los brazos fueron de creación propia, uno fue comprado y otro fue intermedio, los de creación propia tendrán 14 puntos de ventaja sobre el comprado. Por otra parte si todos fueron originales, el menos original tendrá cero puntos y el más 14. Así también puede ganar uno que sólo puso 4 fichas si los demás pusieron 3 o 2, pero tiene mucho peso en puntos el que logre poner las 10, que es el objetivo principal, a pesar de que tarde un poco más y sea un poco menos original. Pero si tarda mucho o no es nada original, puede ser fuertemente penalizado y perder ante otro que logre colocar 9 fichas.

ANEXO 2 Codigos de Matlab®

GATOROBOT.M

```
%variables
fs=44100;
load tuturno
load sgato
load pierdes
turno= -1; %decide quien empieza
fichas=[[0,0,0];[0,0,0];[0,0,0]];
ganar=0;
valido=1;
xtiro=0;
ytiro=0;
tiros=0;
largo=30;
ancho=30;
%Pintargato
figure(15)
pintargato(largo,ancho)
% figure(1)
%quita el brazo
quitarbrazo
%toma tablero
Tomatablerogato;
tomatiro;

while ganar==0
%turno humano
if turno<0
while valido==1
%tirar
%quita el brazo
quitarbrazo;
wavplay(tuturno,fs)
sdad=input('Aprieta enter cuando tires');
tomatiro;
[C,I]=max(cambio);

%buscar en que cuadrante fue el tiro
%buscar columna
tiro=[mod(2+I,3)+1,ceil(I/3)];
col=tiro(1);
ren=tiro(2);
%revisar si fue valido el tiro
% xlabel(['col: ',num2str(col),'ren: ',num2str(ren),'turno: ',num2str(turno)])
if fichas(ren,col)==0
fichas(ren,col)=turno;
turno=-turno;
tiros=tiros+1;
break
end
end
else
[tiro,casilla]=aigato(fichas) ;
col=tiro(1);
ren=tiro(2);
fichas(ren,col)=turno;
```

```
turno=-turno;
tiros=tiros+1;

end

if turno>0
xlabel('turno jugador 1')
else
xlabel('turno jugador 2')
end

%pintar el tiro
[ex ey]=esquinacuadrante(largo,ancho,col,ren);
figure(15)
if turno>0
pintarcirculo(ex,ey,largo,ancho);
else
pintarcruz(ex,ey,largo,ancho);
pintarcruzrobotinvcinematica;
quitarbrazo;
pause(3)
tomatiro;
end

%revisar si gano
ganar= ganador(fichas);

if tiros ==9
break
end

end
if turno>0
xlabel('Algo muy raro pasó aquí');
else
xlabel(';Pierdes Pitufo!');
wavplay(pierdes,fs)
pintaR;
end

if tiros ==9 && ganar==0
xlabel('GATO');
wavplay(sgato,fs)
pintaG;
end

QUITARBRAZO.M

pos1=1070;
pos2=600;
poslb1=mod(real(pos1),256);
poshb1=floor(pos1/256);
arriba=1040;
rampa3=7;
```

```

%max(895)

poslb2=mod(real(pos2),256);
poshb2=floor(pos2/256);
rampa1=20;
rampa2=20;

co1= [33 83 67 1 rampa1 poslb1 poshb1 13];
co2= [33 83 67 3 rampa2 poslb2 poshb2 13];

fwrite(s,co1,'uint8');
fwrite(s,co2,'uint8');

pos3=arriba;
poslb3=mod(real(pos3),256);
poshb3=floor(pos3/256);
co3= [33 83 67 9 rampa3 poslb3 poshb3 13] ;
fwrite(s,co3,'uint8');

rampa1=0;
rampa2=0;
pause(2)

```

PINTARGATO.M

```

function pintargato(largo,ancho)
    color='k';
    clf;
    axis([0, largo, 0, ancho]);
    hold on;
    axis equal;
    axis manual;

    line([0,30],[10,10],'Color',color,'LineWidth',5);
    line([0,30],[20,20],'Color',color,'LineWidth',5);
    line([10,10],[0,30],'Color',color,'LineWidth',5);
    line([20,20],[0,30],'Color',color,'LineWidth',5);

    return

```

TOMATABLEROGATO.M

```

%Prender camara
vid = videoinput('winvideo',1);
set(vid,'ReturnedColorSpace','rgb');

esquina1=[157 42];
esquina2=[];
esquina3=[250 165];
esquina4=[];

%revisar camara
button=0;
itablero= getsnapshot(vid);
itablero=imcrop(itablero,[esquina1(1) esquina1(2)
esquina3(1)-esquina1(1) esquina3(2)-
esquina1(2)]);
tablerogs = rgb2gray(itablero);

colorblanco=max(max(tablerogs))-35;
colorgris=mean(mean(tablerogs));
colomegro=colorblanco-1;
blanco=0;
negro=0;

```

```

H = fspecial('disk',3);
tablerogs = imfilter(tablerogs,H,'replicate');

```

```

for r=1:1:size(tablerogs,1)
    for c=1:1:size(tablerogs,2)
        tablerogscontr(r,c)=colorgris;
        negro(r,c)=0;
        blanco(r,c)=0;
        if tablerogs(r,c)<colornegro
            negro(r,c)=60;
            tablerogscontr(r,c)=0;
        elseif tablerogs(r,c)>colorblanco
            blanco(r,c)=60;
            tablerogscontr(r,c)=255;
        end
    end
end

```

```

end
orillablanca = edge(blanco,'sobel');
orillanegro = edge(negro,'sobel');

```

```

r=15;

```

```

[yb,xb]=find(orillablanca);

```

```

buscacentrost;

```

```

%=====
%=====
%identificacion de los tiros
%=====
%=====
%se toma la foto del tablero vacio (o el color de
los tiros muy diferentes
%al delos tiros
%a partir de los cuadros se utiliza el arlonitmo de
buscar bordes para ver
%las lineas del tablero. se saca el centtro de estos
cuadrados
%para identificar el tiro se analizan 2 ciruclares
para identificar si son
%circulos o cruces con barrido de lineas
horizontales

```

```

%-----
% dividir el tablero en secciones
%-----

```

```

%revisar cada una de las secciones y almacenar
el tipo de caracter
%identificacion de caracter

```

```

tipoficha=idcaract(centrox(i4),centrosy(i4),linea,an
chocuadro,altocuadro)
% almacenar caracter en el tablero

```

```

imcas1=[[0,0];[0,0]];
imcas2=[[0,0];[0,0]];
imcas3=[[0,0];[0,0]];
imcas4=[[0,0];[0,0]];
imcas5=[[0,0];[0,0]];

```

```

imcas6=[[0,0];[0,0]];
imcas7=[[0,0];[0,0]];
imcas8=[[0,0];[0,0]];
imcas9=[[0,0];[0,0]];

```

TOMATIRO.M

```

%Prender camara
vid = videoinput('winvideo',1);
set(vid,'ReturnedColorSpace','rgb');

%revisar camara
button=0;
itablero= getsnapshot(vid);
itablero=imcrop(itablero,[esquina1(1) esquina1(2)
esquina3(1)-esquina1(1) esquina3(2)-
esquina1(2)]);
tablerogs = rgb2gray(itablero);

figure(1)
colormap(gray(256));
image(tablerogs);
colorblanco=max(max(tablerogs))-15;
colomegro=colorblanco-1;%50
colorgris=mean(mean(tablerogs));
blanco=0;
negro=0;

for r=1:1:size(tablerogs,1)
    for c=1:1:size(tablerogs,2)
        tablerogscontr(r,c)=colorgris;
        negro(r,c)=0;
        blanco(r,c)=0;
        if tablerogs(r,c)<colomegro
            negro(r,c)=60;
            tablerogscontr(r,c)=colomegro;
        elseif tablerogs(r,c)>colorblanco
            blanco(r,c)=60;
            tablerogscontr(r,c)=colorblanco;
        end
    end

end
orillablanca = edge(blanco,'sobel');
orillanegro = edge(negro,'sobel');

r=15;

centrosblancos=zeros(size(tablerogs,1),size(tablerogs,2));

%=====
%=====
%identificacion de los tiros
%=====
%=====
%se toma la foto del tablero vacio (o el color de
los tiros muy diferentes
%al delos tiros
%a partir de los cuadros se utiliza el arloritmo de
buscar bordes para ver

```

```

%las lineas del tablero. se saca el centtro de estos
cuadrados
%para identificar el tiro se analizan 2 ciruclares
para identificar si son
%circuitos o cruces con barrido de lineas
horizontales
imcas1o=imcas1;
imcas2o=imcas2;
imcas3o=imcas3;
imcas4o=imcas4;
imcas5o=imcas5;
imcas6o=imcas6;
imcas7o=imcas7;
imcas8o=imcas8;
imcas9o=imcas9;
%%%%%%%%%%
%%%%%%%%%%
i=1;
imcas1=imcrop(orillanegro,[centrosx(i)-
anchocuadro*.4,centrosy(i)-
altocuadro*.4,anchocuadro*.8,altocuadro*.8]);
% subplot(3,3,i);
% imshow(imcas1)
i=i+1;
imcas2=imcrop(orillanegro,[centrosx(i)-
anchocuadro*.4,centrosy(i)-
altocuadro*.4,anchocuadro*.8,altocuadro*.8]);
% subplot(3,3,i);
% imshow(imcas2)
i=i+1;
imcas3=imcrop(orillanegro,[centrosx(i)-
anchocuadro*.4,centrosy(i)-
altocuadro*.4,anchocuadro*.8,altocuadro*.8]);
% subplot(3,3,i);
% imshow(imcas3)
i=i+1;
imcas4=imcrop(orillanegro,[centrosx(i)-
anchocuadro*.4,centrosy(i)-
altocuadro*.4,anchocuadro*.8,altocuadro*.8]);
% subplot(3,3,i);
% imshow(imcas4)
i=i+1;
imcas5=imcrop(orillanegro,[centrosx(i)-
anchocuadro*.4,centrosy(i)-
altocuadro*.4,anchocuadro*.8,altocuadro*.8]);
% subplot(3,3,i);
% imshow(imcas5)
i=i+1;
imcas6=imcrop(orillanegro,[centrosx(i)-
anchocuadro*.4,centrosy(i)-
altocuadro*.4,anchocuadro*.8,altocuadro*.8]);
% subplot(3,3,i);
% imshow(imcas6)
i=i+1;
imcas7=imcrop(orillanegro,[centrosx(i)-
anchocuadro*.4,centrosy(i)-
altocuadro*.4,anchocuadro*.8,altocuadro*.8]);
% subplot(3,3,i);
% imshow(imcas7)
i=i+1;
imcas8=imcrop(orillanegro,[centrosx(i)-
anchocuadro*.4,centrosy(i)-
altocuadro*.4,anchocuadro*.8,altocuadro*.8]);
% subplot(3,3,i);
% imshow(imcas8)
i=i+1;

```

```

imcas9=imcrop(orillanegro,[centrosx(i)-
anchocuadro*.4,centrosy(i)-
altocuadro*.4,anchocuadro*.8,altocuadro*.8]);
% subplot(3,3,i);
% imshow(imcas9)
% i=i+1;

%revisar cada una de las secciones y almacenar
el tipo de caracter
%identificacion de caracter
%
tipoficha=idcaract(centrox(i4),centrosy(i4),linea,an
chocuadro,altocuadro)
% almacenar caracter en el tablero
% fichas(columna,renglon)=tipoficha;
% end
tolerancia=.02;
cambio=[0,0,0,0,0,0,0,0,0];
if mean(mean(imcas1))<mean(mean(imcas1o))-
tolerancia ||
mean(mean(imcas1))>mean(mean(imcas1o))+tole
rancia
cambio(7)=1;
end
if mean(mean(imcas2))<mean(mean(imcas2o))-
tolerancia ||
mean(mean(imcas2))>mean(mean(imcas2o))+tole
rancia
cambio(8)=1;
end
if mean(mean(imcas3))<mean(mean(imcas3o))-
tolerancia ||
mean(mean(imcas3))>mean(mean(imcas3o))+tole
rancia
cambio(9)=1;
end
if mean(mean(imcas4))<mean(mean(imcas4o))-
tolerancia ||
mean(mean(imcas4))>mean(mean(imcas4o))+tole
rancia
cambio(4)=1;
end
if mean(mean(imcas5))<mean(mean(imcas5o))-
tolerancia ||
mean(mean(imcas5))>mean(mean(imcas5o))+tole
rancia
cambio(5)=1;
end
if mean(mean(imcas6))<mean(mean(imcas6o))-
tolerancia ||
mean(mean(imcas6))>mean(mean(imcas6o))+tole
rancia
cambio(6)=1;
end
if mean(mean(imcas7))<mean(mean(imcas7o))-
tolerancia ||
mean(mean(imcas7))>mean(mean(imcas7o))+tole
rancia
cambio(1)=1;
end
if mean(mean(imcas8))<mean(mean(imcas8o))-
tolerancia ||
mean(mean(imcas8))>mean(mean(imcas8o))+tole
rancia
cambio(2)=1;

```

```

end
if mean(mean(imcas9))<mean(mean(imcas9o))-
tolerancia ||
mean(mean(imcas9))>mean(mean(imcas9o))+tole
rancia
cambio(3)=1;
end

```

```
cambio;
```

PINTARCIRCULO.M

```

function pintarcirculo(ex,ey,largo,ancho)

color='b';

rectangle('Position',[ex-largo/12,ey-
ancho/12,largo/6,ancho/6],'Curvature',[1,1],'LineWi
dth',5,'EdgeColor',color)

return

```

PINTARCRUZ.M

```

function pintarcruz(ex,ey,largo,ancho)

color='r';

line([ex-largo/7,ex+largo/7],[ey-
ancho/7,ey+ancho/7],'Color',color,'LineWidth',5);
line([ex-largo/7,ex+largo/7],[ey+ancho/7,ey-
ancho/7],'Color',color,'LineWidth',5);

return

```

ESQUINACUADRANTE.M

```

function
[ex,ey]=esquinacuadrante(largo,ancho,col,ren)
anchcol=largo/3
anchoreg=ancho/3

ex=largo/6+(anchcol)*(col-1)
ey=ancho/6+(anchoreg)*(ren-1)

return

```

AIGATO.M

```

%aigato v1.0
function [tiro,casilla]=aigato(tabl)

num(1)=abs(tabl(1,1))+abs(tabl(1,2))+abs(tabl(1,3)
);
num(2)=abs(tabl(2,1))+abs(tabl(2,2))+abs(tabl(2,3)
);
num(3)=abs(tabl(3,1))+abs(tabl(3,2))+abs(tabl(3,3)
);
num(4)=abs(tabl(1,1))+abs(tabl(2,2))+abs(tabl(3,3)
);
num(5)=abs(tabl(1,1))+abs(tabl(2,1))+abs(tabl(3,1)
);

```

```

num(6)=abs(tabl(1,2))+abs(tabl(2,2))+abs(tabl(3,2))
);
num(7)=abs(tabl(1,3))+abs(tabl(2,3))+abs(tabl(3,3))
);
num(8)=abs(tabl(3,1))+abs(tabl(2,2))+abs(tabl(1,3))
);

```

```

l(1)=tabl(1,1)+tabl(1,2)+tabl(1,3);
l(2)=tabl(2,1)+tabl(2,2)+tabl(2,3);
l(3)=tabl(3,1)+tabl(3,2)+tabl(3,3);
l(4)=tabl(1,1)+tabl(2,2)+tabl(3,3);
l(5)=tabl(1,1)+tabl(2,1)+tabl(3,1);
l(6)=tabl(1,2)+tabl(2,2)+tabl(3,2);
l(7)=tabl(1,3)+tabl(2,3)+tabl(3,3);
l(8)=tabl(3,1)+tabl(2,2)+tabl(1,3);

```

```

if max(num)>0
for i=1:1:8
if l(i)==num(i)
if l(i)<2
f(i)=l(i)^2+.5;
else
f(i)=10
end
else
f(i)=0;
end
if -l(i)==num(i)
c(i)=l(i)^2+.5;
else
c(i)=0;
end
end

```

```

end
f;
c;
puntos(1)=abs(abs(tabl(1,1))-
1)*(c(1)+c(4)+c(5)+f(1)+f(4)+f(5));%(1,1)
%evita jugada principiante
puntos(2)=abs(abs(tabl(1,2))-
1)*((c(1)+c(6)+f(1)+f(6))+(-tabl(1,1)-
tabl(3,3)));%(1,2)
puntos(3)=abs(abs(tabl(1,3))-
1)*(c(1)+c(7)+c(8)+f(1)+f(7)+f(8));%(1,3)
puntos(4)=abs(abs(tabl(2,1))-
1)*(c(2)+c(5)+f(2)+f(5));%(2,1)
puntos(5)=abs(abs(tabl(2,2))-
1)*(c(2)+c(4)+c(6)+c(8)+f(2)+f(4)+f(6)+f(8));%(2,2)
puntos(6)=abs(abs(tabl(2,3))-
1)*((c(2)+c(7)+f(2)+f(7))+(-tabl(3,1)-
tabl(1,3)));%(2,3);
puntos(7)=abs(abs(tabl(3,1))-
1)*(c(3)+c(5)+c(8)+f(3)+f(5)+f(8));%(3,1);
puntos(8)=abs(abs(tabl(3,2))-
1)*(c(3)+c(6)+f(3)+f(6));%(3,2);
puntos(9)=abs(abs(tabl(3,3))-
1)*(c(3)+c(4)+c(7)+f(3)+f(4)+f(7));%(3,3);
puntos
disp(['puntos' num2str(puntos)])
%se puede cambiar a un for para generar
variedad
[C,l] = max(puntos)
%toda esta parte
if C==0
puntos(1)=abs(abs(tabl(1,1))-1);%(1,1)

```

```

puntos(2)=abs(abs(tabl(1,2))-1);%(1,2)
puntos(3)=abs(abs(tabl(1,3))-1);%(1,3)
puntos(4)=abs(abs(tabl(2,1))-1);%(2,1)
puntos(5)=abs(abs(tabl(2,2))-1);%(2,2)
puntos(6)=abs(abs(tabl(2,3))-1);%(2,3)
puntos(7)=abs(abs(tabl(3,1))-1);%(3,1);
puntos(8)=abs(abs(tabl(3,2))-1);%(3,2);
puntos(9)=abs(abs(tabl(3,3))-1);%(3,3);

```

```

end
[C,l] = max(puntos);
%convierto el indice a un valor de ij
l;
 tiro=[mod(2+l,3)+1,ceil(l/3)];
 casilla=l;
 return

```

```

else
 %para darle variedad al inicio
 r=round(rand()*5);
 l=max(r*2-1,1);
 tiro=[mod(2+l,3)+1,ceil(l/3)];
 casilla=l;
 return
end

```

GANADOR

```

function ganar=ganador(fichas)

```

```

l1=abs(fichas(1,1)+fichas(1,2)+fichas(1,3));
l2=abs(fichas(2,1)+fichas(2,2)+fichas(2,3));
l3=abs(fichas(3,1)+fichas(3,2)+fichas(3,3));
l4=abs(fichas(1,1)+fichas(2,2)+fichas(3,3));
l5=abs(fichas(1,1)+fichas(2,1)+fichas(3,1));
l6=abs(fichas(1,2)+fichas(2,2)+fichas(3,2));
l7=abs(fichas(1,3)+fichas(2,3)+fichas(3,3));
l8=abs(fichas(3,1)+fichas(2,2)+fichas(1,3));

```

```

if
max(l1,max(l2,max(l3,max(l4,max(l5,max(l6,max(l7,l8))))))
)==3
ganar=1;
else
ganar=0;
end
return

```

PINTARCRUZROBOTCINEMATICA. M

```

%Pintarcruzrobot
clear trazox trazoy trazox2 trazoy2

```

```

[esquinasx,esquinasy]=esquinacuadranterobot(casilla);

```

```

h=50;
rampa1=15;
rampa2=0;
rampa3=15;

```

```

constantesgato

%interpolacion lineal
for i=1:2:3
xini=esquinasx(i);
yini=esquinasy(i);
xfin=esquinasx(i+1);
yfin=esquinasy(i+1);

res1=(xini-xfin)/h;
res2=(yini-yfin)/h;

xact=xini;
yact=yini;

for j=1:1:h

xcomp=xact+abs(offsetx);
ycomp=yact+abs(offsety);

thetat=acos(xcomp/sqrt(xcomp^2+ycomp^2));
theta1=-acos(-((l2^2-l1^2-
(xcomp^2+ycomp^2))/(2*l1*sqrt(xcomp^2+ycomp^
2))))+thetat;
theta2=-acos((xcomp^2+ycomp^2-l1^2-
l2^2)/(2*l1*l2));

angulo1=((theta1*180/pi)*(pendiente1)+ordenada1
);
pos1=angulo1;
angulo2=((-
theta2*180/pi)*(pendiente2)+ordenada2);
pos2=angulo2;
poslb1=mod(real(pos1),256);
poshb1=floor(pos1/256);
poslb2=mod(real(pos2),256);
poshb2=floor(pos2/256);

%mandar los valores a los servos
co1= [33 83 67 1 rampa1 poslb1 poshb1 13];
co2= [33 83 67 3 rampa2 poslb2 poshb2 13];

fwrite(s,co1,'uint8')
fwrite(s,co2,'uint8')

if i==1

trazox(j)=xact;
trazoy(j)=yact;
else
trazox2(j)=xact;
trazoy2(j)=yact;

end
xact=xact-res1;
yact=yact-res2;

%baja plumon

```

```

% pause(.2)
if j==1
pause(3)
pos3=abajo;
poslb3=mod(real(pos3),256);
poshb3=floor(pos3/256);
co3= [33 83 67 9 rampa3 poslb3 poshb3 13] ;
fwrite(s,co3,'uint8')
pause(3)
rampa1=0;
end

end
%levantar el plumon
rampa3=15;
pos3=arriba;
poslb3=mod(real(pos3),256);
poshb3=floor(pos3/256);
co3= [33 83 67 9 rampa3 poslb3 poshb3 13] ;
fwrite(s,co3,'uint8')
pause(3)
end

% figure(1)
hold on
% plot( trazox,trazoy,'x', trazox2,trazoy2,'o')
% plot(esquinasx,esquinasy)
hold off
% end

```

ESQUINACUADRANTEROBOT.M

```

function [ex,ey]=esquinacuadranterobot(indice)
anchomediacruz=1.5;
altomediacruz=2;

constantesgato

ex(1)=centrosxi(indice)-anchomediacruz;
ex(2)=centrosxi(indice)+anchomediacruz;
ex(3)=centrosxi(indice)+anchomediacruz;
ex(4)=centrosxi(indice)-1.5*anchomediacruz;

ey(1)=centrosyi(indice)+altomediacruz;
ey(2)=centrosyi(indice)-altomediacruz;
ey(3)=centrosyi(indice)+altomediacruz;
ey(4)=centrosyi(indice)-altomediacruz;

ex=ex-1;
return

```

CORRERCOMPLETO.M

```

%abriserial;
vid = videoinput('winvideo', 1);
set(vid,'ReturnedColorSpace','rgb');
pos1=1000;
pos2=300;
poslb1=mod(real(pos1),256);
poshb1=floor(pos1/256);

%max(895)

```

```

poslb2=mod(real(pos2),256);
poshb2=floor(pos2/256);
rampa1=5;
rampa2=5;

```

```

co1= [33 83 67 1 rampa1 poslb1 poshb1 13]
co2= [33 83 67 3 rampa2 poslb2 poshb2 13]

```

```

fwrite(s,co1,'uint8')
fwrite(s,co2,'uint8')
pause(3)
Tomatablerofichas2;
pixcm;
rutas;
moverpiestro;
%fclose(s)

```

ABRISERIAL.M

```

s = serial ('com4');
set(s,'BaudRate',2400);
fopen(s);
s

```

TOMATABLEROFICHAS2.M

```

%Prender camara
%vid = videoinput('winvideo',1);
%set(vid,'ReturnedColorSpace','rgb');
esquina1=[90 16];
esquina2=[];
esquina3=[307 234];
esquina4=[];

```

```

%revisar camara
button=0;
itablero= getsnapshot(vid);
image(itablero);
% xlabel(['dar click boton izquierdo para continuar
derecho para volver a tomar'])
itablero=imcrop(itablero,[esquina1(1) esquina1(2)
esquina3(1)-esquina1(1) esquina3(2)-
esquina1(2)]);
%itablero=imcrop(itablero,[96 75 260 250]);
tablerogs = rgb2gray(itablero);

```

```

figure(1)
colormap(gray(256));
image(tablerogs);
colorblanco=max(max(tablerogs))-30;
colomegro=min(min(tablerogs))+60;%50
colorgris=mean(mean(tablerogs));
blanco=0;
negro=0;

```

```

H = fspecial('disk',3);
tablerogs = imfilter(tablerogs,H,'replicate');
figure(1)
colormap(gray(256));
image(tablerogs);

```

```

for r=1:1:size(tablerogs,1)
    for c=1:1:size(tablerogs,2)

```

```

        tablerogscontr(r,c)=colorgris;
        negro(r,c)=0;
        blanco(r,c)=0;
        if tablerogs(r,c)<colornegro
            negro(r,c)=60;
            tablerogscontr(r,c)=colornegro-50;
        elseif tablerogs(r,c)>colorblanco
            blanco(r,c)=60;
            tablerogscontr(r,c)=colorblanco+30;
        end
    end
end

```

```

end
figure(2)
image(blanco);
axis([0 size(tablerogs,2) 0 size(tablerogs,1)]);
figure(3)
image(negro);
axis([0 size(tablerogs,2) 0 size(tablerogs,1)]);

```

```

figure(4)
image(tablerogscontr);
axis([0 size(tablerogs,2) 0 size(tablerogs,1)]);

```

```

colormap(gray(256));
figure(5)
orillablanca = edge(blanco,'sobel');
orillanegro = edge(negro,'sobel');

```

```

imshow(orillablanca);
figure(6)
imshow(orillanegro);
r=15;

```

```

[xb,yb]=find(orillablanca);
[centrob,Ub,objFcnb] = fcm([xb,yb],5);

```

```

[xn,yn]=find(orillanegro);
[centron,Un,objFcn] = fcm([xn,yn],5);

```

```

hold on
figure(5)
centrosblancos=zeros(size(tablerogs,1),size(tablerogs,2));
for r=1:1:size(tablerogs,1)
    for c=1:1:size(tablerogs,2)
        for i=1:1:5

```

```

            if(and(round(centrob(i,2))==c,round(centrob(i,1))==r))

```

```

                centrosblancos(r,c)=1;

```

```
            end

```

```
        end

```

```
    end

```

```

end
centrosblancos=logical(centrosblancos);
imshow(centrosblancos+orillablanca);

```

```

figure(6)
centrosnegros=zeros(size(tablerogs,1),size(tablerogs,2));
for r=1:1:size(tablerogs,1)
    for c=1:1:size(tablerogs,2)
        for i=1:1:5

```

```

if(and(round(centron(i,2))==c,round(centron(i,1))==
r))
    centrosnegros(r,c)=1;
end
end
end
end
centrosnegros=logical(centrosnegros);
imshow(centrosnegros+orillanegro);
hold off

conv=(mean(size(tablerogs,1),size(tablerogs,2)))/4
0;
pixel0x=round(size(tablerogs,2)/2);
pixel0y=size(tablerogs,1);

fichasx=transpose([centrob(:,2);centron(:,2)]);
fichasy=transpose([centrob(:,1);centron(:,1)]);

for i=1:1:10
    fichasx(i)=(fichasx(i)-pixel0x)/conv+2
    fichasy(i)=(-fichasy(i)+pixel0y)/conv
end

```

RUTAS.M

```

%trazado de rutas
colorfichas=[1 1 1 1 1 0 0 0 0];
cajasx=[-27 27];
cajasy=[20 20];
posinix=0;
posiniy=20;

datosx=[fichasx,cajasx,posinix];
datosy=[fichasy,cajasy,posiniy];
distancias=calcdist(datosx,datosy);

xblancas=zeros(1,5);
ynegras=zeros(1,5);
xblancas=zeros(1,5);
ynegras=zeros(1,5);
j=1;
k=1;
for i=1:1:10
    if colorfichas(i)==1
        xblancas(j)=fichasx(i);
        yblancas(j)=fichasy(i);
        j=j+1;
    else
        xnegras(k)=fichasx(i);
        ynegras(k)=fichasy(i);
        k=k+1;
    end
end

datosbx=[xblancas,cajasx,posinix];
datosby=[yblancas,cajasy,posiniy];
distanciasb=calcdist(datosbx,datosby);
datosnx=[xnegras,cajasx,posinix];
datosny=[ynegras,cajasy,posinix];

```

```

distanciasn=calcdist(datosnx,datosny);

mdn=mean(distanciasn,1)
mdb=mean(distanciasb,1)

disccol=mdn(6)-mdb(6)+mdb(7)-mdn(7);
%cero cuadro es la caja de -20x la negra (6)
if disccol>0
    cajanegrax=27;
    cajablanca=-27;
    colnegra=12;
    colblanca=11;
else
    cajanegrax=-27;
    cajablanca=27;
    colnegra=11;
    colblanca=12;
end

distini=distancias;
distini(13,:)=[];
distini(12,:)=[];
distini(11,:)=[];
ordenpos=zeros(20,2);

[Y,I]=min(distini(:,13))
distini(I,:)=129;
ordenpos(1,1)=fichasx(I);
ordenpos(1,2)=fichasy(I);
if colorfichas(I)==1
    ordenpos(2,1)=cajablanca;
    ordenpos(2,2)=20;
    ultficha=1
else
    ordenpos(2,1)=cajanegrax;
    ordenpos(2,2)=20;
    ultficha=0
end

for i=3:2:19
    if ultficha==1
        colc=colblanca;
    else
        colc=colnegra;
    end

[Y,I]=min(distini(:,colc))
distini(I,:)=129;
ordenpos(i,1)=fichasx(I);
ordenpos(i,2)=fichasy(I);
if colorfichas(I)==1
    ordenpos(i+1,1)=cajablanca;
    ordenpos(i+1,2)=20;
    ultficha=1
else
    ordenpos(i+1,1)=cajanegrax;
    ordenpos(i+1,2)=20;
    ultficha=0
end

end
figure(1)

```



```

hold off

hold on
plot([0;ordenpos(:,1)],[20;ordenpos(:,2)],xnegras,y
negras,'x',xblancas,yblancas,'o',datosx,datosy,'v')
for i=1:2:19
    text(ordenpos(i,1)+1,ordenpos(i,2),
num2str(ceil(i/2)))
end

hold off

```

MOVERPDIESTRO.M

```

%%Salida serial desactivada!

%Inicialisaciones y constantes
%=====
=====
%limpio variables
clf;

%parametros
kx=1.;
ky=1.;
rampa1=5;
rampa2=0;
rampa3=7;
pendiente1=-4.38401298719580;
ordenada1=1031
pendiente2=6.36145227449997
ordenada2=230

arriba=1040
soltar=1120
abajo=250

%-----
%Constnantes
deg=2*pi/360;
l1=28;
l2=27.5;
%dibujos fijos
%marco del tablero
xc=[-20,-20,20,20,-20];
yc=[0,40,40,0,0];
%cruz de origen
offsetx=-1;
offsety=-10;
xo=[offsetx-5,offsetx+5,offsetx,offsetx,offsetx];
yo=[offsety,offsety,offsety,offsety-5,offsety+5];

%condiciones iniciales
% condiciones del punto
xfin=ordenpos(1,1);
yfin=ordenpos(1,2);

xinst=0;
yinst=20;

```

```

xcomp=xinst-abs(offsetx);
ycomp=yinst+abs(offsety);

theta1=acos(xcomp/sqrt(xcomp^2+ycomp^2));
theta1=-acos(-((l2^2-l1^2-
(xcomp^2+ycomp^2))/(2*l1*sqrt(xcomp^2+ycomp^
2))))+theta1;
theta2=-acos((xcomp^2+ycomp^2-l1^2-
l2^2)/(2*l1*l2));

x1=l1*cos(theta1)-offsetx;
y1=l1*sin(theta1)+offsety;
x2=x1+l2*cos(-theta2+theta1);
y2=y1+l2*sin(-theta2+theta1);

%rutina de funcionamiento
%=====
=====
for j=1:1:21 %repite el problema muchas veces
con puntos al azar

    plot(xfin,yfin,'o',xc,yc,xo,yo);
    axis([-32.5,32.5,-15,50]); %minimos y maximos
de la ventana
    axis square;
    axis manual;

    brazo1=line([offsetx,x1],[offsety,y1]);
    set(brazo1,'Color',[1 0 0]);
    set(brazo1,'LineWidth',4);
    set(brazo1,'LineStyle','-');

    brazo2=line([x1,x2],[y1,y2]);
    set(brazo2,'Color',[0 0 1]);
    set(brazo2,'LineWidth',4);
    set(brazo2,'LineStyle','-');

    brazo3=line([offsetx,xinst],[offsety,yinst]);
    set(brazo3,'Color',[0 1 0]);
    set(brazo3,'LineWidth',1);
    set(brazo3,'LineStyle','-');
    %convierto los valores a centimetros

% xprojo=centror(1)*conv-pixel0x
% yprojo=-centror(2)*conv+pixel0y

distanciay=yfin-yinst;
distanciac=xfin-xinst;
%rutina de movimiento
%-----
while(distanciay^2 > 0.01 || distanciac^2 > 0.01);
    %
    i=i+1;
    %logica difusa
    %-----

%defusificacion
oy= distanciac*ky;
ox=distanciac*kx;

%-----

```

```

%calculo de angulos mediante el uso de
cinematica inversa

xinst=xinst+ox;
yinst=yinst+oy;

xcomp=xinst+abs(offsetx);
ycomp=yinst+abs(offsety);

thetat=acos(xcomp/sqrt(xcomp^2+ycomp^2));
theta1=-acos(-((l2^2-l1^2-
(xcomp^2+ycomp^2))/(2*l1*sqrt(xcomp^2+ycomp^
2))))+thetat;
theta2=-acos((xcomp^2+ycomp^2-l1^2-
l2^2)/(2*l1*l2));

x1=l1*cos(theta1)+offsetx;
y1=l1*sin(theta1)+offsety;
x2=x1+l2*cos((-theta2+theta1));
y2=y1+l2*sin((-theta2+theta1));

%meter la conversion de la thetas a valores del los
servos
%pendiente1 ordenada1
angulo1=((theta1*180/pi)*(pendiente1)+ordenada1
)
pos1=angulo1
%pendiente1 ordenada1
angulo2=(-
theta2*180/pi)*(pendiente2)+ordenada2)
pos2=angulo2
xlabel(['pos x : ',num2str(xfin),'pos y :
',num2str(yfin)])
%mandar los valores a los sevos

poslb1=mod(real(pos1),256);
poshb1=floor(pos1/256);

%max(895)

poslb2=mod(real(pos2),256);
poshb2=floor(pos2/256);

co1= [33 83 67 1 rampa1 poslb1 poshb1 13]
co2= [33 83 67 3 rampa2 poslb2 poshb2 13]

fwrite(s,co1,'uint8')
fwrite(s,co2,'uint8')

%dibujar grafica
%_____

set(brazo1,'XData',[offsetx,x1]);
set(brazo1,'YData',[offsety,y1]);

set(brazo2,'XData',[x1,x2]);
set(brazo2,'YData',[y1,y2]);
set(brazo3,'XData',[offsetx,xinst]);
set(brazo3,'YData',[offsety,yinst]);

```

```

pause(.2)

%actualizo la distancia

distanciax=xfin-xinst;
distanciay=yfin-yinst;
%distanciax=xfin-xprojo;
% distanciay=yfin-yprojo;

end
text(1,1.5,';;Alcanzado!!!');

pause(1)
if j/2 == round(j/2)
% xlabel('inpar')
pos3=abajo
poslb3=mod(real(pos3),256);
poshb3=floor(pos3/256);
co3= [33 83 67 9 rampa3 poslb3 poshb3 13]
fwrite(s,co3,'uint8')
pause(1)
pos3=arriba
poslb3=mod(real(pos3),256);
poshb3=floor(pos3/256);
co3= [33 83 67 9 rampa3 poslb3 poshb3 13]
fwrite(s,co3,'uint8')

else

pos3=soltar
poslb3=mod(real(pos3),256);
poshb3=floor(pos3/256);
co3= [33 83 67 9 rampa3 poslb3 poshb3 13]
fwrite(s,co3,'uint8')
pause(1)
% xlabel('par')

end
pause(1);
%un segundo para celebrar la victoria y luego a
buscar un nuevo punto
%al azar:
if j<=20
xfin=ordenpos(j,1)
yfin=ordenpos(j,2)
end

end

return

```