



La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la “ESCUELA POLITÉCNICA NACIONAL” bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia si mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN SISTEMA DE CONTROL PARA UN BRAZO ROBÓTICO MEDIANTE ADQUISICIÓN Y PROCESAMIENTO DE IMÁGENES

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y CONTROL**

JUAN CARLOS BASANTES ORTIZ

jc_basantes@hotmail.com

FREDDY RICARDO TORRES TUFÍÑO

ricardotorres1976@latinmail.com

DIRECTOR: Dr. HUGO BANDA GAMBOA

hbanda@ieee.org

Quito, Febrero 2009

DECLARACIÓN

Nosotros, Juan Carlos Basantes Ortiz y Freddy Ricardo Torres Tufiño, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Juan Carlos Basantes Ortiz

Freddy Ricardo Torres Tufiño

CONTENIDO

RESUMEN	8
PRESENTACIÓN	10
 Capítulo 1	
PLANTEAMIENTO DEL PROYECTO	12
1.1. PROCESAMIENTO DE IMAGEN.....	12
1.1.1. OBJETIVO	12
1.1.2. VARIABLES QUE INFLUYEN EN EL PROCESO.....	12
1.1.3 PLANTEAMIENTO DEL PROBLEMA	13
1.1.4 SOLUCIÓN PROPUESTA.....	14
1.2 DESARROLLO DE UN MANIPULADOR ROBÓTICO.....	15
1.2.1 OBJETIVO	15
1.2.2 VARIABLES QUE INFLUYEN EN EL PROCESO.....	15
1.2.3. PLANTEAMIENTO DEL PROBLEMA	16
1.2.4. SOLUCIÓN PROPUESTA.....	17
 Capítulo 2	
MARCO TEÓRICO DEL PROCESAMIENTO DE IMAGEN	20
2.1 INTRODUCCIÓN	20
2.2 PROCESAMIENTO DIGITAL DE IMAGEN.....	20
2.3 IMÁGENES DIGITALES	22
2.4 ADQUISICIÓN DE DATOS	22
2.5 ETAPAS DE ANÁLISIS DE IMAGEN Y RECONOCIMIENTO DE FORMAS.....	23
2.5.1. PREPROCESAMIENTO	24
2.5.1.1 Representación de Imágenes con Niveles de Gris.....	24
2.5.1.2 Representación de Imágenes Binarias	25
2.5.2 SEGMENTACIÓN.....	25
2.5.2.1 Simplificación.....	26
2.5.2.2 Extracción de Características.....	26
2.5.2.3. Decisión.....	27

2.5.3. METODOLOGÍAS DE SEGMENTACIÓN	27
2.5.3.1. Segmentación Basada en Características	27
2.5.3.2. Segmentación por Detección de Transiciones	28
2.5.3.2.1. <i>Filtros Detectores de Bordes</i>	29
2.5.3.2.2. <i>Etiquetado de una Imagen</i>	29
2.5.3.2.3. <i>Etiquetado de Imágenes Binarias</i>	30
2.5.3.3. Segmentación Basada en Modelos	30
2.5.3.3.1. <i>La Transformada de Hough (TH)</i>	30
2.5.3.3.2. <i>Definición de la Transformada de Hough</i>	31
2.5.3.3.3. <i>Técnica Para Detectar Segmentos Rectos en Imágenes de Grises</i>	35
2.5.3.3.4. <i>Algoritmo de la TH (Imagen binaria)</i>	36
2.5.3.3.5. <i>Transformada de Hough de Puntos Alineados</i>	36
2.5.3.3.6. <i>Método General para Implementar la TH</i>	37
2.5.4. SEGMENTACIÓN BASADA EN MORFOLOGÍA MATEMÁTICA	38
2.5.4.1. Operaciones Morfológicas	38
2.5.4.1.1. <i>Erosión y dilatación</i>	38
2.5.4.1.2. <i>Relleno de Región (Region Filling)</i>	39
2.5.4.1.3. <i>Apertura Y Cierre</i>	39
2.5.4.1.4. <i>Adelgazamiento</i>	39
2.5.5. INTRODUCCIÓN AL ENTORNO MATLAB	40
2.5.5.1. Imágenes RGB	40
2.5.5.2. Imágenes Binarias	40
2.5.6. COMANDOS DE MATLAB UTILIZADOS PARA EL PROCESAMIENTO DE IMÁGENES.	41
2.5.6.1. Funciones Para la Extracción de Bordes	42
2.5.6.2. Operaciones Morfológicas	43
2.5.6.3. Operaciones Basadas en Objetos	45

Capítulo 3

MARCO TEORICO DEL MANIPULADOR ROBÓTICO	47
3.1. CONCEPTOS Y DEFINICIONES	47
3.2. CLASIFICACIÓN Y MORFOLOGÍA DEL ROBOT	49
3.2.1. CLASIFICACIÓN DEL ROBOT	49

3.2.2.	MORFOLOGÍA DEL ROBOT.....	49
3.2.3.	TIPOS DE ARTICULACIONES.....	50
3.2.4.	ELEMENTOS EFECTORES TERMINALES	51
3.3.	MOTORES CON SERVOMECANISMO	52
3.3.1.	SERVO ANALÓGICO PARA MODELISMO	53
3.3.2.	CONTROLADOR ELECTRÓNICO	54
3.4.	MODELO CINEMÁTICO DE LOS MANIPULADORES.....	56
3.4.1.	EL PROBLEMA CINEMÁTICO DIRECTO.....	57
3.4.1.3.	Algoritmo de Denavit-Hartenberg para la obtención del modelo matemático cinemático directo:	63
3.4.2.	EL PROBLEMA CINEMÁTICO INVERSO.....	65

Capítulo 4

	DESARROLLO DEL SISTEMA DE ADQUISICIÓN Y PROCESAMIENTO DE IMAGEN	69
4.1.	DIAGRAMA DE IMPLEMENTACIÓN DEL SISTEMA DE ADQUISICIÓN DE IMAGEN	69
4.2.	ADQUISICIÓN DE DATOS	70
4.3.	PREPROCESAMIENTO	71
4.3.1.	ESCALA DE GRISES.....	72
4.3.2.	BINARIZACIÓN.....	73
4.3.3.	FILTRO MORFOLÓGICO	75
4.3.3.1.	Dilatación y Erosión	77
4.4.	EXTRACCIÓN DE CARACTERÍSTICAS.....	78
4.5.	SELECCIÓN DE CARACTERÍSTICAS	80
4.6.	CLASIFICACIÓN.....	81

Capítulo 5

	DESARROLLO DEL MANIPULADOR CON VISIÓN ROBÓTICA	87
5.1.	DISEÑO E IMPLEMENTACIÓN DE LA ESTRUCTURA	87
5.1.1.	SERVOMOTORES SELECCIONADOS	88
5.1.2.	ESTRUCTURA MECÁNICA	89
5.1.3.	FUENTES DE ALIMENTACIÓN.....	94

5.2	DESARROLLO CINEMÁTICO DEL BRAZO ROBÓTICO	95
5.3	DESARROLLO DEL SOFTWARE DE CONTROL	105
5.3.1	Etapa de Adquisición	105
5.3.2	Etapa de Procesamiento y Cálculo	107
5.3.3	Etapa de Posicionamiento	112
5.3.4	Etapa de Transmisión	118

Capítulo 6

	PRUEBAS Y RESULTADOS	120
6.1	CALIBRACIÓN Y PRUEBAS DEL SISTEMA DE ADQUISICIÓN DE IMÁGENES	120
6.1.1	SELECCIÓN DE CÍRCULOS	123
6.1.2	SELECCIÓN DE CUADRADOS	124
6.1.3	SELECCIÓN DE TRIÁNGULOS	125
6.1.4	CONDICIONES GENERALES PARA LA TRANSFORMADA DE HOUGH	126
6.2	CALIBRACIÓN Y PRUEBAS DEL SISTEMA DE POSICIONAMIENTO DEL BRAZO ROBÓTICO	129

Capítulo 7

	CONCLUSIONES Y RECOMENDACIONES	135
7.1	CONCLUSIONES	135
7.2	RECOMENDACIONES	139
	REFERENCIAS BIBLIOGRÁFICAS	140
	ANEXOS	144

RESUMEN

Desde sus inicios, la humanidad ha buscado el desarrollo de herramientas y maquinarias que faciliten sus tareas sobre todo manuales y repetitivas. Con la revolución industrial y la producción masiva de bienes en las fábricas, se llega a la automatización de procesos y al desarrollo de las tecnologías informáticas que revolucionan la industria a pasos acelerados.

Los sistemas robotizados modernos, empezaron a sustituir a los humanos en las cadenas de montaje a partir de la década de 1970. La mayoría de los robots son diseñados para efectuar trabajos repetitivos que presenten dificultad, peligro o que impliquen un agotamiento físico para los humanos.

En la actualidad, los sistemas industriales requieren con más frecuencia la implementación de sistemas automatizados más allá de las operaciones mecánicas, desarrollando tecnologías que reemplacen otros sentidos humanos como la visión y el tacto.

Las tecnologías de visión robótica artificial se utilizan actualmente en sistemas de control de calidad, clasificación y paletizado, entre muchos otros en los cuales la robótica y la informática se fusionan para la optimización de los procesos productivos.

En su estructura global, el presente proyecto constituye una aplicación tecnológica de un sistema de visión robótica orientado al control de robots manipuladores. Se ha desarrollado un sistema completo para la aplicación de técnicas de captura y procesamiento de imágenes, y la implementación de un prototipo didáctico de un brazo robótico, similar a los utilizados en la industria moderna.

Con el presente proyecto se logra una visión más amplia de los alcances potenciales que se pueden obtener al integrar diferentes disciplinas temáticas como son el procesamiento de imágenes y la robótica aplicada, como un aporte a

la profundización del estudio de éstas y otras ciencias relacionadas con la automatización.

PRESENTACIÓN

En este proyecto se presenta el proceso de diseño y construcción de un sistema de control automático en lazo abierto para un brazo robótico, mediante la captura y procesamiento de una imagen en la que se reconocen los patrones geométricos de piezas colocadas al azar en una plataforma "A" y trasladar los mismos a una plataforma "B" en un orden predeterminado.

En el Capítulo 1 se realiza un estudio del caso para el proyecto planteado, subdividiendo el proyecto en dos temas principales: la captura y procesamiento de la imagen y el desarrollo del manipulador robótico. Para ambas temáticas se establecen los objetivos, variables que influyen en el proceso, planteamiento del problema y la solución propuesta como una introducción detallada de los procedimientos seguidos a lo largo del desarrollo del proyecto.

En el Capítulo 2 se resume el marco teórico del procesamiento de imágenes, revisando las alternativas existentes para este fin y haciendo un estudio de las metodologías para una aplicación específica en el reconocimiento de patrones geométricos mediante la transformada de Hough. Se introduce al estudio de este tipo de aplicaciones en el programa MatLab y los comandos utilizados para el procesamiento de imágenes.

En el Capítulo 3 se resume el marco teórico del manipulador robótico, en el cual se establecen los parámetros morfológicos a seguir para la construcción del prototipo, los motores para el movimiento de las respectivas articulaciones y el sistema de control de los mismos. Adicionalmente se hace un estudio del modelo cinemático para el brazo robótico, estableciendo la metodología para la obtención de los parámetros Denavit-Hartenberg y deducir de éstos, la solución de las ecuaciones cinemáticas inversas que permiten al robot auto-posicionar sus articulaciones en función de la posición final deseada para la pinza en coordenadas X-Y.

En el Capítulo 4 se detalla el desarrollo del sistema de adquisición y procesamiento de imagen que permite la localización, mediante coordenadas, de

los centroides de las imágenes capturadas de las piezas geométricas (cuadrados, triángulos y círculos) ubicadas en una plataforma construida para el efecto.

En el Capítulo 5 se detalla el desarrollo tanto del *hardware* como del *software* para el manipulador robótico, integrándolo con el programa de procesamiento de imágenes implementado en MatLab. El control del robot y la HMI se implementan en el programa LabView y se presenta un método de solución para las ecuaciones cinemáticas inversas de los ángulos de posicionamiento de las articulaciones del brazo robótico.

En el Capítulo 6 se presenta un resumen de las pruebas realizadas para la puesta a punto y la calibración del sistema integrado, tanto en lo que respecta a la adquisición y el procesamiento de imágenes como del sistema de posicionamiento del brazo robótico.

En el Capítulo 7 finalmente, se indican las conclusiones y recomendaciones a las cuales se ha llegado en el presente proyecto, en base a los resultados obtenidos durante todo el proceso de desarrollo.

CAPÍTULO 1

PLANTEAMIENTO DEL PROYECTO

El presente Proyecto: “DESARROLLO DE UN SISTEMA DE CONTROL PARA UN BRAZO ROBÓTICO MEDIANTE ADQUISICIÓN Y PROCESAMIENTO DE IMÁGENES”, consiste en la implementación de un sistema que realice la adquisición y procesamiento de una imagen obtenida a través de una cámara digital, identifique entre un grupo de figuras geométricas, de acuerdo a la opción definida por el usuario (cuadrados, círculos o triángulos), localizadas en una plataforma A, para luego trasladarlas a una plataforma B, mediante el posicionamiento adecuado de un brazo robótico de cinco grados de libertad. Se ha desarrollado en dos etapas claramente definidas, la de Procesamiento de Imagen y la de Desarrollo e Implementación del Manipulador Robótico.

1.1 PROCESAMIENTO DE IMAGEN

1.1.1 OBJETIVO

Adquirir la imagen proveniente de una cámara web, extraer sus principales atributos y distinguir figuras geométricas regulares contenidas en esta imagen, (cuadrados, triángulos y círculos), una vez realizada esta acción, encontrar las coordenadas del centroide de cada una de las figuras, las mismas que serán enviadas al sistema de control de posicionamiento del brazo robótico.

1.1.2 VARIABLES QUE INFLUYEN EN EL PROCESO

La adquisición y procesamiento de una imagen proveniente de una cámara presenta una serie de variables, tales como la resolución de la cámara web, los efectos de la iluminación, la posición de la cámara, el contraste y el brillo de la imagen, los que influyen directamente en el proceso y afectan la calidad de la misma, contribuyendo en la presencia de ruido en la información obtenida. Complementariamente, el adecuado procesamiento que se aplique a la imagen adquirida es parte vital del objetivo que se persigue.

1.1.3 PLANTEAMIENTO DEL PROBLEMA

- a)** Se debe realizar la selección del programa adecuado para realizar el proceso de adquisición y procesamiento de imagen.
- b)** Los efectos de la iluminación son un aspecto decisivo para lograr el aumento de la exactitud en una aplicación de visión artificial, ya que si no se dispone de una iluminación adecuada se pierde exactitud en la imagen; el tipo de iluminación es otro aspecto a resolver ya que la iluminación proveniente de lámparas de sodio, lámparas fluorescentes o la luz natural, introducen una gran cantidad de brillo y sus efectos en la imagen no son los mismos para cada tipo de iluminación.
- c)** La ubicación de la cámara es otro aspecto influyente, puesto que si no se escoge una adecuada posición de la misma, se introducirán parámetros errados en los atributos de las figuras y además su colocación debe permitir el libre movimiento del brazo robótico.
- d)** La superficie en donde se colocarán las figuras puede ser oscura o clara, una superficie clara brinda un menor contraste, mientras que una superficie oscura suele introducir parámetros errados en la imagen.
- e)** Una vez efectuada la adquisición de la imagen, realizar un acondicionamiento previo, para luego proceder a la extracción de sus características.
- f)** Se debe implementar un algoritmo que permita diferenciar, dentro de un conjunto de figuras geométricas regulares, ya sea cuadrados, círculos o triángulos.
- g)** Se debe localizar el centroide de cada una de las regiones encontradas dentro de la imagen capturada por la cámara y enviar sus coordenadas X-Y para un correcto posicionamiento del brazo robótico.

1.1.4 SOLUCIÓN PROPUESTA

- a)** Para la adquisición y el procesamiento de imágenes se selecciona la herramienta computacional MatLab en su versión 7.0 la misma que dispone de una amplia librería de funciones específicas, además de una gran cantidad de información al respecto. Adicionalmente, el software permite la utilización de cualquier tipo de cámara web comercial.
- b)** Para mitigar los efectos de la iluminación y debido a que en esta aplicación no se requiere la extracción de atributos basados en colores, se utilizan las librerías incluidas en el software, transformando así la imagen adquirida RGB a una escala de grises y luego, con la técnica de binarización, se obtiene una imagen en blanco y negro con la cual se han minimizarán los efectos de la iluminación.
- c)** Con el objetivo de poder realizar la adquisición de la imagen de manera tal que se pueda visualizar plenamente la plataforma en la que son colocadas las figuras geométricas para ser identificadas, sin que la imagen sufra distorsión producida por el ángulo de incidencia de la cámara, ésta se debe colocar perpendicularmente al plano de trabajo y a una distancia que permita la libre acción del brazo robótico.
- d)** La superficie de trabajo sobre la cual se colocan las figuras geométricas y que será capturada por medio de la cámara, debe ser oscura lo que permitirá aprovechar la capacidad de realce del contraste en los detalles superficiales, consiguiendo así una mejor identificación de las figuras geométricas.
- e)** Previo a la extracción de las características de las figuras geométricas capturadas en la imagen se utilizan una serie de operaciones morfológicas presentes en las librerías del software (erosión, dilatación, reconstrucción) que brindan ventajas tales como supresión de ruido, simplificación de formas, ampliación, reducción y permiten realizar una descripción cualitativa de los objetos.

- f)** Dentro del procesamiento de imágenes se tienen varias alternativas para la extracción de características de una imagen dada, dependiendo de los requerimientos a ser analizados. Para el objetivo planteado en este trabajo se utilizan las técnicas de etiquetado, segmentación extracción de bordes y para diferenciar entre una forma geométrica y otra, se utiliza la Transformada de Hough.
- g)** Para la localización de los centroides se utilizan las herramientas matriciales con las que cuenta el programa MatLab.

1.2 DESARROLLO DE UN MANIPULADOR ROBÓTICO

1.2.1 OBJETIVO

Diseñar e implementar un sistema manipulador robótico controlado por computador, el cual debe recibir la información de las coordenadas de localización de los centroides de las figuras identificadas y posicionar de manera automática las articulaciones del robot para que sea capaz de tomar las figuras detectadas de una plataforma A, trasladarlas a un arreglo ordenado en una plataforma B y repetir la acción para cualquier par de coordenadas dado.

1.2.2 VARIABLES QUE INFLUYEN EN EL PROCESO

Para el desarrollo de un sistema robótico intervienen una serie de factores que deben tomarse en cuenta para la resolución del objetivo planteado, tales como el tipo de tarea a realizarse, la morfología seleccionada del robot, el elemento efector terminal que utilice, la estructura mecánica del robot y de su entorno de trabajo, el tipo de actuador que pondrá en movimiento las articulaciones, el tipo de control de los actuadores, los algoritmos matemáticos que se utilizan para la posición de las articulaciones robóticas, la fuente de energía y el software correspondiente en el caso de un control computarizado. Adicionalmente el factor económico constituye un elemento decisivo en la selección de la solución que se plantea.

1.2.3 PLANTEAMIENTO DEL PROBLEMA

- a)** El tipo de tarea que el robot realice se involucra directamente en los demás parámetros de influencia, por lo que se debe definir claramente el tipo de trabajo que el robot realice.
- b)** Se debe seleccionar una morfología acorde al entorno de trabajo y la tarea a realizarse.
- c)** La selección del elemento efector terminal debe considerar el tipo de piezas que se manipularán, así como su tamaño y su peso.
- d)** Para la estructura mecánica de la planta y del robot se requieren dos plataformas y un aditamento para la correcta ubicación de la cámara, la cual debe colocarse de manera tal que permita el movimiento libre del robot.
- e)** Se debe seleccionar el sistema de actuadores para poner en movimiento las articulaciones del robot, acorde a la estructura de la planta que se diseña.
- f)** El tipo de control de los actuadores debe seleccionarse tomando en cuenta que se plantea implementar un sistema computarizado como núcleo central del proceso.
- g)** La implementación de un algoritmo matemático que permita el adecuado posicionamiento de las articulaciones del robot constituye un elemento fundamental para conseguir un control autónomo del sistema y para garantizar la confiabilidad del mismo.
- h)** La fuente de energía que permita el movimiento del robot debe elegirse acorde con los actuadores y controladores seleccionados.
- i)** En la elección del software computacional debe tomarse en cuenta la capacidad y experticia del diseñador para realizar la programación del mismo, la facilidad de conseguir versiones actualizadas y la necesidad o no de licencias, la compatibilidad con el sistema operativo del computador en el cual se instalará y se ejecutará su programación y la disponibilidad de librerías adecuadas para la consecución del objetivo.

1.2.4 SOLUCIÓN PROPUESTA

- a)** La tarea que el robot debe realizar es típica de aplicaciones industriales en las cuales se requiere la manipulación de elementos de una cadena de producción como piezas mecánicas, envases, materias primas u otro tipo de objetos. Para este caso se necesita transportar una por una las piezas de las figuras colocadas en una plataforma denominada "A" hacia una plataforma denominada "B" de manera que queden ordenadas por su forma. La selección de las piezas se hace mediante un sistema de coordenadas determinado por el procesamiento de imágenes.
- b)** La morfología que se utiliza en la industria a nivel mundial es la de un brazo robótico antropomórfico, esto se debe principalmente a que esta configuración permite mayor diversidad y flexibilidad en las tareas a realizarse en la mayoría de procesos de producción y se adapta perfectamente a los objetivos planteados para el presente trabajo.
- c)** Debido a que se debe manipular un conjunto de piezas de pequeño tamaño y dado que las respectivas plataformas en las que encuentran dichas piezas no poseen una gran superficie, se selecciona como elemento terminal a la configuración de pinza con dedos deslizantes, la cual se asemeja antropomórficamente a la acción de pinza de los dedos índice y pulgar de la mano humana, de esta manera es más fácil colocar aditamentos que sobresalen de las piezas, justamente en los respectivos centroides de éstas, ya que la ubicación en coordenadas X - Y que entrega el procesamiento de imágenes corresponde a este punto geométrico. Otras topologías de elemento terminal son posibles siempre y cuando se tengan los medios necesarios para su correcta operación, sin embargo para el presente trabajo el factor económico es un limitante principal, por lo que la selección referida se ajusta plenamente al diseño global de la planta.
- d)** Como se ha mencionado anteriormente, los costos de un sistema robótico industrial como el que se plantea son elevados y limitan a que se diseñe un sistema prototipo de mayor envergadura, por lo cual se construye un pequeño brazo robótico basado en modelos similares existentes en el mercado y de

acuerdo a las dimensiones de éste, se implementa una estructura de plataformas colocadas a 90 grados una de otra para la operación del traslado de las piezas. Un accesorio con regulación telescópica se coloca estratégicamente para la ubicación adecuada de la cámara sin que, en primera instancia, se interponga en el paso libre del brazo robótico, sin embargo y debido a que para capturar una imagen adecuada del espacio de trabajo por parte de la cámara, no se puede colocar ésta en un sitio demasiado alto, en las operaciones de movimiento articular del brazo robótico se programan restricciones para evitar que el mismo choque con el soporte de la cámara en algún momento dado.

- e) Como sistema de actuadores se seleccionan servomotores de modelismo, los cuales poseen un sistema de control de posición incorporado y desarrollan el torque suficiente para mover los elementos que constituyen el prototipo de brazo robótico construido.
- f) Existen en el mercado sistemas de control específicos para el tipo de actuadores seleccionado (servomotores) y precisamente se ha seleccionado el controlador *Mini SSC II*, el cual posee ocho salidas de control PWM para servomotores y comunicación serial RS-232 para el control por medio de un computador.
- g) Se utiliza el algoritmo matemático derivado de la convención *Denavit-Hartenberg*, la cual se utiliza universalmente en robótica para el tipo de morfología escogida para el robot. Este algoritmo permite la deducción de soluciones en parámetros geométricos para la posición (en este caso angular) de las articulaciones del brazo robótico de manera que se alcance un punto en el espacio, dadas sus coordenadas X , Y , Z . Debido a que las soluciones se cumplen para todos los puntos del entorno de trabajo establecido, el sistema tiene completa autonomía y una aceptable confiabilidad en el desarrollo de la tarea, confiabilidad reducida únicamente debido a las limitaciones mecánicas tanto de la estructura del robot como de sus actuadores ya que éstos sólo se pueden mover en pasos de 0,72 grados (según el fabricante).

- h)* Debido a que tanto el controlador como los actuadores necesitan de energía eléctrica (9 y 6 voltios, respectivamente), se diseñan e implementan las respectivas fuentes de poder, conociendo de manera teórica y experimental los requerimientos máximos de corriente de los elementos involucrados.
- i)* Para la selección del software computacional, se ha optado por la utilización del sistema de desarrollo LabView 7.0, debido principalmente a que este programa ha sido utilizado en varias ocasiones en el desarrollo de los estudios de la carrera con resultados satisfactorios, se consiguen fácilmente versiones de desarrollo estudiantil que no requieren licencia, es absolutamente compatible con la mayoría de sistemas operativos existentes en el mercado y posee un conjunto de librerías muy potente que se ajustan a los requerimientos del proyecto, tales como MatLab Script y NiVisa las cuales permiten la comunicación con los códigos desarrollados en MatLab para la adquisición y el procesamiento de imágenes y la comunicación a través del puerto serial RS-232, respectivamente. Es importante anotar que las versiones avanzadas de LabView poseen librerías para el procesamiento de imágenes, pero requieren de cámaras específicas de venta exclusiva del fabricante (National Instruments), o a su vez se debe adquirir la respectiva licencia para acceder a los drivers de cámaras comunes, por lo que la opción de la utilización del MatLab en esta etapa del proceso es preferible y en todo caso, más interesante. Adicionalmente, LabView permite un desarrollo más simple de la interfaz hombre-máquina HMI ya que su entorno gráfico posibilita la inclusión de elementos no solo para el control del brazo robótico, sino también para la implementación de una pantalla de usuario atractiva e intuitiva.

CAPÍTULO 2

MARCO TEÓRICO DEL PROCESAMIENTO DE IMAGEN

2.1 INTRODUCCIÓN

Los modelos de datos y señales para imágenes y secuencias de vídeo están experimentando un importante desarrollo. Tradicionalmente, las imágenes digitales se han representado como matrices rectangulares de píxeles y los vídeos como una secuencia de tramas en movimiento. Las nuevas aplicaciones multimedia requieren una representación más cercana al mundo real o que al menos tenga en cuenta parte del proceso que creó la información digital.

El observador humano usa un amplio rango de señales, obtenidas a partir del color, la perspectiva, el sombreado, y una amplia librería de experiencias individuales. La percepción visual depende únicamente de la capacidad humana de realizar juicios. Sin embargo, un sistema de visión artificial no tiene una experiencia de base de la cual tomar decisiones mediante una comparación; todo debe estar específicamente definido. Problemas simples como encontrar un orificio o juzgar puntos finos, se vuelven una tarea complicada en la visión artificial. Para reducir el número de variables, el sistema de visión debe estar provisto con la mejor imagen que sea posible.

En un sistema de visión artificial, la mejor imagen es aquella que tiene el mayor contraste, donde las áreas de interés se destacan del fondo (background) intrascendente. La mejor imagen es aquella que facilita la tarea del sistema de visión.

2.2 PROCESAMIENTO DIGITAL DE IMAGEN

La información consta de incontables elementos, figuras, sonidos, voces y demás elementos visuales. Toda esta información se guardará y transferirá a través de una tecnología que combina productos multimedia y técnicas de digitalización.

Los seres humanos confían fundamentalmente en el sentido de la vista. Recogemos 99 % de nuestra información a través de nuestros ojos. El procesamiento de imágenes es un proceso que concierne tanto a ordenadores como a algoritmos diversos para un propósito específico, una vez que se han adquirido las imágenes de interés a través de una cámara y/o un escáner ^[1].

Los procesos de reconocimiento de patrones, y en particular los que utilizan señales provenientes del mundo real comprenden algunas etapas básicas ^{[2], [3]}:

- a) *Adquisición de Datos (señales temporales, imágenes estáticas o una secuencia de imágenes) empleando dispositivos adecuados:* Usualmente en esta etapa se emplean instrumentos de medición, sistemas computarizados de adquisición y almacenamiento de información.
- b) *Procesamiento previo o procesamiento de datos:* etapa que permite a menudo un cambio en el espacio de representación de la señal, adaptándola para la etapa posterior (segmentación de las imágenes, normalización).
- c) *Extracción de características relevantes:* en esta etapa los métodos que usualmente se aplican pueden agruparse en: no transformados (momentos, modelos paramétricos), y transformados (Transformada de Hough, Transformada de Fourier); elementos de la geometría, y descriptores gráficos.
- d) *Selección de características:* es el paso en el que se optimiza en número las características extraídas en el paso anterior, sin que el patrón pierda los rasgos que ayuden a su posterior identificación.
- e) *Clasificación:* es la identificación de un objeto como miembro de una clase, cuyos componentes se han agrupado en función de similitud.

En el presente proyecto se proponen aportes para la caracterización de información de tipo 2-D (imágenes estáticas) con técnicas de diversos campos (Trasformadas, geometría, procesamiento digital de imágenes).

2.3 IMÁGENES DIGITALES

A partir de los datos adquiridos por medio de una cámara digital, se pasa a una imagen bidimensional de intensidad de luz y color.

Si se define a la imagen como una función $f(x,y)$ el origen del sistema de coordenadas se encuentra en la esquina superior izquierda, el eje X se encontraría definido de derecha a izquierda mientras que el eje Y se encontraría de arriba hacia abajo como se ilustra en la Figura 2.1

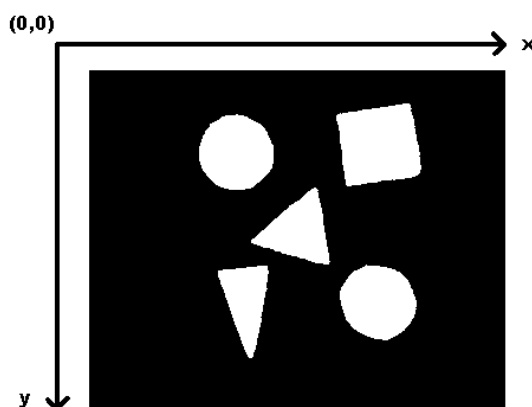


Figura. 2.1 Esquema de Coordenadas en una Imagen [*]

El proceso de muestreo consiste en medir el valor de la función de la imagen $f(x,y)$ en intervalos discretos en el espacio. Cada muestra corresponde a una pequeña área cuadrada de la imagen conocida como **píxel**. Una imagen digital es un arreglo bidimensional de píxeles representados por las coordenadas $(X-Y)$ [2].

2.4 ADQUISICIÓN DE DATOS

El desarrollo constante de nuevas tecnologías de visión artificial ha permitido a tales sistemas ser utilizados en problemas como medición, inspección y monitoreo de procesos en la industria. La llegada de las computadoras rápidas, los integrados de procesamiento digital de señales y los lenguajes de nivel más alto han incrementado la capacidad de la visión artificial. Hoy en día, en la mayoría de las grandes industrias existe un equipo de trabajo que tiene una especialización en las aplicaciones de visión artificial. Los avances en el hardware, software y las cámaras de visión redujeron los costos y el tiempo del desarrollo de la aplicación.

*Imagen creada por los autores.

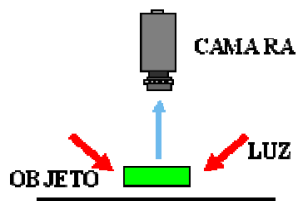


Figura 2.2 Esquema de Adquisición de Imagen ^[*]

Muchas aplicaciones han fallado por la falta de una iluminación apropiada. Los sistemas de iluminación que trabajan correctamente en el laboratorio no siempre lo hacen en la planta industrial donde la iluminación fluctuante del ambiente causa distorsiones.

La visión artificial es una técnica basada en la adquisición de imágenes, generalmente en dos dimensiones, para luego procesarlas digitalmente mediante algún tipo de CPU (computadora, microcontrolador, DSP, etc.), con el fin de extraer y medir determinadas propiedades de la imágenes adquiridas. Se trata, por tanto, de una tecnología que combina las computadoras con las cámaras de video para adquirir, analizar e interpretar imágenes de una forma equivalente a la inspección visual humana.

2.5 ETAPAS DE ANÁLISIS DE IMAGEN Y RECONOCIMIENTO DE FORMAS

Los algoritmos relacionados con visión artificial son muy variados y abarcan numerosas técnicas y objetivos ^[4]. Los pasos más comunes en que consiste el análisis o procesamiento de imagen se muestran en la Figura 2.3

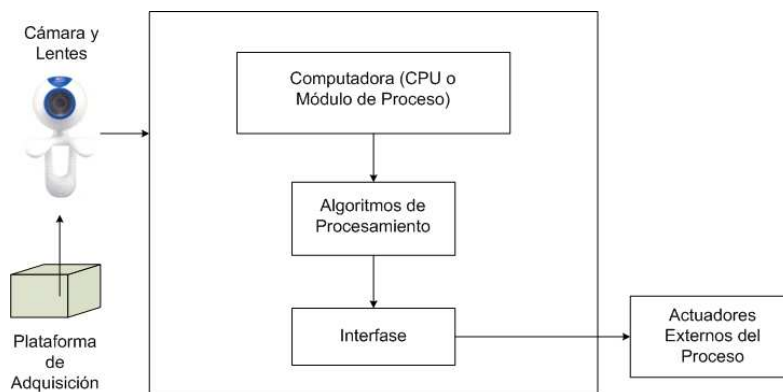


Figura 2.3 Elementos de un Sistema en Tiempo Real ^[*]

* Imagen creada por los autores.

2.5.1 PREPROCESAMIENTO

Cuando se adquiere una imagen mediante cualquier sistema de captura, por lo general ésta no es directamente utilizable por el sistema de visión. La aparición de variaciones en intensidad debidas al ruido, por deficiencias en la iluminación, o la obtención de imágenes de bajo contraste, hace necesario un pre-procesamiento de la imagen con el objetivo fundamental de corregir estos problemas, además de aplicar aquellas transformaciones a la imagen que acentúen las características que se deseen extraer de las mismas, de manera que se facilite las operaciones de las etapas posteriores. A este campo pertenecen las técnicas de transformaciones geométricas, las basadas en el histograma, el filtrado espacial ^[5] y el filtrado frecuencial ^[6]. Aunque existen líneas de investigación abiertas, no es el campo más activo en ese sentido, y las técnicas existentes son bastante útiles para la mayoría de las aplicaciones.

Algunas de las técnicas más habituales son:

- Conversión de los niveles de gris: su objetivo es la mejora de la calidad de la imagen.
- Transformaciones geométricas: cuyos objetivos son la corrección de la perspectiva y la reconstrucción tridimensional de los objetos de la escena.

2.5.1.1 Representación de Imágenes con Niveles de Gris

Utiliza como característica solamente la luminancia de cada píxel, útil si distintos objetos se corresponden con niveles de gris diferentes.

Técnicas de operaciones puntuales (Umbralización).

Técnicas de clasificación basada en amplitud. (Imagen binaria)

En el caso ideal en que el objeto posea un rango estrecho de niveles de gris frente a un fondo uniforme, se puede establecer un nivel de gris intermedio (umbral) para separar objeto y fondo. Para separar por umbral, es útil recurrir al histograma ^[2].

* Imagen creada por los autores.

2.5.1.2 Representación de Imágenes Binarias

Se define a una imagen binaria como una función de dos variables discretas $a[m,n]$ que puede tomar dos valores, '0' o '1', dependiendo del nivel de gris de la imagen (una imagen binaria tiene dos niveles: blanco y negro).

Se puede proponer una definición alternativa si se considera que una imagen consiste en un conjunto de coordenadas discretas. En este sentido, el conjunto corresponde a todos aquellos puntos o píxeles que pertenecen a la imagen. Por lo tanto, se puede decir que en morfología matemática los conjuntos representan objetos en una imagen. Por ejemplo, el conjunto de todos los píxeles negros en una imagen binaria constituye una descripción completa de la misma ^[7].

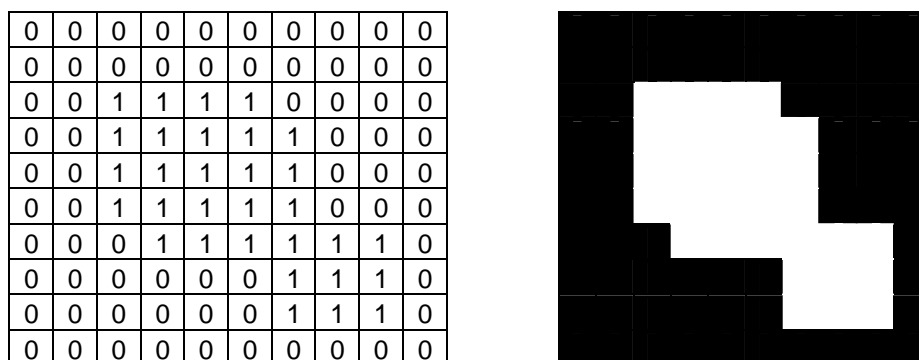


Figura 2.4 Representación de Imágenes Binarias ^[7]

2.5.2 SEGMENTACIÓN

La palabra 'segmentación' tiene un significado que depende del uso y del contexto en el cual se utilice. El objetivo básico de los algoritmos de segmentación es definir una partición del espacio ^[1].

Esta sección revisa los pasos principales implicados en un algoritmo de segmentación y las principales opciones a escoger, adicionalmente se introduce la terminología. Un esquema general para la segmentación se puede considerar como la concatenación de tres pasos principales, representados en la Figura 2.5.

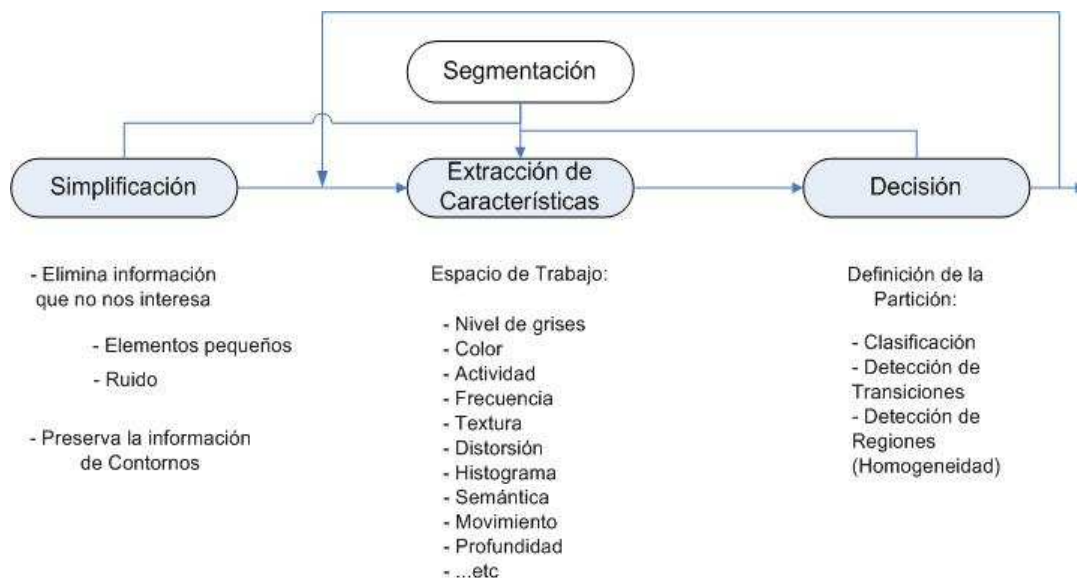


Figura 2.5 Proceso Global de la Segmentación de Imágenes ^[*]

2.5.2.1 Simplificación

La mayoría de las veces, los datos originales en una imagen o en una secuencia de vídeo contienen información que es irrelevante para una aplicación dada. En tales casos, los datos se deben simplificar, removiendo toda aquella información irrelevante. La simplificación controla la cantidad y naturaleza de la información que es preservada. Además los datos simplificados deben contener áreas fáciles de segmentar. Por ejemplo, la simplificación puede reducir la complejidad de texturas, eliminando ruido u objetos muy pequeños para un tamaño dado ^[1].

2.5.2.2 Extracción de Características

La segmentación se realiza sobre características específicas de los datos. La selección del espacio de características conduce al tipo de homogeneidad que se espera en la partición final. En algunas aplicaciones, los datos originales proveen directamente el espacio de características necesario para la segmentación. Por ejemplo, para la segmentación a color, el valor del píxel puede corresponder directamente a la característica de interés. Sin embargo, en otros casos, las características de interés deben ser estimadas a partir de los datos originales. Las características típicas se enumeran debajo del bloque de la extracción de características de la Figura 2.5 ^[1].

* Imagen creada por los autores.

2.5.2.3 Decisión

Finalmente para obtener una partición de los datos, el espacio de características debe ser analizado. El paso de Decisión incide sobre la posición de los contornos que forman la partición en el espacio de decisión. Los contornos separan áreas de datos que contienen elementos con las mismas características en el espacio de características seleccionado. Las técnicas de segmentación son muy dependientes del propósito de la aplicación y del tipo de imágenes a analizar ^[5].

Antes de segmentar es preciso definir qué objetos interesa determinar. Tras la segmentación es posible realizar operaciones de filtrado (a nivel de objetos), así como determinar características que permitan clasificar los objetos.

Una buena segmentación es difícil de evaluar. Fundamentalmente, lo que se busca es que diferentes objetos tengan valores claramente diferentes de la(s) característica(s) discriminante(s).

2.5.3 METODOLOGÍAS DE SEGMENTACIÓN

Existen varias metodologías para realizar la segmentación de una imagen, de entre las cuales se puede citar las siguientes ^[8]:

Segmentación basada en características: Segmentación por niveles de gris, segmentación de imágenes en color y segmentación por texturas.

Segmentación basada en transiciones: Detección de bordes

Segmentación basada en modelos: Transformada de Hough

Segmentación basada en Morfológica Matemática.

2.5.3.1 Segmentación Basada en Características

La Estrategia de Segmentación basada en características queda definida por los siguientes pasos:

- Estudio del espacio de características

- Definición de clases en el espacio de características
- Definición de las Regiones

Las técnicas de Clasificación empiezan creando una partición del espacio de características y entonces traducen esta partición a una partición del espacio de decisión.

Recordar a modo de terminología, que una región creada por un algoritmo de segmentación se define como un conjunto de elementos (píxeles o imágenes) homogéneos en el espacio de características y conexo en el espacio de decisión. Una región puede no tener ningún significado semántico. Por lo contrario, un objeto es una representación visual en 2D de una entidad con significado semántico que puede estar formado a su vez por diversas regiones ^[1].

2.5.3.2 Segmentación por Detección de Transiciones

Los métodos de segmentación basada en transiciones aplican sobre la imagen, o sobre un plano de características, un detector de bordes. Las regiones se definen a partir de las fronteras delimitadas por los bordes detectados.

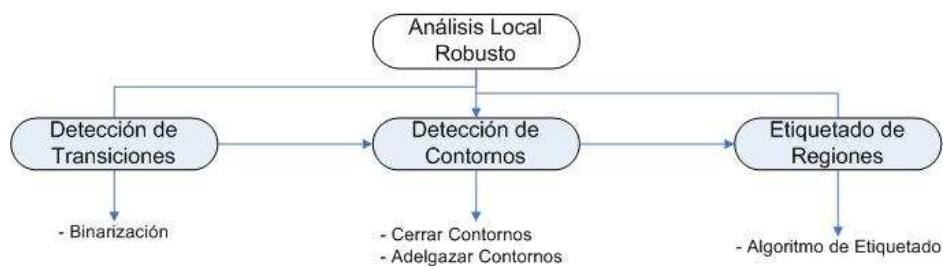


Figura 2.6 Diagrama de Bloques de la Detección de Contornos ^[*]

En un proceso de segmentación, el paso de la decisión debe proporcionar la partición final de los datos originales. Las técnicas basadas en transición proponen estimar la posición de las discontinuidades en el espacio de decisión.

Estas discontinuidades, son evaluadas en el espacio de características. Estas discontinuidades son realizadas por un proceso previo que se considera como un filtrado. La salida del filtro debe contener valores altos en aquellas posiciones de los datos que están cerca de las transiciones y valores bajos en las áreas

* Imagen creada por los autores.

homogéneas. Para esto se utilizan filtros lineales paso alto, filtros no lineales, el uso de operaciones morfológicas como es el cierre en la que se intentan conectar esos contornos parciales para dar lugar a la partición final ^[1].

2.5.3.2.1 Filtros Detectores de Bordes

Realizan otro tipo de operaciones con los píxeles, pero siempre con el resultado de enfatizar los bordes que rodean a un objeto en una imagen, para hacerlo más fácil de analizar. Estos filtros típicamente crean una imagen con fondo gris y líneas blancas y negras rodeando los bordes de los objetos y características de la imagen.

2.5.3.2.2 Etiquetado de una Imagen

Se entiende por etiquetar una imagen a la discriminación de regiones dentro de la misma, cuya región corresponde a una agrupación de píxeles con características similares ^[9]. Se consideran ciertas características para el etiquetado de las regiones.

Vecindad: Determina si dos píxeles son o no vecinos (adyacentes).

Vecindad a 4: Sólo se consideran puntos conectados en direcciones perpendiculares (izquierda, derecha, arriba, abajo), ver Figura 2.7 a).

Vecindad a 8: También se toman en cuenta los puntos adyacentes en diagonal), ver Figura 2.7 b).

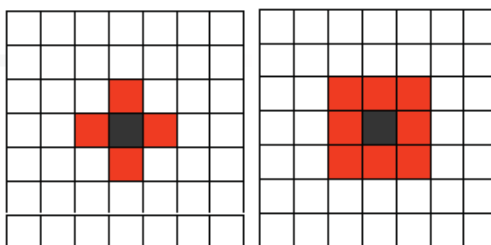


Figura 2.7 a) Vecindad a 4 b) Vecindad a 8 ^[*]

Conectividad: Dos píxeles con la misma etiqueta, *c*, están conectados si existe un camino de uno al otro a través de píxeles vecinos con la misma etiqueta ^[8].

* Imagen creada por los autores.

2.5.3.2.3 Etiquetado de Imágenes Binarias

Suponiendo dos niveles:

- Nivel 1 (primer plano)
- Nivel 0 (fondo)

El etiquetado consiste en identificar grupos conexos de píxeles de primer plano, asignándole a cada uno de ellos una etiqueta diferente ^[8].

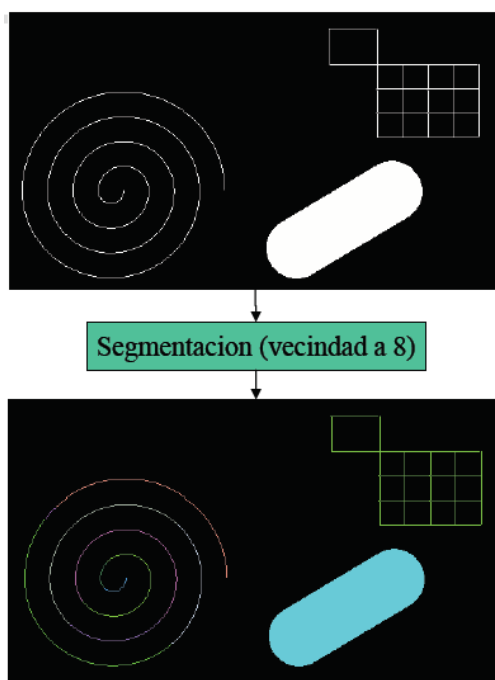


Figura 2.8 Etiquetado de Imágenes Binarias ^[8]

2.5.3.3 Segmentación Basada en Modelos

Las técnicas de segmentación basada en modelos presuponen conocidas algunas características de los objetos o regiones en la imagen: rectas, objetos circulares, etc.

2.5.3.3.1 La Transformada de Hough (TH)

Es una de las técnicas de segmentación, ampliamente difundida. Es utilizada en el procesamiento de imágenes aéreas: en identificación y localización de estructuras en imágenes aéreas (carreteras, edificios, vehículos, etc.). Adicionalmente, encuentra aplicación en biometría, para la identificación del iris y

huellas dactilares, en detección de rasgos faciales y gestos, en seguimiento de objetos, y en procesos de fabricación para detección de defectos ^[10].

Se realiza una transformación de dominio para encontrar curvas paramétricas en una imagen (binaria o de grises). El nuevo espacio transformado se denomina espacio de Hough. Para ajustar distintos tipos de formas se usan distintos espacios de Hough.

El objetivo de la TH es transformar un problema de detección de patrones en un problema de detección de máximos en el espacio de Hough. La ventaja principal de utilizarla es que ésta es poco afectada por ruido o huecos en las curvas. Identificando las curvas que interesan en el espacio de Hough, es posible aislar regiones en la imagen original (filtrado) ^[8].

2.5.3.3.2 Definición de la Transformada de Hough

Supóngase que, para n puntos de una imagen (típicamente una imagen binaria), se desea encontrar subconjuntos de estos puntos que residan en líneas rectas. Una posible solución es encontrar en primer lugar todas las líneas determinadas por cada par de puntos y a continuación encontrar todos los subconjuntos de puntos que están cerca de determinadas líneas. El problema de este procedimiento es que implica encontrar $[n(n-1)/2] \sim n^2$ líneas y, a continuación llevar a cabo $[n(n(n-1))/2] \sim n^3$ comparaciones de cada punto de todas las líneas. Este método es totalmente prohibitivo desde un punto de vista de tiempo de cálculo, pero las aplicaciones son de lo más triviales.

Hough [1962] propuso un método alternativo, al que normalmente se hace referencia como la transformada de Hough. Considérese un punto (x_i, y_i) y todas las líneas que pueden pasar a través de este punto, utilizando la ecuación general de la recta en forma explícita (punto – pendiente), $y_i = a * x_i + b$. A través de (x_i, y_i) pasan un infinito número de líneas, pero todas ellas satisfacen la ecuación $y_i = a * x_i + b$ para diversos valores de a y b . Sin embargo, escribiendo esta ecuación como $b = -x_i * a + y_i$ y considerando el plano ab (también denominado espacio parámetro) se obtiene la ecuación de la única línea para un par determinado (x_i, y_i) . Además, un segundo punto (x_j, y_j) también tiene una línea en

el espacio de parámetro asociado con él, y esta línea corta a la línea asociada con (x_i, y_i) en (a', b') , donde a' es la pendiente y b' la ordenada en el origen de la línea que contiene a (x_i, y_i) y (x_j, y_j) en el plano X-Y. De hecho, todos los puntos contenidos en esta línea tienen líneas en el espacio de parámetro que corta a (a', b') . La Figura 2.9 ilustra estos conceptos [2][3].

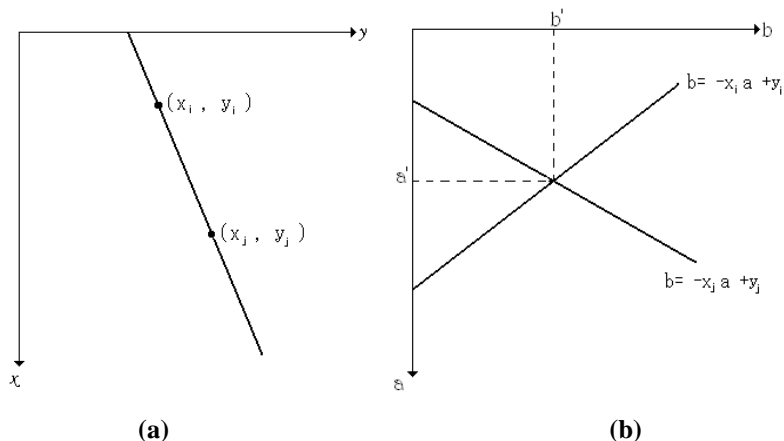


Figura 2.9 a) Plano xy b) Espacio de Parámetro [2]

El atractivo de la transformada de la transformada de Hough surge en la subdivisión del espacio de parámetro en las denominadas células acumulador, que se muestran en la Figura 2.10, donde (a_{\max}, a_{\min}) , (b_{\max}, b_{\min}) son los rasgos esperados de los valores de la pendiente y la ordenada. La célula de coordenadas (i,j) , con valor de acumulador $A(i,j)$, corresponde al cuadrado asociado a las coordenadas del espacio de parámetro (a_i, b_j) . Inicialmente estas células están puestas a cero, después, para cada punto (x_k, y_k) del plano de la imagen, se fija el parámetro a igual a cada uno de los valores permitidos de la subdivisión sobre el eje X, y se resuelve para el b correspondiente utilizando la ecuación $b = -x_i a + y_i$.

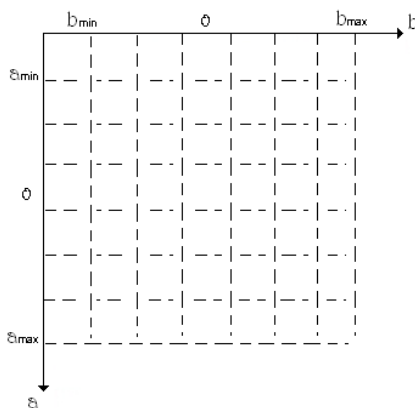


Figura 2.10 División del Espacio de Parámetro en Células Acumulador en el Plano X-Y [2]

Las b resultantes se redondean después al valor más próximo permitido del eje x . Si una elección de a_p resulta ser solución de b_q , se fija $A(p,q) = A(p,q) + 1$. Al final de este procedimiento, un valor de M en $A(i,j)$ correspondiente a M puntos del plano X - Y situados en la línea $y_i = a * x_i + b$.

La precisión de la colinealidad de estos puntos está determinada por el número de subdivisiones del plano a - b .

Obsérvese que al subdividir el eje x en K incrementos dados, para cada punto (x_k, y_k) , K valores de b corresponderán a los posibles valores de a . Con n puntos imagen, esto implica nK cálculos. Por ello el procedimiento que se acaba de exponer es lineal en n , y el producto nK no se acerca al número de cálculos expuesto al principio de esta sección a menos que J se aproxime o supere a n .

Un problema que aparece al utilizar la ecuación $y = a * x + b$ para representar una línea es que tanto la pendiente como la ordenada en el origen tienden al infinito cuando se acerca a la vertical. Una forma de evitar esta dificultad es utilizar la representación normal de una recta.

$$x * \cos(\theta) + y * \text{sen}(\theta) = \rho \quad \text{Ec. 2.1}$$

La Figura 2.11 muestra el significado de los parámetros utilizados en la ecuación 2.1. El empleo de esta representación en la construcción de una tabla de acumuladores es idéntico al método expuesto para la representación punto pendiente, Sin embargo, en vez de líneas rectas, los lugares geométricos son curvas sinusoidales en el plano $\rho\theta$. Como antes, M puntos colineales situados sobre una línea $x * \cos(\theta_j) + y * \text{sen}(\theta_j) = \rho_i$, producen M curvas sinusoidales que aportan a (ρ_i, θ_j) en el espacio de parámetro. Al incrementar θ y resolver para ρ correspondiente a M entradas al acumulador $A(i,j)$ asociado con la célula determinada por (ρ_i, θ_j) . La Figura 2.12 ilustra la subdivisión del espacio de parámetro ^[2].

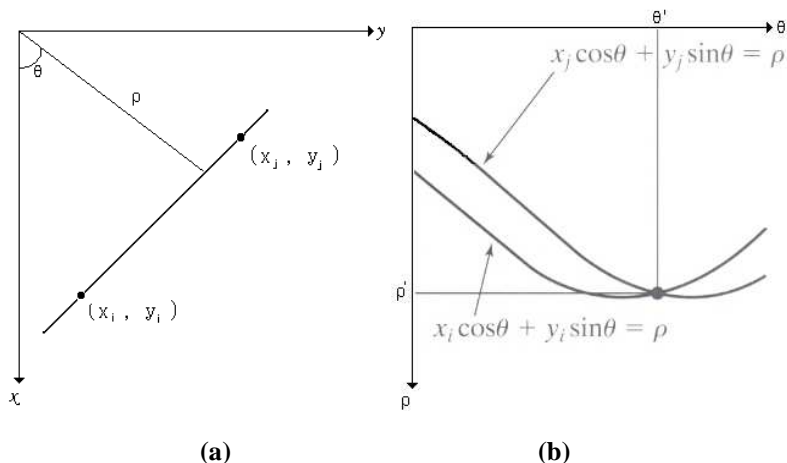


Figura 2.11 a) Parametrización de líneas en el plano xy b) Curvas Sinusoidales en el plano $\rho\theta$ [2]

El rango del ángulo θ es ± 90 , medidos con respecto al eje X . Por ello una línea horizontal tiene un $\theta = 0^\circ$, siendo ρ igual a la x positiva. De forma similar, una línea vertical tiene un $\theta = 90^\circ$, siendo ρ igual a la y positiva, $\theta = -90^\circ$, siendo ρ igual a la y negativa. Cada curva sinusoidal de la Figura 2.11 representa la familia de líneas que pasan a través de un particular punto. El punto de intersección (ρ', θ') corresponde a la línea que pasa a través de los dos puntos (x_i, y_i) y (x_j, y_j) .

El atractivo computacional de la transformada de Hough radica en la subdivisión del plano $\rho\theta$ en las celdas acumuladoras tal y como se ilustra en la Figura 2.12 donde $(\rho_{\min}, \rho_{\max})$ y $(\theta_{\min}, \theta_{\max})$ son los valores esperados. Usualmente, el máximo rango de los valores es $-90 \leq \theta \leq 90$ y $-D \leq \rho \leq D$, donde D es la distancia entre las esquinas de la imagen [2].

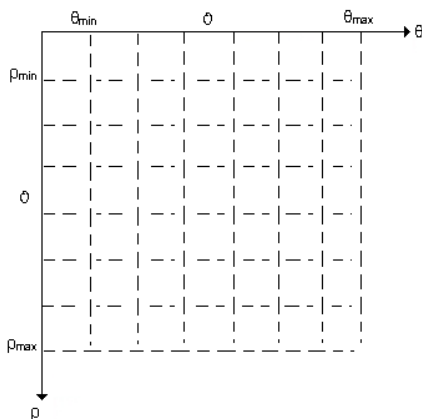


Figura 2.12 División del Espacio de Parámetro en Células Acumulador en el Plano $\rho\theta$ [2]

2.5.3.3.3 Técnica Para Detectar Segmentos Rectos en Imágenes de Grises

Usa la descripción paramétrica de formas geométricas: representación de una recta en coordenadas polares.

$$x * \cos(\theta) + y * \text{sen}(\theta) = \rho \quad \text{Ec. 2.1}$$

Una recta a distancia ρ_j del origen y con orientación θ_i se representa en la Figura 2.13. La transformada de Hough de esta recta es un punto del plano (ρ, θ) .

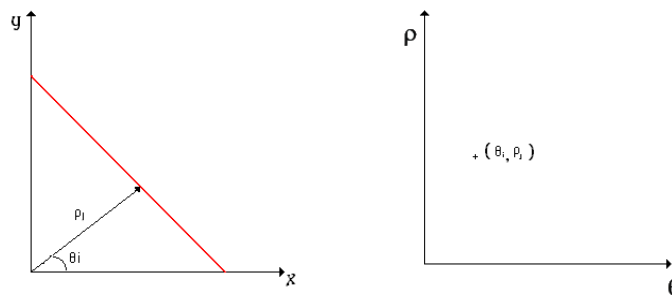


Figura 2.13 Representación Paramétrica de TH ^[2]

Cada punto (x_k, y_k) corresponde con una curva en el plano (ρ_j, θ_i) . Así, calculando las curvas para muchos puntos, los puntos de cruce definen rectas que los unen.

Todas las curvas correspondientes a las componentes colineales intersecan en el mismo punto (θ, ρ) , donde θ y ρ especifican los parámetros de la línea.

En el caso discreto, el dominio de Hough es un array bidimensional que representa valores discretos de θ y ρ . Antes de aplicar la TH es necesario seleccionar una resolución del dominio de Hough ^[8].

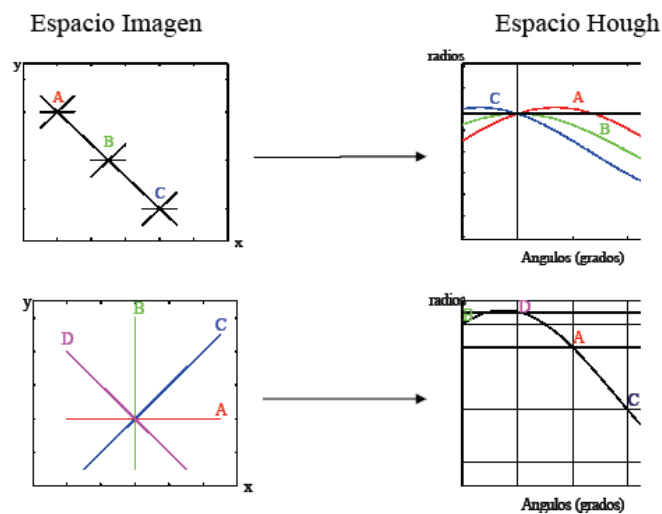


Figura 2.14 Representación Correspondiente a la TH ^[8].

La Cuantificación del plano de parámetros en celdillas acumuladoras TH de una Imagen binaria corresponde a una Matriz Acumuladora. Cada celdilla indica el número de píxeles que componen el segmento recto. Si la imagen no es binaria, la cuenta de píxeles debe ponderarse por el nivel de intensidad.

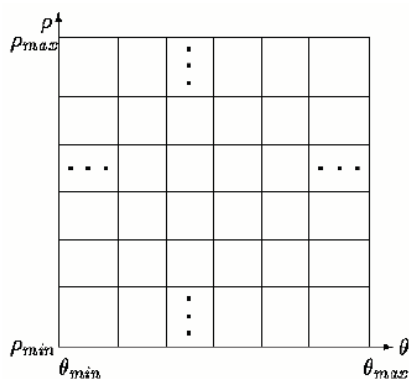


Figura 2.15 Matriz Acumuladora ^[2]

2.5.3.3.4 Algoritmo de la TH (Imagen binaria)

Inicialización de la matriz acumuladora a ceros. Para cada píxel de la imagen con nivel distinto de cero.

Para cada valor de θ se determina el valor de ρ se incrementa la posición (θ, ρ) de la matriz en 1 al finalizar, cada celdilla indica el número de píxeles que forman una recta con esos parámetros (θ, ρ) ^[8].

2.5.3.3.5 Transformada de Hough de Puntos Alineados.

El brillo en la transformada es proporcional al número de puntos que contribuyen.

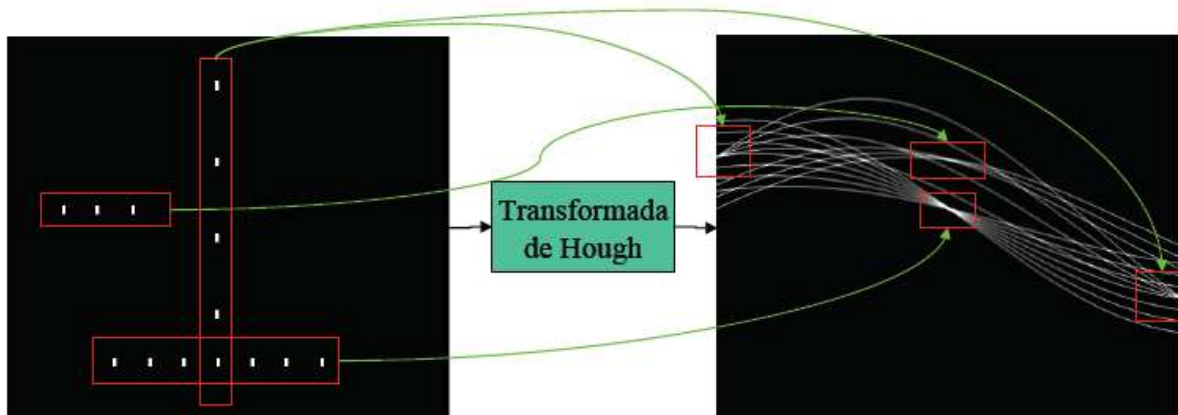


Figura 2.16 Representación de Campos Rectangular y Paramétrico ^[8]

2.5.3.3.6 Método General para Implementar la TH

Determinación de la ecuación correspondiente al rasgo que se desea buscar, así como los parámetros en función de los que se define.

A partir de la imagen original destacar los elementos de la misma que van a dar forma al rasgo buscado, por ejemplo, puntos de la frontera o bordes ^[8].

Discretización del espacio de búsqueda (para que las contribuciones de cada píxel se puedan acumular en celdillas).

Cálculo de la Transformada de Hough Imágenes binarias: todos los píxeles contribuyen por igual. Imágenes de grises: los píxeles con mayor nivel presentan mayor contribución.

Filtrado, determinar los valores de los parámetros cuya característica se desea filtrar en el espacio original.

En esta sección se realizará un ejemplo con una imagen creada en MatLab 7.0, en donde se pueden visualizar 9 puntos, cuatro de ellos colocados en las esquinas de la imagen, 4 de ellos en el interior de la misma y finalmente uno en el centro, obteniendo, mediante TH, lo siguiente:

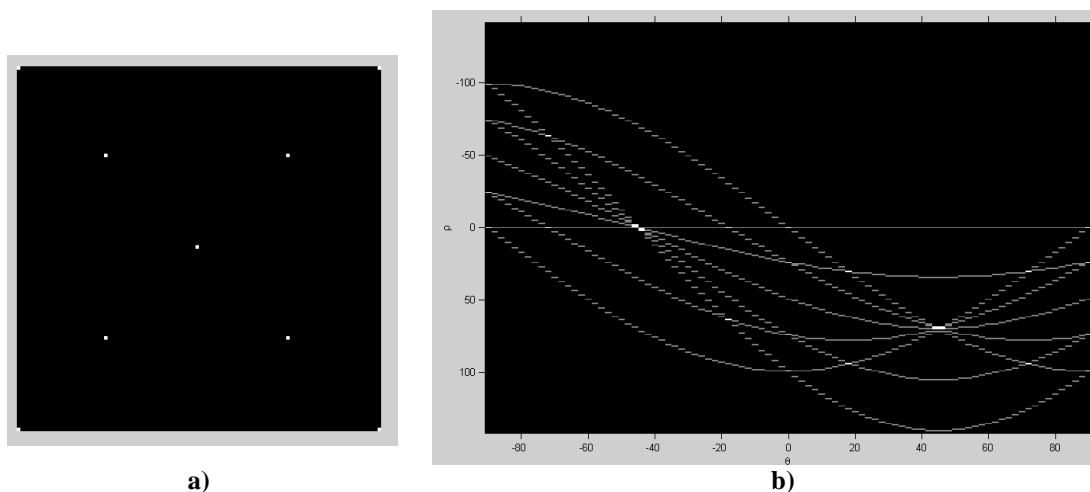


Figura 2.17 a) Imagen Creada en MatLab, b) Resultado de la TH sobre la Imagen ^[*]

* Imagen creada por los autores.

La Figura 2.17 b), muestra el resultado de aplicar la TH a la imagen binaria. La intersección de 5 sinusoidales en ± 45 indica que existen 5 puntos colineales en la imagen binaria, mientras que la intersección de dos curvas sinusoidales en $(\theta, \rho) = (-90,0), (-90,-100), (0,100), (-70,65), (70,100), (70,20), (20,90), (20, 20)$, indica que existen 8 grupos de puntos colineales que representan las líneas horizontales y verticales para unir los límites.

2.5.4 SEGMENTACIÓN BASADA EN MORFOLOGÍA MATEMÁTICA

La morfología matemática es una herramienta muy utilizada en el procesamiento de imágenes. Las operaciones morfológicas pueden simplificar los datos de una imagen, preservar las características esenciales y eliminar aspectos irrelevantes. Teniendo en cuenta que la identificación y descomposición de objetos, la extracción de rasgos, la localización de defectos e incluso los defectos en líneas de ensamblaje están sumamente relacionados con las formas, es obvio el papel de la morfología matemática.

La morfología matemática es una herramienta para extraer componentes de una imagen que sean útiles en la representación y descripción de la forma de una región, tales como contornos, esqueletos y cerco convexo. También son de interés las técnicas morfológicas para el pre o post-procesado, tales como el filtrado morfológico ^[7].

2.5.4.1 Operaciones Morfológicas

2.5.4.1.1 Erosión y dilatación

Los algoritmos de morfología matemática binaria son combinaciones de dos operaciones básicas, la dilatación y la erosión, las cuales consisten en comparar la imagen a analizar A con otra imagen de forma conocida B, denominada elemento estructurante.

La erosión consiste en examinar cada píxel y cambiarlo de 1 a 0 si alguno de sus vecinos está en 0. Normalmente se utilizan como vecinos los ocho que rodean al píxel examinado, aunque para algunas aplicaciones se pueden utilizar

conectividades de 4 vecinos (los 2 verticales y los 2 horizontales) e incluso conectividades de 2 vecinos (los verticales o los horizontales).

La dilatación es el proceso inverso, consiste en cambiar los píxeles de 0 a 1 si alguno de sus vecinos está en 1.

2.5.4.1.2 Relleno de Región (Region Filling)

En ciertas aplicaciones, es conveniente poder reconstruir una imagen que ha sufrido varias erosiones o poder llenar un objeto que está definido por una frontera. Para ello, se puede utilizar una operación llamada relleno de región cuyos algoritmos están basados en dilataciones, complementaciones e intersecciones.

2.5.4.1.3 Apertura Y Cierre

Al resultado de una erosión más una dilatación se le denomina apertura ("*opening*"). El nombre proviene de la tendencia de esta secuencia de operaciones a separar (abrir) puentes de unión entre objetos próximos o a abrir cavidades próximas al borde.

La operación opuesta (dilatación más erosión) es denominada cierre ("*closing*") y puede usarse para conectar objetos muy próximos o para rellenar pequeños huecos.

2.5.4.1.4 Adelgazamiento

El "*thinning*" o adelgazamiento es otro tipo de operación morfológica, consistente en obtener la línea media o eje de un objeto. Este tipo de procedimiento reduce los objetos a un conjunto de líneas sin perderse la conectividad del objeto original. Resulta de gran utilidad su aplicación al estudio de longitudes y orientaciones de estructuras alargadas, por ejemplo: dendritas neuronales, microfisuras en rocas, etc.

2.5.5 INTRODUCCIÓN AL ENTORNO MATLAB

La estructura básica de datos en MatLab es una matriz, un conjunto ordenado de elementos reales. Este tipo es naturalmente adecuado para la representación de imágenes, vistas éstas como un conjunto de píxeles con información de color o intensidad de gris. MatLab almacena la mayoría de tipos de imágenes que soporta como matrices bidimensionales, cada elemento de la cual hace referencia a un píxel de la imagen representada. Por ejemplo, una imagen compuesta por un conjunto de 144 filas y 176 columnas de píxeles puede ser almacenada en MatLab en una matriz bidimensional de 144x176 o tridimensional si es imagen color de 144x176x3. Esta posibilidad de almacenamiento supone que trabajar con imágenes en MatLab sea similar a trabajar con cualquier otro tipo de dato matricial, por lo que sería posible seleccionar un determinado píxel de la imagen mediante el formato típico de acceso a un elemento de una matriz.

2.5.5.1 Imágenes RGB

Una imagen RGB representa el color de cada píxel como un conjunto de tres valores que contienen la intensidad de rojo, verde y azul para conformar el color. En MatLab, las componentes de color rojo, verde y azul de la imagen RGB residen en una matriz $m \times n \times 3$, donde m y n son el número de filas y de columnas de la imagen, y la tercera componente representa la dimensión, indicando que consta en este caso de 3 planos, conteniendo los colores de rojo, verde y azul de cada píxel ^[4].

2.5.5.2 Imágenes Binarias

En una imagen binaria, cada píxel puede tomar uno de dos posibles valores discretos. Esencialmente, estos dos valores corresponden a *on*, *off*. Una imagen binaria es almacenada como una matriz bidimensional de ceros (píxeles *off*) y de unos (píxeles *on*) ^[1].

En el *Toolbox* de Procesado de Imagen cada función que retorna una imagen binaria como una matriz lógica. El *Toolbox* utiliza la etiqueta lógica para significar

que el rango es [0,1]. Si la etiqueta lógica está *off* se asume que el rango es [0,255].

2.5.6 COMANDOS DE MATLAB UTILIZADOS PARA EL PROCESAMIENTO DE IMÁGENES.

Cargar Imágenes en Matrices:

```
A=imread(NombreImagen,Formato);
imagen=imread('vale.tif','tif');
```

Donde *NombreImagen* representa el nombre de la imagen con el *path* completo y *Formato* representa el formato de la imagen ^[6].

Visualizar imágenes a partir de una matriz:

```
imshow(uint8(A))
```

Donde *A* es la matriz que representa la imagen y *uint8* se utiliza para convertir los elementos de *A* en enteros de 8 bits (256 niveles de gris). Si la imagen ya está en formato *uint8* no es necesario usar la función '*uint8*'.

Visualizar más de una imagen en una misma ventana:

```
subplot(2,2,1);imshow(uint8(A))
subplot(2,2,2);imshow(uint8(B))
subplot(2,2,3);imshow(uint8(C))
subplot(2,2,4);imshow(uint8(D))
```

El siguiente comando lee el fichero especificado en formato *TIF* y retorna la imagen en el *array x*, así como su mapa de colores en el *array map*. Nótese que el comando terminado con punto y coma para que no se visualice todo el proceso intermedio que ello conlleva, de lo contrario, se convertiría en una operación que muestra en pantalla todos los valores de luminancia dentro de la imagen ^[6].

```
[imagen,map]=imread('vale.tif','tif');
```

Con la siguiente orden visualizamos la imagen que acabamos de leer (aquí es indistinto terminar o no con punto y coma) ^[10]:

```
imshow(imagen,map)
```

Para convertir la imagen anterior, a una imagen en escala de grises (conteniendo sólo la información de la luminancia) y almacenarla en el *array* *i*, se utiliza la siguiente orden. Los valores de luminancia resultantes van desde 0 (negro) a 1 (blanco) y cada pixel es cuantificado con 8 bits (256 niveles de gris) ^[11].

$$i = \text{rgb2gray}(\text{imagen})$$


Figura 2.18 a) Imagen Original; b) Imagen Escala de Grises ^[*]

2.5.6.1 Funciones Para la Extracción de Bordes

En visión computacional es de utilidad para hacer reconocimiento de objetos o bien para segmentar regiones, extraer los bordes de objetos (que en teoría delimitan sus tamaños y regiones). La función *edge* da la posibilidad de obtener los bordes de la imagen. El formato de esta función es:

$$\text{Imagen_bordes} = \text{edge}(\text{Image}, \text{algoritmo});$$

Donde *Imagen_bordes* es la imagen obtenida con los bordes extraídos, *Image* es la variable que contiene la imagen en escala de grises a la cual se pretende recuperar sus bordes, mientras que *algoritmo* puede ser uno de los utilizados por el MatLab: *Sobel*, *Prewitt*, *Roberts*, *Laplacian of a Gaussian (LoG)*, *Zero Crossings*, *Canny*.

$$[g, t] = \text{edge}(f, \text{'método'}, T, \text{dir});$$

Donde: *f* es la imagen, *T* es el límite (opcional), *método* identifica el algoritmo deseado, *dir* es la dirección de filtrado: *'horizontal'*, *'vertical'* ó *'both'* (opción por defecto); *g* es la imagen de ejes y *t* es el límite usado.

* Imagen creada por los autores.

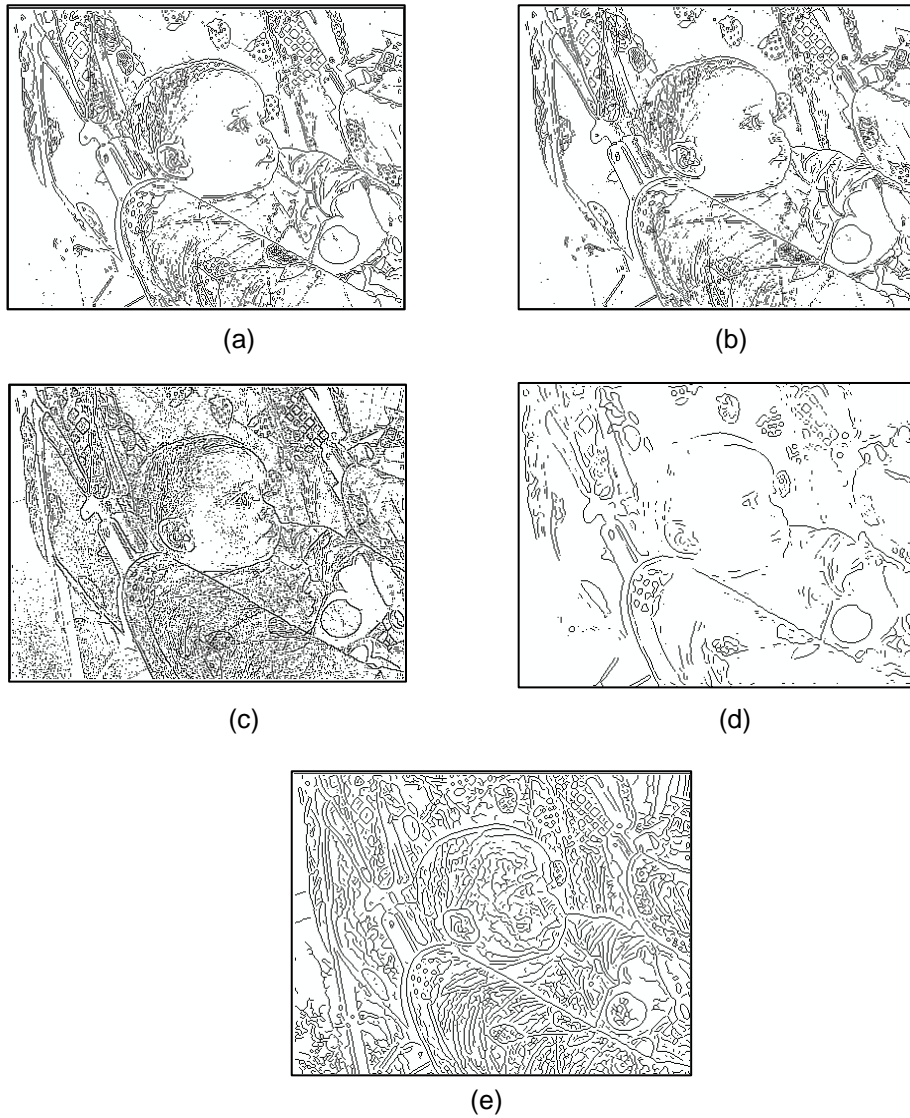


Figura 2.19 a) Extracción de bordes Utilizando el Filtro de Sobel b) Filtro de Prewitt
c) Filtro de Roberts d) Filtro LoG e) Filtro de Canny ^[*]

2.5.6.2 Operaciones Morfológicas

La función *bwmorph* implementa una serie de funciones útiles basadas en la combinación de la dilatación y erosión. Su sintaxis es:

$$g = \text{bwmorph}(f, \text{operación}, n)$$

Donde *f* es una imagen binaria a la cual se va a realizar la operación, operación establece la operación morfológica que se desea realizar sobre la imagen, y *n* es un número positivo que especifica el número de veces que se desea que la

* Imagen creada por los autores.

operación sea repetida, n es opcional y puede ser omitida; en este caso se realizará en una ocasión la operación seleccionada ^[11].

Operación	Descripción
<i>Bridge</i>	Conecta los píxeles separados únicamente por un píxel.
<i>Clean</i>	Remueve píxeles aislados fuera de la región.
<i>Close</i>	Cierra las regiones utilizando un elemento estructurante 3x3
<i>Dilate</i>	Realiza la dilatación utilizando un elemento estructurante de 3x3.
<i>Erode</i>	Realiza la erosión utilizando un elemento estructurante de 3x3.
<i>Fill</i>	Llena píxeles interiores de una región
<i>Open</i>	Abre las regiones utilizando un elemento estructurante 3x3
<i>remove</i>	Remueve píxeles interiores que no tienen conexión
<i>Shrink</i>	Encoje objetos que no contengan huecos en su interior.
<i>Skel</i>	Saca el esqueleto de una imagen
<i>Thin</i>	Aedelgaza regiones sin huecos en su interior

Tabla 2.1 Operaciones Morfológicas en MatLab ^[10]

Continuación se presenta los resultados de realizar algunas operaciones morfológicas utilizando el MatLab.

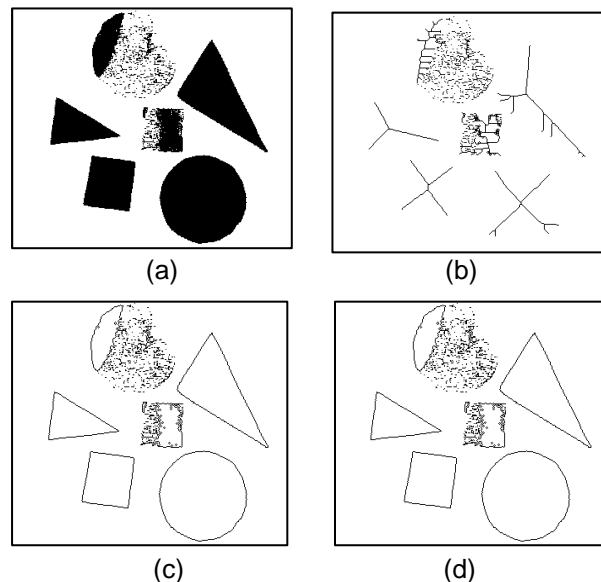


Figura 2.20 a) Imagen Original; b) Imagen Resultado de la Operación Skel
c) Imagen Resultado de la Operación Close; d) Imagen Resultado de la Operación Remove ^[8]

2.5.6.3 Operaciones Basadas en Objetos

En una imagen binaria, puede definirse un objeto como un conjunto de píxeles conectados con valor 1. Para muchas operaciones la distinción de objetos depende de la convención utilizada de conectividad, que es la forma en la que se considera si dos píxeles tienen relación como para considerar que forman parte del mismo objeto. La conectividad puede ser de dos tipos, de conexión-4 ó bien conexión-8. En la Figura 2.21 se esquematizan ambas conectividades. En la conectividad conexión-8 se dice que el píxel rojo pertenece al mismo objeto si existe un píxel de valor uno en las posiciones 1, 2, 3, 4, 5, 6, 7 y 8. Por su parte la conectividad conexión-4 relaciona solo a los píxeles 2, 4, 5 y 7.

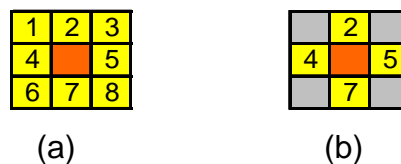


Figura 2.21 a) Conectividad Conexión 8; b) Conectividad Conexión 4 ^[6]

Para las operaciones que consideran conectividad como un parámetro es importante tomar en cuenta que esta determina fuertemente el resultado final del procesamiento puesto que puede dar origen a objetos nuevos en donde si se hubiera elegido otra conectividad no existirán. Para explicar lo anterior se considera la Figura 2.22. Como puede verse, si se elige la conectividad conexión-8, la figura contenida en la imagen sería considerada como una sola, pero si al contrario se elige la conectividad conexión-4, sería vista como 2 objetos diferentes ^[6].

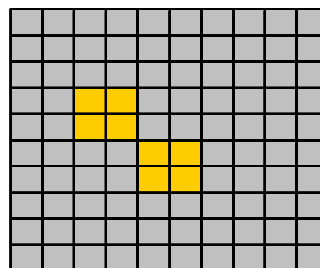


Figura 2.22 Problema de Elegir Conectividad ^[6]

La función *bwlabel* realiza un etiquetado de los componentes existentes en la imagen binaria, la cual puede ser considerada como una forma de averiguar

cuantos elementos (considerando como elementos agrupaciones de unos bajo alguno de los dos criterios de conectividad) están presentes en la imagen. La función tiene el siguiente formato:

$$\text{ImagenR}=\text{bwlablel}(\text{Image})$$

Donde *ImagenR* es la imagen resultado que contiene los elementos etiquetados con el número correspondiente al objeto, *ImagenS* es la imagen binaria que se desea encontrar el número de objetos y conectividad puede ser 4 o 8 (correspondiendo al tipo de conectividad anteriormente explicado). El efecto de esta función puede explicarse fácilmente si se analiza la imagen original y el resultado de aplicar la función *bwlabel*.

Supongamos que la Figura 2.23 (a) es la imagen binaria original (*ImagenR*) y 2.23 (b) es la imagen resultado (*ImagenS*) de ejecutar la función *bwlabel*.

La imagen resultado asigna a cada píxel perteneciente a un determinado objeto según su conectividad la etiqueta perteneciente al número de objeto mientras que los píxeles en cero no tienen efecto en la operación, en la Figura 2.23 esto corresponde a la codificación de colores mostrada en 2.23 (c) ^[10]

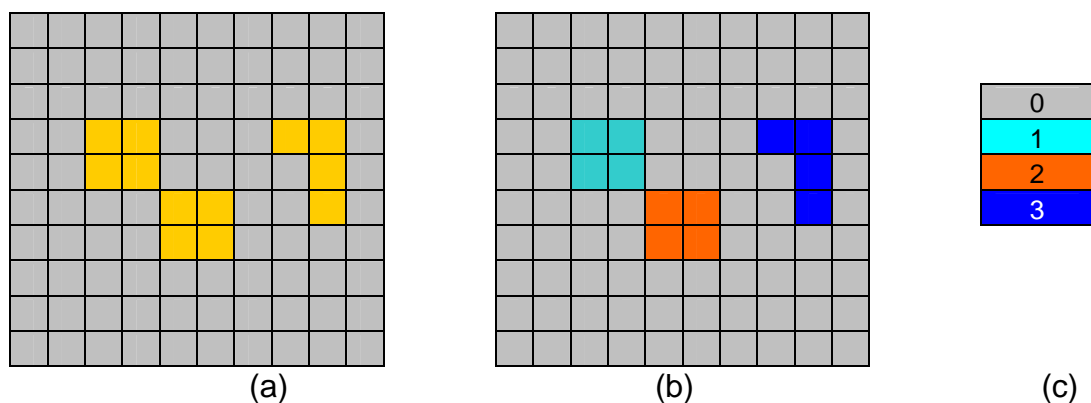


Figura 2.23 a) Imagen Original Binaria; b) Imagen Resultado operación *bwlabel* con Conectividad 4
c) La codificación de Color ^[10]

La imagen resultado es del tipo *double*. Además debido a su contenido (los valores son muy pequeños) no puede ser desplegada por la función *imshow*. Una técnica útil para la visualización de este tipo de matrices es la de utilizar pseudo color en forma de una imagen.

CAPÍTULO 3

MARCO TEORICO DEL MANIPULADOR ROBÓTICO

3.1 CONCEPTOS Y DEFINICIONES

Robot.- El Instituto Norteamericano de Robótica (RIA) define: "Un robot es un manipulador reprogramable multifuncional, diseñado para mover materiales, piezas, herramientas o dispositivos especiales mediante movimientos programados y variables, que permiten realizar diversas tareas" [12]. Las características que distinguen a un robot de otros dispositivos manipuladores, son la multifuncionalidad (versatilidad para ejecutar diversas tareas, no previstas inicialmente por su diseñador) y la programabilidad (capacidad de realización de una tarea a otra cambiando el programa que debe ejecutar) [12].

Robótica.- Se define como "La ciencia que estudia los robots como sistemas que operan en algún entorno real, estableciendo algún tipo de conexión inteligente entre percepción y acción" [13].

Articulación.- Elemento que une dos eslabones consecutivos y les confiere movilidad relativa con uno o más grados de libertad.

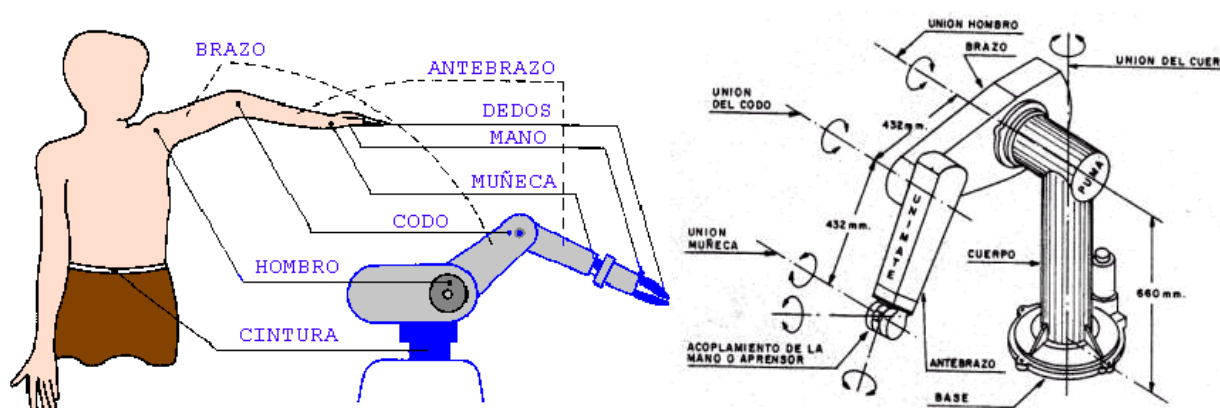


Figura 3.1 Analogía Antropomórfica de Eslabones y Articulaciones [4]

♣ Imagen obtenida de http://dis.um.es/~barzana/Imagenes/Comparación_Robot.gif

Eslabón.- Cada uno de los elementos que, unidos mediante articulaciones, conforman la cadena cinemática del robot.

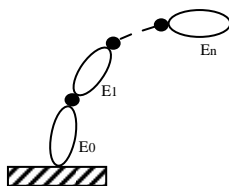


Figura 3.2 Cadena Cinemática Abierta ^[*]

Grado de Libertad.- Cada uno de los movimientos independientes que puede realizar una articulación y que fijan la situación del elemento terminal.

Volumen o Espacio De Trabajo.- Es el conjunto de puntos del espacio alrededor del robot industrial que es capaz de abarcar con el extremo de su elemento terminal, depende de la morfología del robot.

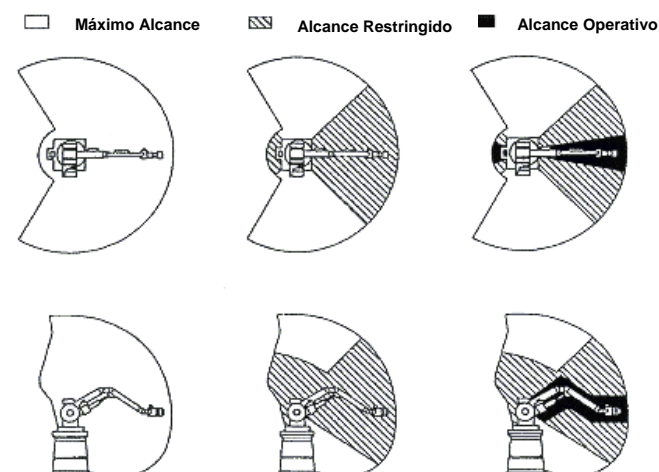


Figura 3.3 Volumen y Alcances de Trabajo de un Manipulador Industrial ^[*]

Capacidad de Carga.- Máxima masa que puede movilizar en cualquier configuración de su estructura mecánica.

Movilidad.- Se determina por el número de grados de libertad del Robot.

Agilidad.- Se mide por dos parámetros: la velocidad máxima y las aceleraciones y deceleraciones máximas.

* Imagen creada por los autores.

♣ Imagen obtenida de www.osha.gov/dts/osta/otm/otm_iv_4fig06.gif

Precisión Y Repetitividad.- La precisión de un robot es su capacidad para posicionar el elemento terminal en un punto arbitrario predeterminado de su espacio de trabajo. La repetitividad es la medida de la habilidad del robot para situar su elemento terminal en un mismo punto repetidas veces ^[14].

Entorno de Trabajo.- La aplicación concreta o el entorno de trabajo al que se destine un robot.

3.2 CLASIFICACIÓN Y MORFOLOGÍA DEL ROBOT

3.2.1 CLASIFICACIÓN DEL ROBOT

Existen varias clasificaciones de acuerdo a diversas fuentes, por ejemplo ^[14]:

- Según el tipo de interacción:
 - Robot de servicios • Robot industrial
- Según como se controlan, los robots industriales se clasifican en:
 - Robots de secuencia controlada
 - Robots teleoperados
 - Robots de trayectoria operada / continua: Robots que operan de acuerdo a las especificaciones de la trayectoria requerida para alcanzar la próxima posición deseada, normalmente alcanzada por interpolación. La trayectoria es continua punto a punto, sin conocimiento sobre su entorno.
 - Robots adaptativos: Robot que tiene un control sensorial, adaptativo o funciones para control mediante aprendizaje.

3.2.2 MORFOLOGÍA DEL ROBOT

Desde el punto de vista morfológico, se clasifica a los robots industriales según dos parámetros ^[14]:

- Según el número de grados de libertad (robots con 3, 4, 5 o más g.d.l.)
- Según el tipo de estructura mecánica:
 - Cartesianos.
 - SCARA.

- Antropomórficos o angulares.
- Paralelos.
- Esféricos.
- Cilíndricos.

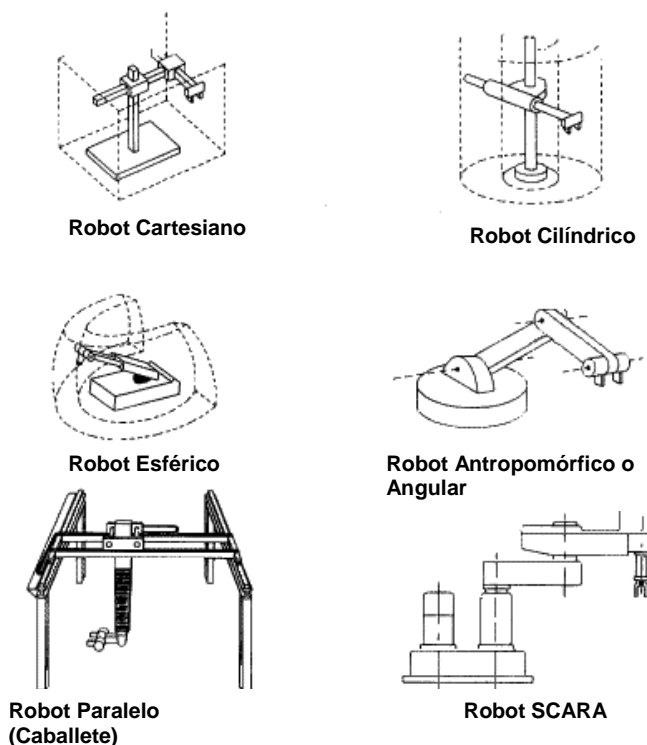
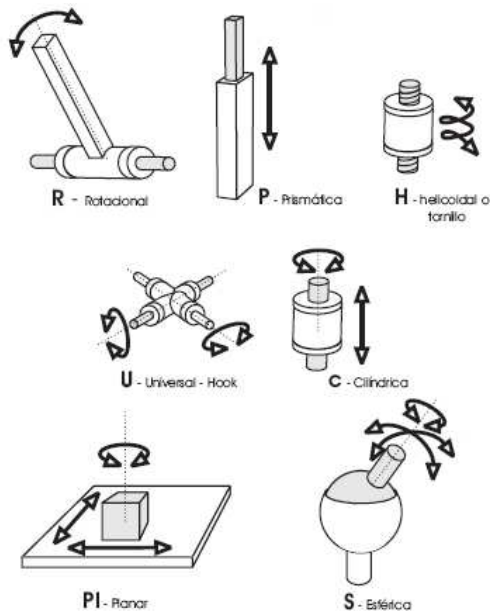


Figura 3.4 Clasificación por Estructura Mecánica [4]

3.2.3 TIPOS DE ARTICULACIONES

Existen diferentes tipos de articulaciones [14], las más utilizadas en robótica son las que se indican en la Figura 3.5.



4 Imagen obtenida de www.osha.gov/dts/osta

Figura 3.5 Tipos de Articulaciones Robóticas^[*]

3.2.4 ELEMENTOS EFECTORES TERMINALES

El órgano o elemento terminal del robot industrial es el dispositivo o herramienta que se coloca en su muñeca (en el caso antropomórfico) y que le permite realizar la función específica para la que ha sido programado.

Los tipos de herramientas de agarre se clasifican según cómo realizan dicha aprehensión y son las siguientes:

3.2.4.1. Pinzas a Presión

El mecanismo que implemente la herramienta proporciona dos variantes de aplicación de presión, según si el movimiento de los dedos es lineal o angular, o bien si éstos son flexibles o adaptables.

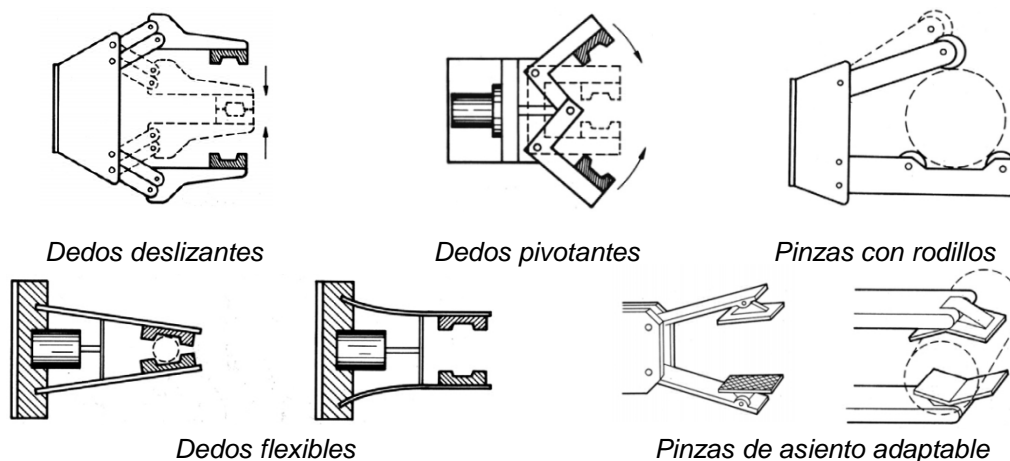


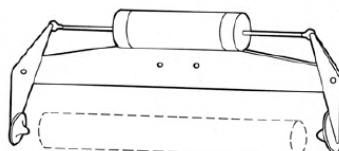
Figura 3.6 Tipología de elementos terminales^[*]

3.2.4.2 Sujeción por Enganche

El órgano terminal dispone de unos elementos de retención que permiten sujetar la carga sin necesidad de realizar presión sobre ella. Dichas cargas disponen

* Imagen obtenida de www.osha.gov/dts/osta/otm/otm_iv_4fig09.gif

♣ Imagen obtenida de www.osha.gov/dts/osta/otm/otm_iv



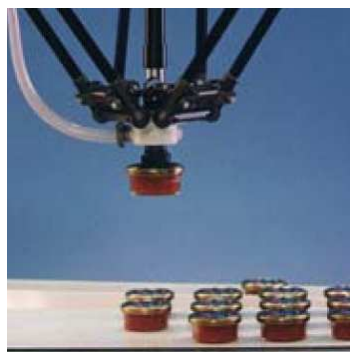
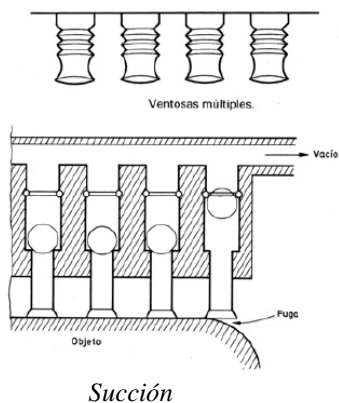
también de unos puntos de sujeción, o bien su geometría es tal que deben ser sujetos por el robot en una orientación determinada.

Figura 3.7 Sujeción por enganche ^[8]

3.2.4.3 Sujeción por Contacto

Si la sujeción no se produce con presión o enganche, se tiene el contacto entre la herramienta y la pieza o carga recurriendo a:

- **Succión:**



Ventosa en un robot IRB 340 'flexpicker' de ABB

Figura 3.8 Sistemas de Succión

- **Atracción magnética:** Este método utiliza un electroimán, aunque a veces también se implementa con un imán permanente que se separa de la pieza por medios neumáticos. Es evidente que la pieza debe ser metálica.

- **Elementos adhesivos:** Se usan compuestos químicos para adherir el elemento terminal a la pieza y trasladarla.

3.3 MOTORES CON SERVOMECANISMO

Un servomecanismo es un actuador mecánico (generalmente un motor con reducción mecánica de velocidad), que posee los elementos de control para monitorizar los parámetros de su actuación mecánica, como su posición, velocidad, torque, etc ^[15].

* Imagen obtenida de www.osha.gov/dts/osta/otm/otm_iv_4fig12.gif



Figura 3.9 Diversas clases de servos, incluyendo lineales ^[15]

Los servomotores que se utilizan comúnmente en modelismo se encuentran disponibles en el comercio local y son económicos, lo que los hace de más fácil acceso cuando se trata de la construcción de proyectos de robótica a baja escala.

3.3.1 SERVO ANALÓGICO PARA MODELISMO

Estos servomotores se componen de un motor de corriente continua, un juego de engranajes para la reducción de velocidad y aumento de torque, un potenciómetro ubicado sobre el eje de salida (que se usa para conocer la posición) y un circuito para el control.



Figura 3.10 Despiece de un servomotor de modelismo ^[15]

Los servos de este tipo utilizan un sistema de realimentación de posición con potenciómetro, el mismo que entrega un valor diferente a su salida según cual sea su posición en grados. Reciben su alimentación de energía para el motor y el circuito del control por los terminales de color rojo y negro (generalmente) y su señal de control, que determina la posición que se requiere, por el terminal blanco o amarillo. La alimentación de estos servos está entre 4,8 y 6 VDC.

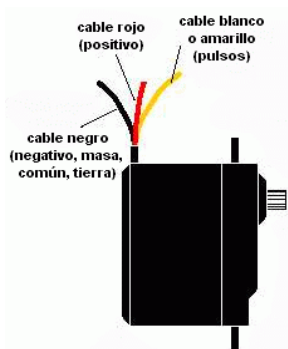


Figura 3.11 Cables terminales de un servomotor de modelismo ^[15]

El estándar de la señal de control para todos los servos de este tipo, elegido para facilitar el uso en radiocontrol, es un pulso de onda cuadrada de 1,5 milisegundos que se repite a un ritmo de entre 10 a 22 ms. Mientras el pulso se mantenga en ese intervalo, el servo se ubicará en la posición central de su recorrido ^[18]. Si el ancho de pulso disminuye, el servo se mueve de manera proporcional hacia un lado. Si el ancho de pulso aumenta, el servo gira hacia el otro lado.

Generalmente el rango de giro de un servo de este tipo cubre entre 90° y 180° de la circunferencia total, según la marca y modelo.

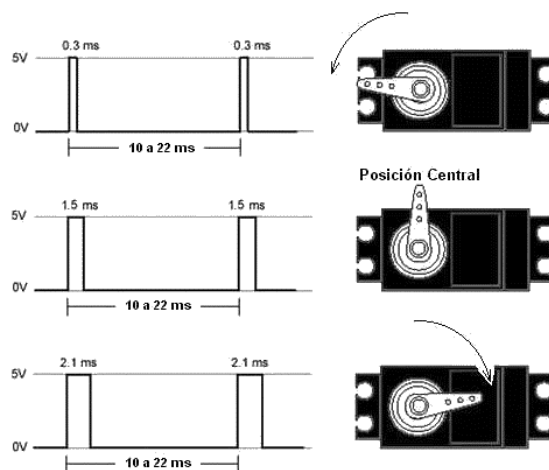


Figura 3.12 Señal PWM típica en un servomotor ^[15]

3.3.2 CONTROLADOR ELECTRÓNICO

La empresa *Scott Edwards Electronics Inc.* ha desarrollado el módulo para control de servomotores *Mini SSC II*, con interfaz RS-232. Este circuito está basado en el microcontrolador PIC 16C621 en encapsulado SMD y posee ocho salidas PWM para el control de un número igual de servomotores simultáneamente ^[16].

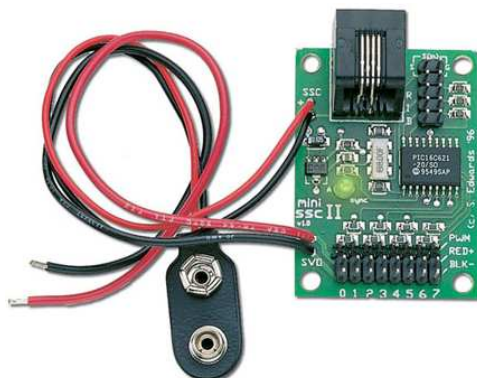


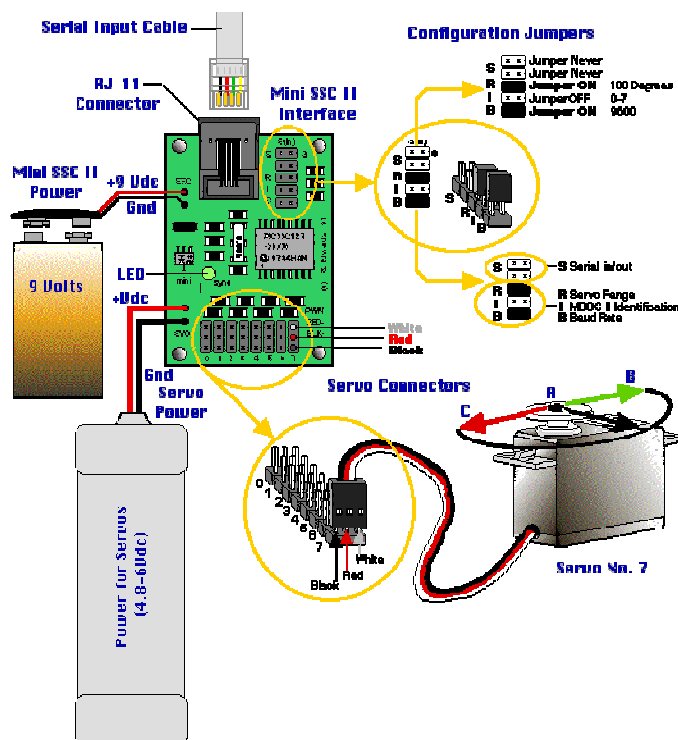
Figura 3.13 Controlador de servos Mini SSC II ^[16]

El módulo permite, mediante una breve manipulación de jumpers de configuración, una rápida implementación del hardware necesario para el control de los servomotores.

El fabricante especifica en su manual de instrucciones las configuraciones que deben realizarse para ajustar el módulo controlador a las condiciones operativas que se requieran, como son el poder seleccionar entre una capacidad de giro de 90 o de 180 grados, la velocidad de transmisión serial y la posibilidad de expandir el sistema hasta 255 servomotores. Para el caso del brazo robótico diseñado, se seleccionaron las siguientes condiciones:

Capacidad de giro:	180°
Velocidad de transmisión:	9800 baudios ^[a]
Máximo número de servos:	8 (0-7)

El sistema debe contar con dos fuentes de alimentación, una de 9 voltios para el circuito y la otra de 4,8 a 6 voltios para los servomotores.



^[a] Unidad de la velocidad de transmisión de señales, equivalente a un bit por segundo.

Figura 3.14 Conexiones y configuración de jumpers^[16]

Para la programación del software, se envían secuencialmente a través del puerto serial, tres Bytes de información que serán interpretados por el módulo de la siguiente forma:

Byte 1	Byte 2	Byte 3
[sync marker (255)]	[servo # (0-254)]	[position (0-254)]

El primer Byte es para sincronización, el segundo indica el servo que será controlado y el tercero brinda la información de la posición deseada del eje del servo en un rango de 0 a 254, lo que dará como resultado (según el manual del fabricante) una respuesta de 0,72 grados de movimiento para cada variación de posición; es importante anotar que la variación de este movimiento es lineal.

3.4 MODELO CINEMÁTICO DE LOS MANIPULADORES.

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia.

Existen dos problemas fundamentales a resolver en la cinemática del robot, el primero se conoce como el *Problema Cinemático Directo* y consiste en determinar cuál es la posición y orientación del extremo final del robot con respecto a un sistema de coordenadas que se toma como referencia, conocidos los valores de las articulaciones y los parámetros geométricos de los elementos del robot; el segundo, denominado *Problema Cinemático Inverso*, resuelve la configuración que debe adoptar el robot, conocidas la posición y orientación del elemento extremo.

Denavit y Hartenberg (DH), en 1955 propusieron un método sistemático para describir y representar la geometría espacial de los elementos de la cadena cinemática de un robot con respecto a un sistema de referencia fijo^[17]. Este método utiliza una matriz de transformación homogénea para describir la relación espacial entre dos eslabones adyacentes, reduciéndose el problema cinemático directo a encontrar una matriz de transformación homogénea 4 x 4 que relacione la localización espacial del extremo del robot con respecto al sistema de coordenadas de su base.

3.4.1 EL PROBLEMA CINEMÁTICO DIRECTO

Un robot se puede considerar como una cadena cinemática formada por objetos rígidos o eslabones unidos entre sí mediante articulaciones, por lo tanto se puede establecer un sistema de referencia fijo situado en la base del robot y describir la localización de cada uno de los eslabones con respecto a dicho sistema de referencia. De esta forma, el problema cinemático directo consiste en encontrar una matriz homogénea de transformación T que relacione la posición y orientación del extremo del robot respecto del sistema fijo situado en la base del mismo. Esta matriz T será función de las coordenadas articulares ^[17].

3.4.1.1 Rotación y Traslación de un Sistema de Coordenadas

Para analizar la rotación de un sistema de coordenadas, se consideran dos sistemas con un mismo origen, rotando uno con respecto al otro (sistema ABC con respecto a XYZ) ^[17]:

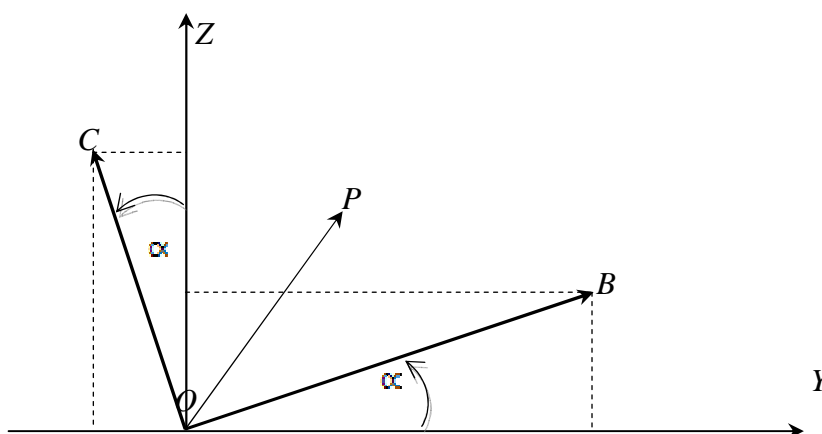


Figura 3.15 Sistema ABC girando sobre XYZ, en α grados sobre el eje X

El punto P sobre los ejes ABC se representa por el vector:

$$P = ai + bj + ck \quad \text{Ec. 3.1}$$

para obtener el vector P en función de las coordenadas XYZ se realiza el producto escalar de P con los vectores unitarios del sistema XYZ . Para la rotación sobre el eje X , respecto al ángulo α , se tiene:

$$x = P \cdot i_x = a \quad \text{Ec. 3.2}$$

$$y = P \cdot j_y = b \cos(\alpha) - c \operatorname{sen}(\alpha) \quad \text{Ec. 3.3}$$

$$z = P \cdot k_z = b \operatorname{sen}(\alpha) - c \cos(\alpha) \quad \text{Ec. 3.4}$$

En forma matricial:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\operatorname{sen}(\alpha) \\ 0 & \operatorname{sen}(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \quad \text{Ec. 3.5}$$

La matriz de 3x3 se denomina matriz de rotación. De la misma forma, se tiene las matrices de rotación sobre los ejes Y, Z:

$$R_{Y,\beta} = \begin{bmatrix} \cos(\beta) & 0 & \operatorname{sen}(\beta) \\ 0 & 1 & 0 \\ -\operatorname{sen}(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad \text{Ec. 3.6}$$

$$R_{Z,\gamma} = \begin{bmatrix} \cos(\gamma) & -\operatorname{sen}(\gamma) & 0 \\ \operatorname{sen}(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 3.7}$$

Si se deben realizar varias rotaciones con respecto al sistema de referencia, la matriz total de rotación se obtiene premultiplicando la primera matriz de rotación por la segunda y así sucesivamente:

$$R_{\text{TOT}} = R_1 \cdot R_2 \cdot \dots \cdot R_{n-1} \cdot R_n \quad \text{Ec. 3.8}$$

Como se requiere la información para la posición y orientación del elemento terminal del robot con respecto al sistema de referencia de las coordenadas de su base, se define el vector de acercamiento \hat{a} como el vector unitario en la dirección de acercamiento del elemento terminal, el vector de orientación \hat{o} como el vector unitario perpendicular tanto a \hat{a} como a los dedos deslizantes del elemento terminal y el vector \hat{u} , definido como el producto cruz entre \hat{a} y \hat{o} :

$$\hat{u} = \hat{o} \otimes \hat{a} \quad \text{Ec. 3.9}$$

Refiriendo el sistema de componentes $\hat{a}, \hat{o}, \hat{u}$ al sistema de coordenadas de referencia de la base, se tiene la matriz de rotación:

$$R = \begin{bmatrix} u_x & o_x & a_x \\ u_y & o_y & a_y \\ u_z & o_z & a_z \end{bmatrix} \quad \text{Ec. 3.10}$$

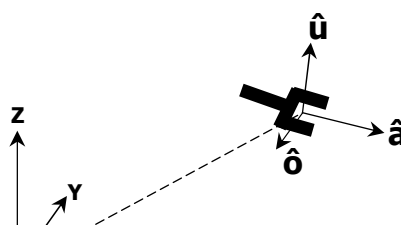


Figura 3.16 Vectores $\hat{a}, \hat{b}, \hat{c}$ del elemento terminal

La traslación del origen de un sistema de coordenadas con respecto a otro se representa con la suma de vectores.

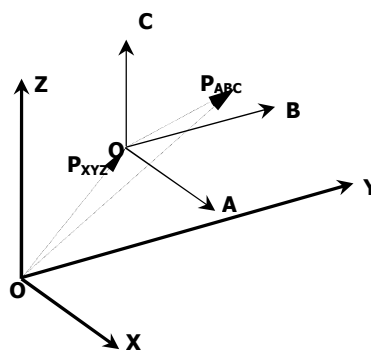


Figura 3.17 Traslación del sistema ABC con respecto a XYZ sin rotación

De la Figura 3.17, si se traslada ABC con respecto a XYZ desde su punto de coincidencia O , se tiene que el punto:

$$P_{abc} = a i_A + b j_B + c k_C \quad \text{Ec. 3.11}$$

se convierte en:

$$P_{XYZ} = (a + d_1) i_X + (b + d_2) j_Y + (c + d_3) k_Z \quad \text{Ec. 3.12}$$

En forma matricial se tiene:

$$P_{XYZ} = \begin{bmatrix} P_X \\ P_Y \\ P_Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d_1 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} \quad \text{Ec. 3.13}$$

La matriz 4x4 es la matriz homogénea de traslación $[T_r]$:

$$P_{XYZ} = [T_r] P_{ABC} \quad \text{Ec. 3.14}$$

Donde,

$$T_r = \begin{bmatrix} 1 & 0 & 0 & d_1 \\ 0 & 1 & 0 & d_2 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 3.15}$$

En robótica, para representar rotación y traslación en una sola matriz se utilizan las denominadas matrices homogéneas de transformación, donde en la última fila los elementos son cero, excepto el último que es igual a uno ^[17].

De la Figura 3.16, utilizando la forma de matriz homogénea, la orientación del elemento terminal con respecto al sistema de coordenadas de referencia, donde el superíndice indica la referencia y el subíndice el sistema desplazado, es:

$${}^0R_m = \begin{bmatrix} u_x & o_x & a_x & 0 \\ u_y & o_y & a_y & 0 \\ u_z & o_z & a_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 3.16}$$

Igualmente, la posición del origen de ABC con respecto al de XYZ es:

$${}^0P_m = \begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 3.17}$$

De esta manera, el elemento terminal ha llegado a la orientación y posición indicadas por medio de una rotación, seguida por una traslación respecto al mismo sistema de referencia, por lo que para obtener la matriz total con ambas operaciones se premultiplica de la siguiente forma:

$${}^0T_m = {}^0T_{r_m} {}^0R_m = \begin{bmatrix} u_x & o_x & a_x & P_x \\ u_y & o_y & a_y & P_y \\ u_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} [R] & P \\ 0 & 1 \end{bmatrix} \quad \text{Ec. 3.18}$$

donde la submatriz 3x3 corresponde a la matriz de rotación y la submatriz 3x1 corresponde a la matriz de traslación. Esta matriz homogénea de transformación, conocida también como matriz de equivalencia simbólica, transforma los puntos dados en el sistema de coordenadas del elemento terminal P_m al sistema de coordenadas base P_o :

$$P_o = {}^0T_m P_m \quad \text{Ec. 3.19}$$

Por facilidad, normalmente se premultiplican las matrices en el orden en que se efectúan las operaciones de movimiento, pero esto no es estrictamente necesario si las operaciones se realizan con el sistema de referencia en movimiento.

3.4.1.2 Parámetros Denavit-Hartenberg

Denavit y Hartenberg desarrollaron en 1955 las matrices de parámetros eslabón para el análisis de las operaciones con respecto al sistema en movimiento.

En general, un robot de n grados de libertad está formado por n eslabones unidos por n articulaciones, de forma que cada par articulación-eslabón constituye un grado de libertad. A cada eslabón se le puede asociar un sistema de referencia solidario a él y, utilizando las transformaciones homogéneas, es posible representar las rotaciones y traslaciones relativas entre los distintos eslabones que componen el robot [18].

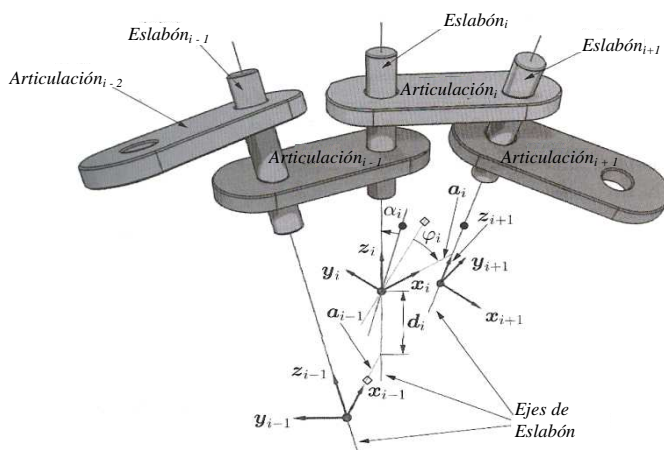


Figura 3.18 Pares articulación-eslabón [*]

Normalmente, la matriz de transformación homogénea que representa la posición y orientación relativa entre los sistemas asociados a dos eslabones consecutivos del robot se denomina matriz ${}^{i-1}A_i$. Así pues, 0A_1 describe la posición y orientación del sistema de referencia solidario al primer eslabón con respecto al sistema de referencia solidario a la base, 1A_2 describe la posición y orientación del segundo

* Imagen obtenida de http://webcourse.cs.technion.ac.il/236927/Winter2007-2008/en/ho_Lectures.html

eslabón respecto del primero, etc. Del mismo modo, se denomina 0A_k a las matrices resultantes del producto de las matrices ${}^{i-1}A_i$ (con i desde 1 hasta k), se puede representar de forma total o parcial la cadena cinemática que forma el robot. Así, por ejemplo, la posición y orientación del sistema solidario con el segundo eslabón del robot con respecto al sistema de coordenadas de la base se puede expresar mediante la matriz 0A_2 :

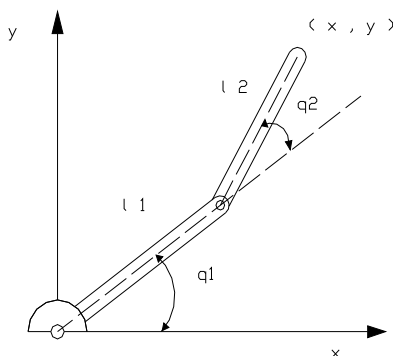


Figura 3.19 Robot Planar de 2 Grados de Libertad

$$\begin{aligned}
 {}^{i-1}A_i &= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & 0 \\ s\theta_i & c\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_i & -s\alpha_i & 0 \\ 0 & s\alpha_i & c\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\
 &= \begin{bmatrix} c\theta_i & c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

Ec. 3.20

Donde $\theta_i, a_i, d_i, \alpha_i$ son los parámetros D-H del eslabón i .

De este modo, basta con identificar los parámetros $\theta_i, a_i, d_i, \alpha_i$ para obtener las matrices A y relacionar así todos y cada uno de los eslabones del robot.

Para que la matriz ${}^{i-1}A_i$ relacione los sistemas S_i y S_{i-1} , es necesario que los sistemas se hayan escogido de acuerdo a unas normas determinadas. Éstas, junto con la definición de los 4 parámetros de Denavit-Hartenberg, conforman el siguiente algoritmo para la resolución del problema cinemático directo ^[18] (ejemplo

aplicado al prototipo del modelo de brazo robótico diseñado para el presente proyecto).

3.4.1.3 Algoritmo de Denavit-Hartenberg para la obtención del modelo matemático cinemático directo:

- Elegir un sistema de coordenadas fijo (X_0, Y_0, Z_0) asociado a la base del robot,

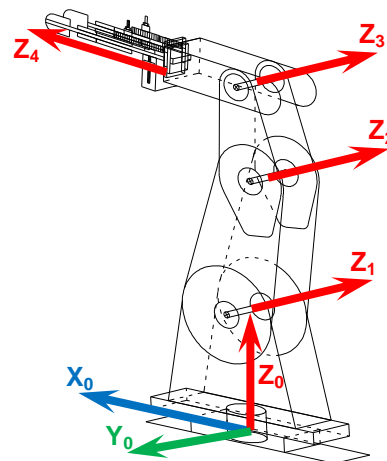
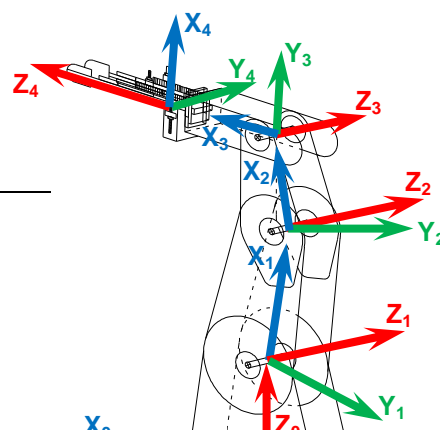


Figura 3.20 Elección de sistemas de coordenadas ^[8]

- Localizar el eje de cada articulación Z_i : Si la articulación es rotativa, el eje será el propio eje de giro, si es prismática, el eje lleva la dirección de deslizamiento.
- Situar los ejes X_i en la línea normal común a Z_{i-1} y Z_i . Si éstos son paralelos, se elige sobre la línea normal que corta a ambos ejes. El eje Y_i debe completar un triedro dextrógiro.



* Imagen creada por los autores.

Figura 3.21 Ubicación de los ejes ^[*]

- Parámetro α_i : ángulo que forman los ejes X_{i-1} y X_i girando sobre el eje Z_{i-1} positivamente. El signo lo determina la regla de la mano derecha. Para una articulación rotatoria, α es variable y los otros parámetros constantes.

ARTICULACIÓN	α_i	a_i	θ_i	d_i
1	α_1			
2	α_2			
3	α_3			
4	α_4			
5	α_5			

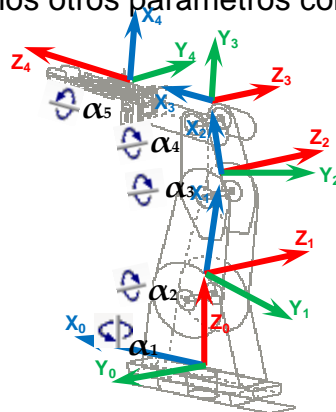
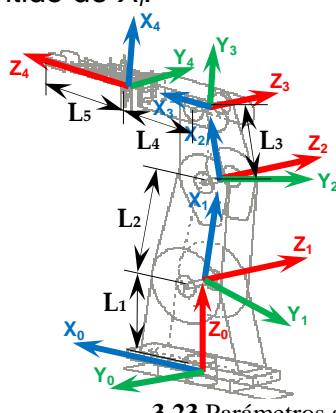


Figura 3.22 Parámetros α_i ^[*]

- Parámetro a_i : distancia entre los ejes Z_{i-1} y Z_i a lo largo de la dirección positiva de X_i . El signo lo define el sentido de X_i .

ARTICULACIÓN	α_i	a_i	θ_i	d_i
1	α_1	L_1		
2	α_2	L_2		
3	α_3	L_3		
4	α_4	L_4		
5	α_5	L_5		



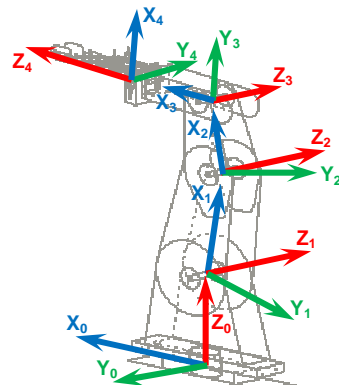
3.23 Parámetros a_i ^[*]

Figura

* Imagen creada por los autores.

- Parámetro θ_i : ángulo entre el eje Z_{i-1} y Z_i , sobre la dirección positiva de X_i . El signo lo determina la regla de la mano derecha.

ARTICULACIÓN	α_i	a_i	θ_i	d_i
1	α_1	L_1	90°	
2	α_2	L_2	0	
3	α_3	L_3	0	
4	α_4	L_4	0	
5	α_5	L_5	0	



3.24 Parámetros θ_i [8]

Figura

- Parámetro d_i : distancia a lo largo del eje Z_{i-1} desde el origen del sistema S_{i-1} hasta la intersección del eje Z_{i-1} con el eje X_i . En el caso de articulaciones prismáticas será la variable de desplazamiento.

ARTICULACIÓN	α_i	a_i	θ_i	d_i
1	α_1	L_1	90°	L_1
2	α_2	L_2	0	0
3	α_3	L_3	0	0
4	α_4	L_4	0	0
5	α_5	L_5	0	0

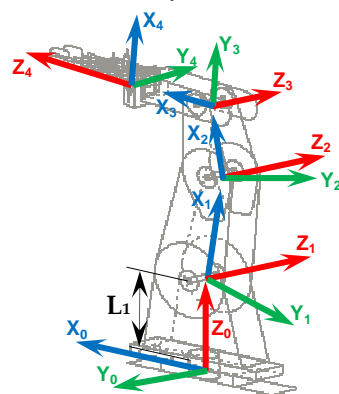


Figura 3.25 Parámetros d_i [8]

- Se obtiene la matriz de transformación T desde el sistema $i-1$ hasta el i

$${}^{i-1}A_i = \text{Rot}(Z_{i-1}, \theta_i) \cdot \text{Tras}(0,0, d_i) \cdot \text{Tras}(a_i, 0,0) \cdot \text{Rot}(X_i, \alpha_i)$$

Ec. 3.21

$$T = {}^0A_n = {}^0A_1 \cdot {}^1A_2 \cdot \dots \cdot {}^{i-1}A_i \dots {}^{n-1}A_n$$

Ec. 3.22

3.4.2 EL PROBLEMA CINEMÁTICO INVERSO

El objetivo del problema cinemático inverso consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot para que su extremo se posicione y oriente según una determinada localización espacial. Es más

* Imagen creada por los autores.

complicado que la cinemática directa ya que las ecuaciones no son lineales, por lo que algunas de sus soluciones no siempre tienen una solución real ^[19].

Se han desarrollado algunos procedimientos genéricos de programación, de modo que una computadora pueda, a partir de la cinemática del robot, obtener los n valores articulares que posicionan y orientan su extremo, el inconveniente de estos procedimientos es que se trata de métodos numéricos iterativos, cuya velocidad de convergencia e incluso su convergencia en sí, no está garantizada para todos los casos. También existe la posibilidad de múltiples soluciones.

La existencia o no de la solución lo define el espacio de trabajo de un robot dado, la ausencia de una solución significa que el robot no puede alcanzar la posición y orientación deseada, porque se encuentra fuera del espacio de trabajo del robot o fuera de los rangos permisibles de cada una de sus articulaciones.

El hecho de que un manipulador posea múltiples soluciones puede causar problemas debido a que el sistema tiene que ser capaz de elegir una de ellas.

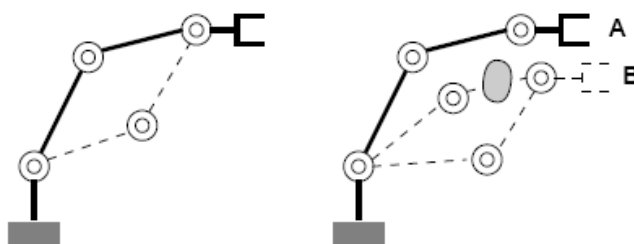


Figura 3.26 Múltiples soluciones ^[*]

En el manipulador de la Figura 3.26 se observa el problema de la solución múltiple, en este caso, “codo arriba” o “codo abajo”.

Para pasar del punto A al B, una buena solución (en ausencia de obstáculos) sería la que minimiza los valores de las articulaciones que es preciso mover, para esto se podría escoger la configuración superior (“codo arriba”), por lo que un argumento de partida para el problema cinemático inverso podría ser la posición en el punto A del manipulador. En el caso que se requiera evitar un obstáculo en la trayectoria, implicaría la elección de la configuración inferior (“codo abajo”).

* Imagen creada por los autores.

Las elecciones de estos casos implican que, en general, se necesita calcular todas las soluciones posibles teniendo en cuenta que están en función de los parámetros de D-H asociados (α_i , a_i , y d_i para articulación angular), y del rango permitido de movimiento de las articulaciones ^[19].

3.4.2.1 Resolución del Problema Cinemático Inverso

En la cinemática se considera que “todos los sistemas con articulaciones prismáticas y de revolución que tengan un total de seis grados de libertad a lo largo de una cadena de elementos simples en serie, es *resoluble*”.

No existe un algoritmo general que pueda emplearse para resolver un sistema de ecuaciones no lineales. Al resolver el problema cinemático inverso es mucho más adecuado encontrar una solución cerrada, es decir, encontrar una relación matemática explícita de la forma: $q_k = f_k(x, y, z, \alpha, \beta, \gamma)$ donde $k = 1 \dots n$ y representa los grados de libertad ^[19].

Si se consideran sólo los tres primeros grados de libertad de muchos robots, los tres primeros elementos quedan contenidos en un plano. Esta circunstancia facilita la resolución del problema.

Asimismo, en muchos robots se da la circunstancia de que los últimos tres grados de libertad, dedicados fundamentalmente a orientar el extremo del robot, corresponden a giros sobre ejes que se cortan en un punto.

Para la solución de estos sistemas, los métodos más utilizados son:

- Métodos geométricos (análisis geométrico del robot).
- Métodos algebraicos (basado en Transformaciones Homogéneas)

Los métodos geométricos permiten normalmente obtener los valores de las variables articulares, que son las que consiguen posicionar el robot, para ello se utilizan relaciones trigonométricas y geométricas sobre los elementos del robot. Se recurre a la resolución de triángulos formados por los elementos y articulaciones del robot.

Los métodos algebraicos consisten en intentar despejar las variables de articulación del conjunto de relaciones extraídas de un grupo de transformaciones, no siendo un método sistemático, sino heurístico. Pueden obtenerse distintas ecuaciones con diferentes parámetros para una misma articulación y ciertos parámetros pueden no aparecer en ninguna ecuación.

CAPÍTULO 4

DESARROLLO DEL SISTEMA DE ADQUISICIÓN Y PROCESAMIENTO DE IMAGEN

4.1 DIAGRAMA DE IMPLEMENTACIÓN DEL SISTEMA DE ADQUISICIÓN DE IMAGEN

La Figura 4.1 muestra el diagrama de flujo de la etapa de adquisición y procesamiento de imagen

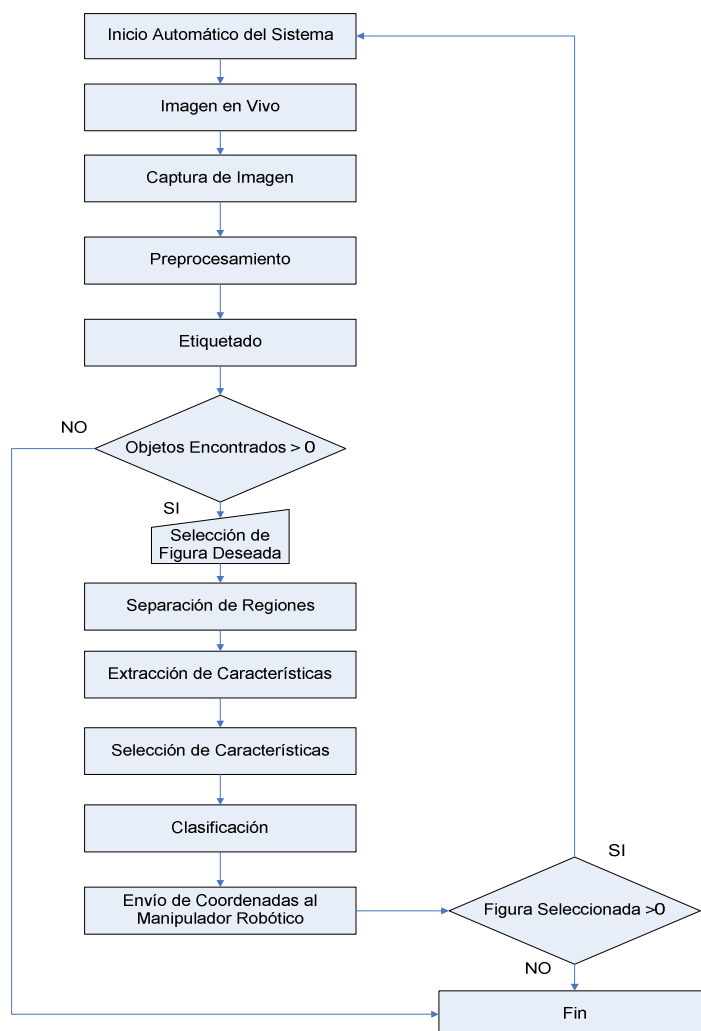


Figura 4.1 Diagrama de Flujo Etapa de Adquisición y Procesamiento de Imagen ^[*]

* Imagen creada por los autores.

Para la etapa de adquisición y procesamiento de la imagen capturada por la imagen se seguirán los siguientes pasos:

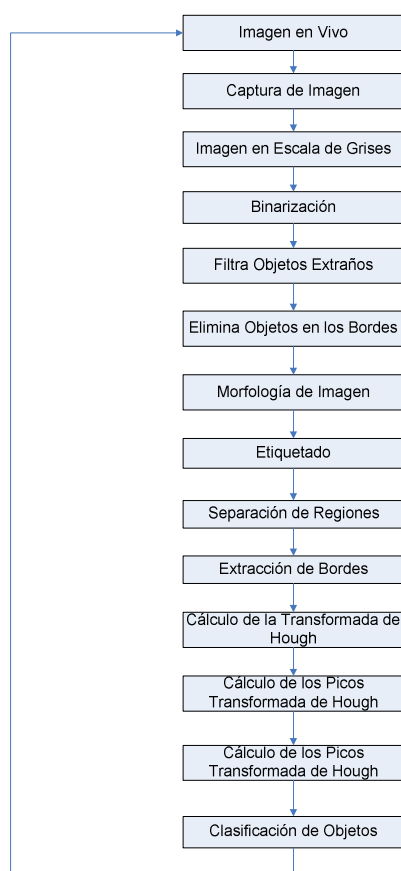


Figura 4.2 Pasos para el Proceso Adquisición y Procesamiento de Imagen ^[*]

Se escoge una cámara web comercial ya que su precio es muy económico y ofrece características muy buenas para la aplicación.

4.2 ADQUISICIÓN DE DATOS

Para la adquisición de la imagen se decidió utilizar el paquete computacional MatLab 7.0 ya que este ofrece una serie de funciones sumamente útiles en el proceso de adquisición y procesamiento de imagen.

Para la adquisición de la imagen se abre una secuencia de video en vivo (*preview*) dentro del MatLab, la cual envía una secuencia de video mostrando los objetos ubicados en la plataforma de inicio A, a continuación se captura una de las tramas la cual servirá como fuente para el procesamiento de imagen, para que

* Imagen creada por los autores.

la imagen contenga todos sus atributos es necesario dejar el tiempo necesario para que el lente de cámara se pueda abrir totalmente, en caso de no dejar esta pausa se obtendrá una imagen de color negro. Este proceso se realiza cada vez que el manipulador robótico realiza una acción con esto se evita introducir un error en el caso de que el brazo haya desplazado cualquier figura y la posición de la misma haya cambiado su nueva ubicación será captada nuevamente por la cámara.

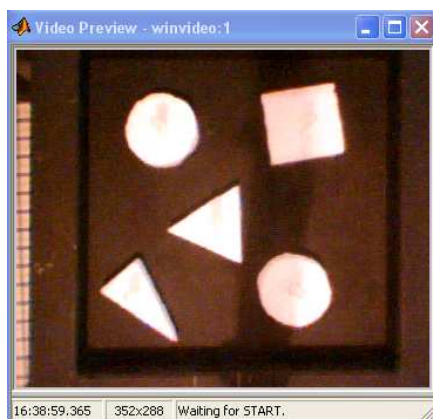


Figura 4.3 Imagen Adquirida por la Cámara [*]

Luego de ser capturada la imagen, se cierra la pantalla donde se encuentra la secuencia de video (*preview*), para luego comenzar con la etapa de análisis de la imagen.

4.3 PREPROCESAMIENTO

El interés en el procesamiento digital de imágenes se basa esencialmente en dos aspectos, mejorar la información contenida en una imagen para la interpretación humana y en el tratamiento de los datos de una escena para la percepción autónoma por una máquina.

* Imagen creada por los autores.

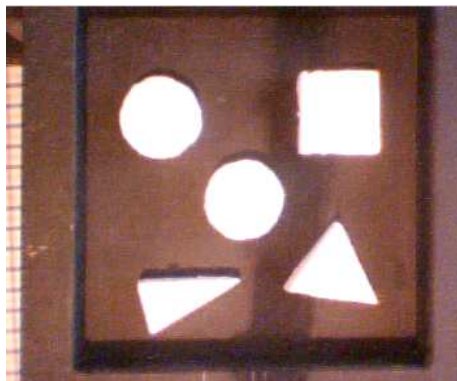


Figura 4.4 Imagen Adquirida por la Cámara ^[*]

Como se puede observar en la Figura 4.4 esta contiene niveles de contraste y brillo bastante grande; debido al efecto de la iluminación y a la colocación de la cámara, se puede visualizar la sombra del soporte que sirve para fijación de la misma.

4.3.1 ESCALA DE GRISES

Como fase previa a la etapa de procesamiento se decidió cambiar el campo de acción de la imagen capturada, la imagen captada por la cámara es del tipo *RGB* es decir una imagen a color se transforma en una imagen binaria. La forma más común de generar imágenes binarias es mediante la utilización del valor umbral de una imagen a escala de grises; es decir se elige un valor límite (o bien un intervalo) a partir del cual todos los valores de intensidades mayores serán codificados como 1 mientras que los que estén por debajo serán codificados a cero.

Para dicha transformación se utiliza la función *rgb2gray*, la cual como fuente tiene la imagen capturada por la captura y retorna una imagen en escala de grises.

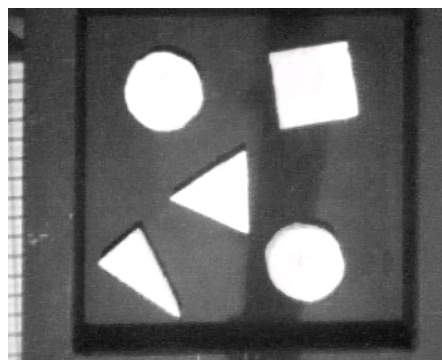


Figura 4.5 Imagen en Escala de Grises [*]

4.3.2 BINARIZACIÓN

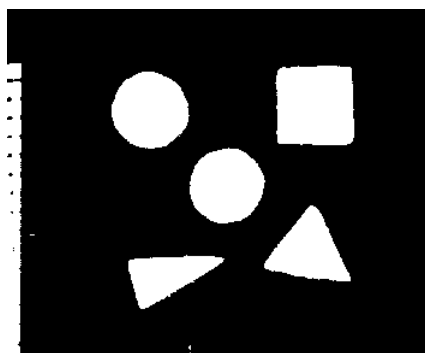
En visión computacional el trabajo con imágenes binarias es muy importante ya sea para realizar segmentación por intensidad de la imagen, para generar algoritmos de reconstrucción o reconocer estructuras.

Una imagen binaria es una imagen en la cual cada píxel puede tener solo uno de dos valores posibles 1 o 0. Como es lógico suponer una imagen en esas condiciones es mucho más fácil encontrar y distinguir sus características estructurales.

En visión computacional el trabajo con imágenes binarias es muy importante ya sea para realizar segmentación por intensidad de la imagen, para generar algoritmos de reconstrucción o reconocer estructuras.

La forma mas común de generar imágenes binarias es mediante la utilización del valor umbral de una imagen a escala de grises; es decir se elige un valor limite (o bien un intervalo) a partir del cual todos los valores de intensidades mayores serán codificados como 1 mientras que los que estén por debajo serán codificados a cero.

Además por medio de la binarización se consigue eliminar el efecto del brillo en la imagen y prepararla para la etapa de procesamiento y extracción de características. Para esto se utiliza la función *im2bw* que cambia la imagen en escala de grises a una imagen binaria.

**Figura 4.6** Imagen Binaria [*]

* Imagen creada por los autores.

En la imagen binaria obtenida se puede observar figuras en la frontera de la misma debido al hecho de que una porción de la imagen obtenida, representa un espacio que se encuentra fuera de la plataforma de trabajo estos objetos se eliminan, además se evita que cualquier objeto incompleto que se ubique en los límites de la imagen pueda interferir o agregar un error al momento de realizar el proceso.

En la Figura 4.6 se pueden observar espacios pequeños extraños que no son parte de los atributos de la figura y que aparecen debido a efectos de iluminación y al hecho de que la cámara ubicada en una posición de 90° con respecto a la superficie de trabajo introduce objetos no deseados dentro de la imagen.

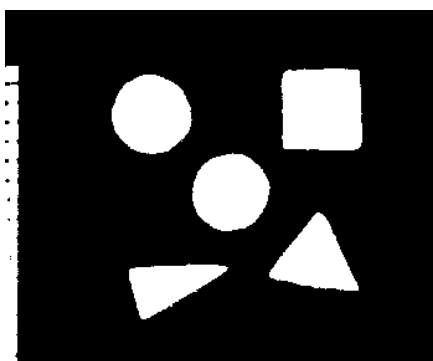


Figura 4.7 Imagen Sin Objetos con más de 30 Píxeles ^[*]

Para eliminar estos objetos extraños se eliminó todos los objetos dentro de la imagen que posean un número menor a 30 píxeles. Utilizando la herramienta *bwareaopen* la cual nos permite eliminar todos los componentes conectados que contengan un número menor que Q píxeles, en este caso el valor de Q es 30. Se decidió realizar esta operación ya que en el momento de realizar el etiquetado de la imagen estos elementos introducían regiones que no son de interés dentro del proceso.

* Imagen creada por los autores.

* Imagen creada por los autores.

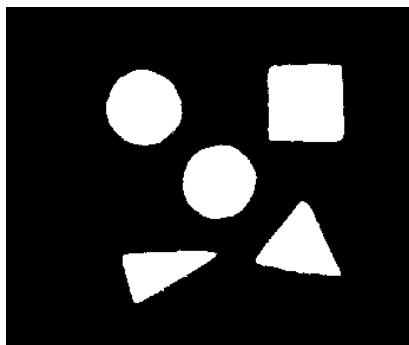


Figura 4.8 Imagen Sin Objetos con en la Frontera ^[*]

4.3.3 FILTRO MORFOLÓGICO

A pesar de que para el ojo humano cada una de las regiones dentro de la imagen se encuentran completas en ocasiones las mismas presentan discontinuidades que provocan espacios separados o vacíos dentro de una misma imagen, por lo cual es recomendable realizar un filtro morfológico utilizando las funciones *imfill* e *imclose* incluidas en el *Toolbox* de imagen del MatLab.

Las operaciones morfológicas aplican un elemento estructural a la imagen de entrada, sin cambiar el tamaño de la imagen de salida. En una operación morfológica, el valor de cada píxel en la imagen de salida depende del valor de ese píxel en la imagen de entrada y su relación con la vecindad. Seleccionando el tamaño y forma de la vecindad (definido a través de un elemento estructural) se puede crear una operación morfológica, que altera el valor del píxel en la imagen de salida.

La acción del elemento estructurante se explica en la Figura 4.9, ésta muestra una imagen binaria que será sometida a la operación de relleno y el elemento estructurante, cuyos elementos están 4-conectados y constituyen la frontera de una región. Comenzando por un punto *p* que se encuentre dentro de la frontera, el objetivo es rellenar completamente la región con 1's.

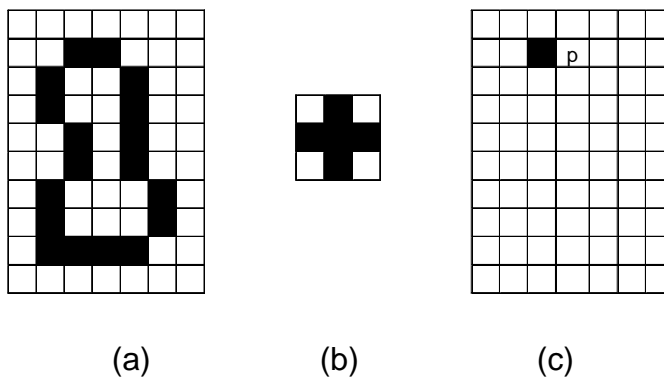


Figura 4.9 a) Relleno de la Región; b) Elemento Estructurante; c) Punto de Inicio en la Frontera ^[7]

Comenzando por el punto p ubicado en la frontera, se observa los pasos del relleno de la región cuando el elemento estructurante se desplaza a través de los puntos de la frontera colocando todos los puntos de la imagen binaria que se encontraban en 0s se colocan en 1s ^[7].

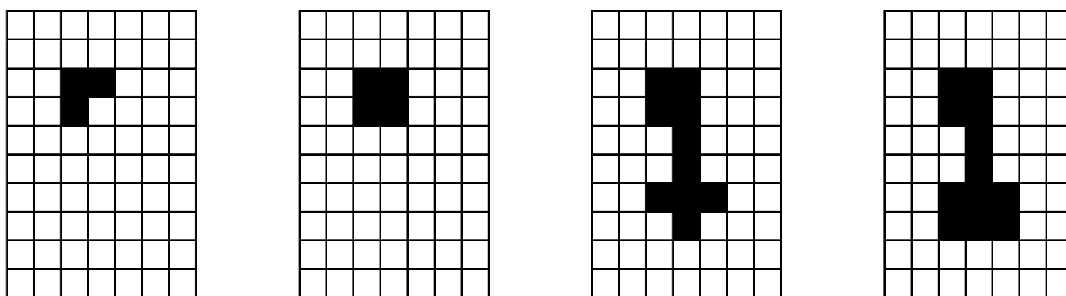


Figura 4.10 Varias Pasos del Relleno ^[7]

La Figura 4.11 muestra el resultado de la operación morfológica relleno.

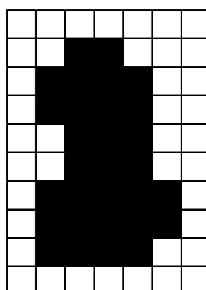


Figura 4.11 Resultado del Relleno ^[7]

La Figura 4.12 muestra el resultado de la utilización del filtro morfológico.

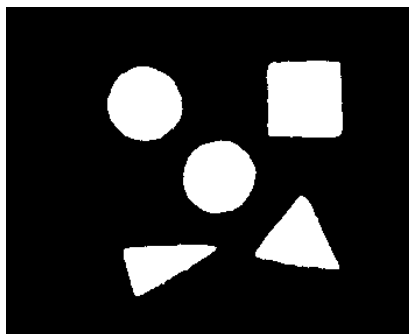


Figura 4.12 Imagen Sometida a las Operaciones *imclose* e *imfill* ^[8]

Para finalizar la etapa de pre-procesamiento se realiza la reconstrucción de la imagen utilizando varias operaciones morfológicas como *imreconstruct*, *imopen*, *imerode*, *imdilate*.

4.3.3.1 Dilatación y Erosión

La dilatación se basa en aumentar el nivel de los valores de los píxeles en el entorno de los objetos presentes en la imagen. La erosión se basa en reducir el nivel de los píxeles del entorno de un objeto. El número de píxeles a los que se aumenta o reduce el nivel depende del tamaño y forma del elemento estructural usado para procesar la imagen. La dilatación y la erosión “expande” y “contrae” respectivamente la imagen.

Para calcular la dilatación se superpone el píxel central del elemento estructural a cada píxel de la imagen de entrada, entonces el píxel de la imagen de entrada se altera en función de los valores de los píxeles del entorno, definidos por el elemento estructural. El valor del píxel de salida será el máximo entre todos los píxeles presentes en la vecindad.



* Imagen creada por los autores.

(a) (b)
Figura 4.13 a) Imagen de Fondo; b) Imagen sin Ruido ^[*]

Para eliminar el efecto de concavidad de la cámara se realiza una resta de imágenes en la primera se tiene el fondo, la segunda representa la imagen original como resultado se obtiene una imagen sin el efecto de la concavidad de la cámara.



(a) (b)
Figura 4.14 a) Imagen Dilatada; b) Imagen Procesada ^[*]

Sometiendo a la imagen a la operación morfológica dilatación para eliminar cualquier reflexión horizontal presente en la imagen y luego se utiliza la función *imreconstruct* y se obtiene la imagen de mejor calidad para la extracción de atributos.

4.4 EXTRACCIÓN DE CARACTERÍSTICAS

Con el objetivo de encontrar el número de regiones presentes dentro de la imagen, se somete a la imagen a un proceso de etiquetado, cada una de las regiones es representada mediante un color distinto como muestra la Figura 4.15.

El Fondo, corresponde a las regiones de la imagen que no interesan. Se les suele asignar la etiqueta 0, y se representan con un nivel cero en la capa de segmentación.

El Primer Plano, que determina los píxeles de la imagen que corresponden a objetos de interés.

* Imagen creada por los autores.

Etiqueta, que es el número que identifica a cada píxel, indicando a qué objeto pertenece.

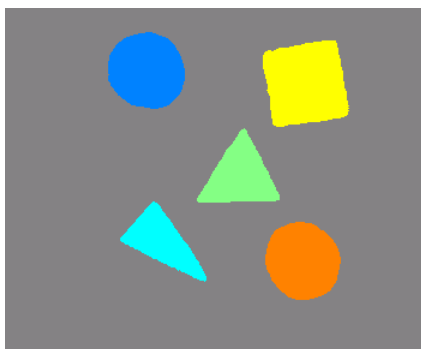


Figura 4.15 Imagen Etiquetada ^[8]

La imagen muestra claramente 6 regiones, etiquetadas cada una de ellas por medio de un diferente color, el fondo de color plomo y cada una de las figuras por los colores amarillo, azul, verde, naranja y turquesa.

Ya que se dispone del número de regiones se calcula las coordenadas de los centroides de cada una de las regiones encontradas, los mismos que una vez realizada la etapa de procesamiento de imagen serán enviados para el posicionamiento del brazo robótico.

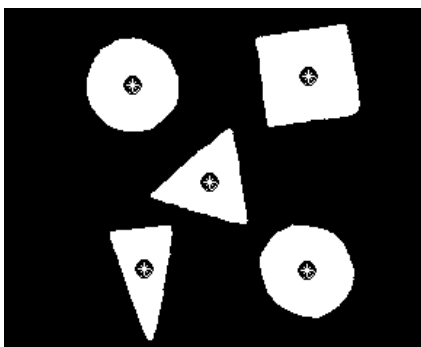


Figura 4.16 Imagen Binaria y Obtención de Centroides ^[8]

* Imagen creada por los autores.

* Imagen creada por los autores.

```
Command Window
>> eliminacion_ruido
>> morfologia2
>> centro

centro_figura =

    124.9430    70.1641

centro_figura =

    133.2147    195.4865

centro_figura =

    177.6234    136.3494

centro_figura =

    245.6827    63.5398

centro_figura =

    244.9818    196.1831

>>
```

Figura 4.17 Coordenadas de los Centroides de las Regiones Encontradas

4.5 SELECCIÓN DE CARACTERÍSTICAS

Ahora se procede a separar cada una de las regiones y visualizarlas independientemente para iniciar la etapa de extracción de características.

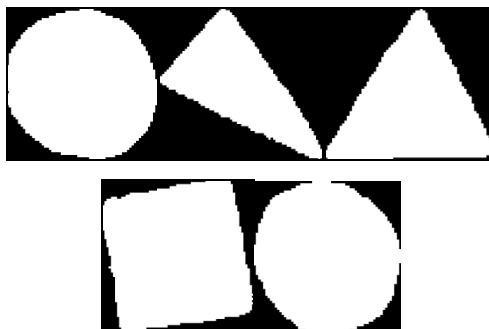


Figura 4.18 Separación de Regiones Independientes ^[*]

Finalizado la separación de cada una de las regiones se extrae los bordes de las mismas.

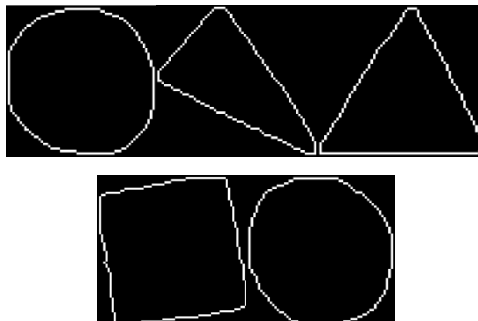


Figura 4.19 Extracción de Bordes ^[*]

Para acentuar los bordes se somete a cada una de las regiones a una dilatación con esto podemos visualizar mejor todas las características de las figuras.

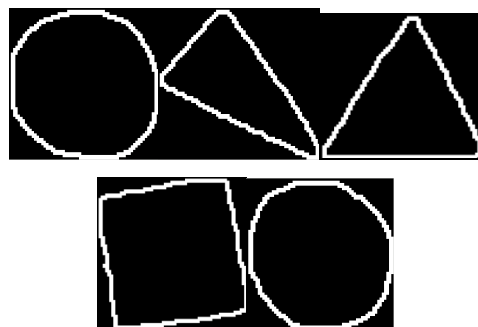


Figura 4.20 Imagen Dilatada ^[*]

4.6 CLASIFICACIÓN

Una vez obtenidas las regiones con la dilatación de los bordes se procede a someter cada una de las regiones a la Transformada de Hough (TH). Aplicando la TH se convierte la representación $[x,y]$ a $[\rho,\theta]$. Utilizando la ecuación paramétrica de la recta.

$$x * \cos(\theta) + y * \text{sen}(\theta) = \rho \quad \text{Ec. 4.1}$$

La TH es independiente de la posición de cada una de las figuras, es decir, se puede identificar la cantidad de picos de la transformada sin importar si la figura

* Imagen creada por los autores.

se encuentra rotada con respecto al eje de proyección del campo de la plataforma.

4.6.1 Clasificación Triángulos

Las Figuras 4.21 y 4.22 muestran la TH de los dos triángulos presentes en la imagen, en las que se puede visualizar la existencia de tres puntos en donde se interceptan las curvas sinusoidales, lo cual indica la presencia de tres líneas las mismas que conforman el perímetro del triángulo. Para determinar los valores donde se interceptan las curvas sinusoidales

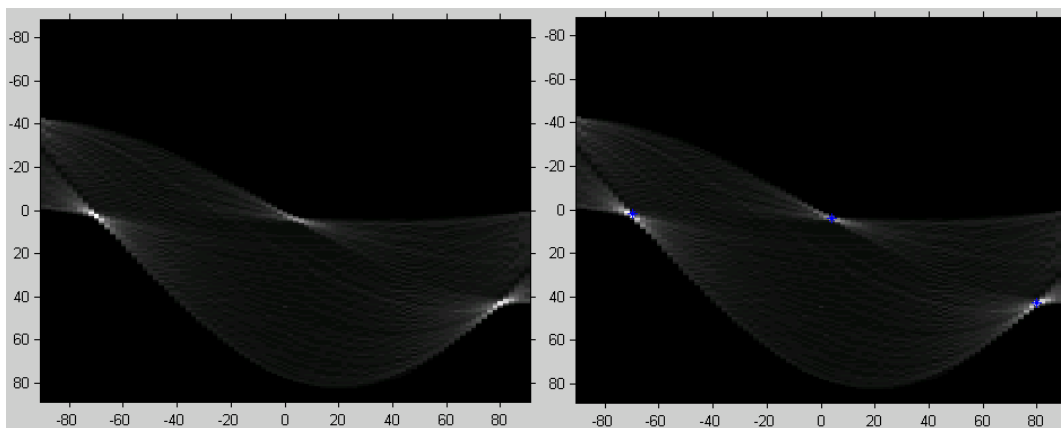


Figura 4.21 TH del Primer Triángulo y Extracción de Picos [*]

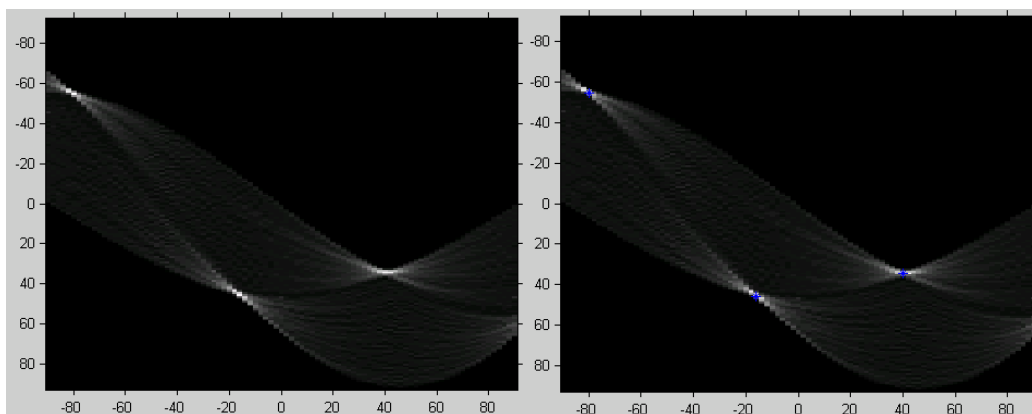


Figura 4.22 TH del Segundo Triángulo y Extracción de Picos [*]

Dentro de la etapa de clasificación se envían a la etapa de posicionamiento del brazo robótico las siguientes variables: el número de objetos presentes en la imagen adquirida por la cámara digital, las coordenadas de los centroides de las

* Imagen creada por los autores.

* Imagen creada por los autores.

regiones correspondientes a la opción del usuario, y por último la cantidad de figuras encontradas. En el caso de la selección de triángulos, por ejemplo se tiene que el número de regiones es cinco, las coordenadas de los centroides son para el primer triángulo (133.2147;195.4865), mientras que para el segundo triángulo se tiene (177.6234;136.3494), y el número total de triángulos encontrados es dos.

En la Figura 4.23 además se encuentran los valores de las coordenadas de los picos que para el primer triángulo son $[\rho, \theta] = (2.0230; -70)$, $(42.4828; 80)$, $(4.0460; 4)$, mientras que para el segundo triángulo los picos se encuentran en las coordenadas $[\rho, \theta] = (34.3736; 40)$, $(-54.5934; -80)$, $(45.4945; -16)$.

```

Command Window
>> eliminacion_ruido
>> morfologia1
>> clasificacion_triángulos

numero_objetos =

    5

rho =

    2.0230    42.4828    4.0460

theta =

   -70    80    4

centro_triángulo =

   133.2147   195.4865

rho =

    34.3736   -54.5934    45.4945

theta =

    40   -80   -16

centro_triángulo =

   177.6234   136.3494

triángulos_total =

    2

```

Figura 4.23 Variables Enviadas a la Etapa de Posicionamiento y TH

4.6.2 Clasificación Cuadrados

En caso de que la selección del usuario sea la de cuadrados se realiza un proceso análogo pero en este caso se buscan cuatro picos dentro de la TH, cada pico representa el número de lados de la región analizada. La Figura 4.24 muestra la TH de un cuadrado y la extracción de los picos dentro de la transformada.

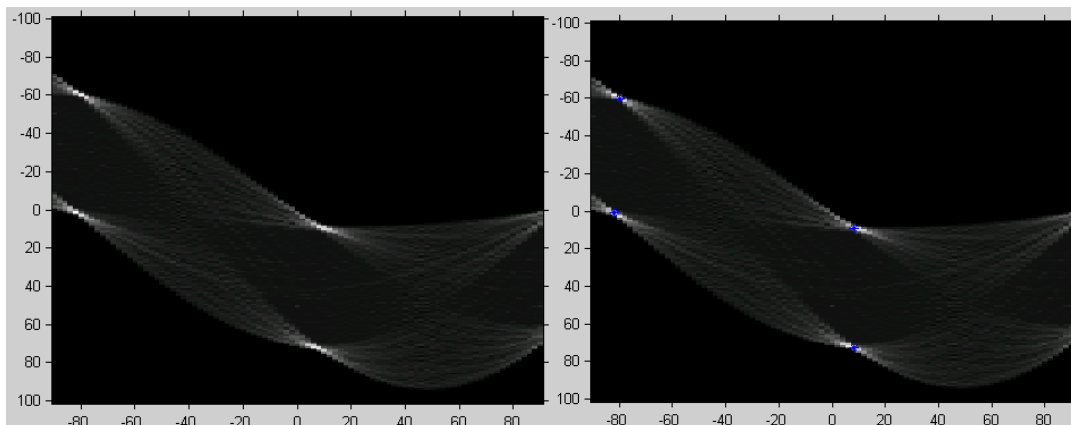


Figura 4.24 TH un Cuadrado y Extracción de Picos ^[*]

Para la selección de cuadrados dentro de la imagen se encontró que el número de objetos es cinco, la coordenada del Centroide es (245.6827;63.5398), y la cantidad total de cuadrados fue de uno. Además las coordenadas de los picos $[\rho, \theta] = (9.0909; 8), (69.6970; 8), (1.0101; -82), (-61.6162; -80)$.

```

Command Window
>> eliminacion_ruido
>> morfologial
>> paso3

numero_objetos =

     5

rho =

     9.0909     69.6970     1.0101    -61.6162

theta =

     8     8    -82    -80

centro_cuadrado =

    245.6827    63.5398

cuadrados_total =

     1
  
```

Figura 4.25 TH para un Cuadrado y Extracción de Picos

4.6.3 Clasificación Círculos

Finalmente, para la selección de círculos se obtiene que el número de picos es bastante alto, debido a que al círculo se le considera como una figura geométrica con un infinito número de lados, mientras mayor sea el número de picos encontrados se acerca en un mayor grado a un círculo perfecto.

* Imagen creada por los autores.

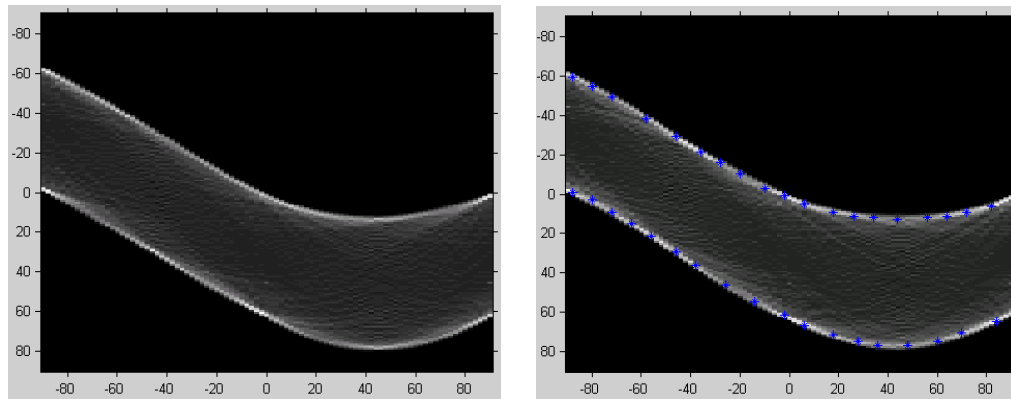


Figura 4.26 TH del Primer Círculo y Extracción de Picos [*]

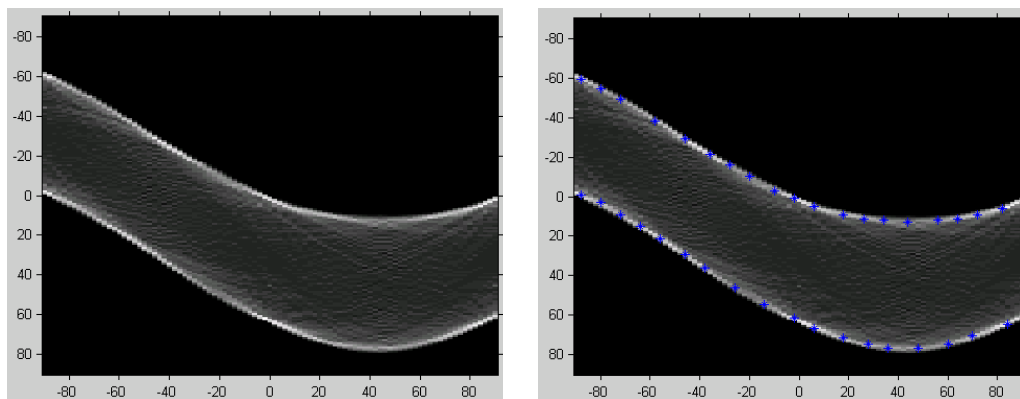


Figura 4.27 TH del Segundo Círculo y Extracción de Picos [*]

Se obtiene que el número de objetos encontrados igual a cinco, las coordenadas de los centroides del primer y segundo círculo (124.9430;70.1641), (244.9818;196.1831) respectivamente y el número total círculos igual a dos.

```

Command Window
>> eliminacion_ruido
>> morfologial
>> clasificacion_circulos

numero_objetos =

     5

rho =

Columns 1 through 7
    60.6818   -59.6705   -1.0114    75.8523    5.0568    63.7159    32.3636

Columns 8 through 14
   -19.2159   -54.6136    24.2727   -13.1477     1.0114    65.7386   -49.5568

Columns 15 through 21
   -43.4886    -5.0568    14.1591    11.1250    56.6364    75.8523    13.1477

Columns 22 through 28
    38.4318     6.0682    11.1250    18.2045   -38.4318    15.1705    14.1591

Columns 29 through 30
    75.8523     5.0568
  
```

* Imagen creada por los autores.

```

theta =
Columns 1 through 12
    -2   -88   -88    44    84    86   -44   -32   -80   -54   -24   -2
Columns 13 through 24
     6   -72   -64   -14    36    68   -10    36    60   -36    8    22
Columns 25 through 30
   -62   -56    44    52    52   -80

centro_circulo =
124.9430  70.1641

rho =
Columns 1 through 7
   -1.0112    3.0337   44.4944   -6.0674   -43.4831    75.8427   -61.6854
Columns 8 through 14
  -49.5506   15.1685   38.4270   50.5618   76.8539    8.0899   29.3258
Columns 15 through 21
  -12.1348   56.6292    3.0337   13.1461   75.8427   -2.0225   -56.6292
Columns 22 through 28
  -37.4157   11.1236   13.1461   -18.2022   61.6854   67.7528   67.7528
Columns 29 through 30
     8.0899   72.8090

theta =
Columns 1 through 12
    -6   -80   -24   -14   -62    36   -88   -70   -62   -32   -16   46
Columns 13 through 24
   -72   -44   -22    -8    2    54    56   -90   -80   -54    24    44
Columns 25 through 30
   -30    0    10    82    14    22

centro_circulo =
244.9818  196.1831

```

Figura 4.28 TH del Segundo Círculo y Extracción de Picos

CAPÍTULO 5

DESARROLLO DEL MANIPULADOR CON VISIÓN ROBÓTICA

Para la construcción de este modelo, se inicia con el diseño y la implementación de la estructura mecánica de la planta completa y luego de establecido el modelo de su sistema cinemático (mediante parámetros Denavit-Hartenberg), se definen los algoritmos de control por software del brazo robótico para la realización de la tarea, objeto de este proyecto.

5.1 DISEÑO E IMPLEMENTACIÓN DE LA ESTRUCTURA

Se escogen las siguientes características generales para el brazo robótico:

Tipo de robot:	Antropomórfico
Grados de libertad:	Cinco
Clasificación:	Robot industrial / trayectoria operada-continua
Morfología:	Configuración angular
Articulación:	Rotacional
Elemento terminal:	Dedos deslizantes

Existen muchos modelos de robots industriales que morfológicamente cumplen con los requerimientos del proyecto objeto de este trabajo, por lo que para implementar la estructura mecánica correspondiente, se ha realizado la construcción del modelo emulando prototipos y diseños ya existentes.

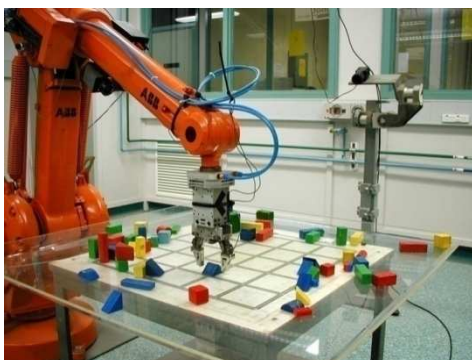


Figura 5.1 Brazos robóticos comerciales ^[*]

* Imágenes obtenidas de www.abb.com/robotics/irb40 y de www.lynxmotion.com.

Para la construcción del modelo, se escogió como material de la estructura el acrílico, debido a que es liviano, económico, resistente y fácil de trabajar.

5.1.1 SERVOMOTORES SELECCIONADOS

Debido a las dimensiones de la planta diseñada, los servomotores de modelismo se ajustan a los objetivos del proyecto. El bajo costo y la posibilidad de consecución en el comercio local de modelos con características diversas de torque resultan en la selección de los siguientes servomotores:

Servomotor de “Base”:



Hitec HS-755HB

Voltaje:	4,8 – 6 V
Torque @ 6VDC:	13,2 Kg·cm
Velocidad @ 6VDC:	0,23 seg/60° sin carga
Peso:	110 g

Servomotor de “Hombro”, “Codo”, “Muñeca” y “Mano”:



Hitec HS-311

Voltaje:	4,8 – 6 V
Torque @ 6VDC:	3,5 Kg·cm
Velocidad @ 6VDC:	0,15 seg/60° sin carga
Peso:	43 g

Para la articulación “Hombro” se utilizan dos servomotores en paralelo para aumentar el torque.

Servomotor de “Pinza”:



Hitec HS-81

Voltaje:	4,8 – 6 V
Torque @ 6VDC:	3 Kg·cm
Velocidad @ 6VDC:	0,09 seg/60° sin carga
Peso:	16,6 g

5.1.2 ESTRUCTURA MECÁNICA

El modelo consiste en un brazo robótico antropomórfico, por lo que la estructura se construye conforme a patrones de robots existentes comercialmente.

En primer lugar se concibe un bosquejo de la estructura para tener una idea de lo que se va a construir, utilizando la gráfica de cadenas cinemáticas se tiene el siguiente modelo:

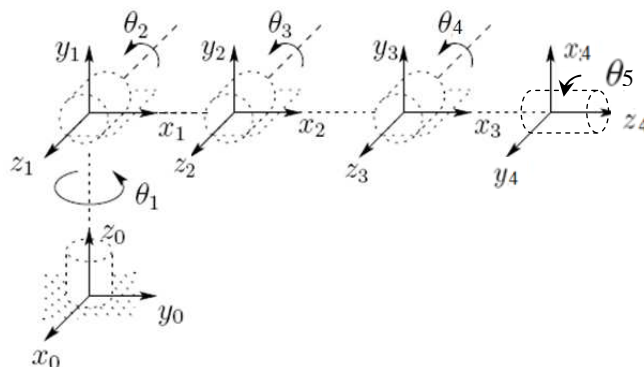


Figura 5.2 Cadena cinemática propuesta ^[8]

Como se puede apreciar, el modelo consta de una base con giro angular, tres articulaciones angulares y una articulación angular en el elemento terminal (pinza). La construcción se realiza mediante el empleo de servomotores de modelismo para el movimiento angular de las articulaciones y debido a que se requiere un material liviano para la estructura de soporte, se utiliza acrílico de 2 mm de espesor.

La estructura, además del brazo robótico propiamente dicho, debe contar con un soporte para la cámara de captura de imágenes y las plataformas que deben contener las figuras geométricas que serán identificadas, localizadas y trasladadas de un sitio a otro. La planta de esta estructura debe ser lo suficientemente robusta para soportar las tareas repetitivas que se darán del proceso, por lo que se utilizan perfiles de hierro.

Para planificar la construcción y distribución de los componentes, se realiza un dibujo de la estructura indicando las dimensiones propuestas para toda la planta a construirse.

* Imágenes obtenidas de www.cs.dartmouth.edu/~donaldclass/Bio/current/Papers/chap3-forward-kinematics.pdf

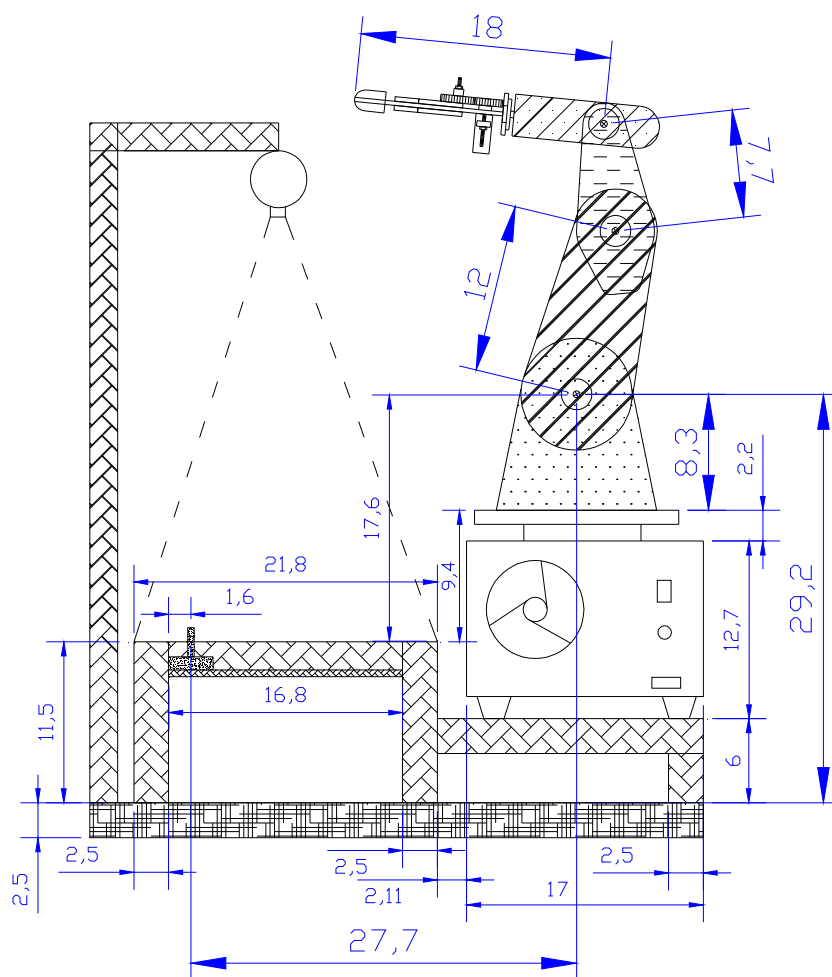


Figura 5.3 Dibujo de estructura, vista lateral ^[*]

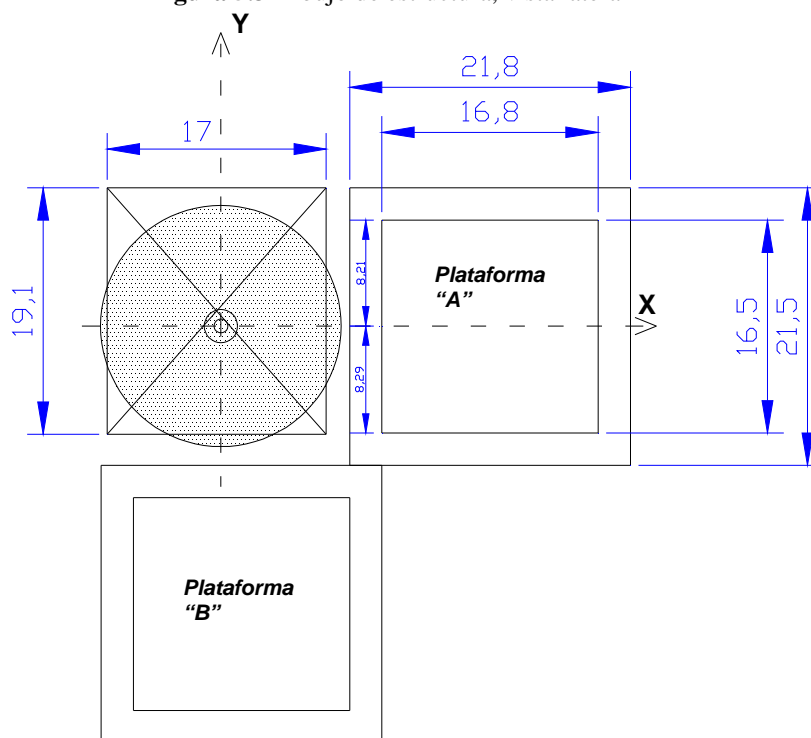


Figura 5.4 Dibujo de estructura, vista de arriba ^[*]

* Imagen creada por los autores.

En la base se coloca el servomotor correspondiente y se aprovecha para la instalación de las fuentes de poder que requiere la circuitería del control del robot, de esta manera se tiene la robustez y firmeza necesaria para el soporte del resto de la estructura. Se acopla una base de acrílico de 10 mm solidaria al eje del servomotor, la cual sirve de sustento principal del brazo como tal.



Figura 5.5 Caja metálica para la base



Figura 5.6 Soporte de acrílico de 10 mm de espesor

En este soporte se colocan dos marcos de acrílico para los servomotores del hombro, los cuales deben mover la carga total del resto del brazo. Esta es la mayor carga del sistema por lo que colocan ambos servos, los que brindan la suma de sus respectivos torques.

Unidos a los ejes de los servomotores del hombro se colocan las placas que corresponden al brazo. Se unen las respectivas partes mediante pernos.

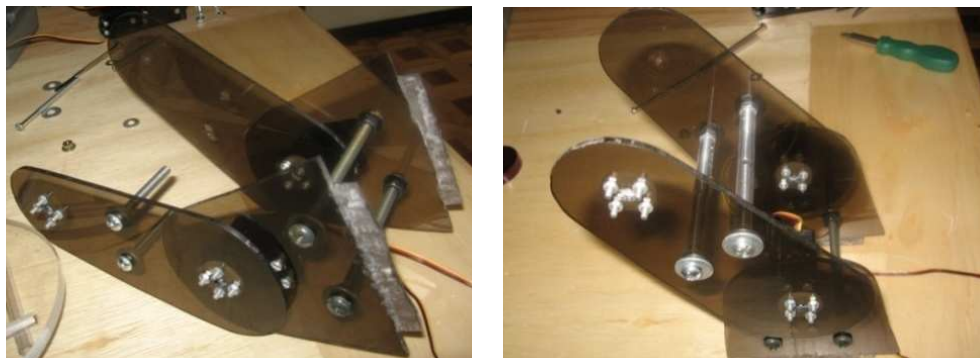


Figura 5.7 Hombro y brazo

Los componentes del antebrazo son más pequeños y, así como el brazo, requieren sólo un servomotor para soportar la carga.



Figura 5.8 Antebrazo

La construcción de la muñeca incluye dos servomotores ya que se acopla también el servomotor para el giro del elemento terminal (mano).

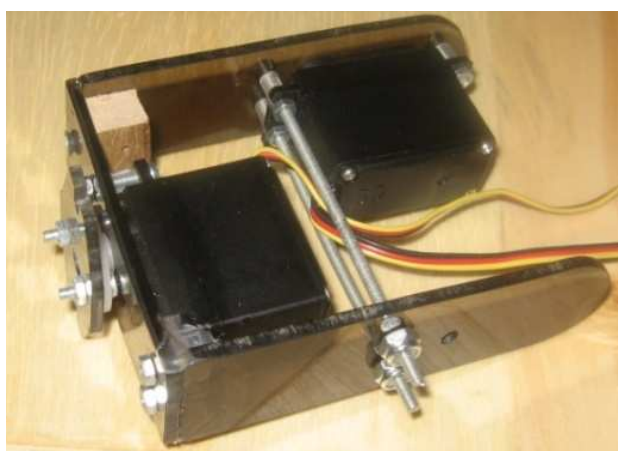


Figura 5.9 Muñeca

El elemento terminal consiste en una pinza con mecanismo de dedos deslizantes en la que coloca un servomotor muy pequeño para obtener la fuerza de sujeción de los dedos.

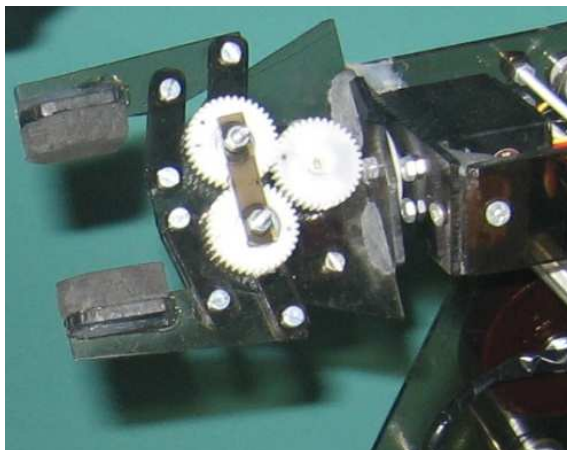


Figura 5.10 Mano (pinza)

Finalmente, la estructura de soporte para la cámara y las plataformas se construyen conforme a las dimensiones de los planos y se pinta de color negro mate para evitar que el reflejo de la luz sobre la superficie genere imágenes erróneas.



Figura 5.11 Planta completa construida

5.1.3 FUENTES DE ALIMENTACIÓN

Son necesarias dos fuentes de alimentación para este diseño, una fuente de 9 voltios para la circuitería del módulo de control de los servomotores SSC II y una fuente de 6 voltios para los servomotores propiamente dichos. Se escogió este voltaje debido a que se obtiene un mejor torque en los motores seleccionados.

El circuito del módulo de control tiene un consumo máximo de aproximadamente 300 mA, medidos experimentalmente con el sistema en funcionamiento. Para la fuente de 9 voltios correspondiente, se tiene el siguiente diagrama de diseño:

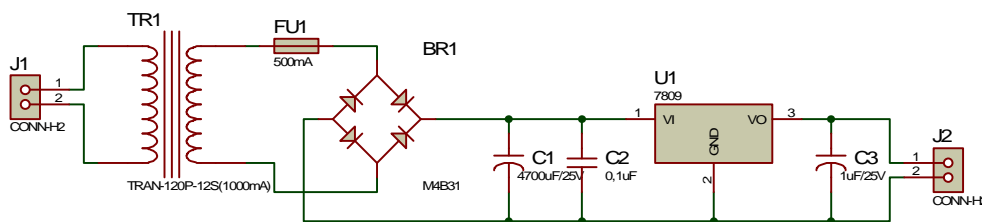


Figura 5.12 Fuente de poder de 9 voltios [8]

Igualmente, de manera experimental con el sistema en funcionamiento, se estableció un consumo máximo de aproximadamente 2,5 A para el sistema de servomotores funcionando todos simultáneamente y con la carga mecánica real implementada, por lo que se utilizó el regulador para tres amperios LM350K, con lo que se tiene el siguiente diagrama de diseño para la fuente de 6 voltios correspondiente:

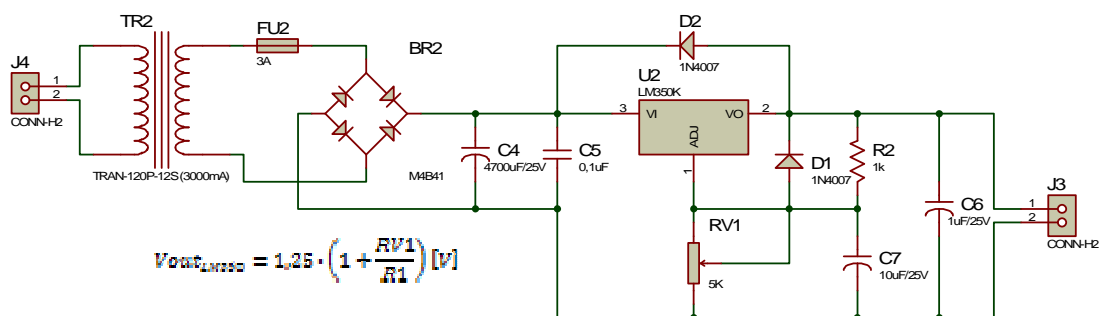


Figura 5.13 Fuente de poder para 6 voltios [8]

Se colocan los circuitos de las fuentes de alimentación, sus respectivos transformadores y un elemento de ventilación, así como la tarjeta de control de los servomotores en la caja metálica destinada para la base del brazo robótico.

* Imagen creada por los autores.

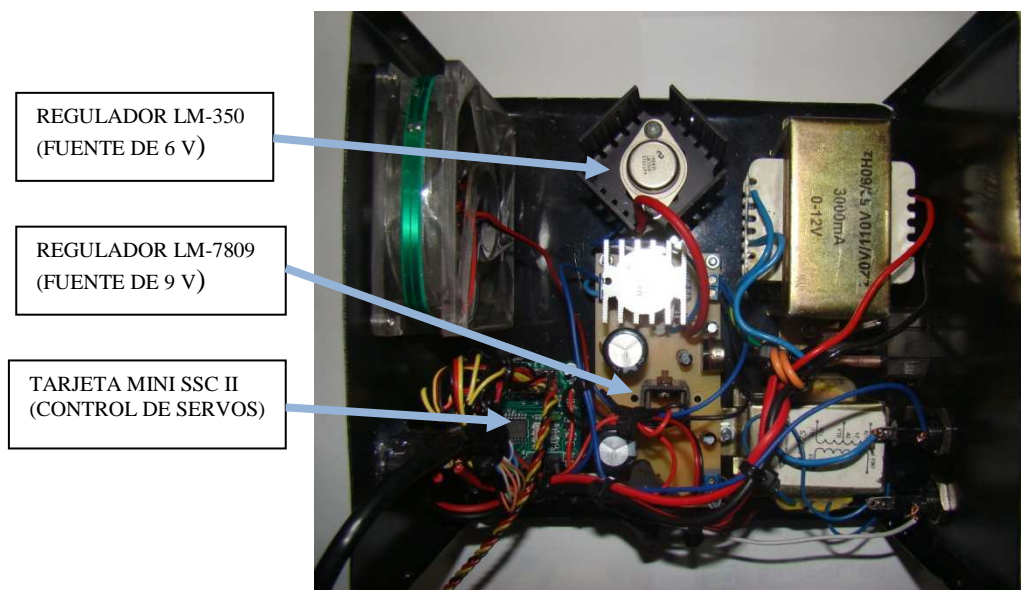


Figura 5.14 Ubicación de la tarjeta de control de servos y fuentes de poder

5.2 DESARROLLO CINEMÁTICO DEL BRAZO ROBÓTICO

El objetivo de esta parte del presente proyecto es desarrollar un sistema computacional que permita el control automático de la posición del elemento terminal del brazo robótico, dada por la localización espacial de una figura geométrica, la cual es determinada por el procesamiento de la imagen capturada por medio de una cámara web.

Mediante el procesamiento de la imagen se logra identificar la posición del centroide de la figura identificada en un plano X-Y, obteniéndose de esta forma la localización de la posición final deseada del elemento terminal para el brazo robótico.

Como se menciona en el marco teórico (3.4.2): “el objetivo del problema cinemático inverso consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot para que su extremo se posicione y oriente según una determinada localización espacial”, por lo tanto al requerirse un sistema de control automático para los servomotores de las articulaciones del robot, en esta sección del proyecto se utilizan las representaciones cinemáticas utilizadas en robótica para la obtención de los parámetros Denavit-Hartenberg, para consecuentemente llegar a deducir las ecuaciones cinemáticas inversas correspondientes a cada articulación; desarrollando un algoritmo computacional

con dichas ecuaciones, el sistema posee la capacidad para auto-configurar los ángulos de posición de las articulaciones del brazo robótico y colocar el elemento terminal sobre el centro de una pieza geométrica a ser localizada, en cualquier posición al azar en la que ésta se encuentre.

Utilizando las definiciones establecidas en la sección 3.4, las matrices de posicionamiento del sistema del brazo robótico se definen del siguiente proceso:

Parámetros de DENAVIT-HARTENBERG para el brazo robótico:

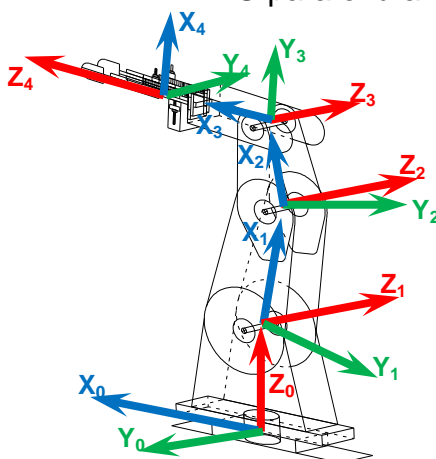


Figura 5.15 Ejes de referencia [8]

ARTICULACIÓN	α_i	a_i	θ_i	d_i
1	α_1	L_1	90°	L_1
2	α_2	L_2	0	0
3	α_3	L_3	0	0
4	α_4	L_4	0	0
5	α_5	L_5	0	0

Tabla 5.1 Parámetros

Denavith-Hartenberg

Tomando en cuenta que las posiciones finales de las articulaciones del brazo robótico, matemáticamente responden a una serie de operaciones matriciales de traslación y rotación de sus respectivos sistemas de coordenadas, se puede resolver el problema cinemático inverso de una manera más simple al reducir el número de incógnitas (ángulos de posición de los ejes rotacionales) con la traslación del origen del sistema de coordenadas X_0, Y_0, Z_0 hasta la intersección con el origen del sistema X_1, Y_1, Z_1 ; realizando las respectivas operaciones se obtiene, utilizando la matriz homogénea de equivalencia simbólica, la

* Imagen creada por los autores.

determinación del ángulo de posición rotacional “ θ ” (giro de la “base”) en función de las coordenadas en X - Y de la localización de una pieza geométrica a través del programa de captura y procesamiento de imágenes.

Para la determinación de los parámetros correspondientes, se elige una posición inicial de referencia para el brazo robótico, de preferencia se utilizan los ángulos rectos para facilitar los cálculos al momento de usar las funciones trigonométricas.

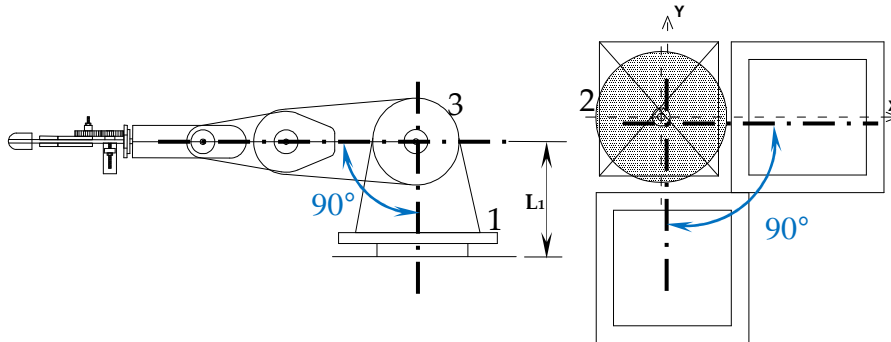


Figura 5.16 Parámetros de traslación de sistemas de coordenadas [*]

	α_i	a_i	θ_i	d_i
1	0	0	0	0
2	0	0	$\pi/2$	0
3	$\pi/2$	0	0	L_1

Tabla 5.2 Parámetros para la traslación del sistema de referencia

Matrices de transformación para la traslación del sistema de referencia X_0, Y_0, Z_0 al sistema X_1, Y_1, Z_1 :

$${}^0T_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ec. 5.1 Ecuación de traslación hasta L_1

$${}^1T_2 = \begin{bmatrix} \cos\theta & -\text{sen}\theta & 0 & 0 \\ \text{sen}\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ec. 5.2 Giro alrededor del eje Z

$${}^2T_3 = \begin{bmatrix} \cos\theta & 0 & \text{sen}\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\text{sen}\theta & 0 & \cos\theta & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

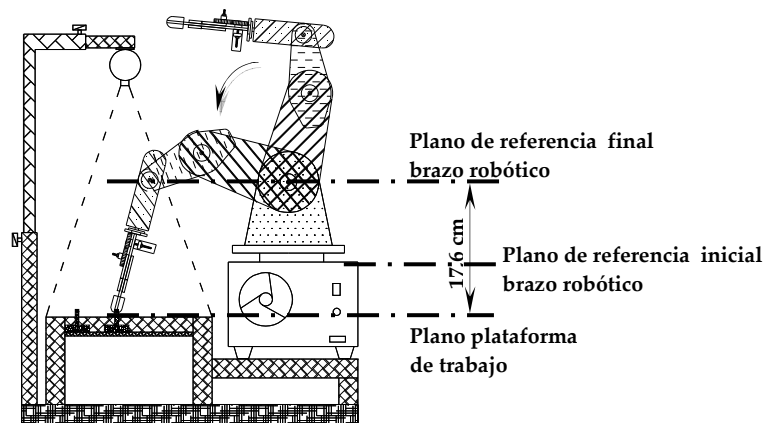
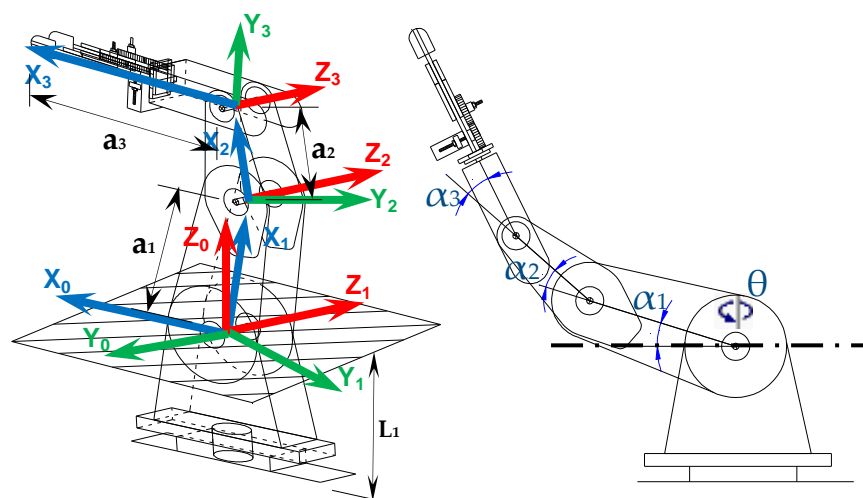
Ec. 5.3 Rotación de 90° sobre el eje Y

* Imagen creada por los autores.

$${}^0T_3 = \begin{bmatrix} 0 & -\text{sen } \theta & \text{cos } \theta & 0 \\ 0 & \text{cos } \theta & \text{sen } \theta & 0 \\ -1 & 0 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ec. 5.4 Matriz de traslación resultante

Complementariamente se tiene un nuevo plano de referencia para el sistema de articulaciones del robot con lo cual, el problema se reduce a la deducción de tres ángulos de posición ya que además, no se requiere la rotación del eslabón correspondiente a la “mano” del robot, ya que la configuración de las plataformas de trabajo y de las piezas construidas mantienen en un mismo plano los elementos a ser manipulados por la pinza del brazo robótico. Cabe resaltar que esta última situación se determinó mediante la observación del comportamiento del sistema completo implementado, durante la etapa inicial de pruebas.

Figura 5.17 Planos de referencia inicial y final ^[*]Figura 5.18 Sistemas de coordenadas y definición de parámetros ^[*]

* Imagen creada por los autores.

Con la notación de Denavit-Hartenberg y la matriz de equivalencia simbólica para sistemas robóticos, se deducen las ecuaciones cinemáticas inversas para las posiciones angulares de las articulaciones del brazo implementado:

	α_i	\mathbf{a}_i	θ_i	\mathbf{d}_i
1	α_1	a_1	0	0
2	α_2	a_2	0	0
3	α_3	a_3	0	0

Tabla 5.3 Parámetros Denavith-Hartenberg para la sección de rotación

Sección de Rotación, giros de eslabón y distancias entre articulaciones:

$${}^3T_4 = \begin{bmatrix} \cos \alpha_1 & 0 & \text{sen } \alpha_1 & a_1 \cos \alpha_1 \\ 0 & 1 & 0 & 0 \\ -\text{sen } \alpha_1 & 0 & \cos \alpha_1 & a_1 \text{sen } \alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 5.5}$$

$${}^4T_5 = \begin{bmatrix} \cos \alpha_2 & 0 & \text{sen } \alpha_2 & a_2 \cos \alpha_2 \\ 0 & 1 & 0 & 0 \\ -\text{sen } \alpha_2 & 0 & \cos \alpha_2 & a_2 \text{sen } \alpha_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 5.6}$$

$${}^5T_6 = \begin{bmatrix} \cos \alpha_3 & 0 & \text{sen } \alpha_3 & a_3 \cos \alpha_3 \\ 0 & 1 & 0 & 0 \\ -\text{sen } \alpha_3 & 0 & \cos \alpha_3 & a_3 \text{sen } \alpha_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 5.7}$$

Con lo que se obtiene:

$${}^3T_6 = \begin{bmatrix} \cos(\alpha_1 + \alpha_2 + \alpha_3) & 0 & \text{sen}(\alpha_1 + \alpha_2 + \alpha_3) & a_1 \cos \alpha_1 + a_2 \cos(\alpha_1 + \alpha_2) + a_3 \cos(\alpha_1 + \alpha_2 + \alpha_3) \\ 0 & 1 & 0 & 0 \\ -\text{sen}(\alpha_1 + \alpha_2 + \alpha_3) & 0 & \cos(\alpha_1 + \alpha_2 + \alpha_3) & a_1 \text{sen} \alpha_1 + a_2 \text{sen}(\alpha_1 + \alpha_2) + a_3 \text{sen}(\alpha_1 + \alpha_2 + \alpha_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 5.8}$$

Determinación de valores a través de matrices de equivalencia simbólica:

$${}^i T_n = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 5.9}$$

Las matrices finales tanto de rotación como de traslación, se igualan a la matriz de equivalencia simbólica y se despejan los valores de las incógnitas requeridas.

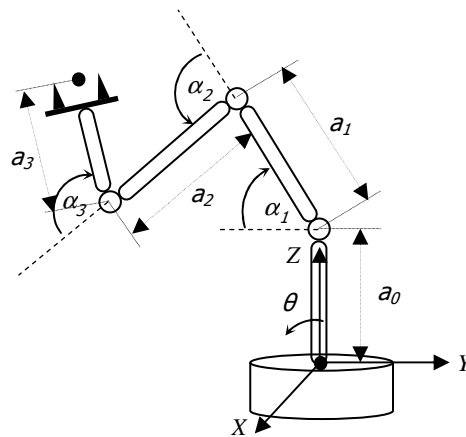


Figura 5.19 Ángulos de movimiento [8]

Tomando las Ecuaciones 5.4 y 5.9 e igualando sus elementos correspondientes, se obtiene:

$$\frac{a_x}{a_y} = \frac{\cos \theta}{\text{sen} \theta} \quad \text{Ec. 5.10}$$

$$\frac{\text{sen} \theta}{\cos \theta} = \frac{a_y}{a_x} \quad \text{Ec. 5.11}$$

$$\tan \theta = \frac{a_y}{a_x} \quad \text{Ec. 5.12}$$

$$\theta = \tan^{-1} \frac{a_y}{a_x} \Rightarrow \theta = \tan^{-1} \frac{C_y}{C_x} \quad \text{Ec. 5.13}$$

Donde:

$a_x = C_x$ = localización en X del objeto identificado en la plataforma de trabajo mediante el procesamiento de imagen.

$a_y = C_y$ = localización en Y del objeto identificado en la plataforma de trabajo mediante el procesamiento de imagen.

De la matriz de rotación, Ecuación 5.8 y 5.9, se obtienen las siguientes igualdades ($P_y=0$):

$$P_x = a_1 \cos \alpha_1 + a_2 \cos(\alpha_1 + \alpha_2) + a_3 \cos(\alpha_1 + \alpha_2 + \alpha_3) \quad \text{Ec. 5.14}$$

$$P_z = a_1 \text{sen} \alpha_1 + a_2 \text{sen}(\alpha_1 + \alpha_2) + a_3 \text{sen}(\alpha_1 + \alpha_2 + \alpha_3) \quad \text{Ec. 5.15}$$

Sea Θ el ángulo total resultante al extremo del último eslabón, entonces:

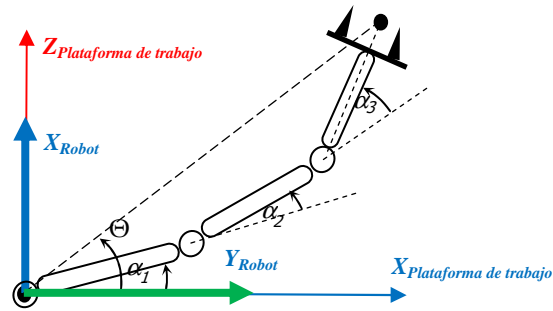


Figura 5.20 Ángulo Θ y ejes ^[8]

$$\Theta = \alpha_1 + \alpha_2 + \alpha_3 \quad \text{Ec. 5.16}$$

$$\Theta = \tan^{-1} \frac{r_x}{r_y} \quad \text{Ec. 5.17}$$

$$r_x = C_z \quad \text{Ec. 5.18}$$

$$r_y = \sqrt{C_x^2 + C_y^2} \quad \text{Ec. 5.19}$$

Donde:

r_x = Componente en el eje X del sistema del robot, eje Z respecto al sistema de la plataforma de trabajo.

r_y = Componente en el eje Y del sistema del robot, eje X respecto al sistema de la plataforma de trabajo.

Entonces:

$$\Theta = \tan^{-1} \frac{C_z}{\sqrt{C_x^2 + C_y^2}} \quad \text{Ec. 5.20}$$

Donde:

C_z = localización en Z del objeto, valor constante = 17,6 cm en dirección negativa (dato de construcción de la planta).

$$\Theta = \alpha_1 + \alpha_2 + \alpha_3 \quad \text{Ec. 5.21}$$

$$\text{sen} \Theta = \text{sen}(\alpha_1 + \alpha_2 + \alpha_3) \quad \text{Ec. 5.22}$$

$$\text{cos} \Theta = \text{cos}(\alpha_1 + \alpha_2 + \alpha_3) \quad \text{Ec.}$$

$$5.23 \quad \tan \Theta = \frac{\text{sen} \Theta}{\text{cos} \Theta} \quad \text{Ec. 5.24}$$

$$\tan \Theta = \frac{C_z}{\sqrt{C_x^2 + C_y^2}} \quad \text{Ec. 5.25}$$

$$\cos \Theta = \frac{1}{\sqrt{1 + \tan^2 \Theta}} \quad \text{Ec. 5.26}$$

$$\cos \Theta = \frac{C_x^2 + C_y^2}{\sqrt{C_x^2 + C_y^2 + C_z^2}} \quad \text{Ec. 5.27}$$

$$\text{sen} \Theta = \frac{1}{\sqrt{1 + \text{ctg}^2 \Theta}} \quad \text{Ec. 5.28}$$

$$\text{sen} \Theta = \frac{C_z}{\sqrt{C_x^2 + C_y^2 + C_z^2}} \quad \text{Ec. 5.29}$$

Se reemplaza en las ecuaciones 5.14 y 5.15:

$$P_X = a_1 \cos \alpha_1 + a_2 \cos(\alpha_1 + \alpha_2) + a_3 \frac{C_x^2 + C_y^2}{\sqrt{C_x^2 + C_y^2 + C_z^2}} \quad \text{Ec. 5.30}$$

$$P_Z = a_1 \text{sen} \alpha_1 + a_2 \text{sen}(\alpha_1 + \alpha_2) + a_3 \frac{C_z}{\sqrt{C_x^2 + C_y^2 + C_z^2}} \quad \text{Ec. 5.31}$$

$$C_x^2 + C_y^2 - a_3 \frac{C_x^2 + C_y^2}{\sqrt{C_x^2 + C_y^2 + C_z^2}} = a_1 \cos \alpha_1 + a_2 \cos(\alpha_1 + \alpha_2) \quad \text{Ec. 5.32}$$

$$C_z - a_3 \frac{C_z}{\sqrt{C_x^2 + C_y^2 + C_z^2}} = a_1 \text{sen} \alpha_1 + a_2 \text{sen}(\alpha_1 + \alpha_2) \quad \text{Ec. 5.33}$$

Ahora, sea:

$$C_x^2 + C_y^2 - a_3 \frac{C_x^2 + C_y^2}{\sqrt{C_x^2 + C_y^2 + C_z^2}} = A \quad \text{Ec. 5.34}$$

$$C_z - a_3 \frac{C_z}{\sqrt{C_x^2 + C_y^2 + C_z^2}} = B \quad \text{Ec. 5.35}$$

Para las ecuaciones 5.34 y 5.35 se realiza la evaluación matemática de elevación al cuadrado y suma de las ecuaciones obteniéndose lo siguiente:

$$A^2 + B^2 = (a_1 \cos \alpha_1 + a_2 \cos(\alpha_1 + \alpha_2))^2 + (a_1 \text{sen} \alpha_1 + a_2 \text{sen}(\alpha_1 + \alpha_2))^2 \quad \text{Ec. 5.36}$$

$$A^2 + B^2 = (a_1 \cos \alpha_1)^2 + (a_2 \cos(\alpha_1 + \alpha_2))^2 + 2 \cdot (a_1 \cos \alpha_1 \cdot a_2 \cos(\alpha_1 + \alpha_2)) + (a_1 \sin \alpha_1)^2 + (a_2 \sin(\alpha_1 + \alpha_2))^2 + 2 \cdot (a_1 \sin \alpha_1 \cdot a_2 \sin(\alpha_1 + \alpha_2))$$

Ec. 5.37

$$A^2 + B^2 = a_1^2 + a_2^2 + 2 \cdot a_1 \cdot a_2 \cdot (\cos \alpha_1 \cdot \cos(\alpha_1 + \alpha_2) + \sin \alpha_1 \cdot \sin(\alpha_1 + \alpha_2))$$

Ec. 5.38

$$A^2 + B^2 = a_1^2 + a_2^2 + 2 \cdot a_1 \cdot a_2 \cdot (\cos(\alpha_1 - \alpha_1 - \alpha_2))$$

Ec. 5.39

$$A^2 + B^2 = a_1^2 + a_2^2 + 2 \cdot a_1 \cdot a_2 \cdot \cos \alpha_2$$

Ec. 5.40

$$\cos \alpha_2 = \frac{A^2 + B^2 - a_1^2 - a_2^2}{2 \cdot a_1 \cdot a_2}$$

Ec. 5.41

Se obtiene entonces, la ecuación para el ángulo α_2 :

$$\alpha_2 = \cos^{-1} \left(\frac{A^2 + B^2 - a_1^2 - a_2^2}{2 \cdot a_1 \cdot a_2} \right)$$

Ec. 5.42

Ahora, de la igualdad:

$$\sin^2 \alpha_2 + \cos^2 \alpha_2 = 1$$

Ec. 5.43

$$\sin \alpha_2 = \sqrt{1 - \cos^2 \alpha_2}$$

Ec. 5.44

$$\tan \alpha_2 = \frac{\sqrt{1 - \cos^2 \alpha_2}}{\cos \alpha_2}$$

Ec. 5.45

se tiene (auxiliarmente):

$$\alpha_2 = \tan^{-1} \left(\frac{\sqrt{1 - \cos^2 \alpha_2}}{\cos \alpha_2} \right)$$

Ec. 5.46

Entonces, para el primer ángulo:

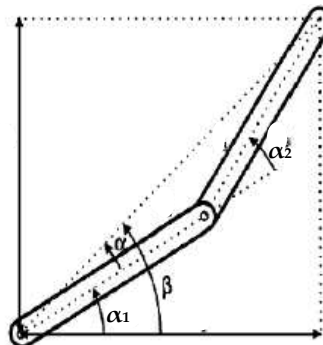


Figura 5.21 Ángulos α , α_1 , β y α_2

$$\alpha_1 = \beta - \alpha \quad \text{Ec. 5.47}$$

$$A = a_1 \cos \alpha_1 + a_2 \cos (\alpha_1 + \alpha_2) \quad \text{Ec. 5.48}$$

$$B = a_1 \operatorname{sen} \alpha_1 + a_2 \operatorname{sen} (\alpha_1 + \alpha_2) \quad \text{Ec. 5.49}$$

$$\tan \beta = \frac{B}{A} \quad \text{Ec. 5.50}$$

$$\beta = \tan^{-1} \left(\frac{B}{A} \right) \quad \text{Ec. 5.51}$$

$$\tan \alpha = \frac{a_2 \operatorname{sen} \alpha_2}{a_1 + a_2 \cos \alpha_2} \quad \text{Ec. 5.52}$$

$$\alpha = \tan^{-1} \left(\frac{a_2 \operatorname{sen} \alpha_2}{a_1 + a_2 \cos \alpha_2} \right) \quad \text{Ec. 5.53}$$

Reemplazando, se obtiene la ecuación para el ángulo α_1 :

$$\alpha_1 = \tan^{-1} \left(\frac{B}{A} \right) - \tan^{-1} \left(\frac{a_2 \operatorname{sen} \alpha_2}{a_1 + a_2 \cos \alpha_2} \right) \quad \text{Ec. 5.54}$$

Para el tercer ángulo (α_3) se tiene:

$$\Theta = \alpha_1 + \alpha_2 + \alpha_3 \quad \text{Ec. 5.55}$$

$$\alpha_3 = \Theta - \alpha_1 - \alpha_2 \quad \text{Ec. 5.56}$$

$$\tan \Theta = \frac{C_z}{\sqrt{C_x^2 + C_z^2}} \quad \text{Ec. 5.57}$$

$$\Theta = \tan^{-1} \left(\frac{C_z}{\sqrt{C_x^2 + C_z^2}} \right) \quad \text{Ec. 5.58}$$

Reemplazando:

$$\alpha_3 = \tan^{-1} \left(\frac{C_z}{\sqrt{C_x^2 + C_z^2}} \right) - \tan^{-1} \left(\frac{B}{A} \right) + \tan^{-1} \left(\frac{a_2 \operatorname{sen} \alpha_2}{a_1 + a_2 \cos \alpha_2} \right) - \tan^{-1} \left(\frac{\sqrt{1 - \cos^2 \alpha_2}}{\cos \alpha_2} \right) \quad \text{Ec. 5.59}$$

5.3 DESARROLLO DEL SOFTWARE DE CONTROL

El sistema completo consiste en realizar la adquisición de la imagen que contenga las piezas geométricas colocadas en la Plataforma A, procesar dicha imagen mediante los algoritmos apropiados en MatLab y generar la ubicación en coordenadas X-Y de los respectivos centroides de las figuras geométricas. Hecho esto se utiliza la información para generar, mediante los algoritmos derivados de los parámetros de Denavit-Hartenberg (Capítulo 5.1.4), las posiciones angulares de las articulaciones del brazo robótico de tal manera que, el elemento terminal (dedos deslizantes), pueda tomar las piezas detectadas y las pueda trasladar hacia su posición final en la Plataforma B.

Para el control del brazo robótico por medio de un computador, se ha desarrollado la interfaz hombre-máquina (HMI) y los algoritmos de control, con el programa LabView 7.0, debido a que esta versión es compatible con la del programa MatLab 7.0 que realiza el procesamiento de imagen, además el sistema de desarrollo en LabView permite una programación rápida y sencilla por medio de un entorno completamente gráfico.

El software consta de cuatro etapas principales: etapa de adquisición, etapa de procesamiento y cálculo, etapa de posicionamiento y etapa de transmisión.

5.3.1 Etapa de Adquisición

El programa LabView, posee la aplicación llamada MatLab Script, que permite establecer comunicación con MatLab, ya sea para escribir directamente en el lenguaje de programación o para invocar aplicaciones desarrolladas con las extensiones “.m”

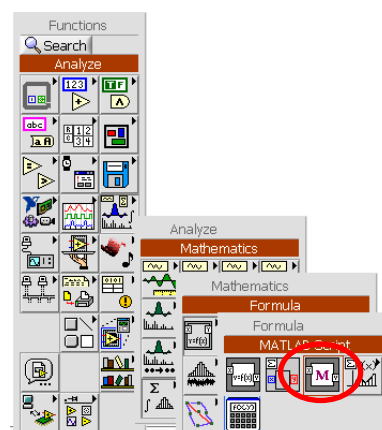


Figura 5.22 Ubicación de la aplicación MATLAB Script en la paleta de Funciones de LabView

En el software desarrollado solamente se necesita obtener la información de localización del centroide en un plano X-Y, generada por el algoritmo de procesamiento de imágenes del MatLab (variable: centro) y una bandera que indica que se ha terminado de realizar la operación (variable: terminado).

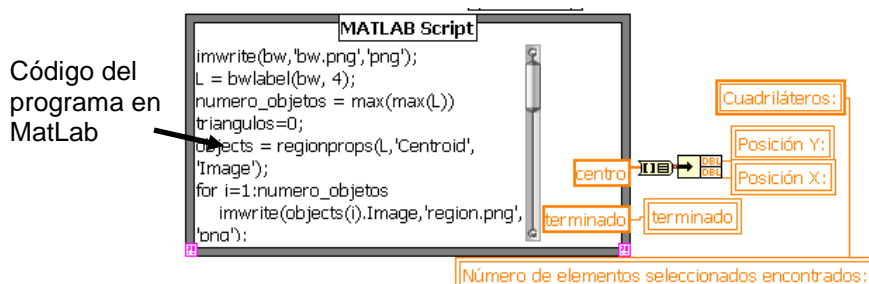


Figura 5.23 Script de MatLab en el programa LabView

Esta información indica la correcta localización del centroide de la figura seleccionada en pixeles, por lo que se realiza la conversión a centímetros con las funciones matemáticas del LabView y de acuerdo a las dimensiones de la planta.

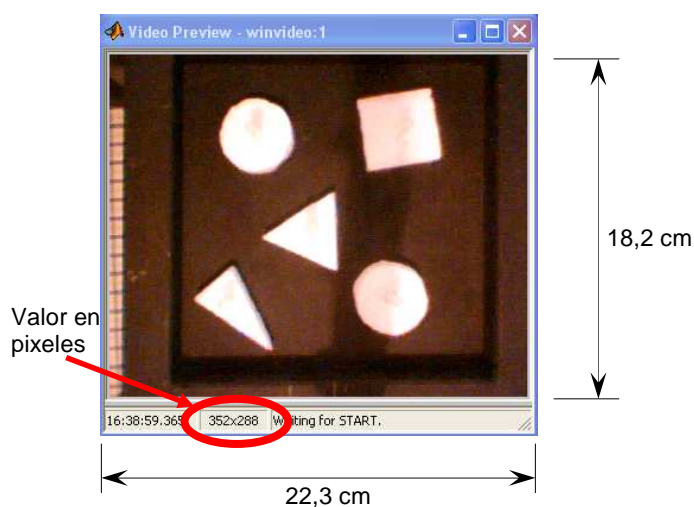


Figura 5.24 Valor en pixeles de la cámara web y valores en cm de la planta

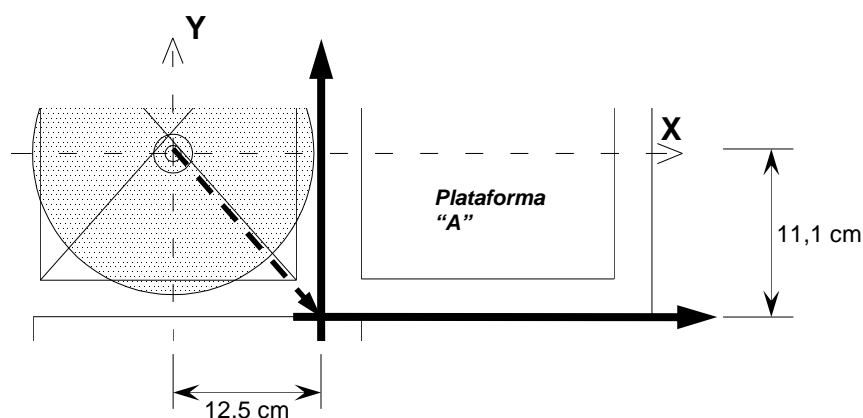


Figura 5.25 Distancias para la traslación del eje en la planta

Efectuando una regla de tres simple y trasladando los ejes se tiene que:

$$Posición X(cm) = \frac{Posición X (píxeles) \cdot 18,2 (cm)}{288 (píxeles)} + 12,5 (cm) \quad \text{Ec. 5.60}$$

$$Posición Y(cm) = \frac{Posición Y (píxeles) \cdot 22,3 (cm)}{352 (píxeles)} - 11,1 (cm) \quad \text{Ec. 5.61}$$

La implementación de esta conversión se observa en la Figura 5.26

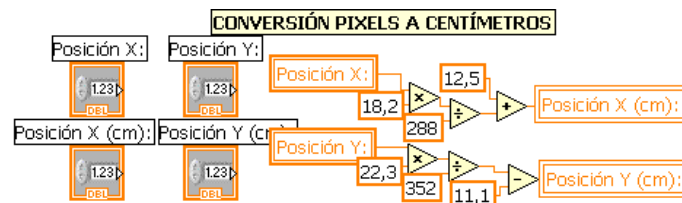


Figura 5.26 Conversión de píxeles a centímetros en LabView

5.3.2 Etapa de Procesamiento y Cálculo

Con la información de posición en el plano X-Y en centímetros, se aplican las ecuaciones cinemáticas inversas obtenidas en la sección 5.1.4:

Ecuación 5.13 para el ángulo No.4 (“base”):

$$\theta = \tan^{-1} \frac{C_y}{C_x}$$

Ecuación 5.54 para el ángulo No.1 (“hombro”):

$$\alpha_1 = \tan^{-1} \left(\frac{B}{A} \right) - \tan^{-1} \left(\frac{a_2 \operatorname{sen} \alpha_2}{a_1 + a_2 \cos \alpha_2} \right)$$

Ecuación 5.42 para el ángulo No.2 (“codo”):

$$\alpha_2 = \cos^{-1} \left(\frac{A^2 + B^2 - a_1^2 - a_2^2}{2 \cdot a_1 \cdot a_2} \right)$$

Ecuación 5.59 para el ángulo No.3 (“muñeca”):

$$\alpha_3 = \tan^{-1} \left(\frac{C_z}{\sqrt{C_x^2 + C_y^2}} \right) - \tan^{-1} \left(\frac{B}{A} \right) + \tan^{-1} \left(\frac{a_2 \operatorname{sen} \alpha_2}{a_1 + a_2 \cos \alpha_2} \right) - \tan^{-1} \left(\frac{\sqrt{1 - \cos^2 \alpha_2}}{\cos \alpha_2} \right)$$

Donde:

$$A = C_x^2 + C_y^2 - a_3 \frac{C_x^2 + C_y^2}{\sqrt{C_x^2 + C_y^2 + C_z^2}} \quad (\text{Ec. 5.34})$$

$$B = C_z - a_3 \frac{C_z}{\sqrt{C_x^2 + C_y^2 + C_z^2}} \quad (\text{Ec. 5.35})$$

C_x = coordenada en x (cm)

C_z = coordenada en z (cm) = -17,6 cm (valor fijo de la planta)

a_1 = valor del eslabón No.1 = 12cm (valor fijo de la planta)

a_2 = valor del eslabón No.3 = 7,7cm (valor fijo de la planta)

a_3 = valor del eslabón No.3 = 18cm (valor fijo de la planta)

Se implementan estas ecuaciones en LabView:

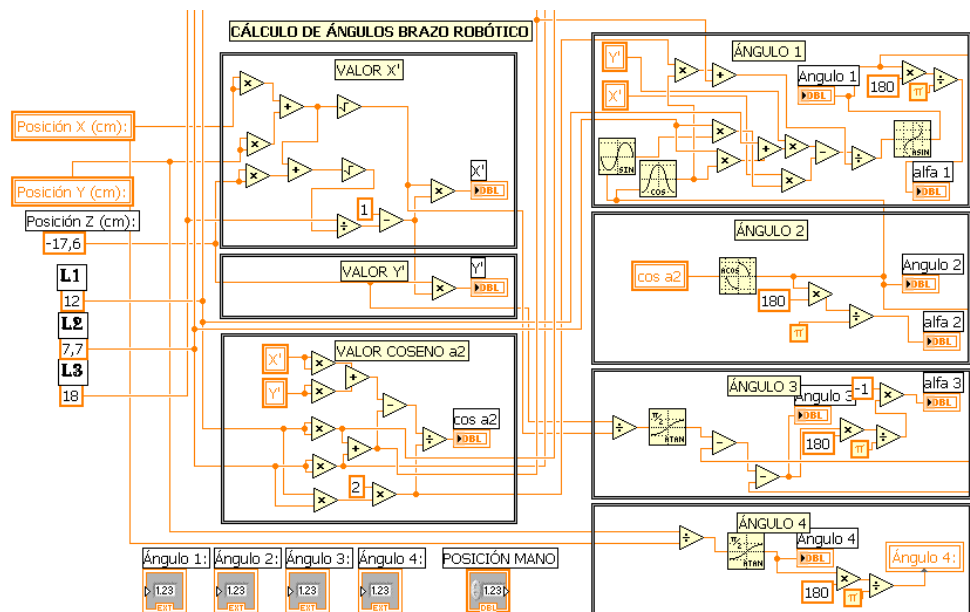


Figura 5.27 Implementación de las ecuaciones Denavit-Hartenberg en LabView

Sin embargo, con los resultados de los ángulos obtenidos con estos cálculos se tiene el problema de la imposibilidad física real de que la planta pueda realizar en la práctica estos movimientos, por lo que se hace necesario realizar una corrección geométrica para que los eslabones y articulaciones del brazo robótico implementado puedan colocarse en posiciones alcanzables.

De manera experimental, se introdujeron los datos en una hoja en Excel tomando como referencia objetos cuyos centroides se suponen ubicados en la plataforma "A" a 14,1cm, 20,9 cm y 27,9 cm del eje X de la base del brazo robótico y se dibujaron (con la ayuda del AutoCAD) los eslabones con sus respectivos ángulos generados por las ecuaciones, manteniendo las dimensiones del sistema real en una representación vectorial gráfica. Utilizando la herramienta "Mirror" del AutoCAD, se observó que es posible generar ángulos a los que en adelante se denomina como "ángulos espejo", los cuales permitían una colocación real de los eslabones del brazo robótico y en una posición que requiere menos esfuerzo mecánico por parte de los servomotores, por lo que se implementó una corrección geométrica aplicable a los resultados de las ecuaciones anteriormente obtenidas.

Esta corrección implica resolver los "ángulos espejo" de los triángulos que forman las posiciones de los respectivos eslabones.

Del dibujo de simulación generado en AutoCAD, se observa:

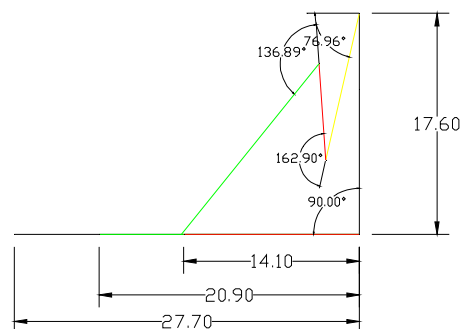


Figura 5.28 Simulación de posiciones en AutoCAD

Utilizando la herramienta "Mirror" a los dos primeros eslabones se obtiene:

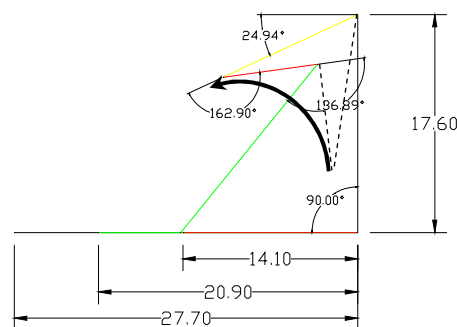


Figura 5.29 Utilización de herramienta mirror

Tomando como datos de ejemplo los ángulos dibujados, se reduce el problema a la solución de la gráfica:

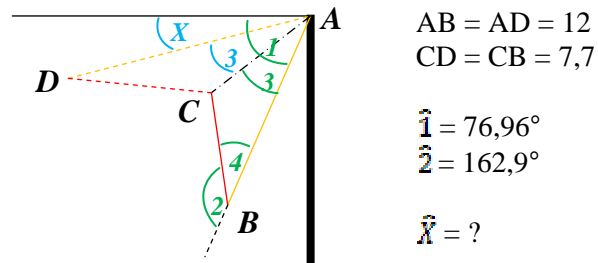


Figura 5.30 Planteamiento del problema geométrico

$$\hat{X} + \hat{3} + \hat{3} = 76,96^\circ \quad \text{Ec.5.62}$$

$$\hat{4} = 180^\circ - \hat{2} = 17,1^\circ \quad \text{Ec.5.63}$$

$$AC^2 = CB^2 + AB^2 - 2(AB)(CB)\cos(\hat{4}) \quad \text{Ec.5.64}$$

$$AC^2 = 7,7^2 + 12^2 - 2(12)(7,7)\cos(17,1^\circ) \quad \text{Ec.5.65}$$

$$AC = 5,16 \quad \text{Ec.5.66}$$

$$\cos(\hat{3}) = -\frac{CB^2 + AC^2 + AB^2}{2(AC)(AB)} \quad \text{Ec.5.67}$$

$$\cos(\hat{3}) = -\frac{7,7^2 + 5,16^2 + 12^2}{2(5,16)(12)} \quad \text{Ec.5.68}$$

$$\hat{3} = 25,96^\circ \quad \text{Ec.5.69}$$

$$2 \cdot \hat{3} = 51,93 \quad \text{Ec.5.70}$$

$$\hat{X} = \hat{1} - (2 \cdot \hat{3}) \quad \text{Ec.5.71}$$

$$\hat{X} = 25^\circ \quad \text{Ec.5.72}$$

Este resultado es coincidente con el obtenido mediante el AutoCAD, por lo que para generalizar, se deduce el siguiente algoritmo de corrección geométrica:

$$I) \quad \hat{4} = 180^\circ - \hat{2} \quad \text{Ec.5.73}$$

$$II) \quad AC = \sqrt{CB^2 + AB^2 - 2(AB)(CB)\cos(\hat{4})} \quad \text{Ec.5.74}$$

$$III) \quad \cos(\hat{3}) = -\frac{CB^2 + AC^2 + AB^2}{2(AC)(AB)} \quad \text{Ec.5.75}$$

$$IV) \quad \hat{X} = \hat{1} - (2 \cdot \hat{3}) \quad \text{Ec.5.76}$$

Aplicando la herramienta “*Mirror*” para los demás casos de estudio se tienen los resultados de la Figura 5.31:

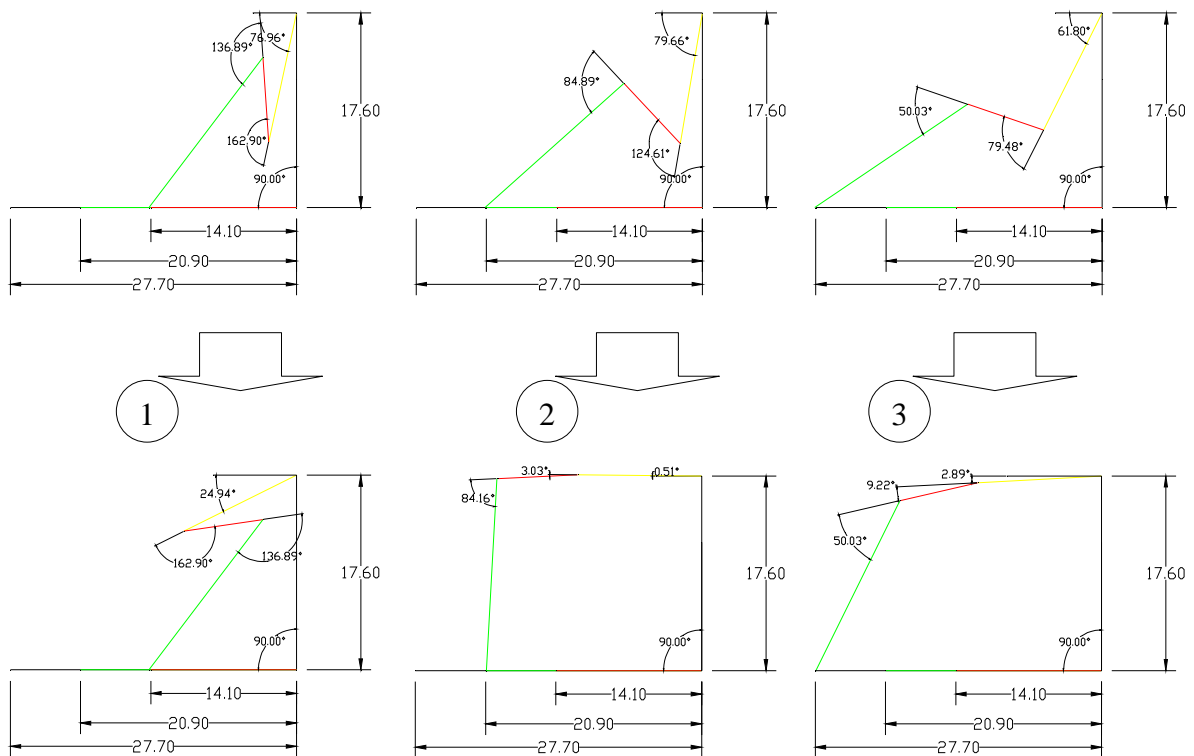


Figura 5.31 “Ángulos espejo” obtenidos mediante la corrección geométrica en AutoCAD

Como se puede apreciar, para los casos 2 y 3 se obtienen teóricamente posiciones más adecuadas para los eslabones del brazo robótico, sin embargo, en el caso 1 se nota que es necesario realizar una operación adicional de la herramienta “*Mirror*”, pero en este caso se aplica primero en los eslabones correspondientes al antebrazo y mano-pinza y luego, por segunda vez en los eslabones correspondientes al brazo y antebrazo; experimentalmente se pudo comprobar que esta situación se presenta para todos los casos de elementos localizados a distancias inferiores a 20,65 cm del eje de la base del brazo robótico, esto se debe a que en las ecuaciones deducidas de los parámetros de Denavit-Hartenberg, todos los valores menores a esta distancia caen en un cuadrante distinto, debido a la redundancia de las ecuaciones trigonométricas utilizadas.

Con esta consideración, para el caso 1 se tiene la siguiente configuración:

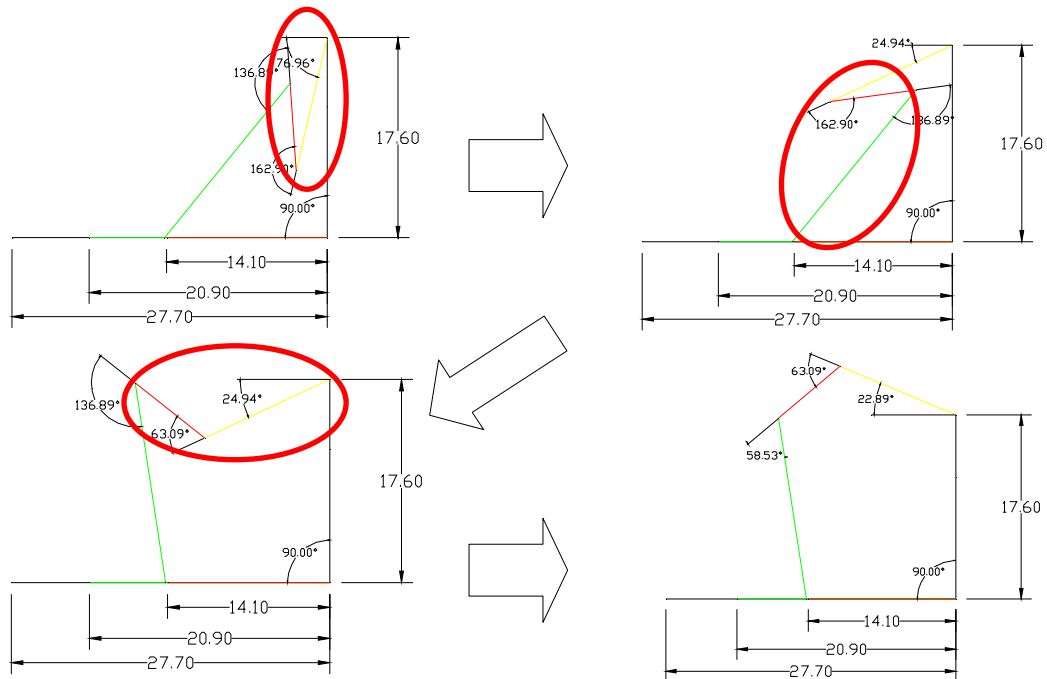


Figura 5.32 “Ángulos espejo” obtenidos para el caso 1

Como se puede apreciar, para este caso específico se debe aplicar el algoritmo de ángulos espejo tres veces para obtener posiciones articulares físicamente viables. Una vez establecida la secuencia a seguir del procedimiento de corrección geométrica, se lo implementa en el programa LabView.

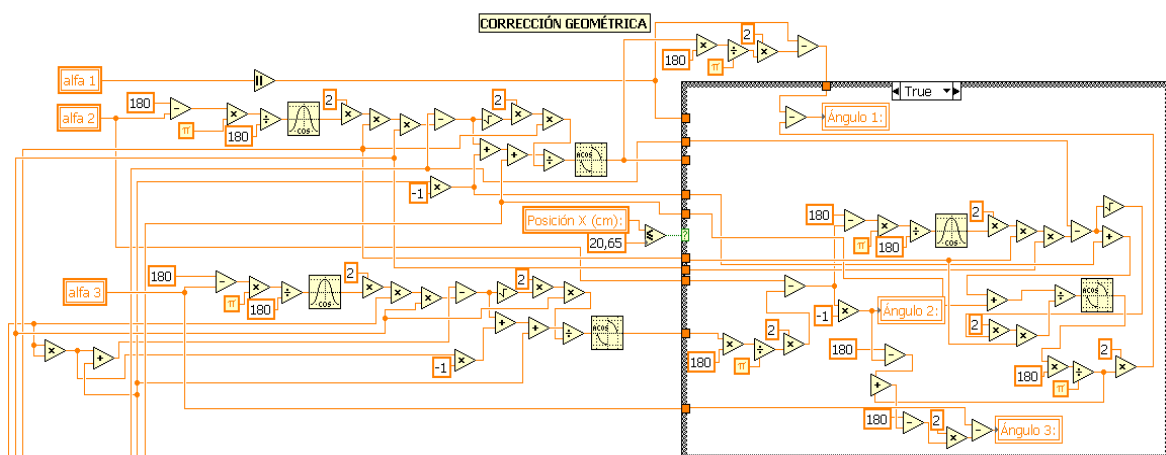


Figura 5.33 Corrección geométrica implementada en LabView

5.3.3 Etapa de Posicionamiento

Esta etapa del programa toma las variables generadas de los ángulos de cada una de las articulaciones del brazo robótico y los ordena en una secuencia para que el posicionamiento de los elementos del robot sea el adecuado y

posteriormente se coloque cada una de las figuras en un orden preestablecido y organizadas de acuerdo a su forma.

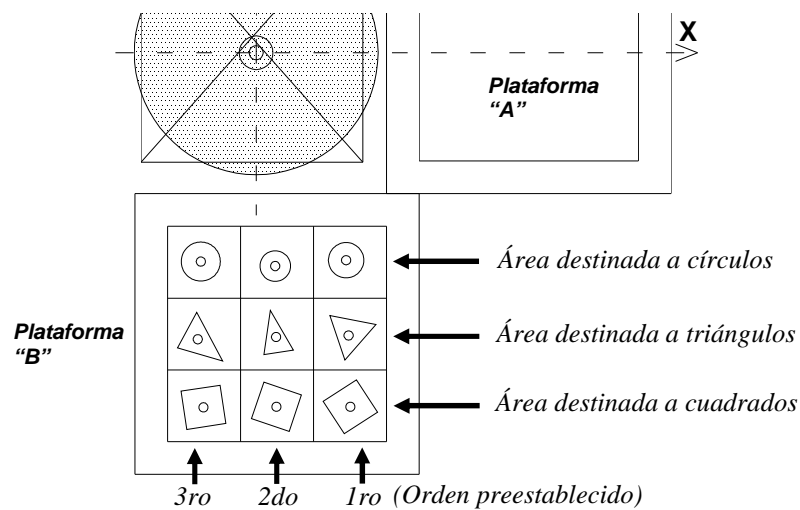


Figura 5.34 Orden de las figuras de acuerdo a su forma

Se han previsto dos tipos de control: manual y automático, a su vez en el modo de control automático se cuenta con la modalidad semiautomática y automática.

En el modo de control manual, existen seis controles deslizantes, uno para cada una de las articulaciones del brazo robótico, incluyendo el comando para abrir y cerrar la pinza del elemento terminal.

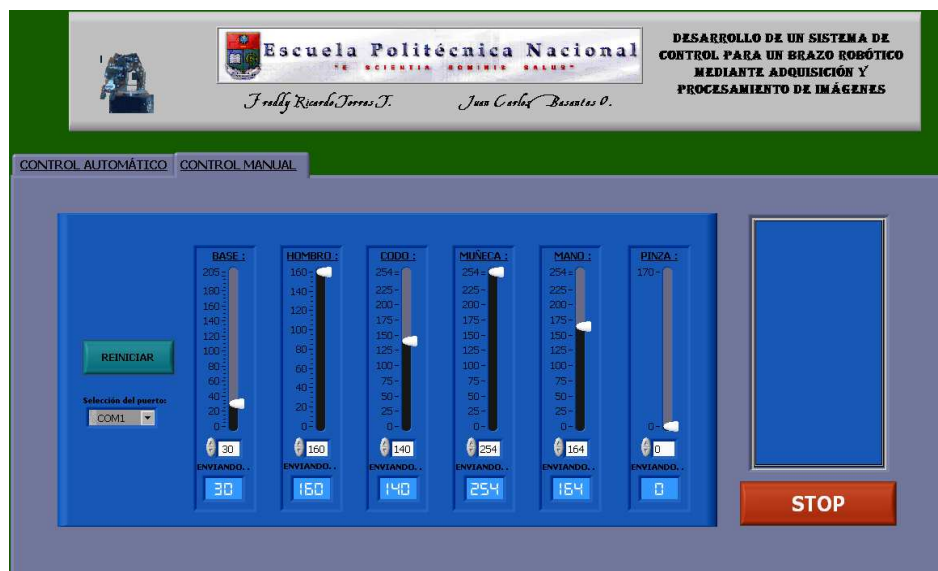


Figura 5.35 HMI implementada en control manual

Con estos controles se envía una señal en secuencia ascendente o descendente en pasos de uno hacia el puerto serial, de manera que se controlen las señales PWM de los respectivos servomotores en el robot.

Es necesario retardar la secuencia de la señal PWM debido a que se comprobó que, cuando ésta es enviada sin retardo, los motores del robot se mueven demasiado rápido y pueden romperse sus engranajes internos debido a la inercia que se ejerce al mover toda la estructura del brazo robótico.

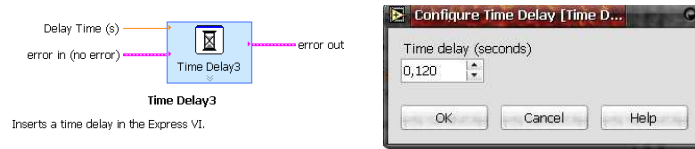


Figura 5.36 Retardo de tiempo configurable (Time Delay)

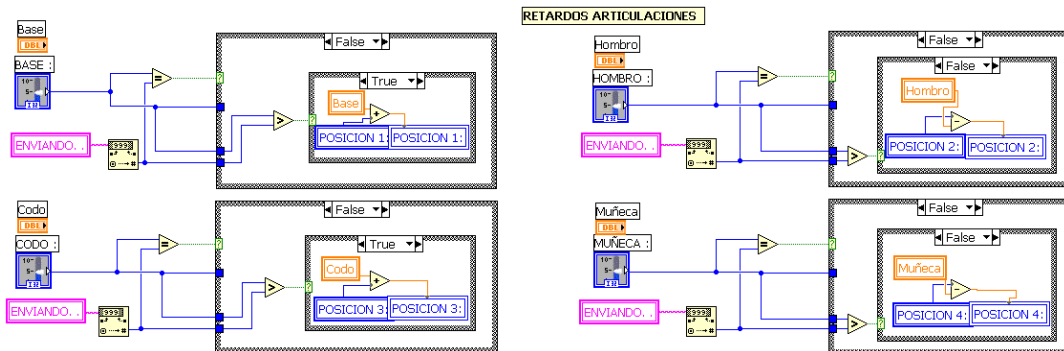


Figura 5.37 Ascenso y descenso secuencial implementado

Con esta opción de control individual, se puede experimentar con las posiciones de cada una de las articulaciones del brazo robótico por separado, de esta forma se consiguieron los valores a ser enviados a los servomotores para predefinir las posiciones finales del robot al colocar ordenadamente las piezas geométricas en la plataforma denominada “B”.

POSICIÓN FINAL PLATAFORMA B								
RNAL CUADRADOS BASE 1	RNAL TRIANGULOS BASE 1	RNAL CIRCULOS BASE 1	RNAL CUADRADOS HOMBRO 1	RNAL TRIANGULOS HOMBRO 1	RNAL CIRCULOS HOMBRO 1	RNAL CUADRADOS CODO 1	RNAL TRIANGULOS CODO 1	RNAL CIRCULOS CODO 1
166	162	158	DB1	DB1	DB1	DB1	DB1	DB1
42	46	56	DB1	DB1	DB1	DB1	DB1	DB1
90	100	138	DB1	DB1	DB1	DB1	DB1	DB1
200	222	222	DB1	DB1	DB1	DB1	DB1	DB1
104	104	104	DB1	DB1	DB1	DB1	DB1	DB1
RNAL CUADRADOS BASE 2	RNAL TRIANGULOS BASE 2	RNAL CIRCULOS BASE 2	RNAL CUADRADOS HOMBRO 2	RNAL TRIANGULOS HOMBRO 2	RNAL CIRCULOS HOMBRO 2	RNAL CUADRADOS CODO 2	RNAL TRIANGULOS CODO 2	RNAL CIRCULOS CODO 2
178	178	178	DB1	DB1	DB1	DB1	DB1	DB1
44	46	60	DB1	DB1	DB1	DB1	DB1	DB1
90	100	148	DB1	DB1	DB1	DB1	DB1	DB1
206	228	222	DB1	DB1	DB1	DB1	DB1	DB1
104	104	104	DB1	DB1	DB1	DB1	DB1	DB1
RNAL CUADRADOS BASE 3	RNAL TRIANGULOS BASE 3	RNAL CIRCULOS BASE 3	RNAL CUADRADOS HOMBRO 3	RNAL TRIANGULOS HOMBRO 3	RNAL CIRCULOS HOMBRO 3	RNAL CUADRADOS CODO 3	RNAL TRIANGULOS CODO 3	RNAL CIRCULOS CODO 3
190	194	200	DB1	DB1	DB1	DB1	DB1	DB1
44	46	60	DB1	DB1	DB1	DB1	DB1	DB1
90	100	148	DB1	DB1	DB1	DB1	DB1	DB1
206	226	222	DB1	DB1	DB1	DB1	DB1	DB1
104	104	104	DB1	DB1	DB1	DB1	DB1	DB1

Figura 5.38 Valores predeterminados de posiciones para plataforma “B”

Para el control automático se han implementado dos modalidades: la primera con el nombre de “Semi-Automático” para demostrar la secuencia de operación del programa y la segunda “Automático” para una mínima intervención del usuario.

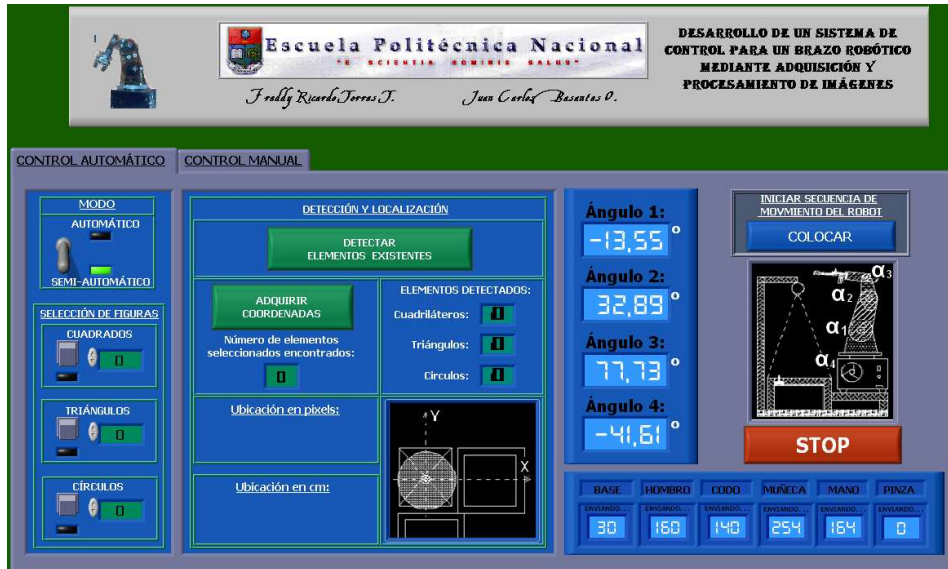
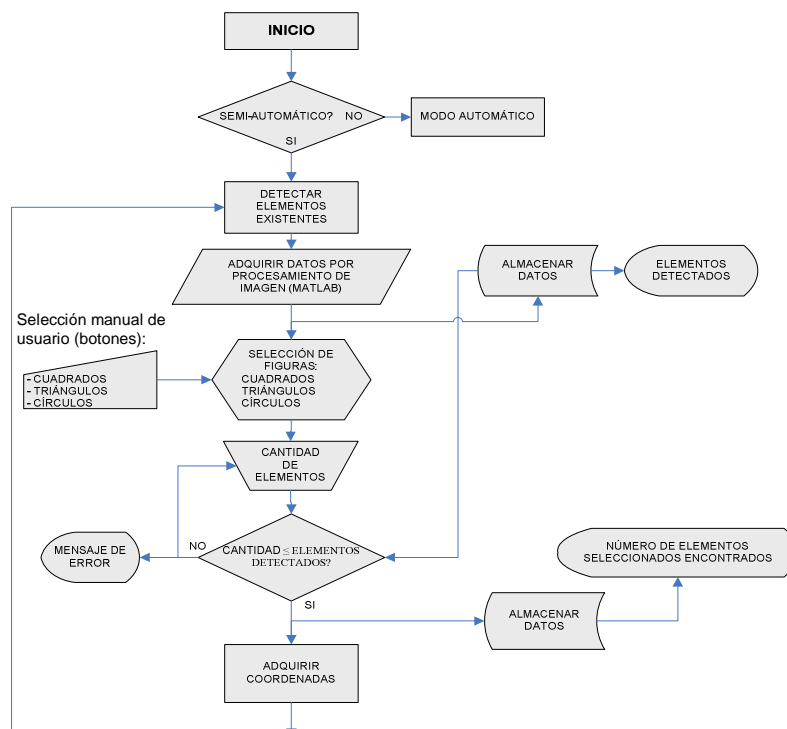


Figura 5.39 HMI implementada en control Semi-Automático

En el modo “Semi-Automático” la HMI presenta un panel del control con los botones de control necesarios para controlar la secuencia de etapas de operación una a la vez y permitir realizar al usuario una exposición y un análisis de la metodología utilizada para la implementación del sistema integrado cámara-robot.

La secuencia de operación es la siguiente:



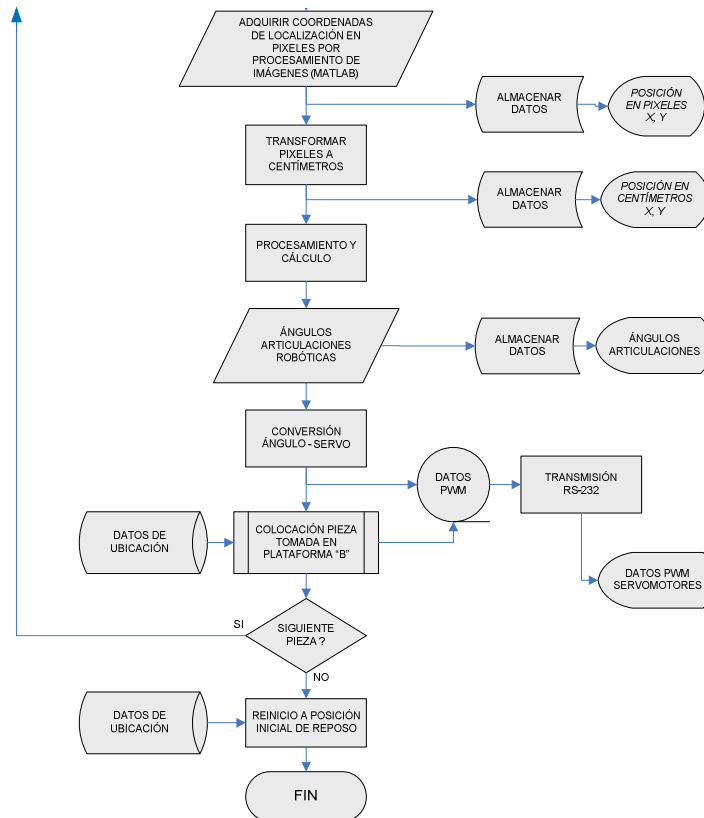


Figura 5.40 Diagrama de flujo del sistema de control por software [*]

En el modo “Automático”, el usuario selecciona el tipo de figura que se desee sólo pulsando el botón correspondiente en la sección “Selección de Figuras”.

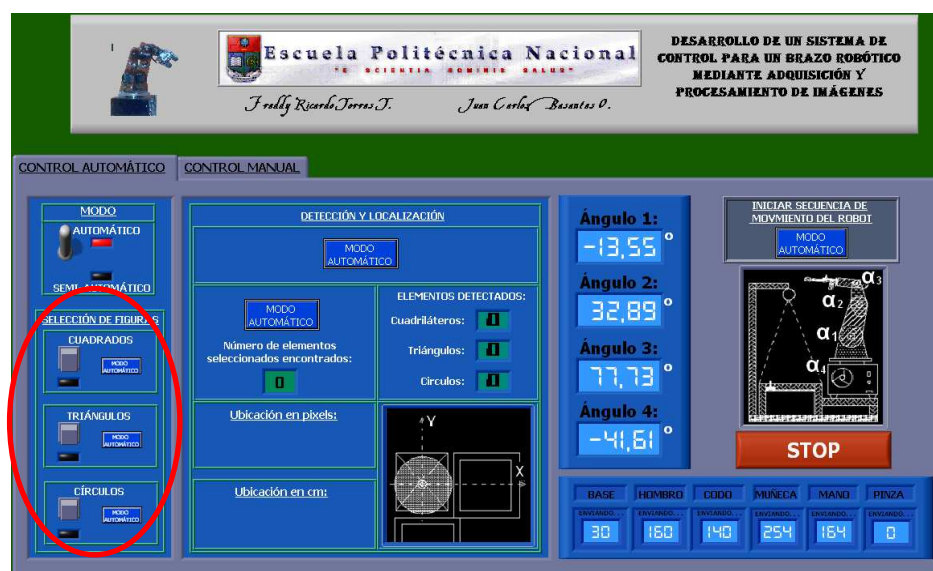


Figura 5.41 HMI implementada en control Automático

* Imagen creada por los autores.

Realizando esta acción, el software anula los controles del modo “Semi-Automático” y toma el control de las secuencias de operación del sistema integrado cámara-robot hasta completar el proceso de detección y transporte de todas las figuras seleccionadas que la cámara encuentre en la plataforma.

La adecuada secuencia de datos se genera mediante una sub-etapa del programa, la cual tiene preestablecido el movimiento inicial del brazo hasta colocarlo en una posición cercana a la plataforma de las piezas geométricas, para luego tomar los datos generados y corregidos de las posiciones angulares que hacen posible la colocación del elemento terminal encima del centro de la figura seleccionada y detectada, hecho esto el programa sigue una secuencia de movimiento del brazo hasta la plataforma “B” y, mediante una serie de parámetros de posicionamiento definidos previamente, coloca las piezas en orden de acuerdo a su respectiva forma. Terminada esta acción, si se encuentra en modo “Semi-Automático”, el usuario debe seleccionar otra figura o reiniciar la secuencia; si está en modo “Automático”, el sistema verificará la existencia de más piezas de la figura seleccionada y procederá a tomar la siguiente figura o, si ya no se detectan más, volverá a colocar el brazo robótico en su posición inicial de reposo.

En cualquier momento mientras el brazo robótico está operando, se tiene un sistema de parada de emergencia, el cual detiene los movimientos del robot y genera una señal de aviso con un cuadro de diálogo, el mismo que tiene las opciones de reiniciar el robot a su estado inicial de reposo o de salir del programa.



Figura 5.42 Parada de emergencia y cuadro de diálogo

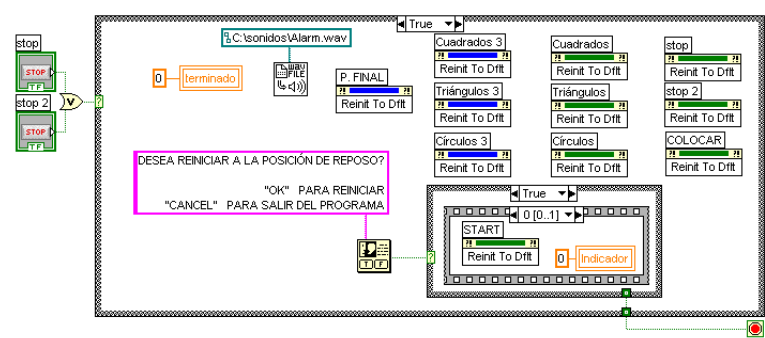


Figura 5.43 Programación de la parada de emergencia (en el caso “False” no realiza ninguna acción)

La rutina de reinicio vuelve todos los valores del software a su estado inicial y permite, ya sea en modo manual o automático, volver a configurar las opciones del usuario.

5.3.4 Etapa de Transmisión

El programa LabView posee una librería de comunicación con la interfaz RS-232 del puerto serial de un computador denominada *NI-VISA (National Instruments - Virtual Instrument Software Architecture)*, la cual se utiliza para enviar una señal a través de este puerto a la tarjeta controladora de los servomotores.

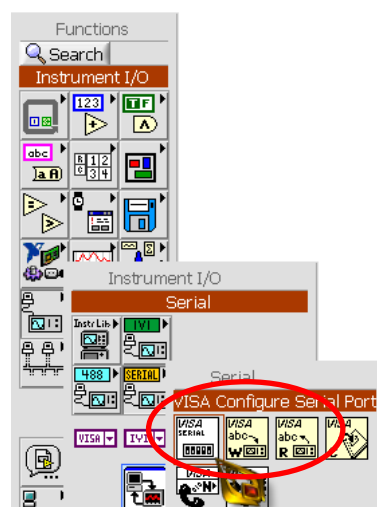


Figura 5.44 Ubicación de la aplicación *NI-VISA* en la paleta de Funciones de LabView

En el presente proyecto se utilizaron las funciones *VISA Configure Serial Port.vi* (para configurar los parámetros de transmisión) y *VISA Write* (para el envío de datos del computador a la tarjeta de control).

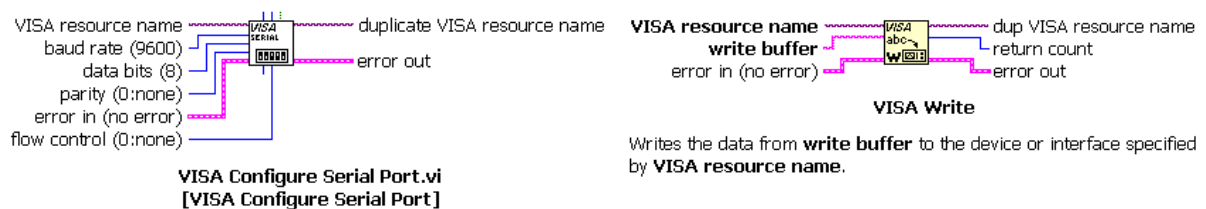


Figura 5.45 Funciones *VISA Configure Serial Port.vi* y *VISA Write*

El software para la transmisión de datos por el puerto serial se construye a partir de secuencias para enviar los respectivos Bytes de información hacia la tarjeta de control de los servomotores Mini SSC II.

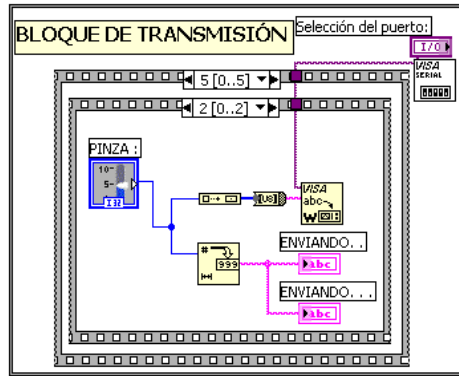


Figura 5.46 Bloque de programa para transmisión serial RS-232

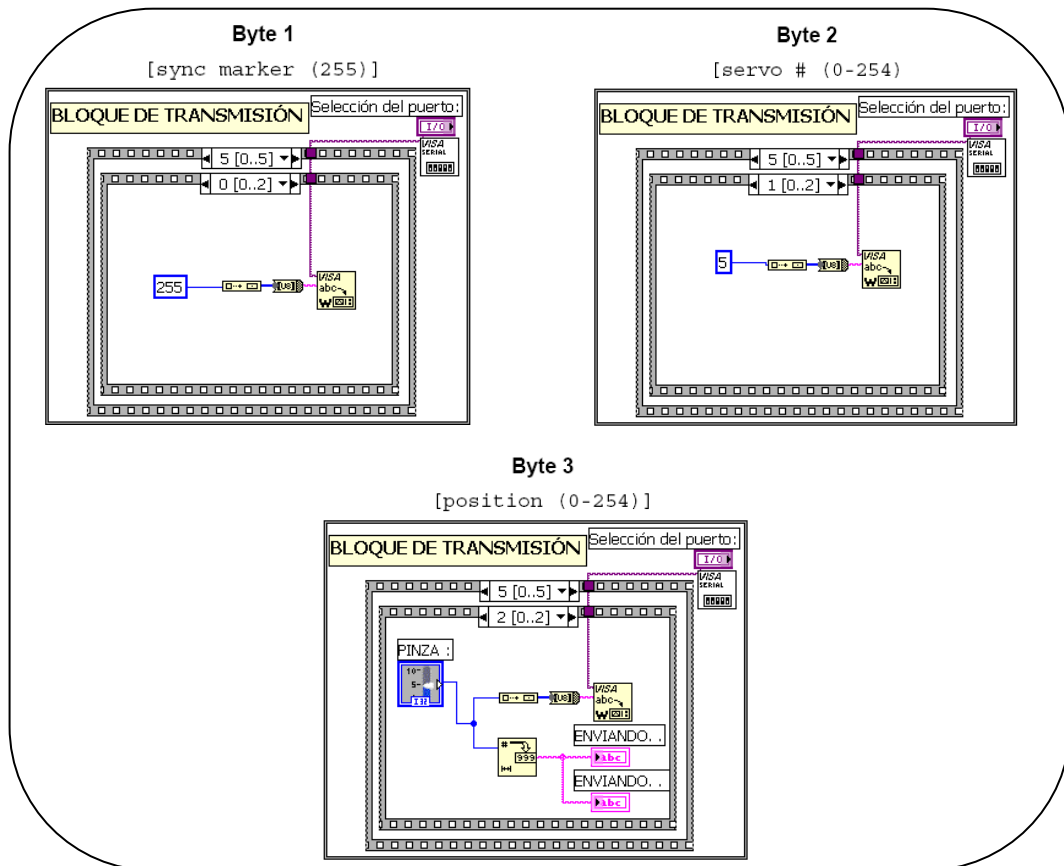


Figura 5.47 Secuencia de transmisión (para el control de la Pinza)

CAPÍTULO 6

PRUEBAS Y RESULTADOS

Una vez desarrollados los sistemas tanto de la captura y procesamiento de imágenes, como de la estructura y control del brazo robótico, se procedió a la integración de todos los componentes y a su calibración para poner en funcionamiento el sistema completo.

6.1 CALIBRACIÓN Y PRUEBAS DEL SISTEMA DE ADQUISICIÓN DE IMÁGENES

Se construyó la estructura de la planta de tal forma que se pueda colocar la cámara en un sitio libre del tránsito del brazo robótico y complementariamente se colocaron elementos telescópicos para ajustar la correcta posición de la cámara a fin de que pueda capturar de la mejor manera la imagen de la plataforma *A* que contiene las piezas a identificar y trasladar. Ajustando estos elementos telescópicos y el enfoque de la cámara web utilizada (*Logitech, QuickCam Orbit/Sphere*), se logró una excelente visualización de la plataforma y sus elementos a una altura de 34 cm y a 10 cm desde su borde.

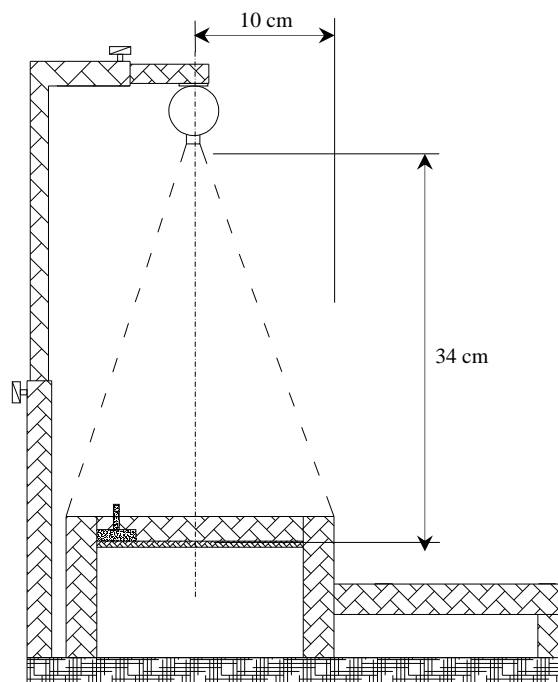


Figura 6.1 Distancias para la colocación de la cámara ^[*]

* Imagen creada por los autores.

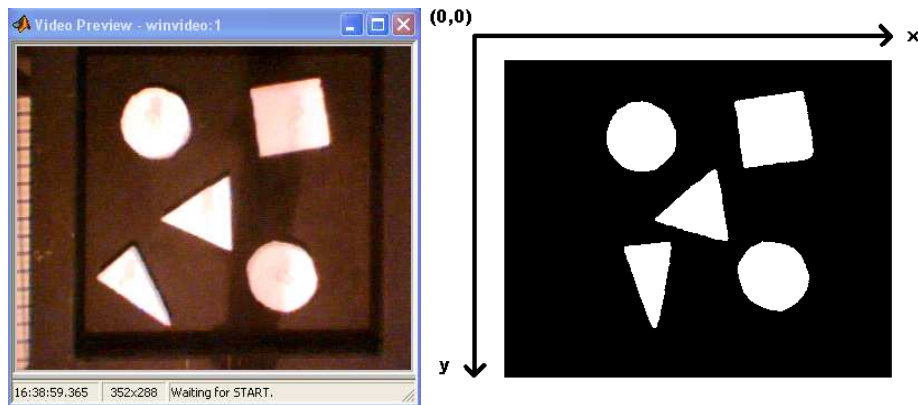


Figura 6.2 Imágenes obtenidas y coordenadas X-Y^[*]

La imagen capturada por la cámara se somete a un pre-procesamiento, en donde se cambia el campo de acción de la imagen, primero en una escala de grises para luego ser binarizada.

La imagen binarizada en donde los 1s representan la parte blanca dentro de la imagen mientras que los 0s representan el fondo de la misma. Durante todo el proceso se usa una imagen de 352x288 píxeles.

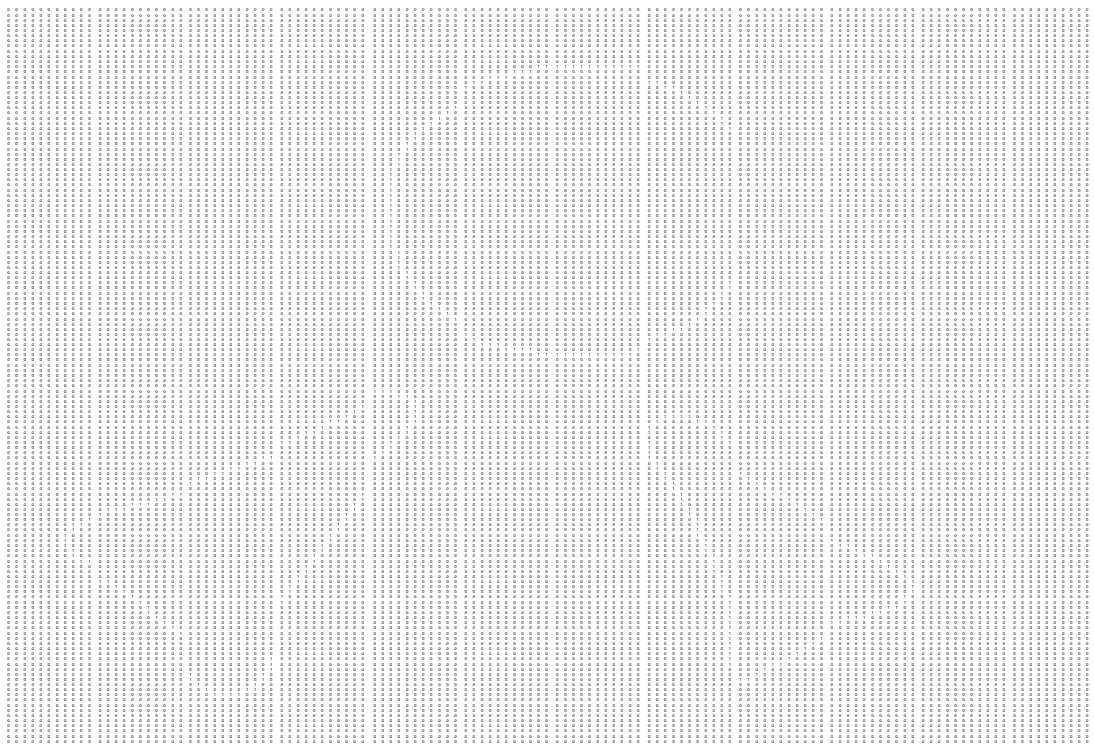


Figura 6.3 Matriz de Imagen Binaria^[*]

* Imagen creada por los autores.

En la Figura 6.3 muestra la matriz de una imagen binaria en donde se puede visualizar los contornos de un círculo y dos triángulos representados por 1's mientras que el fondo y el interior de las figuras se encuentran representados por medio de 0's.

La Figura 6.4 muestra la imagen pre-procesada de la cual se extraerán las características para distinguir figuras geométricas regulares tales como círculos, cuadrados y triángulos.

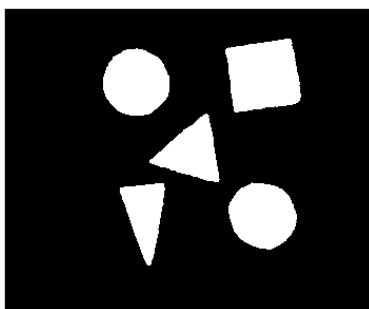


Figura 6.4 Imagen Pre-procesada ^[*]

6.1.1 SELECCIÓN DE CÍRCULOS

Como se describió en el capítulo tres, la imagen capturada por la cámara se somete a eliminación de ruido, a un proceso de morfología binaria y luego a una etapa de etiquetado; posteriormente se detectan los centroides de cada una de las regiones y se visualizan únicamente en las figuras seleccionadas por el usuario, en este caso se muestran los centroides correspondientes a los círculos.

Para el ejemplo de la Figura 6.6, en los dos círculos se detectan los centroides de cada una de las regiones de interés en los píxeles $X=124.94$, $Y=70.16$ para el primer círculo, mientras que el centroide del segundo círculo se encuentra en la ubicación $X=244.98$, $Y=196.18$, además se obtiene la cantidad de círculos detectados y el número de figuras encontradas dentro de la imagen.

* Imagen creada por los autores.

```

Command Window
>> eliminacion_ruido
>> morfologial
>> etiquetadocirculos

centro_circulo =

    124.9430    70.1641

centro_circulo =

    244.9818    196.1831

circulosfinal =

     2

```

Figura 6.5 Variables enviadas, correspondientes a la Etapa de Posicionamiento “Círculos”

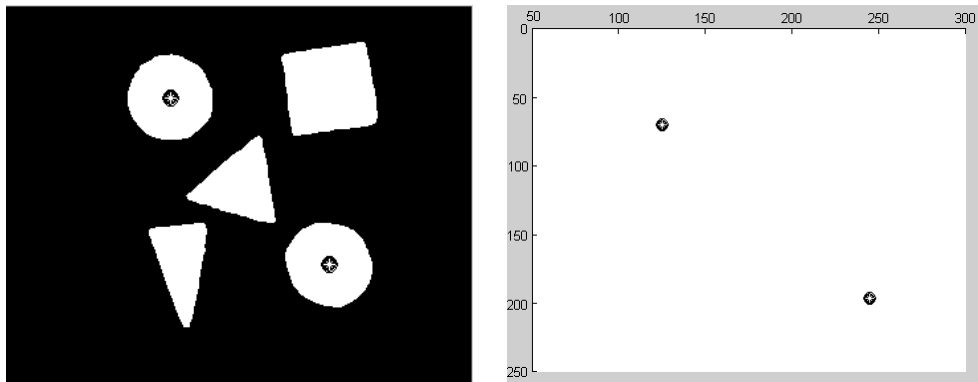


Figura 6.6 Centroides de los Círculos Detectados [*]

6.1.2 SELECCIÓN DE CUADRADOS

De manera análoga a la selección de círculos, para la alternativa de cuadrados se realiza un proceso similar, en donde se envían las coordenadas de los centroides encontrados y la cantidad de cuadrados contenidos dentro de la imagen.

```

Command Window
>> eliminacion_ruido
>> morfologial
>> etiquetadocuadrados

centro_cuadrado =

    245.6827    63.5398

cuadrosfinal =

     1

>>

```

Figura 6.7 Variables enviadas, correspondientes a la Etapa de Posicionamiento “Cuadrados”

* Imagen creada por los autores.

La Figura 6.7 muestra que se encontró un cuadrado, en las coordenadas $X=245.68$, $Y=63.53$.

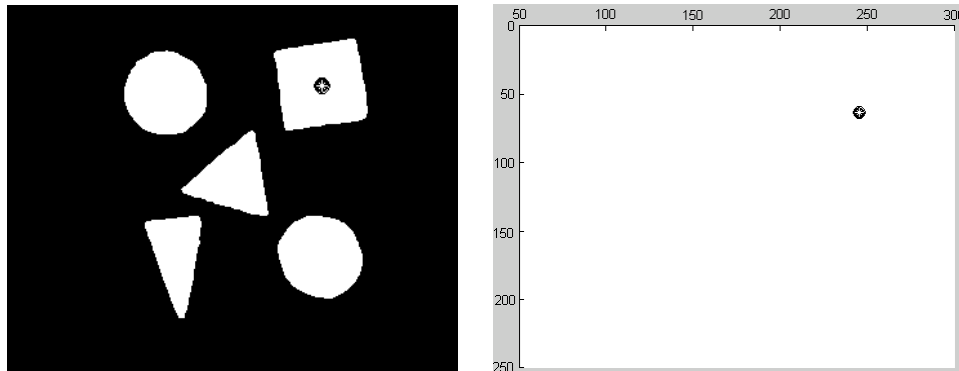


Figura 6.8 Centroide del Cuadrado Detectado [8]

6.1.3 SELECCIÓN DE TRIÁNGULOS

De igual manera que los casos anteriores se realiza un proceso previo de procesamiento de imagen, y luego se envían las coordenadas, en este caso de los triángulos encontrados dentro de la imagen, la cantidad de figuras encontradas y la cantidad total de triángulos detectados.

```

Command Window
>> eliminacion_ruido
>> morfologia1
>> etiquetadotriangulos2

numero_objetos =

    5

centro_triangulos =

    133.2147  195.4865

centro_triangulos =

    177.6234  136.3494

triangulos_final =

    2

```

Figura 6.9 Variables enviadas, correspondientes a la Etapa de Posicionamiento “Triángulos”

En este caso se encontraron dos triángulos ubicados en las coordenadas $X=133.21$, $Y=195.48$ y $X=177.62$, $Y=136.34$ mostrados en Figura 6.9, así como las cinco figuras encontradas en total.

* Imagen creada por los autores.

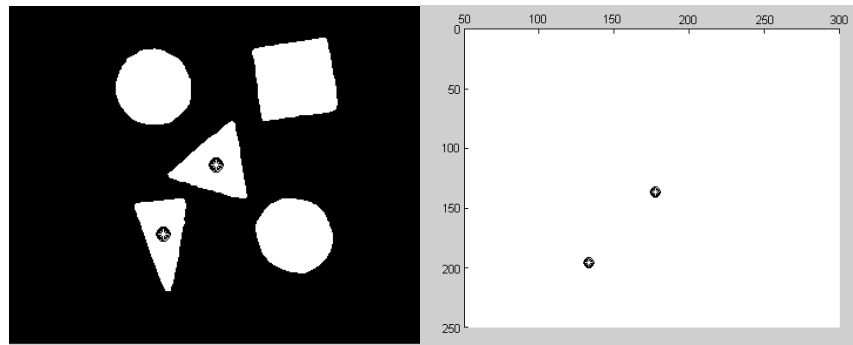


Figura 6.10 Centroides de los Triángulos Detectados [*]

6.1.4 CONDICIONES GENERALES PARA LA TRANSFORMADA DE HOUGH

Se debe tener en consideración al momento de la extracción de TH, una adecuada selección del espacio para cada celda acumuladora ya que en casos de que las celdas sean muy pequeñas, los picos dentro de la transformada se superponen lo cual hace que para la aplicación propuesta se generen errores en la etapa de extracción de características.

En la Figura 6.11 se puede visualizar la TH de un triángulo, en donde para cada uno de los picos que representan cada uno de los lados del triángulo se visualizan una concentración de valores probables dentro las celdas acumuladoras. Este problema se soluciona haciendo que el tamaño del elemento estructurante de los parámetros de vecindad alrededor de cada pico sea de mayor tamaño, en la figura se muestra una cantidad de celdas acumuladoras igual al tamaño de la imagen, dividido por 50.

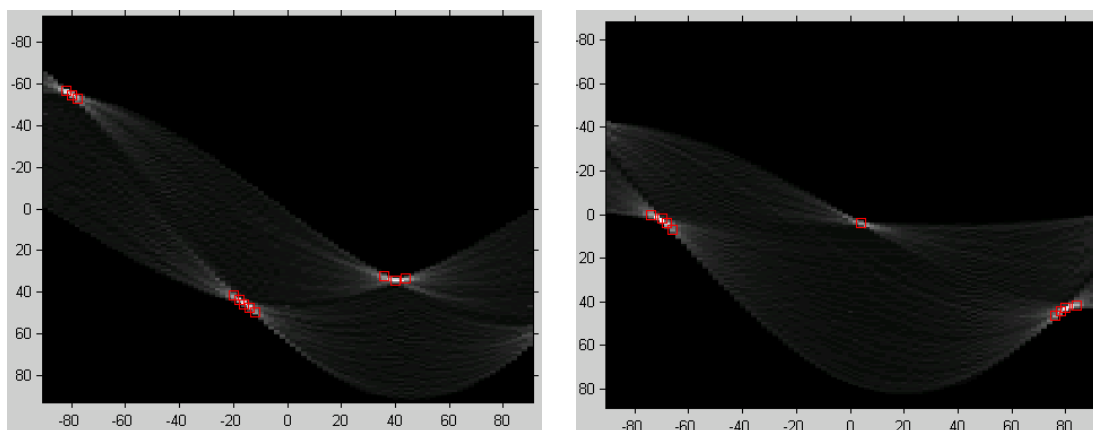


Figura 6.11 TH de un Triángulo con Superposición de Picos [*]

* Imagen creada por los autores.

Mediante el uso de la Transformada de Hough, se puede detectar la cantidad de líneas dentro de una imagen por medio del número de picos, representados por la intersección de sinusoidales dentro de la gráfica de la transformada, en la Figura 6.12 se muestran cuatro objetos dentro de la imagen, esta imagen se somete a la detección de cuadrados, donde se detectan tres de los mismos y una figura de ocho lados que no corresponde a ninguna de las opciones de usuario dentro de la pantalla de selección.

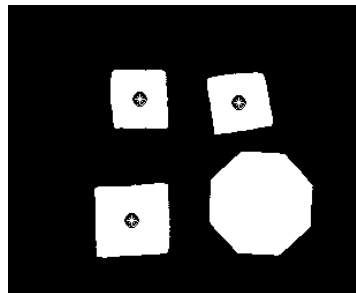


Figura 6.12 Imagen Procesada [8]

La Figura 6.13 muestra las variables enviadas a la etapa de posicionamiento del brazo robótico: número de objetos detectados, coordenadas de los centroides y el número de cuadrados detectados.

```

Command Window
>> eliminacion_ruido
>> morfologia2
>> paso2

numero_objetos =
    4
centro_cuadrado =
    108.4063  199.8869

centro_cuadrado =
    114.5901  95.0016

centro_cuadrado =
    201.3956  97.9581

cuadrados_total =
    3
  
```

Figura 6.13 Variables enviadas, correspondientes a la Etapa de Posicionamiento “Cuadrados”

La Figura 6.14 muestra la TH del objeto restante en donde se pueden visualizar ocho picos correspondientes al número de lados del objeto, en este proyecto constituye como alcance la detección de triángulos, cuadrados y círculos, pero el uso de la TH no solo se limita a este tipo de figuras, sino que también se puede realizar una ampliación al proyecto y detectar figuras regulares de mayor cantidad de lados.

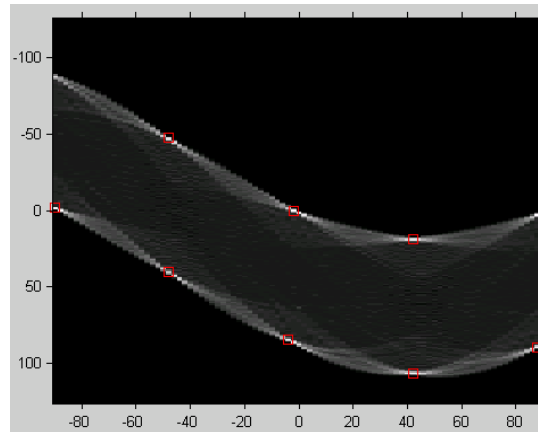


Figura 6.14 TH de Figura de Ocho Lados

Poniendo en funcionamiento el sistema completo, se verificó que se obtiene una imagen clara y completa de la plataforma. Tomando en cuenta la lectura que el sistema de procesamiento de imagen hace de las coordenadas X, Y de la figura identificada y, realizada la conversión de pixeles a centímetros, el software de control procesa esta información y muestra en la pantalla de la HMI los ángulos que deberían tener las respectivas articulaciones del brazo robótico.

6.2 CALIBRACIÓN Y PRUEBAS DEL SISTEMA DE POSICIONAMIENTO DEL BRAZO ROBÓTICO

Una vez que se obtienen los ángulos para cada una de las articulaciones, el software transforma esta información a parámetros de 0 a 254 para la señal PWM que se envía a cada uno de los servomotores a través del puerto serial.

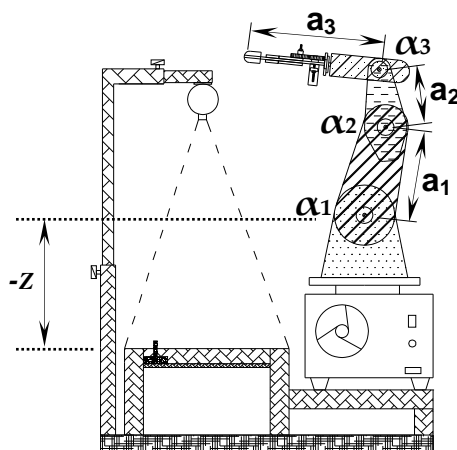


Figura 6.15 Parámetros para la generación de los ángulos de posicionamiento [*]

Para comprobar si estos ángulos permiten la correcta colocación de las articulaciones del brazo robótico hasta posicionar el elemento terminal sobre la pieza identificada y, conociendo las dimensiones tanto de la estructura del robot como de las plataformas, se procedió primero a la verificación de las ecuaciones cinemáticas inversas obtenidas en la sección 5.1.4, para los ángulos del brazo robótico; para la comprobación se creó una tabla en *MS EXCEL* que permita visualizar el resultado para un par de coordenadas predeterminadas y conocidas para realizar la calibración.

L1	L2	L3	L1+L2+L3
12	7,7	18	37,7
x	y	z	
14,01	0	-17,6	

CORRECCIÓN GEOMÉTRICA		CORRECCIÓN GEOMÉTRICA SI X ≤ 20,65	
α1=	-64,954 grados	α1=	38,00 °
α2=	172,181 grados	α2=	-116,27 °
α3=	-158,706 grados	α3=	158,71 °
α4=	0,000 grados	α4=	0,00 °

SEN A2	COS A2	SEN A2(-1)	A2	A2(-1)	PRIMER CUADRANTE		CUARTO CUADRANTE	
0,136040616	-0,990703261	-0,136040616	-0,136463768	0,136463768	A2	A2(-1)	A2	A2(-1)
L2 * SEN A2(-1)	L1 + L2 * COS A2	ATAN(M1)	ATAN(M2)	A1	A1(-1)	A1	A1(-1)	
-1,04751274	4,371584892	0,235184254	-0,235184254	-1,133669515	-0,663301006	-64,95447855	-38,00434819	
A3(-1)	A3	A3(-1)	A3	A3(-1)	A3	A3(-1)		
-0,371648022	21,29386312	-21,29386312	-2,769944632	2,769944632	-158,7061369	158,7061369		
X*(L2 * SEN A2)	X*(L2 * SEN A2(-1))	A/B	A/B	A1	A1(-1)	A1	A1(-1)	
2,932696029	-2,932696029	-0,905971731	-0,615721276	-1,133669515	-0,663301006	-64,95447855	-38,00434819	

Figura 6.16 Tabla en *MS EXCEL* utilizada para la comprobación del algoritmo matemático [*]

* Imagen creada por los autores.

Se seleccionaron inicialmente cinco pares de coordenadas ubicadas en las esquinas y en el centro de la plataforma denominada "A", que contiene inicialmente las figuras a ser identificadas y localizadas, tomando además como referencia las dimensiones de una figura patrón colocada en cada uno de estos puntos. Los pares de coordenadas correspondientes a la figura patrón (cuadrado de 1,6 x 1,6 cm) se etiquetaron como 1, 2, 3, 4 y 5 como se ve en la Figura 6.17.

Siendo conocidas las distancias de los puntos centroides de la pieza patrón para calibración en los ejes X, Y con el valor del eje Z como una constante de la planta construida, se ingresan los datos en la tabla y se obtienen los respectivos ángulos de posicionamiento. Complementariamente se utilizó como herramienta de simulación gráfica el *AutoCAD* para colocar con precisión los datos obtenidos.

En el plano X-Y, los datos para el ángulo α_4 que eventualmente controla la posición de la Base del brazo robótico, coinciden perfectamente en la simulación gráfica para todos los pares de coordenadas previstos en la calibración, sin embargo en la calibración realizada con la estructura real, no siempre coinciden los ángulos calculados con la posición física final obtenida.

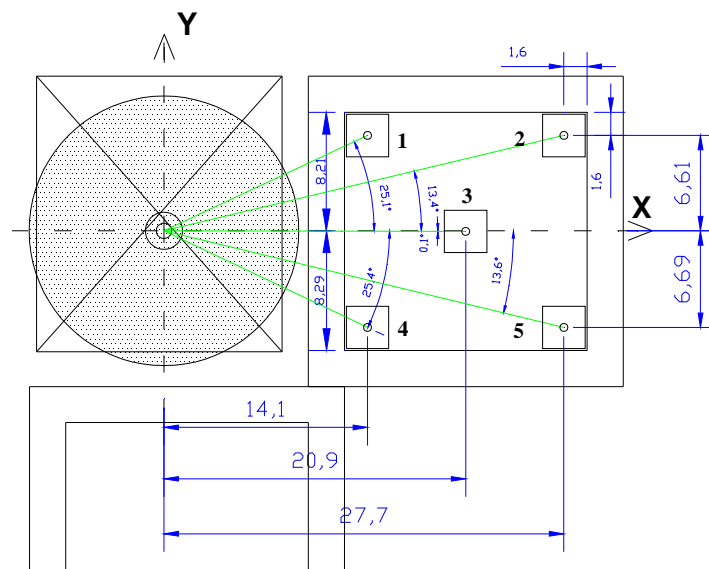


Figura 6.17 Simulación gráfica en *AUTOCAD* para α_4

Estos errores de posición se deben principalmente a que los servomotores se posicionan con una señal PWM en forma discreta, es decir de 0 a 254 en pasos de 1 en 1. De acuerdo a los datos del fabricante, cada paso de 1 en la señal implica una variación angular de 0,72 grados, no obstante se pudo comprobar en

la práctica que este dato no es fijo y varía (experimentalmente) entre 0,70 y 0,80 grados aproximadamente, dependiendo del servomotor y de su posición, la cual influye dados los torques que deben manejar dependiendo de su función.

Para tener control sobre este factor, en la calibración se introdujo una variable local en el software, asumiendo además que la variación del movimiento angular de los servos es lineal, de esta forma se estableció una regla de tres simple para convertir el valor del ángulo respectivo en un parámetro de control PWM, lo cual permitió manipular en tiempo real los movimientos del brazo robótico hasta que, mediante algunas pruebas, se consiguió una precisión aceptable en primera instancia.

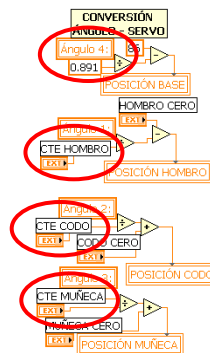


Figura 6.18 Variables para calibración de posición en servomotores

Otro factor influyente en el error de posición final es el resultado de la operación de procesamiento de imagen que entrega el programa *MatLab*, ya que como se puede apreciar en la siguiente tabla, para las mismas localizaciones de prueba con la figura patrón, se registran veinte corridas en las cuales se apuntan los resultados de posición X-Y (en pixeles) respecto a las cinco posiciones preestablecida para pruebas, apreciándose claramente que los mismos no se repiten, restando precisión al sistema integral.

	1		2		3		4		5	
	X	Y	X	Y	X	Y	X	Y	X	Y
1	41,4212	283,426	235,611	282,736	143,469	185,382	42,8716	92,5806	237,682	90,578
2	39,5611	284,061	237,602	283,136	140,439	187,486	41,7007	91,4257	238,892	89,0529
3	41,7196	283,833	235,752	282,252	136,478	186,034	42,3295	92,6809	237,861	90,5371
4	39,9195	284,323	237,235	283,095	140,135	186,890	40,7834	91,2264	239,008	89,572
5	40,2296	282,519	236,001	281,569	139,667	187,947	41,5368	92,4300	237,569	90,750
6	40,5951	284,138	238,077	283,489	141,906	186,086	41,4667	90,7675	238,779	88,8402
7	40,5538	282,368	235,986	284,447	141,698	186,035	42,2066	92,7631	237,757	91,1731
8	39,9786	284,852	237,523	283,646	139,955	188,214	40,9164	91,0165	238,829	88,9888
9	42,1370	283,624	235,738	282,768	137,733	187,796	42,4829	92,5179	236,441	90,7594

10	40,0424	284,728	237,585	283,221	139,906	185,608	40,7093	91,2376	238,926	89,9907
11	40,7361	282,710	236,089	281,551	138,521	184,892	42,3216	93,0899	237,323	91,3705
12	43,0860	283,899	236,293	281,790	129,836	192,145	42,8126	91,9506	236,948	91,5863
13	40,7249	284,611	237,055	283,516	129,970	190,895	41,2564	90,7291	239,655	88,9407
14	41,2704	283,086	235,964	281,800	135,935	185,454	42,5781	90,5402	239,146	90,49
15	40,5709	285,597	237,382	283,521	130,217	184,783	40,5018	89,5547	240,098	88,847
16	42,1276	283,458	235,917	282,113	133,758	184,936	42,6035	90,7169	239,443	90,260
17	40,2850	284,899	237,085	284,620	134,661	186,877	41,4788	89,0852	239,207	89,5153
18	43,5637	282,523	235,569	282,023	135,528	180,781	43,1846	89,5379	238,270	89,3281
19	40,7808	284,875	236,317	283,888	135,471	184,614	41,5156	89,3472	239,419	88,7154
20	42,3387	283,352	235,392	282,319	137,448	181,039	42,8871	90,5482	238,506	89,0556

Tabla 6.1 Resultados de posición generados por *MatLab* para diferentes localizaciones de la figura patrón.

Para contar con referencias de posición fijas para calibrar el sistema, se obtuvo un valor promedio de los anteriores datos:

1		2		3		4		5	
X	Y	X	Y	X	Y	X	Y	X	Y
41,0821	283,844	236,508	282,875	137,136	186,194	41,9072	91,1873	238,488	89,9176

Tabla 6.2 Valores de posición promedio (píxeles) para calibración.

Adicionalmente es muy importante resaltar el hecho de que una vez que el algoritmo para obtener los ángulos de posición del brazo entrega su resultado, éste se somete a una transformación lineal para conseguir valores discretos de 0 a 254 para el control de los servomotores, esto impide que se pueda colocar la pinza del brazo robótico con toda la precisión matemática resultante, ya que el software acerca al valor entero inmediato superior o inferior todo dato fraccionario, por lo que esta circunstancia absolutamente necesaria contribuye a la acumulación del error de posición del efector final.

Con los datos referenciales se realizaron pruebas del sistema completo colocando la pieza patrón en los puntos de calibración, comprobando que la pinza del brazo robótico alcanzaba el objetivo con una precisión aceptable sólo en los puntos 1, 2 y 3, pero en los puntos 4 y 5 se manifestaba un error de posición significativo que hacía fallar al robot en todas las posiciones aledañas a los puntos mencionados. Este error no se pudo corregir aún mediante el cambio en los valores de las variables de calibración de los servomotores, por lo que se optó por utilizar un método complementario de calibración del sistema.

Mediante la observación del comportamiento del sistema integral a lo largo de un número indeterminado de pruebas, se llegó a la conclusión de que para unos valores determinados de las variables para la calibración de los servomotores, el brazo robótico se posicionaba correctamente sólo en cierto sector y únicamente cambiando los valores de dichas variables el robot acertaba en un sector diferente. Con la experimentación se identificaron en la plataforma seis zonas donde los valores recurrentes de las variables permitieron un posicionamiento correcto del robot, por lo que se procedió a dividir el espacio físico en seis sectores y en cada uno de ellos se realizaron las pruebas correspondientes, similares al primer procedimiento.



Figura 6.19 División del espacio físico de la Plataforma “A” para pruebas ^[*]

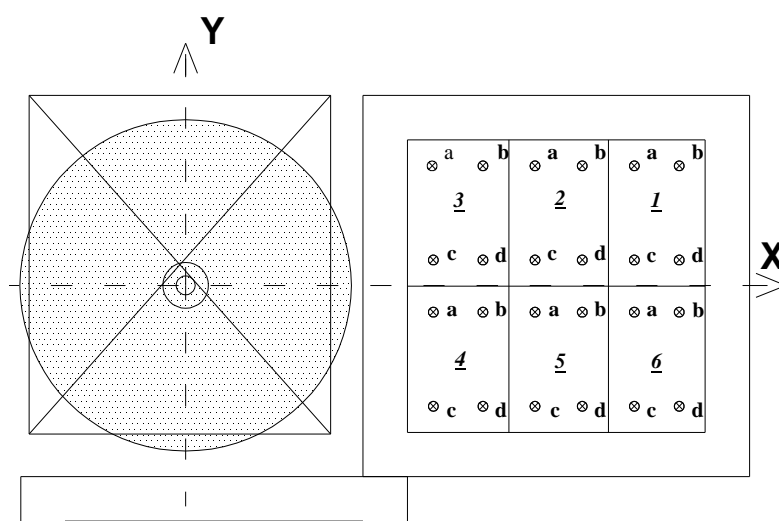


Figura 6.20 Ubicación de los puntos de prueba en la Plataforma “A” por sectores ^[*]

* Imagen creada por los autores.

Para lograr la sectorización mediante software, se tomaron las distancias limítrofes de los sectores y se implementaron lazos condicionantes dependiendo de dónde se haya localizado la figura a ser trasladada por el brazo robótico y para cada caso se fueron calibrando las variables de control de los servomotores hasta conseguir el correcto posicionamiento del brazo robótico en cualquier punto de la Plataforma "A".

	a		b		c		d	
	X	Y	X	Y	X	Y	X	Y
1	24,00	7,23	27,50	7,21	24,00	1,10	27,6	1,10
2	18,65	7,20	23,50	7,20	18,63	1,00	23,6	1,00
3	14,30	7,24	18,12	7,20	14,36	0,90	18,63	1,06
4	14,36	0,60	18,10	0,58	14,40	-5,80	18,24	-5,80
5	18,63	0,55	23,50	0,63	18,70	-5,80	23,60	-5,81
6	24,00	0,74	27,60	0,70	24,03	-5,81	27,70	-584

Tabla 6.3 Valores de posición promedio (centímetros) obtenidos en los sectores.

Con los datos obtenidos, se realizaron las pruebas con el control en modo manual, logrando calibrar las variables de los servomotores a valores que permiten un correcto posicionamiento del brazo robótico.

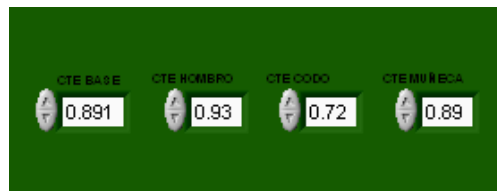


Figura 6.21 Valores de variables para servomotores

Calibradas las variables, el sistema se configura en modo automático (y semi-automático), obteniendo un funcionamiento adecuado del sistema y consiguiendo la totalidad de los objetivos del presente proyecto con resultados satisfactorios.

CAPÍTULO 7

CONCLUSIONES Y RECOMENDACIONES

7.1 CONCLUSIONES

- Se ha desarrollado satisfactoriamente una aplicación que permite distinguir dentro de un grupo de figuras localizadas en una plataforma A, que corresponden a figuras geométricas regulares tales como círculos cuadrados y triángulos, para luego transportarlas de una manera ordenada hacia una plataforma B, con la ayuda de un adecuado posicionamiento de las articulaciones del brazo robótico.
- Se ha logrado vincular los resultados del proceso de adquisición de imagen (centroides de las figuras identificadas), con el sistema de posicionamiento del brazo robótico de acuerdo al objetivo propuesto. La aplicación enlaza dos sistemas de desarrollo computacional avanzados como son el MatLab y el LabView los cuales, durante la ejecución del software, no muestran incompatibilidad ni retardos en el proceso computacional del equipo utilizado (Laptop GATEWAY MA3; 1.8 GHz, RAM 1Gb, S.O. Windows XP).
- Este proyecto se desarrolló dentro de una concepción de “prototipo” y se muestra un perfil tecnológico multidisciplinario en el desarrollo de un sistema de visión artificial y su vinculación con la robótica, cuya portabilidad permitió realizar pruebas bajo diferentes condiciones. La portabilidad de este prototipo también permitirá profundizar la enseñanza de adquisición y procesamiento de imagen aplicada a procesos en donde sea necesario realizar una recopilación de información proveniente del mundo real.
- Inicialmente, este proyecto se concibió para procesar imágenes mediante técnicas de inteligencia computacional (redes neuronales, lógica difusa, etc); sin embargo al realizar las primeras investigaciones se determinó que para este tipo de aplicaciones, específicamente para imágenes dinámicas (en continuo cambio), estas técnicas no son eficientes por cuanto toma mucho tiempo de

cómputo el procesamiento de dichas imágenes y, en el caso de las redes neuronales de aprendizaje, se requiere además un amplio número de casos de entrenamiento para que se puedan cubrir todos los puntos de localización posibles. El proyecto se orientó posteriormente al desarrollo de los algoritmos computacionales referentes al proceso de selección mediante técnicas de visión artificial que involucran las etapas de adquisición, procesamiento y reconocimiento de imágenes, estas técnicas demostraron desde el inicio ser mucho más eficaces que otras para los propósitos establecidos en el presente proyecto, en cuanto al tiempo de desarrollo del software y su complejidad.

- El sistema de visión implementado es capaz de incorporar nuevas tareas o adaptarse a cambios utilizando los medios existentes (por ejemplo distinción imágenes por colores), sin requerir demasiado trabajo por parte del programador. El presente trabajo permitiría encontrar figuras de mayor número de lados, añadiendo lazos en el cálculo del número de picos en la Transformada de Hough.
- Las cámaras web son herramientas de entrada para la adquisición de imagen y actualmente sus costos son relativamente económicos. Por lo tanto la introducción de un sistema de visión artificial, en aplicaciones basadas en computadoras es un objetivo factible de implementar. La interfaz desarrollada usa la visión artificial como canal de comunicación desde el usuario a la computadora. Se debe recordar que ninguna imagen proveniente de una cámara digital puede ser utilizada directamente, ésta debe ser sometida a un proceso de mejoramiento de calidad para su interpretación y extracción de características.
- En un sistema de visión artificial no existe un modelo óptimo o general para todas las aplicaciones. Cada aplicación deberá determinar las técnicas adecuadas para una extracción eficiente de los atributos requeridos dentro de la imagen.
- En relación a eliminación de brillos o reflexiones se ha podido comprobar cómo una apertura por reconstrucción, favorece la atenuación progresiva del ruido

provocado por fuentes luminosas. En ese sentido una erosión precede a una dilatación. Este proceso puede llegar a simplificar demasiado la imagen y así eliminar atributos requeridos para su clasificación (cavidades, irregularidades, texturas, etc).

- Un algoritmo adecuado para la detección de líneas y curvas basado en la transformada de Hough fue propuesto en el software de este proyecto. Como se puede ver, la expresión de superficie-plano ρ - θ de la imagen no solamente es una técnica eficaz para la detección de líneas rectas, también puede ser generalizado a curvas de mayor grado.
- Con los resultados que se obtuvieron en la aplicación de la transformada de Hough, se puede concluir que su efectividad es bastante buena, sin embargo, el tiempo de procesamiento está en función del número de puntos que tengan los bordes, por ello el contraste y una buena detección de bordes en la imagen son deseables para aplicar la transformada de Hough.
- El tiempo que le toma al método encontrar cada una de las figuras en la secuencia de imágenes es muy corto y depende de la velocidad de proceso del computador utilizado. Por eso se ve que su aplicación inmediata podría darse en la industria, para el control de robots seleccionadores o aparatos semejantes. Cabe anotar que en el presente proyecto se introdujeron retardos de 3 segundos en cada una de las etapas del procesamiento y su correspondiente visualización gráfica, para permitir apreciar el proceso completo propiamente dicho, por lo que se podría interpretar erróneamente que el sistema es lento.
- Se ha evaluado el funcionamiento del sistema con diferentes posiciones de las figuras ubicadas en la plataforma, éste responde satisfactoriamente en la mayoría de casos, sin embargo se manifiestan errores de posicionamiento debido a factores mecánicos y de software. Los principales factores que inciden en el error de posicionamiento del elemento terminal del brazo robótico sobre el centro de la pieza geométrica son: su estructura mecánica, la precisión de los servomotores implementados y la precisión del cómputo iterativo del software

de procesamiento de imágenes para localizar las coordenadas X-Y del centro de la figura.

- La estructura mecánica no se implementó con toda la precisión milimétrica deseable debido a que el prototipo fue construido con maquinarias y elementos artesanales, lo que le resta exactitud respecto a los planos y a las dimensiones diseñadas, las mismas que son consideradas en el análisis matemático para la obtención de las ecuaciones cinemáticas inversas, que consecuentemente se utilizan para el posicionamiento de las articulaciones del robot. No obstante, los resultados finales en cuanto a precisión y repetitividad fueron bastante aceptables.
- Tanto la tarjeta de control de servos como los servomotores seleccionados se utilizan en muchos modelos comerciales de robots didácticos, sin embargo la precisión con la que cuentan estos elementos no resultó ser muy confiable debido a que, según los datos del fabricante (ver Anexo 2), éstos poseen una capacidad de giro de 180 grados en el eje del servomotor y se posicionan en pasos de 0,72 grados con su respectiva señal PWM, pero en las pruebas realizadas, se comprobó que el rango de giro llegó desde unos 180 grados hasta más de 200 grados en algunos servos y, por lo tanto, se obtuvieron pasos desde 0,72 hasta 0,93 grados para su posicionamiento, lo que complicó la tarea de calibración.
- El hecho de que se debe controlar la posición de los servos con una señal PWM de 0 a 254 y por ende, en pasos discretos de 0,72 (ó 0,93) grados es el mayor limitante de precisión del sistema, ya que el algoritmo matemático que genera los ángulos de posición del brazo robótico es bastante preciso, pero al enviar este dato a los servomotores, se debe convertir el valor angular en un número entero equivalente PWM (0-254), por lo que se pierde exactitud.
- Otro factor influyente en el error de posicionamiento se presenta cuando se mantiene operando el robot en funcionamiento continuo por largos períodos de tiempo, debido al sobrecalentamiento de los servomotores, los que no están

diseñados específicamente para este tipo de tareas en funcionamiento continuo.

7.2 RECOMENDACIONES

Si se requiere un sistema de alta precisión, los algoritmos computacionales implementados en el presente proyecto, tanto el de procesamiento de imágenes como el de control del robot, demuestran ser lo suficientemente eficientes, por lo que se recomienda su utilización en un brazo robótico industrial comercial o en un proyecto construido bajo parámetros de diseño mecánico y utilizando motores más potentes con sistemas de retroalimentación de posición, velocidad, etc., y se podría ampliar este proyecto a un análisis mediante ecuaciones dinámicas para conseguir un control más exacto de otras variables físicas.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Editor de Imágenes basado en Regiones. Aplicación en entorno MatLab. Autor: Samira Hervella Azouzi, I.T.Telecomunicaciones,esp, 2006, pp. 7-18. <https://upcommons.upc.edu/pfc/bitstream/2099.1/3860/1/54960-1.pdf>
- [2] **GONZÁLEZ**, Rafael C.; **WOODS**, Richard E.; Digital Image Processing, 2nd Edition. Prentice Hall. Upper Saddle River, NJ. 2002, pp 17-25.
- [3] **PRATT**, William K.; Digital Image Processing, 3th Edition. PixelSoft Inc. Los Altos, California. 2001, pp 3-32.
- [4] Aspectos de un Sistema de Visión Artificial Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial, 2005, pp. 5-6 http://www.aitex.es/estudios/Vision_Artificial1.pdf
- [5] Filtrado en el Dominio de la Frecuencia, Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial, 2005, pp 1-13. <http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Filtrado%20en%20el%20Dominio%20de%20la%20Frecuencia.pdf>
- [6] Filtrado Espacial Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial, 2005, pp. 1-14 <http://iaci.unq.edu.ar/Materias/vision/archivos/apuntes/Filtrado%20Espacial.pdf>
- [7] Operaciones Morfológicas en Imágenes Binarias, Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial, 2005, pp 1-3. <http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Operaciones%20Morfológicas%20en%20Imágenes%20Binarias%20-%20parte%201.pdf>
- [8] Métodos de Análisis de Imágenes, Extracción de Características Autores:José Luis Alba –Univ. de VigoJesús Cid – Univ. Carlos III de Madrid Inmaculada Mora – Univ. Rey Juan Carlos, 2006, pp 42-75 <http://www.gts.tsc.uvigo.es/pi/Analisis%20de%20imagenes.pdf> Páginas
- [9] Visión por Computador utilizando MatLab y Toolbox de Procesamiento Digital de Imágenes, Erik Valdemar Cuevas Jimenez, Daniel Zaldivar Navarro. 2006, pp. 22-25 <http://proton.ucting.udg.mx/tutorial/vision/cursovision.pdf>
- [10] Image Processing Toolbox For Use with MATLAB User's Guide Version 2.1, pp. 107-259 <http://www.mathworks.com/image/toolbox.pdf>
- [11] Image Processing Toolbox For Use with MATLAB User's Guide Version 2.1, Getting Started pp. 5-25

<http://www.mathworks.com/image/processingtoolboxstarted.pdf>

- [12] **DOMINGO**, Juan; Robótica, Apuntes para la Asignatura, Universidad de Valencia, España, 2001, pp.3.
- [13] **SANZ VALERO**, Pedro J., Introducción a la Robótica Inteligente, pp.12-16.
<http://icc.uji.es/asignatura/obtener.php?letra=l&codigo=S67&fichero=1107949061IS67-robóticaintro.pdf>
- [14] **TORRES**, F.; **POMARES**, J.; **GIL**, P.; **PUENTE**, S.T.; **ARACIL**, R, Robots y Sistemas Sensoriales, PEARSON EDUCACIÓN S.A., Madrid, 2002, pp.12-14, pp.31-37.
- [15] Robots-Argentina, Servos-Características Básicas.
http://robots-argentina.com.ar/Prueba_ServoRC01.htm
- [16] Scott Edwards Electronics, Inc., Mini SSC II Serial Servo Controller, User's Manual.
- [17] **IÑIGO**, Rafael; **VIDAL**, Enric; Robots Industriales Manipuladores, ALFAOMEGA GRUPO EDITOR, S.A., México, D.F., 2004, pp.35-56.
- [18] **DOMINGO**, Juan; Robótica, Universidad de Valencia, España, 2001, pp.10-24.
- [19] **CRAIG**, John J., ROBÓTICA, Tercera Edición, PEARSON EDUCACIÓN S.A., México, 2006, pp. 100-121.

Bibliografía adicional

RUSS, John C.; The Image Processing Handbook, 3th Edition. CRC Press, Florida. 1999.

HOUGH, P.V.C.; Methods and Means for Recognizing Complex Patterns. Patent 3,069,654. U.S. 1962.

CALVO, Raúl Alonso; Temas Avanzados en Tratamiento y Análisis de Imágenes y en Morfología Matemática, 2003 / 2004.

LAJARA, José R.; **PELEGRÍ**, José, LabVIEW. Entorno Gráfico de Programación, ALFAOMEGA GRUPO EDITOR, S.A., México, D.F., 2007.

Image Processing Toolbox, Users Guide, Version 6, The MathWorks Inc. Natick, MA. 2005.

LabVIEW, User Manual, April 2003 Edition, National Instruments Co. Austin, Texas. 2003.

Artículos Internet

Editor de Imágenes basado en Regiones. Aplicación en entorno MatLab. Autor: Samira Hervella Azouzi, I.T.Telecomunicaciones,esp.-Terrassa 2006.

<https://upcommons.upc.edu/pfc/bitstream/2099.1/3860/1/54960-1.pdf>

Aspectos de un Sistema de Visión Artificial Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial Cátedra: Visión Artificial, 2005.

http://www.aitex.es/estudios/Vision_Artificial1.pdf

Visión por Computador utilizando MatLab Y el Toolbox de Procesamiento Digital de Imágenes, Erik Valdemar Cuevas Jimenez, Daniel Zaldivar Navarro. 2006

<http://proton.ucting.udg.mx/tutorial/vision/cursovision.pdf>

Filtrado en el Dominio de la Frecuencia, Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial Cátedra: Visión Artificial Octubre de 2005.

<http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Filtrado%20en%20el%20Dominio%20de%20la%20Frecuencia.pdf>

Filtrado Espacial, Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial Cátedra: Visión Artificial Septiembre de 2005.

<http://iaci.unq.edu.ar/Materias/vision/archivos/apuntes/Filtrado%20Espacial.pdf>

Operaciones Morfológicas en Imágenes Binarias, Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial, 2005

<http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Operaciones%20Morfológicas%20en%20Imágenes%20Binarias%20-%20parte%201.pdf>

Procesamiento Global empleando la Transformada de Hough Universidad Nacional de Quilmes – Ing. en Automatización y Control Industrial Cátedra: Visión Artificial Septiembre de 2005

<http://iaci.unq.edu.ar/materias/vision/archivos/apuntes/Transformada%20de%20Hough.pdf>

Comandos de MatLab utilizados para el procesamiento de imágenes. Autores: José Luis Alba -Universidad de Vigo, Jesús Cid -Universidad Carlos III de Madrid, Inmaculada Mora -Universidad Rey Juan Carlos, marzo de 2006

www.gts.tsc.uvigo.es/pi/Transformadas.pdf

Visión Artificial Extracción de Características F.Escolano, O.Colomina, M.A. Cazorla 2006-2007

<http://www.eduonline.ua.es/jornadas2007/comunicaciones/2C1.pdf?PHPSESSID=60eea1fc6e87876974bbf8a85b30bd76>

Procesamiento de imágenes para Ingeniería de Alimentos, Domingo Mery

www.ing.puc.cl/~dmery/iman.html

Procesamiento de Imágenes

<http://www.itc.nl/external/unesco-rapca/Casos%20de%20estudios%20SIG/02%20Sensores%20remotos/sensores%20remotos.pdf>

Visión por computadora y procesamiento de imágenes con FPGAs

<http://ccc.inaoep.mx/fpgacentral/reconfig/2003/pdf/Marias-apl.pdf>

IEEE Xplore: Signal Processing Magazine, IEEE
<http://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=79>

Tratamiento Digital De La Imagen
http://campusvirtual.uma.es/tdi/www_netscape/TEMAS/Tdi_30/

Captura y Procesamiento Digital de Señales e Imágenes
http://cpdsi-fich.wdfiles.com/local--files/practicas/practica_02b.pdf

IEEE Signal Processing Magazine
<http://www.ieee.org/organizations/pubs/magazines/sp.htm>

Curso de Vision Computacional
<http://ccc.inaoep.mx/~esucar/vision.html>

Image Processing Toolbox Release Notes, pp 30-76
<http://www.mathworks.com/image/processingtoolbox/notes.pdf>

Fundamentos de Procesamiento de Imágenes
<http://fcqi.tij.uabc.mx/docentes/esqueda/cursoimagenes.pdf>

Cinemática Directa
http://wiphala.net/courses/090666/2006-II/class/class_b1_direct_kinematic.ppt-

Forward Kinematics: The Denavit-Hartenberg convention
www.cs.dartmouth.edu/~donaldclass/Bio/current/Papers/chap3-forward-kinematics.pdf

Datasheets de servomotores
<http://www.hitecrd.com/servos/show?name=HS-81.pdf>
<http://www.hitecrd.com/servos/show?name=HS-311.pdf>
<http://www.hitecrd.com/servos/show?name=HS-755HB.pdf>

Presentación de powerpoint, Robotica 1 y 2
http://webcourse.cs.technion.ac.il/236927/Winter2007-2008/en/ho_Lectures.html

Parametros Denavit-Hartenberg
http://maqlab.uc3m.es/PROYECTOS/SIDEMAR/para_dh.pdf

Dpt. Teoría de la Señal, Telemática y Comunicaciones, Introducción a la Robótica
<http://ceres.ugr.es/~angel/docencia/robin/Apuntes/Tema%201.pdf>
<http://ceres.ugr.es/~angel/docencia/robin/Apuntes/Tema%202.pdf>

5 Servomotor Robotic Arm Kit
<http://scientificsonline.com/product.asp?pn=3082117>

JAMECO Robotstore
<http://www.robotstore.com>

ANEXO 1

PROGRAMA EN MATLAB: CAPTURA Y
PROCESAMIENTO DE IMÁGENES

PROGRAMA EN MATLAB: CAPTURA Y PROCESAMIENTO DE IMÁGENES

ADQUISICIÓN DE IMAGEN

Subrutina que permite adquirir la imagen proveniente de la cámara web, MatLab despliega una pantalla de video, luego captura la imagen para ser procesada. Se recomienda dejar una pausa después de desplegar el *preview* de video con el objetivo de que el lente de la cámara pueda abrirse completamente. La pantalla de video es cerrada después de adquirir y guardar la imagen:

```

adquirir_imagen_camara =          % Se crea una entrada de video winvideo
videoinput('winvideo',1);         corresponde al controlador de la cámara.

preview(adquirir_imagen_camara);  % Se despliega la pantalla de video con la
                                  secuencia de tramas capturada por la cámara.
pause(3)                          % Se propone una pausa de 3 seg. para una
                                  adecuada apertura den lente de la cámara.
imagentesis = getsnapshot(vidobj); % Se captura dentro de la secuencia de video
                                  la imagen que luego será procesada.
imwrite(imagentesis,'imgcamara.png','png'); % Se guarda la imagen capturada.

imageRes = vidobj.VideoResolution;

imageWidth = imageRes(1);

imageHeight = imageRes(2);

imagentesis = imread('imgcamara.png'); % Se lee la imagen previamente guardada.

delete(adquirir_imagen_camara)

clear adquirir_imagen_camara      % Se cierra la pantalla de secuencia de video.

imshow(imagentesis)

title(['Imagen Adquirida por la Camara ']); % Muestra la imagen para ser procesada.

```

PREPROCESAMIENTO DE IMAGEN

Subrutina de pre-procesamiento, en la cual se cambia la imagen capturada por la cámara y se transforma el campo de acción, primero a una imagen en escala de grises, y luego a una imagen binaria. La imagen binaria se guarda para su posterior utilización:

```

veces=1; %Inicialización de Variables

cuadrados=0;

triangulos=0;

circulos=0;

I = rgb2gray(imagentesis); % Se convierte la imagen RGB en una imagen
                           % en escala de grises
imagbinaria = graythresh(I); % Se calcula el valor de umbral más adecuado
                             % para binarización
bw = im2bw(I,imagbinaria); % Se binariza la imagen en escala de grises

imwrite(bw,'binaria.png','png'); % Se guarda la imagen binarizada

imshow (bw); % Se muestra la imagen binaria obtenida

title(['Imagen Binarizada']);

```

MORFOLOGÍA MATEMÁTICA

Subrutina de morfología matemática en donde se realizan operaciones tales como erosión, dilatación apertura y cierre. Una apertura seguida de un cierre elimina espacios vacíos dentro de la imagen, y una erosión seguida de una dilatación elimina objetos pequeños provenientes de la reflexión de la cámara, a este proceso se conoce como reconstrucción de imagen:

```

bw_obr = imreconstruct(imerode(bw,ones(1,91)),bw); = %Para eliminar las reflexiones y efectos de
                                                    % concavidad de la cámara se realiza una
                                                    % reconstrucción en varios pasos el primero es
                                                    % erosionar la imagen.

bw_o = imopen (bw,ones(1,91)); % Luego se realiza una apertura

bw_thr = imsubtract(bw,bw_o); % Se substraen las dos imagen con esto se
                              % eliminan las reflexiones horizontales

g_obr = imreconstruct(imerode(bw_thr,ones(1,91)),bw_thr); = % Se realiza una erosión para eliminar
                                                            % reflexiones horizontales
g_obrd = imdilate(g_obr,ones(1,11)); % se realiza una dilatación para mantener la
                                      % imagen y cerrar los bordes

f2 = imreconstruct(min(g_obrd,bw_thr),bw_thr); % Se realiza una reconstrucción utilizando
                                                % con la imagen final sin reflexión

bw = f2;

imshow(bw) % se muestra el resultado

title(['Imagen Preprocesada']);

imwrite(bw,'bw.png','png');

```



```

B=fliplr(theta);
theta=[theta - B];
ntheta=length(theta);
D=sqrt((M-1)^2+(N-1)^2);
q=ceil(D/drho);
nrho=2*q-1;
rho=linspace(-q*drho,q*drho,nrho);
[x,y,val]=find(f);
x=x-1; y=y-1;
h=zeros(nrho,length(theta));
for k=1:ceil(length(val)/1000);
first=(k-1)*1000+1;
last=min(first+999,length(x));
x_matrix= repmat(x(first:last),1,ntheta);
y_matrix= repmat(y(first:last),1,ntheta);
val_matrix= repmat(val(first:last),1,ntheta);
theta_matrix= repmat(theta,size(x_matrix,1),1)*pi/180;
rho_matrix=x_matrix.*cos(theta_matrix)+
y_matrix.*sin(theta_matrix);
slope=(nrho-1)/(rho(end)-rho(1));
rho_bin_index=round(slope*(rho_matrix-rho(1))+1);
theta_bin_index= repmat(1:ntheta,size(x_matrix,1),1);
h=h+full(sparse(rho_bin_index(:),theta_bin_index(:),
val_matrix(:),nrho,ntheta));
end

```

% Se realiza un cambio la matriz theta ahora su primer elemento es 0 hasta -90
% Se define los limites de theta desde -90 hasta 90
% Se encuentra la longitud de theta
% Calculamos la variación de Rho
% Se define el espacio de valores de rho
% Devuelve un vector columna con los valores tanto de x e y
% Espacio de transformado de rho y theta (celdas de acumulación)
% Redondeado de rho desplazando cada casillero
% Se crea una matriz con todos los valores de x
% Se crea una matriz con todos los valores de y
% Matriz auxiliar para transformación de coordenadas desplazamiento por los casilleros
% Transformación de coordenadas senos y cosenos
% Redondeado de RO desplazando cada casillero
% LLeno las grilla con los puntos
% Si esta dentro de la grilla escribo el valor en h

CÁLCULO DE LOS PICOS EN LA T.H.

Subrutina que encuentra los picos dentro de la TH, cada uno de los picos detectados significa una línea dentro de la región encontrada. Se eliminan elementos más cercanos a los picos, ya que un valor muy pequeño de las celdas de acumulación (valores más probables) ocasiona problemas en la detección:

```

threshold = 0.5*max(h(:));

hnew=h;

r=[];
c=[];

while~done
y=size(h);
nhood=size(h)/20;

[p,q]=find(hnew == max(hnew(:)));

if hnew(p,q)>=threshold
r(end+1)=p; c(end+1)=q;
p1=p-(nhood(1)-1)/2;
p2=p+(nhood(1)-1)/2;
q1=q-(nhood(2)-1)/2;
q2=q+(nhood(2)-1)/2;
[pp,qq]=ndgrid(p1:p2, q1:q2);
badro=find((pp<1) | (pp>size(h,1)));
pp(badro)=[]; qq(badro)=[];

theta_too_low=find(qq<1);
qq(theta_too_low)=size(h,2)+qq(theta_too_low);
pp(theta_too_low)=size(h,1)-
pp(theta_too_low)+1;
theta_too_high=find(qq>size(h,2));
qq(theta_too_high)=qq(theta_too_high)-
size(h,2);
pp(theta_too_high)=size(h,1)-
pp(theta_too_high)+1;
hnew(sub2ind(size(hnew),pp,qq))=0;

done=length(r)==numpicos;

else
done=true;
end
end

```

% Valor umbral para la detección de picos, corresponde el tamaño de elementos que no serán considerados como picos
% Es la matriz con la TH con los elementos vecinos suprimidos
% Es la fila en donde se detecta el pico
% Es la columna en donde se encuentra el pico

% Se encuentra el tamaño de la matriz
% nhood corresponde es un vector que contiene el tamaño de los elementos vecinos que serán suprimidos, los elementos vecinos a un pico son suprimidos para evitar falsos picos.
%Se encuentran los máximos de cada una de los senoidales de la TH
%Se suprimen los máximos y los elementos vecinos mas cercanos

%Se desecha las coordenadas de los elementos vecinos que se encuentran fuera in la dirección del eje rho
% Para coordenadas que se encuentran fuera del limite umbral en la dirección de theta, considerando la simetría de la TH, +/- 90.

% Convierte todos los valores que no sean picos en cero
% Se define en el calculo un valor de 50 picos se realiza este proceso hasta llegar al número de picos solicitados, si el número de picos encontrados es menor, se devuelve un vector con el número de picos encontrado.

ANEXO 2

HOJAS TÉCNICAS:

- TARJETA DE CONTROL MINI SCCII

- SERVOMOTORES :
 - *Hitec HS-755HB*
 - *Hitec HS-311*
 - *Hitec HS-81*

ANEXO 3

COSTO DEL SISTEMA

COSTO DEL SISTEMA

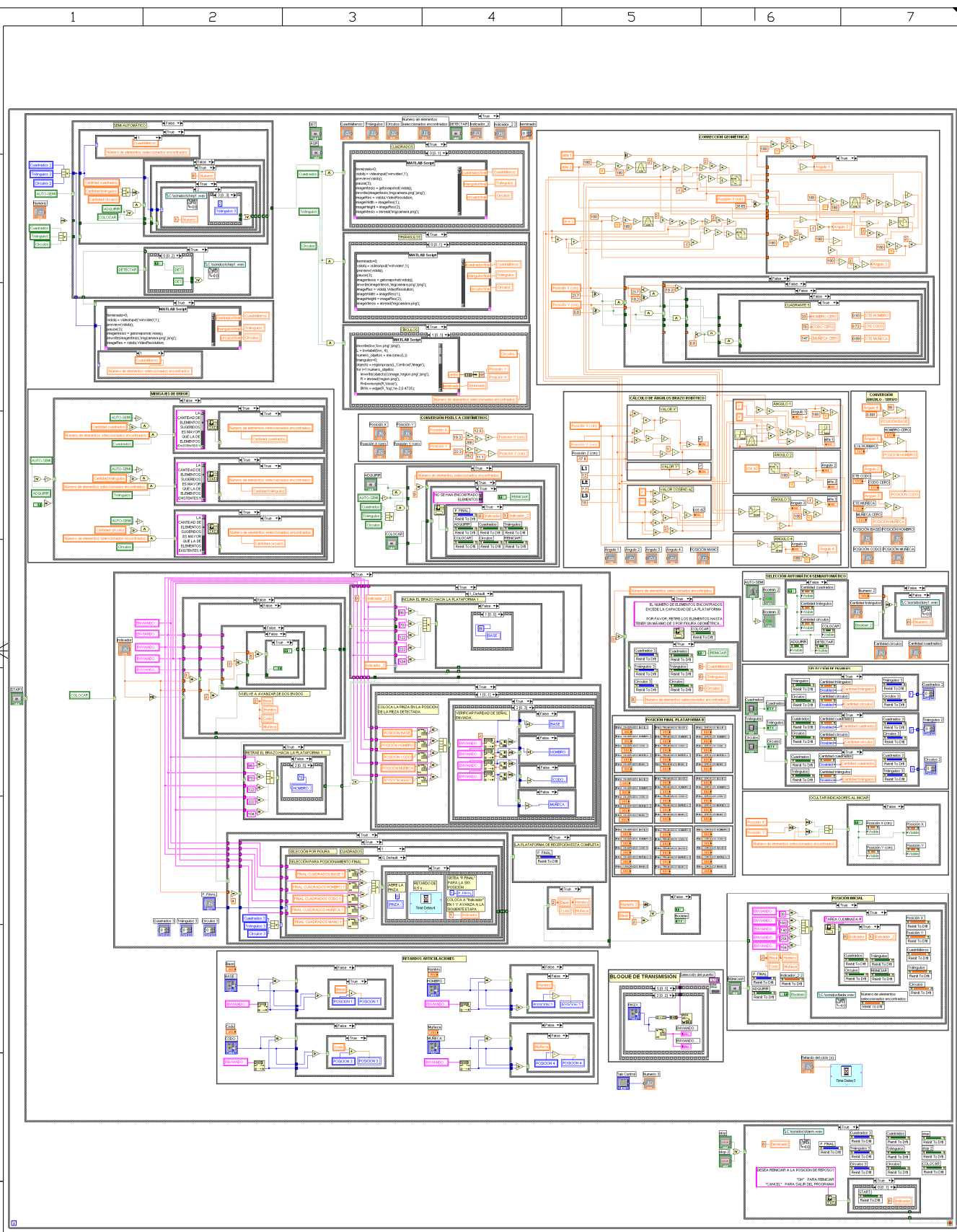
Se describe un detalle de los componentes utilizados en el desarrollo del proyecto, analizando los costos de los materiales, herramientas y equipos

DESARROLLO DE UN SISTEMA DE CONTROL PARA UN BRAZO ROBÓTICO MEDIANTE ADQUISICIÓN Y PROCESAMIENTO DE IMÁGENES				
	COMPONENTE	CANTIDAD	COSTO UNITARIO (\$USD)	COSTO TOTAL (\$USD)
1	Tarjeta Controladora Servomotores <i>Mini SSCII</i>	1	\$ 57.95	\$ 57.95
2	Gigant Robotic Ball Bearing Servo <i>HITECHS-755HB</i>	1	\$ 64.95	\$ 64.95
3	High Efficiency Robotic Servo <i>HITECHS-311</i>	5	\$ 25.95	\$ 129.75
4	Sub Micro Servo <i>HITECHS-81</i>	1	\$ 40.00	\$ 40.00
5	Plancha acrílico cobre esp. 2 mm, 1m x 1,50 m	1	\$ 25.00	\$ 25.00
6	Base circular acrílico transp. 10 mm	1	\$ 15.00	\$ 15.00
7	Juego pernos + tuercas espiga *	8	\$ 0.50	\$ 4.00
8	Juego pernos + tuercas milimétrico *	50	\$ 0.15	\$ 7.50
9	Rodamiento Base	1	\$ 35.00	\$ 35.00
10	Caja metálica Base	1	\$ 25.00	\$ 25.00
11	Transformador 120VAC- 12VAC/ 3A	1	\$ 15.00	\$ 15.00
12	Transformador 120VAC- 9VAC/ 1A	1	\$ 5.00	\$ 5.00
13	Circuito Fuente 6 VDC*	1	\$ 17.00	\$ 17.00
14	Circuito Fuente 9 VDC*	1	\$ 12.50	\$ 12.50
15	Ventilador + led	1	\$ 11.30	\$ 11.30
16	Planta metálica Plataformas A y B + soporte cámara	1	\$ 65.00	\$ 65.00
17	Cámara web Logítec Quick-Cam	1	\$ 28.00	\$ 28.00
18	Cableado misceláneo *	1	\$ 10.00	\$ 10.00
19	Uso herramientas *	1	\$ 70.00	\$ 70.00
20	Misceláneos (switches, pintura, pegamento, tape, etc.)*	1	\$ 25.00	\$ 25.00
21	Adaptador USB a Serial	1	\$ 37.50	\$ 37.50
22	Uso equipos (multímetro, pinza amperimétrica, etc.) *	1	\$ 20.00	\$ 20.00
			TOTAL:	\$ 720.45

* Costos aproximados

ANEXO 4

PROGRAMA EN LABVIEW: CONTROL POR
COMPUTADOR DE BRAZO ROBÓTICO



PROGRAMA EN LABVIEW 7.0

ELABORADO POR:
BASANTES, JUAN CARLOS
TORRES, FREDDY RICARDO

DESARROLLO DE UN SISTEMA DE CONTROL
PARA UN BRAZO ROBOTICO MEDIANTE ADQUISICION
Y PROCESAMIENTO DE IMAGENES