

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DISEÑO, CONSTRUCCIÓN Y CONTROL DE UN ROBOT INDUSTRIAL CON ARQUITECTURA ANTROPOMÓRFICA DOTADO CON VISIÓN ARTIFICIAL

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

ARÉVALO MOLINA PEDRO JOSÉ

pedro_qnk@yahoo.es

HERRERA ANDA SANTIAGO ANDRÉS

santiago_ha@hotmail.com

DIRECTOR: NELSON SOTOMAYOR, MSc.

n.sotomayor@hotmail.com

Quito, febrero 2010

DECLARACIÓN

Nosotros, Pedro José Arévalo Molina y Santiago Andrés Herrera Anda, 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.

Pedro José Arévalo Molina

Santiago Andrés Herrera Anda

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Pedro José Arévalo Molina y Santiago Andrés Herrera Anda, bajo mi supervisión.

Nelson Sotomayor, MSc.

DIRECTOR DEL PROYECTO

AGRADECIMIENTO

La realización de este proyecto no hubiera sido posible sin la participación de varias personas que colaboraron en el desarrollo del mismo. Quisiera agradecer a National Instruments Ecuador por el apoyo y auspicio brindado desde el inicio hasta la culminación del proyecto; así como al ingeniero Nelson Sotomayor director del proyecto. Finalmente quisiera agradecer y dedicar el esfuerzo plasmado en este trabajo a personas muy especiales que han jugado un rol decisivo en el desarrollo de este proyecto, puesto que han sido fuente de inspiración y modelo de vida: Orlando, Livia, Vanessa, Marco, Andrea, Belén, Ma. José, Sebastián, Carlos Eduardo, Ricardito, este esfuerzo es para ustedes. Gracias.

Pedro José Arévalo Molina

Me gustaría dedicar ésta tesis a toda mi familia, sobre todo a mis padres por su apoyo incondicional y pilares fundamentales en mi educación y formación como persona; por los valores y principios que me han enseñado con su ejemplo de honestidad, sacrificio y esfuerzo. Gracias a ellos y a mi familia he logrado conseguir las metas propuestas en mi vida con empeño y perseverancia. Me considero afortunado por tener unos padres como ellos, una familia y amigos que sin duda han sido grandes aliados para lograr alcanzar mis objetivos. A todos ellos gracias.

Santiago Andrés Herrera Anda

CONTENIDO

RESUMEN

PRESENTACIÓN

CAPÍTULO 1

FUNDAMENTOS BÁSICOS	1
1.1 INTRODUCCIÓN A LA ROBÓTICA	2
1.2 DEFINICIÓN DEL ROBOT	2
1.2.1 DEFINICIÓN DE ROBOT INDUSTRIAL MANIPULADOR.....	5
1.2.2 DEFINICIONES ADICIONALES	5
1.2.2.1 Robots teleoperados (NASA).....	5
1.2.1.2 Cobot.....	6
1.3 CLASIFICACIÓN DE LOS ROBOTS.....	7
1.3.1 CLASIFICACIÓN DE ACUERDO A LA GENERACIÓN.....	7
1.3.2 CLASIFICACIÓN ATENDIENDO AL ÁREA DE APLICACIÓN	9
1.3.3 CLASIFICACIÓN ATENDIENDO AL TIPO DE ACTUADORES	10
1.3.4 CLASIFICACIÓN ATENDIENDO AL NÚMERO DE EJES.....	11
1.3.5 CLASIFICACIÓN ATENDIENDO A LA CONFIGURACIÓN.....	11
1.3.6 CLASIFICACIÓN ATENDIENDO AL TIPO DE CONTROL.....	12
1.3.6.1 Robot secuencial.....	12
1.3.6.2 Robot controlado por trayectoria	12
1.3.6.3 Robot adaptativo	12
1.3.6.4 Robot teleoperado.....	12
1.4 morfología del robot.....	13
1.4.1 ESTRUCTURA MECÁNICA DE UN ROBOT INDUSTRIAL	13
1.4.2 TRANSMISORES	15
1.4.2.1 Reductores.....	16
1.4.2.2 Accionamiento Directo	17
1.4.3 ACTUADORES.....	18
1.4.3.1 Actuadores neumáticos.....	18

1.4.3.2	Actuadores hidráulicos	18
1.4.3.3	Actuadores eléctricos	19
1.4.3.3.1	Servomotores	19
1.4.3.3.2	Composición del servomotor	19
1.4.3.3.3	Control de posición	21
1.4.3.3.4	Utilización de un servomotor	22
1.4.3.3.5	Modificaciones posibles en un servomotor	22
1.4.3.3.6	Servos digitales	23
1.5	INTRODUCCIÓN A LA VISIÓN ARTIFICIAL	24
1.5.1	CONFIGURACIÓN BÁSICA DE UN SISTEMA DE VISIÓN ARTIFICIAL ...	25
1.5.1.1	Etapas Básicas de una aplicación en Visión Artificial	25
1.6	PROCESAMIENTO DIGITAL DE IMÁGENES	27
1.6.1	ETAPA DE PRE-PROCESAMIENTO	27
1.6.1.1	Representación de una imagen a escala de grises	28
1.6.2	ETAPA DE SEGMENTACIÓN	29
1.6.2.1	Representación de una imagen en formato binario	29
1.6.2.2	Segmentación por detección de bordes	31
1.6.2.2.1	Técnicas basadas en el operador Gradiente	32
1.6.2.2.2	Operadores basados en el Laplaciano	32
1.6.2.2.3	Operadores de Canny	33
1.6.2.3	Segmentación basada en morfología	33
1.6.2.3.1	Conectividad	33
1.6.2.3.2	Erosión	34
1.6.2.3.3	Dilatación	35
1.6.2.3.4	Apertura y Cierre	35
1.6.2.4	Segmentación basada en modelos	37
1.6.2.4.1	Transformada de Hough	37
1.6.2.4.2	Modelo General para implementar la Transformada de Hough	37
1.6.3	ETAPA DE REPRESENTACIÓN Y DESCRIPCIÓN	38
1.6.3.1	Búsqueda de objetos	38
1.6.3.2	Medición de objetos	39
1.6.4	ETAPA DE RECONOCIMIENTO E INTERPRETACIÓN	39

CAPÍTULO 2

DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO	40
2.1 DISEÑO MECÁNICO.....	40
2.1.1 SOLIDWORKS	40
2.1.2 DISEÑO REALIZADO EN SOLIDWORKS PARA EL ROBOT	43
2.1.2.1 Piezas que conforman la base	43
2.1.2.2 Piezas que conforman el primer eslabón	44
2.1.2.3 Piezas que conforman el segundo eslabón.....	46
2.1.2.4 Piezas que conforman el tercer eslabón	47
2.1.2.5 Piezas que conforman el cuarto eslabón	48
2.1.3 COSMOSWorks	49
2.1.3.1 Criterio de von Mises.....	53
2.1.3.2 Interpretación del Factor de Seguridad (FDS).....	53
2.1.3.3 Análisis de Deformación Unitaria	54
2.1.4 ANÁLISIS DE ESFUERZOS DEL ROBOT	55
2.1.4.1 Análisis de Esfuerzos del Tercer Eslabón	55
2.1.4.2 Análisis de Esfuerzos del Segundo Eslabón	56
2.1.4.3 Análisis de Esfuerzos del Primer Eslabón.....	57
2.1.4.4 Análisis de Esfuerzos de la Base	58
2.2 DISEÑO ELECTRÓNICO	59
2.2.1 COSMOSMotion	60
2.2.2 ANÁLISIS DE MOVIMIENTO DEL ROBOT	63
2.2.2.1 Configuración del movimiento de las articulaciones.....	64
2.2.3 HARDWARE UTILIZADO PARA LE CONTROL DE POSICIÓN	68
2.2.3.1 Realimentación de Posición	68
2.2.3.2 Adquisición de Datos	70
2.2.3.3 Circuito de Potencia para el Electroimán	70
2.2.3.4 Señales de Control.....	71
2.2.4 HARDWARE UTILIZADO PARA EL SISTEMA DE VISIÓN ARTIFICIAL ...	73
2.2.5 SISTEMA DE ALIMENTACIÓN	75
2.2.6 MONTAJE DEL HARWARE PARA LE SISTEMA DE CONTROL	75

CAPÍTULO 3

DESARROLLO DEL PROGRAMA DE CONTROL.....	77
3.1 CINEMÁTICA DEL ROBOT.....	77
3.2 EL PROBLEMA CINEMÁTICO DIRECTO.....	78
3.2.1 RESOLUCIÓN DEL PROBLEMA CINEMÁTICO DIRECTO MEDIANTE MÉTODOS GEOMÉTRICOS	79
3.2.2 ALGORITMO DE DENAVIT-HARTENBERG PARA LA OBTENCIÓN DEL MODELO CINEMÁTICO DIRECTO	82
3.3 EL PROBLEMA CINEMÁTICO INVERSO.....	88
3.3.1 RESOLUCIÓN DEL PROBLEMA CINEMÁTICO INVERSO POR MÉTODOS GEOMÉTRICOS	91
3.4 DESARROLLO DEL SOFTWARE DE CONTROL EN LabVIEW	97
3.4.1 SOFTWARE PARA PROCESAMIENTO DE IMAGEN	98
3.4.2 SOFTWARE PARA CONTROL DE POSICIÓN.....	107

CAPÍTULO 4

PRUEBAS Y RESULTADOS.....	117
4.1 PRUEBAS DEL SISTEMA DE VISIÓN ARTIFICIAL.....	117
4.1.1 CORRECCIÓN DE DEFORMACIÓN ESFÉRICA Y CONVERSIÓN PÍXELES - CENTÍMETROS	117
4.1.2 DETECCIÓN DE OBJETOS EXTRAÑOS EN LA ESCENA	120
4.1.3 EXTRACCIÓN DE COORDENADAS Y FILTRADO DE FORMA.....	122
4.1.3.1 Programa de Selección de Forma.....	122
4.1.3.2 Programa de clasificación	126
4.2 PRUEBAS DEL SISTEMA DE CONTROL DE POSICIÓN	127
4.2.1 ACONDICIONAMIENTO DE SEÑAL DE POSICIÓN	128
4.2.2 SISTEMA DE CONTROL DE POSICIÓN ANGULAR.....	130
4.2.3 SISTEMA DE CONTROL DE POSICIÓN ESPACIAL.....	133
4.2.3.1 Modo Manual	133
4.2.3.2 Modo Automático	134

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES	136
5.1 CONCLUSIONES	136
5.2 RECOMENDACIONES.....	140

REFERENCIAS BIBLIOGRÁFICAS	142
---	------------

ANEXO A

MANUAL DE USUARIO

ANEXO B

PLANOS DE LAS PIEZAS QUE COMPONEN AL ROBOT

ANEXO C

USO DE LabVIEW MECHATRONICS TOOLKIT

ANEXO D

DISEÑO Y CONSTRUCCIÓN DE CIRCUITOS

RESUMEN

La robótica en los últimos años ha logrado un lugar importante en la industria moderna, siendo actualmente los robots industriales, herramientas imprescindibles para el funcionamiento y progreso de muchas industrias.

Gracias a la amplia gama de ventajas que ofrece un robot frente a un operador humano, fundamentalmente en su desempeño durante tareas repetitivas, largos periodos de trabajo, o control de calidad visual a grandes velocidades a través de la Visión Artificial; de ha contribuido en la reducción de los costos de producción y el mejoramiento de la calidad de los productos en la industria.

El desarrollo del software y la tecnología ha permitido realizar sistemas robóticos más autónomos y con controles mucho más avanzados, que además del control clásico, incorporan inteligencia artificial lo que permite al robot trabajar con una mayor interacción con el medio que lo rodea. Tratando cada vez más, de emular el comportamiento humano, tanto en su actuar con el medio, mediante los sentidos (visión, tacto, etc.) como la toma de decisiones en base a las condiciones y restricciones.

El sector donde se utilizan la mayor parte de los robots, es sin duda el industrial. Por tal motivo el presente trabajo ha sido enfocado a éste sector, a través de toda la ingeniería que interviene en el diseño, la construcción y el control de un prototipo de robot industrial con arquitectura antropomórfica, desde su diseño CAD hasta el desarrollo de características avanzadas como el sistema de visión artificial, el cual a través de una cámara permite la localización de objetos en el plano de trabajo, y además la detección de interferencias generadas por objetos extraños dentro del área de trabajo, lo cual permite generar alarmas para que el robot cambie a otro estado, ya sean de reposo o movimiento. Esta capacidad adicional de Visión Artificial da al robot una característica muy útil dentro de procesos de clasificación o detección de fallas en la producción de una industria.

En el presente proyecto se abordan diferentes temáticas relacionadas con sistemas de visión artificial y robótica industrial, que convergen en el desarrollo de una aplicación específica orientada a la adquisición y procesamiento de imagen,

como medio de información para el control de un manipulador industrial en lazo cerrado.

PRESENTACIÓN

En el presente trabajo se abordan los procesos de diseño, validación, construcción y control en lazo cerrado de un prototipo de robot industrial de cuatro grados de libertad dotado con visión artificial orientado a la detección, clasificación y traslado de objetos presentes en un espacio definido como área de trabajo.

En el Capítulo 1 se realiza una introducción a todo el marco teórico necesario para el desarrollo del proyecto. Se trata de una forma general los conceptos relacionados con robótica, actuadores, transmisores y la teoría referida a las herramientas y etapas para el procesamiento de imágenes por computador.

En el Capítulo 2 se describe el proceso de diseño mecánico realizado en el software de diseño CAD SolidWorks, y la validación del modelo mediante análisis de esfuerzos realizado en la herramienta de SolidWorks, llamada COSMOSWorks. Adicionalmente, se describe el proceso de diseño electrónico, el cual empieza con el dimensionamiento de los actuadores mediante COSMOSMotion (SolidWorks), y continúa con la descripción del hardware y metodología utilizados para la implementación del sistema de control del robot.

En el Capítulo 3 se explica el funcionamiento del software de control diseñado mediante LabVIEW, tanto para Visión Artificial como para el control de posición. Para ello se partió del desarrollo de la cinemática directa e inversa para el posicionamiento del extremo del robot. Posteriormente, se describen más detalladamente los algoritmos y subprocesos utilizados para la implementación del software de Control de Posición y Visión Artificial, que en conjunto permiten el manejo del robot y la detección de obstáculos en el área de trabajo.

En el Capítulo 4 se presentan las diferentes pruebas realizadas para evaluar el funcionamiento del sistema. Inicialmente se realizaron pruebas de subprocesos simples como la corrección de la deformación esférica en la imagen adquirida y la linealización de actuadores en el control de posición. Posteriormente se evaluó el funcionamiento de todo el sistema, en cuanto a la detección, selección, clasificación de objetos y su posterior traslado.

En Capítulo 5 se presentan las conclusiones y recomendaciones, que se han podido extraer durante todo el proceso de desarrollo del presente trabajo.

CAPÍTULO 1

FUNDAMENTOS BÁSICOS

1.1 INTRODUCCIÓN A LA ROBÓTICA

La historia de la robótica ha estado unida a la construcción de “artefactos”, que trataban de materializar el deseo humano de crear seres a su semejanza que lo descargasen del trabajo. Los griegos tenían una palabra específica para denominar a estas máquinas: *automatos*. De esta palabra deriva la actual **autómata**: máquina que imita la figura y movimientos de un ser animado [1].

En realidad *Karel Čapek*, un escritor checo, acuñó en 1921 el término "Robot" en su obra dramática "Rossum's Universal Robots / R.U.R.", a partir de la palabra checa “robotá”, que significa servidumbre o trabajo forzado [1].

El término robótica es acuñado por Isaac Asimov, famoso escritor y bioquímico ateo judío nacionalizado estadounidense nacido en Bielorrusia, exitoso y excepcionalmente prolífico autor de obras de ciencia ficción, historia y divulgación científica, fue también quien definió a la ciencia que estudia a los robots. Asimov creó además las Tres Leyes de la Robótica [1].

En octubre de 1942 publicó en la revista *Galaxy Science Fiction* una historia titulada “*The Caves of Steel*”, en la que por primera vez anunció sus **tres leyes de la robótica** [1]:

1. Un robot no puede perjudicar a un ser humano, ni con su inacción permitir que un ser humano sufra daño.
2. Un robot ha de obedecer las órdenes recibidas de un ser humano, excepto si tales órdenes entran en conflicto con la primera ley.
3. Un robot debe proteger su propia existencia mientras tal protección no entre en conflicto con la primera o segunda ley.

1.2 DEFINICIÓN DEL ROBOT

Muchas veces un robot es entendido como un humanoide que emula al ser humano, pero no es del todo cierto, ya que el término engloba un amplio campo de dispositivos, que hoy en día, son reconocidos.

Es por ello, complicado acotar lo que debe ser entendido por un robot. En la Tabla 1.1 se recogen algunas de las definiciones consideradas por los diccionarios y enciclopedias de más prestigio [1]:

Tabla 1.1 Definición de robot según algunos diccionarios y enciclopedias, tomado de [1].

<p>Enciclopedia Británica</p> <p><i>Máquina operada automáticamente que sustituye el esfuerzo de los humanos, aunque no tiene por qué tener apariencia humana o desarrollar sus actividades a manera de humanos.</i></p>
<p>Diccionario Merrian Webster</p> <p><i>Máquina que se asemeja a los humanos y desarrolla como ellos tareas complejas como andar o hablar. Un dispositivo que desarrolla de manera automática tareas complicadas, a menudo de manera repetitiva, guiado por control automático.</i></p>
<p>Diccionario de la Real Academia Española</p> <p><i>Máquina o ingenio electrónico programable, capaz de manipular objetos y realizar operaciones antes reservados sólo a las personas.</i></p>

En el estado tecnológico actual de la robótica, incluso estas definiciones generalistas resultan insuficientes para abarcar el conjunto de sistemas que son entendidos hoy en día por **robot** [1].

Por eso para poder lograr diferenciar, al término robot se le añade un adjetivo, el cual permite dar un mayor detalle de sus características y campo de aplicación; como por ejemplo: robot aéreo, robot industrial antropomórfico de 4 grados de libertad, etc.

La Tabla 1.2 muestra los tipos de robots incluidos en las líneas de actividad investigadora en robótica que considera el Grupo de Robótica (GtROb) del Comité Español de Automática [2].

Tabla 1.2 Tipos de robots considerados por el GtRob del Comité Español de Automática CEA, tomado de [2].

Robots aéreos
Robots asistenciales
Robots autónomos
Robots caminantes
Robots cooperantes
Robots de inspección
Robots de educación
Robots de entretenimiento
Robots escaladores
Robots espaciales
Robots de exteriores

Robots de servicio
Robots flexibles
Robots humanoides
Robots manipuladores
Robots marinos
Robots bio-médicos
Robots móviles
Robots paralelos
Robots personales
Robots submarinos
Sistemas multi robots
Tele robots
Manipuladores móviles
Mini y Micro robots

Esta lista ha ido creciendo a lo largo de los años, ya que hasta finales de los años ochenta, los únicos considerados robots eran aquellos dedicados a la manufactura. El trabajo que este tipo de robot, han realizado básicamente el de manipular piezas o herramientas en entorno industrial, por lo que se le denomina **robot industrial manipulador**. Sobre él ha sido posible establecer una precisa definición y clasificación según diferentes criterios, como se verá posteriormente [1].

Se van a presentar a continuación las definiciones estandarizadas del robot industrial manipulador, así como diferentes criterios para su clasificación. Por otra parte y en la medida de lo posible, se definirán y clasificarán algunos de los otros tipos de robots [1].

1.2.1 DEFINICIÓN DE ROBOT INDUSTRIAL MANIPULADOR

Según *Robot Institute of America*, actualmente *Robotic Industries Association*

“Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materiales, piezas, herramientas o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas” [1].

De acuerdo a ISO 8373, se define a un robot como “un manipulador de tres o más ejes, con control automático, reprogramable, multiaplicación, móvil o no, destinado a ser utilizado en aplicaciones de automatización industrial. Incluye al manipulador (sistema mecánico y accionadores) y al sistema de control (software, hardware de control y potencia) [1].

Reprogramable: Aquellos en los que los movimientos programados o las funciones auxiliares pueden cambiarse sin modificación física [1].

Modificación física: Modificación de la estructura mecánica o del sistema de control (se excluyen cambios en los soportes de memoria: Disco, cinta, ROM, etc.) [1].

1.2.2 DEFINICIONES ADICIONALES

Una temprana definición, de los **robots teleoperados** es la dada por la NASA (1978) y recogida sin apenas modificaciones en el *Handbook of Industrial Robotics* [1].

1.2.2.1 Robots teleoperados (NASA)

Dispositivos robóticos con brazos manipuladores, sensores y cierto grado de movilidad, controlados remotamente por un operador humano de manera directa o a través de un ordenador [1].

Esta definición, igual que la dada en su día por el RIA para los robots, queda limitada a los robots manipuladores, excluyendo a todos los otros que, sin tener necesariamente la capacidad de manipuladores, pueden ser operados a distancia. Así en la actualidad, se debe entender por telerobot; a cualquier robot cuya tarea pueda ser monitorizada y reprogramada a distancia por el ser humano [1].

La Norma ISO, define a un Robot Teleoperado como un robot que puede ser controlado remotamente por un operador humano, extendiendo las capacidades sensoriales y motoras de éste a localizaciones remotas [1].

En ambas definiciones se deja abierta la posibilidad de teleoperar cualquier tipo de robot.

1.2.2.2. Cobot

Un último concepto derivado de la telerobótica son los denominados Cobots, es decir, aquel dispositivo robótico que manipula objetos en colaboración física directa con el ser humano. Los Cobots, ideados para la colaboración estrecha entre humano y robot, se utilizan en tareas donde es preciso guiar al operador por caminos sometidos a restricciones (superficies o restricciones virtuales) o donde éste debe mover cargas notables. El cobot, limita las trayectorias que puede hacer el humano o soporta la carga, mientras éste dirige el movimiento [1].

Otras definiciones de interés relacionadas con los robots son las referentes a los útiles que puede portar el robot en su extremo. La UNE EN ISO 8373:1998 define así:

Elemento terminal de robot: *Objetivo unido a la interface mecánica dispuesta en el extremo del robot, para dotarle a éste de la funcionalidad necesaria para realizar su tarea. Se incluyen pinzas, herramientas y dispositivos para facilitar la conexión con otros elementos [1].*

Pinza: *Elementos Terminal concebido para coger y sujetar [1].*

1.3 CLASIFICACIÓN DE LOS ROBOTS

Existen diferentes maneras de clasificar a los Robots, que van a depender de su forma o construcción, de la aplicación, de la tarea a que se destinan, etc. A continuación se puede observar algunos tipos de clasificación.

1.3.1 CLASIFICACIÓN DE ACUERDO A LA GENERACIÓN

La generación de un robot hace referencia al momento tecnológico en que éste aparece. De éste modo se puede considerar que se pasa de una generación a la siguiente cuando se da un impulso que supone un avance significativo en las capacidades de los robots. En la Tabla 1.3 se muestra una posible clasificación por generaciones:

Tabla 1.3 Clasificación de los Robots según generaciones, tomado de [1].

1.^a Generación	Repite la tarea programada secuencialmente. No toma en cuenta las posibles alteraciones de su entorno.
2.^a Generación	Adquiere información limitada de su entorno y actúan en consecuencia. Puede localizar, clasificar (visión) y detecta esfuerzos y adaptar sus movimientos en consecuencia.
3.^a Generación	Su programación se realiza mediante el empleo de un lenguaje natural. Posee capacidad para la planificación automática de tareas.

La tercera generación está desarrollándose en estos días, siendo, por lo tanto, objeto de un futuro cercano.

La Tabla 1.4 resume una clasificación propuesta por Knasel en el año 1986. A pesar de su antigüedad los aspectos recogidos en ella no han perdido validez, permitiendo tener una visión del pasado, presente y futuro de la robótica.

Tabla 1.4 Características de las generaciones de los robots, tomado de [1]

Generación	Nombre	Tipo de control	Grado de movilidad	Usos más frecuentes
1.^a (1982)	Pick & place	Fines de carrera, aprendizaje	Ninguno	Manipulación, servicio de máquinas
2.^a (1984)	Servo	Servocontrol, trayectoria continua, programación Condicional	Desplazamiento por vía pintura	Soldadura
3.^a (1989)	Ensamblado	Servos de precisión, visión, tacto, programación Off-line	AGV Guiado por vía	Ensamblado Desbarbado
4.^a (2000)	Móvil	Sensores inteligentes	Patas Ruedas	Construcción Mantenimiento
5.^a (2010)	Especiales	Controlados con técnicas de IA	Andante Saltarín	Uso Militar Uso Especial

1.3.2 CLASIFICACIÓN ATENDIENDO AL ÁREA DE APLICACIÓN

Otro modo de clasificación es a través del uso que se le dé al robot, esto es en base al sector económico en el que trabajan, el tipo de aplicación o tarea que desarrollen. Las actividades económicas pueden ser clasificadas de acuerdo al estándar internacional de clasificación de sectores de las actividades económicas (ISIC). En la Tabla 1.5 se recoge sintetizada la clasificación.

Tabla 1.5 Clasificación de las aplicaciones de los robots industriales manipuladores, según IFR, tomado de [1].

000	Sin especificar
110	Manipulación en fundición
130	Manipulación en moldeo de plásticos
140	Manipulación en tratamientos térmicos
150	Manipulación en la forja y estampación
160	Soldadura
170	Aplicación de materiales
180	Mecanización
190	Otros procesos
200	Montaje
210	Empaquetado
220	Medición, inspección, control de calidad
230	Manipulación de materiales
240	Formación, enseñanza e investigación
900	Otros

También se considera el grado de interacción con el humano, como se muestra en la Tabla 1.6.

Tabla 1.6 Clasificación de los robots de servicio por Categoría y tipo de Interacción según IFR, tomado de [1].

510	Robots de servicios a humanos (personales, seguridad, entretenimiento, etc.) Interactivos No interactivos Otros
520	Robots de servicios a equipos (mantenimiento, reparación, limpieza, etc.) Interactivos No interactivos Otros
530	Otros robots desarrollando funciones autónomas (vigilancia, transporte, adquisición de datos, etc.) y/o robots de servicio que no entran en 510 y 520.

1.3.3 CLASIFICACIÓN ATENDIENDO AL TIPO DE ACTUADORES

Dependiendo de cuál sea el tipo de energía utilizada por los ejes principales del robot, éste puede ser clasificado como:

- Robot Neumático
- Robot Hidráulico
- Robot Eléctrico

La mayor parte de los robots actuales son de accionamiento eléctrico, pero pueden encontrarse casos particulares de robots con accionamiento hidráulico o neumático [1].

Los accionamientos hidráulicos en particular se deben utilizar cuando se necesite de una elevada relación capacidad de carga-peso del robot o cuando sea necesario aislamiento eléctrico entre el robot y el resto del sistema [1].

1.3.4 CLASIFICACIÓN ATENDIENDO AL NÚMERO DE EJES

Esta característica es aplicable a los robots o telerobots con cadena cinemática (es decir, sería aplicable a los robots manipuladores, pero no lo sería, por ejemplo, a los robots móviles). Se entiende por eje cada uno de los movimientos independientes con que está dotado el robot. Puesto que de acuerdo a la definición ISO el robot manipulador industrial debe tener al menos 3 ejes y extendiendo esta condición a los robots de servicio manipuladores, se podrán encontrar robots de cualquier número de ejes superior o igual a 3. En la práctica, la mayor parte de los robots tienen 6 ejes, seguidos por los de 4. Los robots con más de 6 ejes son poco frecuentes, estando justificado este número para aumentar la capacidad de maniobra del robot y siendo en muchas ocasiones telerobots [1].

1.3.5 CLASIFICACIÓN ATENDIENDO A LA CONFIGURACIÓN

Esta clasificación solamente es aplicable a robots y telerobots con cadena cinemática, donde la configuración que adopte el robot quedará definida por el tipo de movimientos permitidos entre dos eslabones consecutivos de la cadena. De acuerdo a esto se tienen los tipos de configuraciones indicados en la Tabla 1.7.

Tabla 1.7 Tipos de Configuraciones de Robots, tomado de [1].

Cartesiano
Cilíndrico
Polar o Esférico
Articular
SCARA
Paralelo

1.3.6 CLASIFICACIÓN ATENDIENDO AL TIPO DE CONTROL

Atendiendo al tipo de Control, la norma ISO 8373 y, en correspondencia la IFR, distingue entre los siguientes:

1.3.6.1 Robot secuencial

Robot con un sistema de control en el que un conjunto de movimientos se efectúa eje a eje en un orden dado, de tal forma que la finalización de un movimiento inicia el siguiente [1].

1.3.6.2 Robot controlado por trayectoria

Robot que ejecuta un procedimiento controlado por el cual los movimientos de tres o más ejes controlados, se desarrollan según instrucciones que especifican en el tiempo, la trayectoria requerida para alcanzar la siguiente posición (obtenida normalmente por interpolación) [1].

Los robots controlados por trayectoria permiten la realización de movimientos en los que puede ser especificado, toda la trayectoria de manera continua [1].

1.3.6.3 Robot adaptativo

Robot que tiene funciones de control con sensores, control adaptativo, o funciones de control de aprendizaje.

De este modo el robot puede modificar su tarea de acuerdo a la información captada del entorno [1].

1.3.6.4 Robot teleoperado

Un robot que puede ser controlado remotamente por un operador humano, extendiendo las capacidades sensoriales y motoras de éste a localizaciones remotas [1].

1.4 MORFOLOGÍA DEL ROBOT

Un robot está formado principalmente por una estructura mecánica, transmisores, sistemas de accionamiento, sistema sensorial, sistema de potencia y control, y elementos terminales.

1.4.1 ESTRUCTURA MECÁNICA DE UN ROBOT INDUSTRIAL

La estructura mecánica de un robot, es básicamente una serie de eslabones unidos mediante articulaciones que permiten el movimiento relativo entre dos eslabones consecutivos.

Este ensamble de eslabones y juntas interconectados, proporcionan un movimiento de salida controlado en respuesta a un movimiento de entrada, esto es lo que se conoce como una **cadena cinemática** [1].

Debido a la similitud que algunos robots manipuladores industriales, tienen con la anatomía del cuerpo humano, muchas partes de estos robots toman nombres como cuerpo, brazo, hombro, codo, muñeca.

Cada uno de los movimientos independientes que puede realizar cada articulación respecto a la anterior, se denomina **grado de libertad** (GDL).

Cuando dentro de una cadena cinemática se puede llegar de un eslabón a otro a través de dos caminos diferentes dentro de la cadena, se dice que se trata de una cadena cinemática cerrada.

En el caso de que sólo haya un camino posible se dirá que se trata de una cadena cinemática abierta [1].

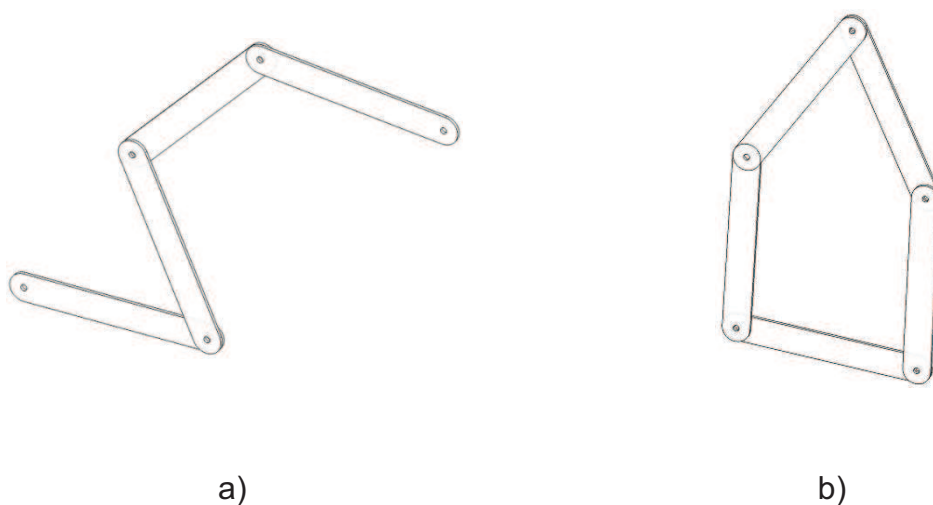


Figura 1.9 Cadena cinemática: a) abierta y b) cerrada

El número de grados de libertad de una cadena cinemática se puede obtener mediante la fórmula de Grübler, según la cual:

$$NGDL = \lambda \cdot (n - j - 1) + \sum_{i=1}^j f_i$$

Donde:

- λ : GDL del espacio de trabajo (típicamente tres en el plano, seis en el espacio)
- n : Número de eslabones (debe incluirse el eslabón fijo o base)
- j : Número de articulaciones
- f_i : Grados de libertad permitidos a la articulación i .

Los robots manipuladores, como es el caso del presente trabajo, en la mayor parte de los casos, son cadenas cinemáticas abiertas con articulaciones de tipo rotacional (con un solo grado de libertad cada uno), por lo que el número de articulaciones coincide con el número de grados de libertad.

La mayor parte de los robots manipuladores actuales, responden a la estructura angular, también conocida como “articular”, representando aproximadamente un 45%.

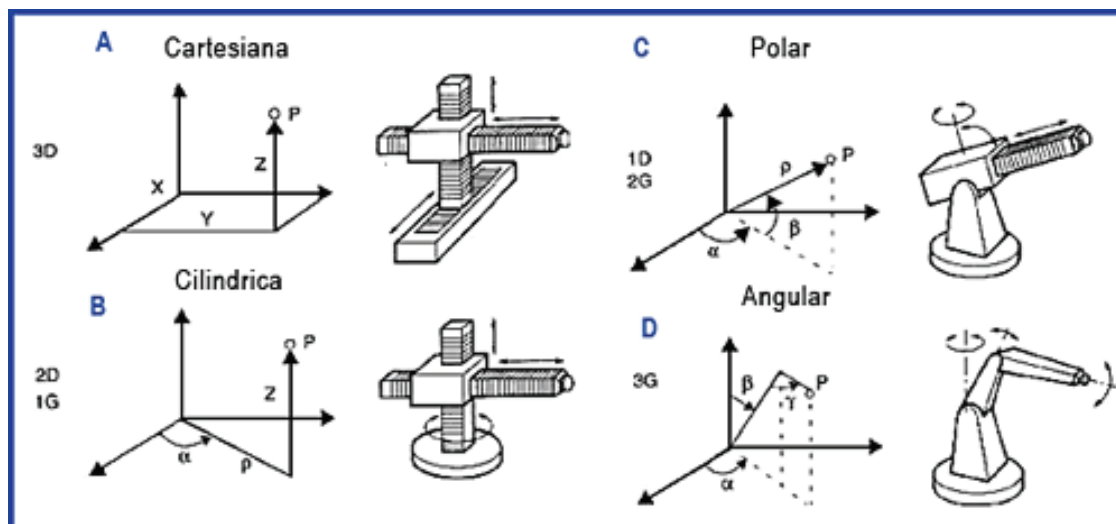


Figura 1.10 Configuraciones más frecuentes de robots, tomado de [3].

Para posicionar y orientar un cuerpo de cualquier manera en el espacio es necesario definir al menos 6 GDL para lograr que el robot posicione y oriente su extremo de cualquier manera en el espacio, aunque muchos por el trabajo que desempeñan no necesitan más que 4 o 5 GDL.

Cuando el número de GDL excede a los GDL necesarios para realizar una determinada tarea, se dice que el robot es **redundante**.

1.4.2 TRANSMISORES

Los transmisores como su nombre lo indica, permiten transmitir el movimiento desde los actuadores a las articulaciones, junto a los transmisores están muchas veces los reductores encargados de adaptar el torque y la velocidad de salida del actuador a los valores adecuados para el movimiento de cada una de las articulaciones del robot.

Muchas veces el peso, el momento de inercia de los elementos del robot y los torques estáticos, son un problema en el desempeño del robot. Por lo que son necesarios sistemas de transmisión que permitan llevar el movimiento hasta las

articulaciones, donde se procura que los actuadores, especialmente los pesados, se encuentren lo más cerca a la base del robot, y en muchas ocasiones se opta por utilizar cadenas o bandas para transmitir el movimiento desde la base del robot.

Aunque también existen actualmente robots industriales con acoplamiento directo, los cuales ofrecen muchas ventajas, entre las cuales están: el control más preciso ya que se evita que pasen por cajas reductoras o engranes, y sobre todo por su montaje simple, lo cual disminuye cualquier posibilidad de error.

1.4.2.1 Reductores

La utilización de reductores en la robótica es importante, sobre todo en los robots industriales, ya que se busca tanto precisión como velocidad de posicionamiento.

Muchos son utilizados para aumentar el torque, pero con una disminución evidente de la velocidad, buscando además cumplir con un bajo peso, reducido tamaño y bajo rozamiento. Existe una limitación en cuanto al torque de salida nominal permisible (T_2) que depende del torque de entrada (T_1) y de la relación de transmisión a través de la relación:

$$T_2 = \eta \cdot T_1 \cdot \frac{\omega_1}{\omega_2}$$

Donde:

T_2 : Torque de salida.

T_1 : Torque de entrada.

η : Rendimiento.

ω_1 : Velocidad angular de entrada.

ω_2 : Velocidad angular de salida.

Debido a que muchas veces los robots trabajan en ciclos cortos que implican continuos arranques y paradas, es muy importante que los reductores sean capaces de soportar picos elevados de torques.

1.4.2.2 Accionamiento directo

Como ya se indicó anteriormente, existen robots que poseen lo que se llama **accionamiento directo** (*Direct Drive DD*), en el que el eje del actuador se acopla directamente a la carga o la articulación que va a mover, sin la utilización de un reductor intermedio.

Las aplicaciones que entran dentro de este grupo, son aquellas donde se necesita combinar precisión con alta velocidad.

Entre las principales ventajas de la utilización de accionamientos directos se tiene las siguientes:

- Posicionamiento rápido y preciso.
- Mejor controlabilidad del sistema.
- Y sobre todo, la simplificación del sistema mecánico.

Mientras que los problemas comunes que se presentan son los siguientes:

- Motor a emplear con un torque elevado y con bajas revoluciones, manteniendo la máxima rigidez posible.
- En algunos casos representa mayores costos, en lo que se refiere al tipo de motor que se debe usar.
- El peso que representan los motores junto con la de los eslabones, afectan enormemente al funcionamiento de la cinemática del robot, especialmente en aquellos cuya arquitectura también es afectada por la gravedad y el peso de sus elementos, lo que no sucede con el robot de tipo SCARA.

Los motores que generalmente son usados en el accionamiento directo son motores síncronos y de continua sin escobillas, ambos con imanes permanentes.

También se utilizan motores de reluctancia variables. Pero todos representan un mayor costo respecto a motores que se utilizarían en su lugar, conjuntamente con transmisores y reductores.

1.4.3 ACTUADORES

Los actuadores son los encargados de generar el movimiento y de éstos depende el funcionamiento de un robot. Antiguamente era común la utilización de la hidráulica para el movimiento de los robots industriales, actualmente, con las ventajas que representa el uso de la electricidad, casi todos los robots utilizan energía eléctrica como fuente de energía.

1.4.3.1 Actuadores neumáticos

En ellos la fuente de energía es aire a presión entre 5 y 10 bar. Existen dos tipos de actuadores neumáticos:

- Cilindros Neumáticos
- Motores Neumáticos

1.4.3.2 Actuadores hidráulicos

Este tipo de actuadores son similares a los neumáticos, pero en lugar de aire utilizan aceite a una presión comprendida normalmente entre los 50 y 100 bar, incluso llegando hasta los 300 bar. Existen también actuadores como en el caso de los neumáticos, de tipo cilindro y del tipo motores de aletas y pistones.

A diferencia de los neumáticos, se puede lograr un control continuo debido a que la compresibilidad del aceite es inferior al aire, además de poder desarrollar elevadas fuerzas y torques.

Entre otras ventajas se tiene que presentan una estabilidad frente a cargas estáticas, es decir, puede soportar peso o presión ejercida sin necesidad de energía. Pero presenta constantes fugas de aceite debido a las elevadas presiones a las que trabajan, además de que la instalación para su funcionamiento es relativamente complicada.

1.4.3.3 Actuadores eléctricos

Las características de control, sencillez y precisión de los accionamientos eléctricos han hecho que sean los más usados en los robots industriales actuales.

Dentro de los actuadores eléctricos pueden distinguirse tres tipos:

- Motores de corriente continua (DC)
 - Controlado por voltaje de armadura
 - Controlados a través de la corriente de excitación
- Motores de corriente alterna (AC)
 - Síncronos
 - Asíncronos
- Motores paso a paso.

1.4.3.3.1 Servomotores

Los Servomotores son un tipo especial de motores DC, que se caracterizan por su capacidad para colocarse de forma casi inmediata en cualquier posición dentro de su intervalo de operación (0° a 180°) [4].

Se dice que el servo es un dispositivo con un eje de rendimiento controlado ya que puede ser llevado a posiciones angulares específicas al enviar una señal codificada. Con tal que exista una señal codificada en la línea de entrada, el servo mantendrá la posición angular del engranaje. Cuando la señal codificada cambia, la posición angular de los piñones cambia. En la práctica, se usan servos para posicionar elementos de control como palancas, pequeños ascensores y timones [4].

1.4.3.3.2 Composición del servomotor

Un servomotor común, lleva por lo general consigo un motor DC, que realiza la función de actuador al aplicarse un voltaje entre sus dos terminales, donde el

motor gira en un sentido a alta velocidad, pero produciendo un bajo par. Para aumentar el par del dispositivo se utiliza un juego de piñones en una caja reductora, que transforma gran parte de la velocidad de giro en torque y finalmente está la circuitería de control.

Además se pueden ver los 3 cables de conexión externa:

- Uno rojo, que es para la alimentación, V_{cc} ($\sim +6V$)
- Otro negro o marrón, para conexión a tierra, Gnd
- El último que puede ser blanco, amarillo, o naranja que corresponde a la línea de control por la que se envía la señal codificada para comunicar el ángulo en el que se debe posicionar.

Tabla 1.9 Colores de los terminales para algunas marcas comerciales, tomado de [4]

Fabricante	Voltaje positivo	Tierra	Señal de control
Futaba	Rojo	Negro	Blanco
Dong Yang	Rojo	Marrón	Naranja
Hitec	Rojo	Negro	Amarillo
JR	Rojo	Marrón	Naranja
Airtronics	Rojo	Negro	Naranja
Fleet	Rojo	Negro	Blanco
Krafr	Rojo	Negro	Naranja
E-Sky	Rojo	Negro	Blanco

1.4.3.3.3 Control de posición

Como ya se ha dicho, los servomotores utilizan un circuito de control interno, el cual permite ubicar al eje en una posición angular específica. Éste control es de tipo **proporcional**.

El set-point se ingresa mediante una señal cuadrada, donde el ancho que dura el pulso es proporcional al ángulo de giro que se desea.

La señal PWM que sirve para el control del motor, se convierte en una señal análoga de voltaje que será comparada con la del potenciómetro que es la señal de realimentación que indica el valor actual de posición y mediante éstas dos señales, se obtiene la señal de error que se amplifica con una ganancia y posteriormente se aplica a los terminales del motor.

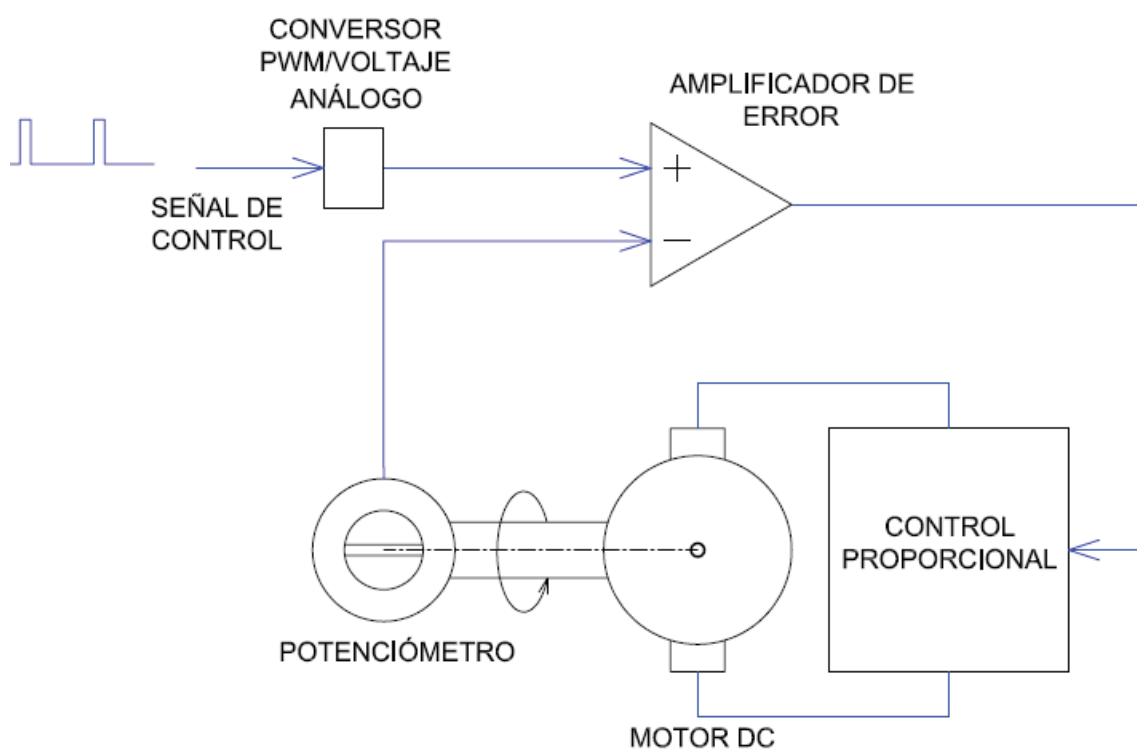


Figura 1.11 Diagrama del circuito de control interno de un servomotor, tomado de [4].

1.4.3.3.4 Utilización de un servomotor

Dependiendo del modelo o marca del servo, el voltaje de alimentación puede estar comprendido entre 4 y 8 voltios.

El ángulo para la ubicación angular del servo, depende del tiempo en alto de la señal, cuyos rangos de operación van a depender de cada servo, ya sea por marca o como esté ensamblado internamente. Por lo que es necesario antes de su utilización, calibrar el rango de valores para obtener un movimiento de 0° a 180°; ya que si los valores están fuera del rango, el servo tiene un límite mecánico que traba el eje, y si se llega a tener por prolongados periodos de tiempo valores fuera del rango, además de tener un zumbido por el esfuerzo excesivo del servo, puede ocasionar que se dañen los piñones. Aproximadamente el rango de tiempo en alto para lograr un movimiento de 180°, está comprendido entre 1 y 2 ms. Para lo cual se podría aplicar la siguiente fórmula:

$$t = 1 + \frac{\emptyset}{180}$$

Donde t está en milisegundos y \emptyset en grados.

Para lograr mantener al servo en una posición específica, es necesario mantener la misma relación de trabajo de la señal PWM, para mantener así el control operando continuamente, sin importar que el eje esté sometido a torques externos, éste mantendrá la posición indicada hasta los límites de su torque; lo que no sucede cuando se lo desconecta de la alimentación o del control, ya que cualquier torque externo podrá mover el eje.

1.4.3.3.5 Modificaciones posibles en un servomotor

El potenciómetro del sistema de control del servo es un potenciómetro de menos de una vuelta, de modo que no puede dar giros completos en un mismo sentido. Para evitar un daño del potenciómetro, por lo general los fabricantes añaden un pequeño tope en el sistema de engranajes de la caja reductora, lo cual impide que éste gire más de lo permitido. Para los servos de éste tipo, es imposible realizar

un control a lazo cerrado, debido a que no se tiene una realimentación de posición de ningún tipo.

Para todas las limitaciones antes expuestas, existen posibles modificaciones realizables.

El primero es la completa eliminación del sistema electrónico de control, para conservar únicamente el motor de corriente continua y el sistema de engranes reductores. Con esto se obtiene simplemente un motor de corriente continua con caja reductora en un mismo empaquetado, útil en aplicaciones donde no se necesite del control de posición incorporado en el servo.

La segunda modificación realizable consiste en un cambio en el sistema de control, de modo que se obtenga un sistema de control de velocidad. Para ello, se desacopla el potenciómetro de realimentación del eje del motor, y se hace que permanezca estático en una misma posición. Así, la señal de error del sistema de control dependerá directamente del valor deseado que se ajuste (que seguirá indicándose mediante pulsos de duración variable). Ambos tipos de modificación requieren que se elimine físicamente la pestaña limitadora de la caja reductora.

El tercer tipo de modificación consiste simplemente en soldar dos cables en el potenciómetro interno, para obtener una señal de voltaje diferencial, el cual permitirá conocer en todo momento y con exactitud, la posición angular del eje del servomotor, lo cual es útil si se quiere hacer un control a lazo cerrado, sin la necesidad de adicionar sensores externos, que puede ser un tanto complicado sobre todo en el acople mecánico de los mismos. Para éste caso no se necesita hacer ninguna modificación mecánica, ya que se le puede dejar todo el sistema interno del servo totalmente intacto y solamente sacar los cables indicados para realizar un control de cualquier tipo, pero externamente.

1.4.3.3.6 Servos digitales

Los servos digitales son similares a los servos convencionales (analógicos), pero cuentan con ciertas ventajas como lo son un mayor par, una mayor precisión, un tiempo de respuesta menor, y la posibilidad de modificar parámetros básicos de

funcionamiento (ángulos, máximo y mínimo de trabajo; velocidad de respuesta; sentido de giro y posición central; entre otros). Además de un mayor costo, tienen la desventaja de que requieren más energía para su funcionamiento, lo cual es crítico cuando se utilizan en aplicaciones que requieren el máximo ahorro de energía posible, tales como robots robustos o aviones radiocontrolados.

1.5 INTRODUCCIÓN A LA VISIÓN ARTIFICIAL

Uno de los sentidos más importantes de los seres humanos es la visión. Ésta es empleada para obtener la información visual del entorno físico.

Un sistema de visión artificial trata de emular la función del sentido de la visión en un ser humano; y para ello utiliza variados algoritmos y métodos que permiten procesar la imagen adquirida. La actividad de extraer información de las imágenes se lleva a cabo desde que se inventó la fotografía, pero no es sino hasta la década de los 80, con la revolución de la electrónica, las cámaras de video CCD y los microprocesadores, junto con la evolución de las Ciencias de la Computación, que se hace factible la Visión Artificial [5]. La Visión Artificial, también llamada Visión por Computador, pretende capturar información del entorno físico para extraer características visuales relevantes, utilizando procedimientos automáticos [5].

En un sistema de visión artificial, un factor muy importante a considerar es la naturaleza de la luz, ya que los foto-receptores dentro de las cámaras captan la información de la imagen en función de la incidencia de luz sobre sí mismos [5]. La luz es una onda electromagnética que se propaga como un conjunto de dos ondas vectoriales que son perpendiculares. De estas dos ondas, una representa el campo eléctrico y otra el campo magnético. El ojo humano percibe solo una porción del conjunto de ondas electromagnéticas que forman el espectro electromagnético completo. Estas ondas emanan radiaciones de longitudes de onda comprendidas entre los 380 nm y los 780 nm.

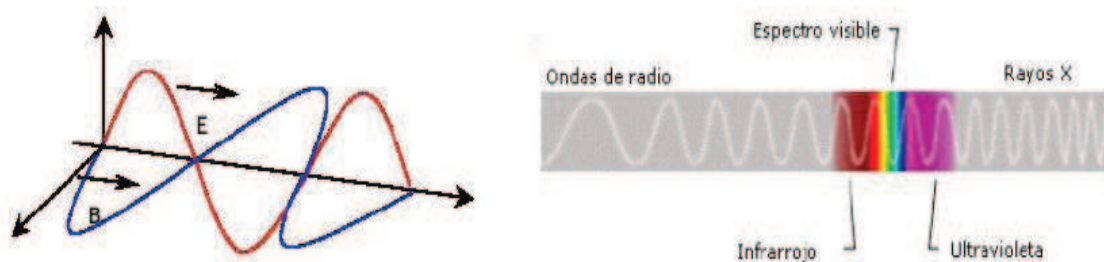


Figura 1.12 a) Campo electromagnético, b) Espectro de la luz, tomado de [5].

1.5.1 CONFIGURACIÓN BÁSICA DE UN SISTEMA DE VISIÓN ARTIFICIAL

Los dos pilares de un sistema físico de visión artificial son: el sistema de formación de las imágenes y el sistema de procesamiento de éstas. El primer sistema está constituido por el subsistema de iluminación, de captación de la imagen y de adquisición de la señal en el computador. Mientras que el segundo, está constituido por el sistema de procesamiento digital de la imagen dentro del computador.

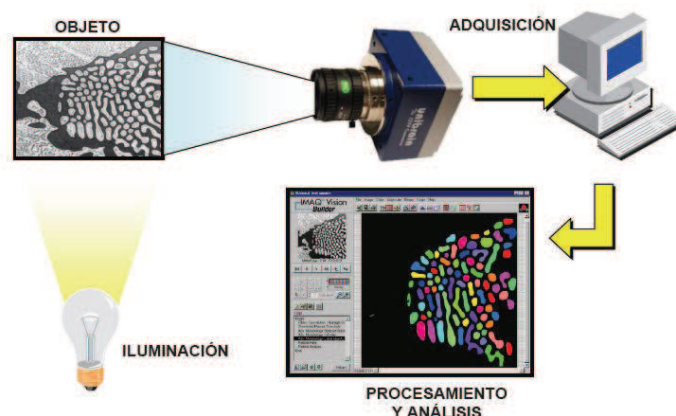


Figura 1.13 Subsistemas físicos de un equipo de visión artificial, tomado de [6].

1.5.1.1 Etapas Básicas de una Aplicación en Visión Artificial

Toda aplicación de Visión Artificial tiene sus especificidades, pero se puede decir que existe un tronco común de etapas entre ellas. No necesariamente debe cubrirse todas en una implementación concreta. En ciertos casos únicamente se tiene un subconjunto de las etapas que se van a citar. Para facilitar la comprensión, se asumirá que las diferentes etapas tienen una secuencia, pero

esta simplificación no es real, ya que en la práctica siempre se encuentra realimentación entre las distintas etapas [5].

La primera etapa es la construcción del sistema de formación de las imágenes. Su objetivo es realzar, mediante técnicas fotográficas (iluminación, óptica, cámaras, filtros, pantallas, etc.) las características visuales de los objetos (formas, texturas, colores, etc.) [5]. El éxito de muchas aplicaciones depende de un buen diseño en esta primera etapa.

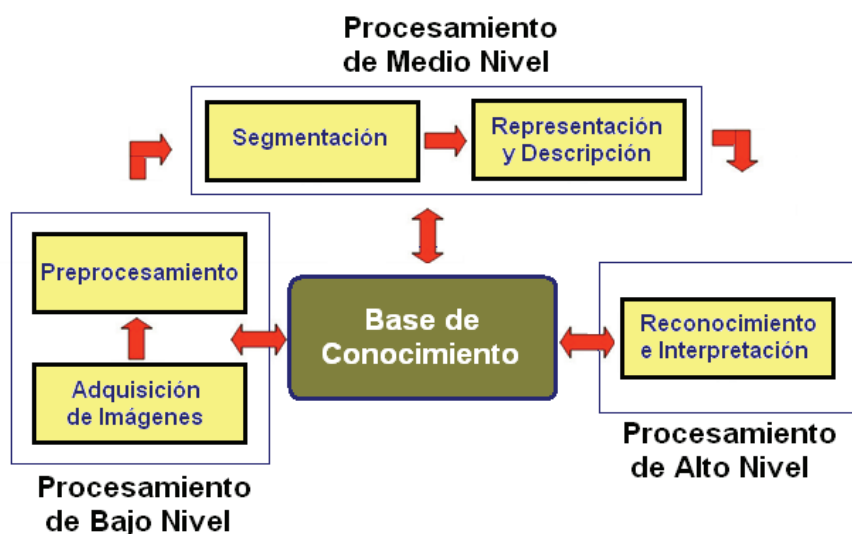


Figura 1.14 Etapas de una aplicación de Visión Artificial

Una vez adquirida la imagen se pasa a la etapa de Pre-procesamiento cuyo objetivo es mejorar la calidad informativa de la imagen adquirida. Se incluyen operaciones de mejora de la relación señal-ruido (denoising), SNR, de atenuación de imperfecciones de la adquisición debido a la función de transferencia del sistema de captación de imágenes (deconvolution), de regularización de la imagen, de mejora de contraste o de optimización de la distribución de intensidad (enhancement) o de realce de algunas características de la imagen, como bordes o áreas [5].

Luego se pasa a la Segmentación, que es la fase donde se particiona la imagen en áreas con significado [5]. Por ejemplo, en una imagen de satélite se determina las zonas de agua, de cultivo, urbanas, carreteras, etc. Existen varias técnicas:

umbralizaciones, discontinuidades, crecimiento de regiones, uso del color o de movimiento, etc.

Una vez dividida la imagen en zonas con características de más alto nivel se pasa a la extracción de sus características. Básicamente son de tipo morfológico, tales como área, perímetro, excentricidad, momentos de inercia, esqueletos, pero también se puede emplear características basadas en la textura o en el color. Luego del procesamiento se ha pasado de una información visual primaria a algo más elaborado. Con las características analizadas de cada región se debe clasificar e interpretar esta información. Por tanto, se diseñarán algoritmos clasificadores que le den a cada área segmentada una etiqueta de alto nivel, como por ejemplo, en una imagen aérea qué zonas son tierras de cultivo, áreas urbanas, etc.

1.6 PROCESAMIENTO DIGITAL DE IMÁGENES

Una vez que se ha adquirido la imagen con el sistema de Visión Artificial, el siguiente paso es procesar la información para obtener una imagen digital que se presenta en forma de matriz. Esta etapa de procesado de bajo nivel, se conoce como pre-procesamiento y pretende mejorar la imagen o bien destacar algún atributo primario; para ello se realizan ciertas operaciones sobre la imagen, que permiten corregir posibles fallas del sistema de iluminación, eliminar ruido, aumentar contraste, realzar bordes, acentuar texturas, etc.

Las imágenes digitales son obtenidas por el proceso de muestreo y cuantificación de la imagen adquirida. El 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 un píxel, lo que indica que una imagen digital es un arreglo bidimensional de píxeles.

1.6.1 ETAPA DE PRE-PROCESAMIENTO

Como ya se mencionó anteriormente, la etapa de pre-procesamiento, tiene como objetivo mejorar la calidad de la imagen, o bien realzar algún atributo primario de la misma, de tal forma que la información deseada de la imagen adquirida no se

perda. La imagen adquirida por el sistema de visión artificial, puede tener varios formatos como por ejemplo escala de grises de 8, 16 o 32 bits, imagen a color RGB, etc.

En general en un sistema de visión por computadora, se procesan imágenes en escala de grises, pero no siempre la cámara utilizada permite adquirir una imagen en este formato, por lo cual se hace necesaria la conversión a un formato adecuado.

1.6.1.1 Representación de una Imagen en Escala de Grises

Para la representación de una imagen en escala de grises, se utiliza la luminancia de cada píxel, y se la coloca dentro de una escala que depende del tamaño de la imagen; así por ejemplo, una imagen representada en escala de grises de 8 bits, tendrá píxeles cuya luminancia oscila entre 0 y 255.

Una vez que la imagen ha sido representada en escala de grises, se puede realizar sobre la misma diversos análisis que permiten extraer únicamente la información de interés. El análisis más básico que se puede realizar a este nivel, es la extracción del *Histograma* de la imagen. El Histograma es una función, que indica con qué frecuencia se repite cada nivel de gris dentro de la imagen [7]. Generalmente, el histograma de una imagen se presenta como una gráfica de frecuencia, como la mostrada en la Figura 1.15.

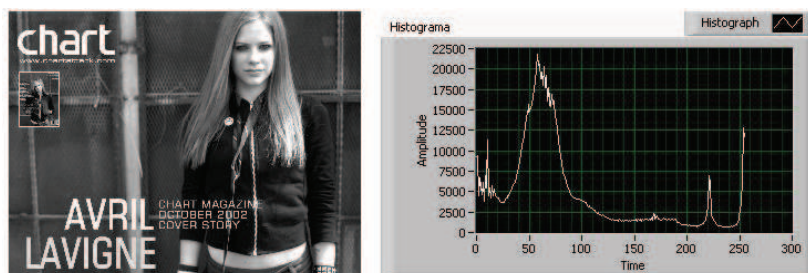


Figura 1.15 Histograma de una Imagen en Escala de Grises

Para el caso de imágenes RGB, es posible obtener un histograma de cada color (rojo, azul y verde), o bien extraer la luminancia de la imagen, y luego obtener el histograma para escala de grises.

Mediante el histograma se obtiene una descripción general de la imagen, que ayuda a identificar las zonas con mayor contenido de información, y también zonas que pueden ser catalogadas como ruido.

Dentro de la etapa de pre-procesamiento, también se pueden incluir operaciones de filtrado y mejoramiento de imagen, como son filtros espaciales o de convolución. Los filtros más utilizados son filtros lineales de convolución por *Gradiente, Laplaciano, Gauss, Suavizado*; y filtros no lineales como los de *Prewitt, Roberts, Sobel, Sigma, Pasa-bajos*.

1.6.2 ETAPA DE SEGMENTACIÓN

La segmentación es un proceso que permite extraer objetos o información de interés de la imagen. En esta etapa se trata de agrupar los píxeles en base a algún criterio de homogeneidad como geometría, área, centro de masa, etc. para poder segmentar la imagen en regiones de interés. En un sistema de visión artificial, generalmente se realiza la segmentación de imágenes binarias por ser su procesamiento más rápido, debido a que existen solo dos intensidades (0 y 1).

1.6.2.1 Representación de una Imagen en formato Binario

Mediante el histograma se puede obtener la distribución de la imagen en cuanto a cada valor de gris (0: negro; 255: blanco). Luego mediante umbrales se puede representar la imagen como una función discreta $a [m, n]$ que puede tomar valores '0' o '1' dependiendo del nivel de gris de cada píxel. A esta representación se le llama imagen *binaria*.

Generalmente se definen dos umbrales: superior e inferior; para definirlos correctamente se utiliza el histograma, de tal forma que dentro de los límites de los umbrales superior e inferior este contenida la mayor distribución de la imagen. Una vez hecho esto se representa como '1' a los píxeles que tengan un valor de gris dentro de los límites, y como '0' a los píxeles que tengan un valor en gris, fuera de estos.

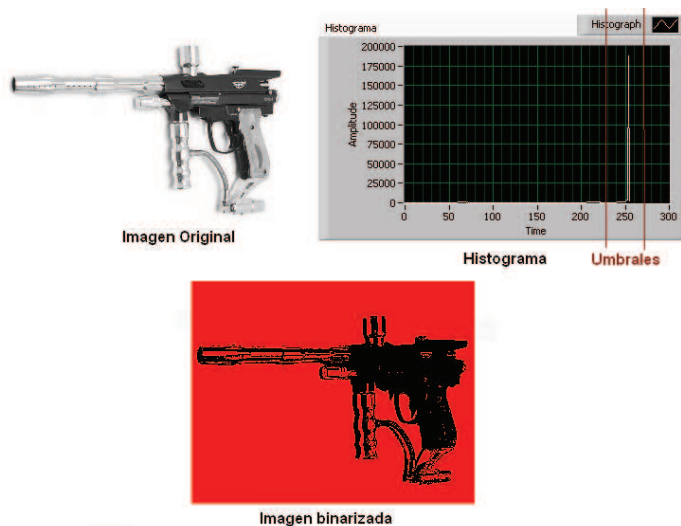


Figura 1.16 Imagen binaria

Una vez que la imagen haya sido particionada, la unidad de la misma dejará de ser el píxel, para ser ahora la agrupación de píxeles que forman un objeto, muchas veces llamada *BLOB (Binary Large Object)*.

La etapa de segmentación puede llegar a ser bastante compleja dependiendo de la imagen adquirida, nivel de iluminación, información deseada, etc. por lo que en la actualidad los métodos de segmentación están en constante desarrollo; sin embargo se pueden resaltar dos procedimientos principales dentro de esta etapa:

- Extracción de bordes o localización de las fronteras geométricas de los objetos.
- Búsqueda de regiones homogéneas en función de criterios de similitud.

A continuación se describirán tres de los métodos más comunes de segmentación:

- Segmentación por detección de bordes
- Segmentación basada en Morfología
- Segmentación basada en modelos (Transformada de Hough)

1.6.2.2 Segmentación por detección de bordes

El borde o contorno puede ser catalogado como una región donde existe una fuerte variación del nivel de intensidad en los píxeles adyacentes. Su causa principal es originada por la intersección de varios objetos, con diferentes niveles de reflectancia, que al ser proyectados sobre la cámara generan discontinuidades de intensidad en los píxeles correspondidos. Sin embargo, éstas discontinuidades también aparecen de forma no deseada por la presencia del ruido, por efecto de sombras en los objetos, etc. [5].

El fundamento para la extracción de bordes es aplicar el operador derivada de primer o segundo orden en un entorno de vecindad. Al analizar una fila de una imagen aplicando la primera derivada, esta tendrá un valor mínimo o máximo justamente cuando se esté en el punto de inflexión del borde; y en caso de aplicar la segunda derivada, los puntos de inflexión de la primera derivada, serán cruces por cero. Estos dos criterios son los más aplicados para la detección de bordes.

Para el caso de las imágenes reales no binarias, no se obtienen buenos resultados al aplicar los criterios antes propuestos; en este caso, se usan funciones más avanzadas como operaciones basadas en el operador gradiente, en el Laplaciano, Laplaciano de Gauss o el operador de Canny.

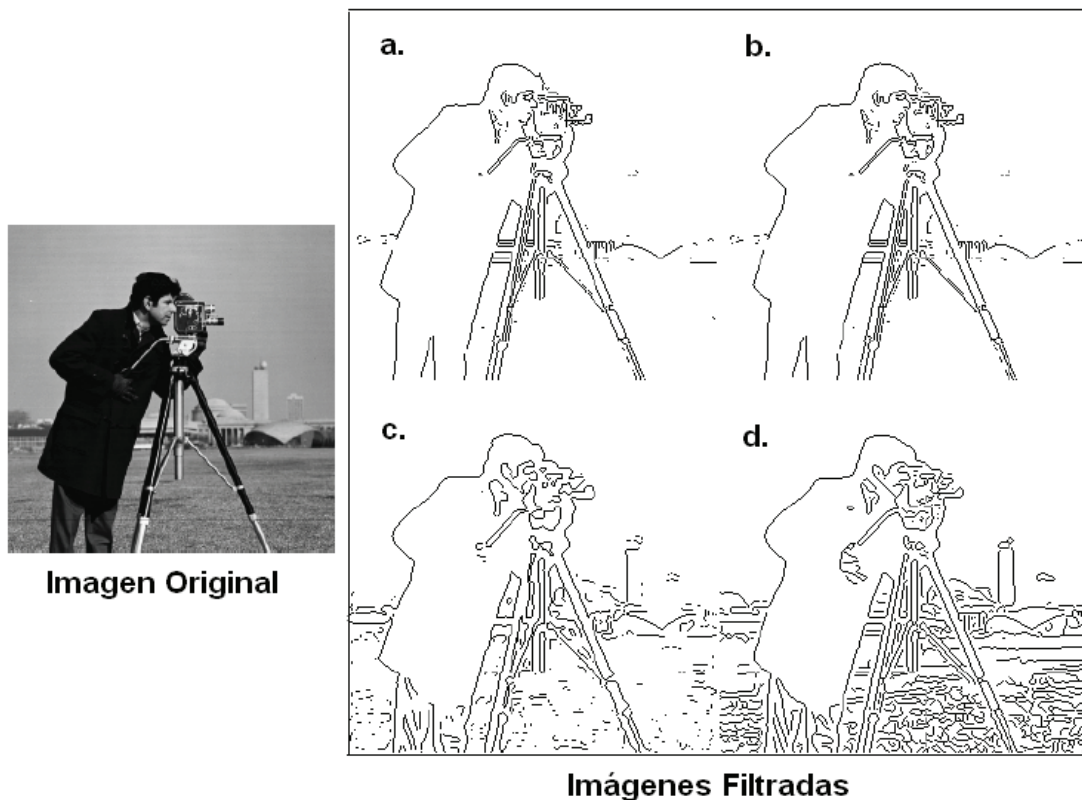


Figura 1.17 Detección de Bordes: a) Prewitt, b) Sobel, c) LOG, d) Canny, tomado de [5]

Todos estos métodos también pueden usarse en imágenes binarias, brindando excelentes resultados. A continuación se describen brevemente algunos de estos métodos.

1.6.2.2.1 *Técnicas basadas en el operador Gradiente*

El operador gradiente aplicado sobre un píxel (x, y) de la imagen, retorna un vector que indica la dirección de máxima variabilidad de la intensidad luminosa y nivel de variación [5]. Las técnicas que se basan en el operador gradiente son el filtro de Prewitt, de Sobel y el operador isotrópico o de Frei-Chen.

1.6.2.2.2 *Operadores basados en el Laplaciano*

Para la detección de bordes utilizando el Laplaciano, se consideran los píxeles que pasen por cero. A diferencia del operador gradiente, el Laplaciano no retorna un vector de intensidad sino un escalar, el cual será nulo en un área con intensidad homogénea; y por el contrario, tendrá valores positivos o negativos al aplicarlo sobre un borde [5].

1.6.2.2.3 Operador de Canny

Este algoritmo es ampliamente utilizado en localización de contornos. Se caracteriza por evitar la ruptura de los bordes de los objetos. Su fundamento se basa en un proceso de maximización teniendo en cuenta los siguientes objetivos a maximizar [5]:

- Aumentar la relación señal-ruido de la imagen.
- Disminuir todo lo posible la distancia entre el borde detectado y el borde real.
- No identificar un borde por un único píxel, sino por un conjunto de píxeles que tengan cierta conectividad.

1.6.2.3 Segmentación basada en Morfología

El procesamiento morfológico de la imagen se basa en el realce de la geometría y forma de los objetos. Se realiza en base a una serie de transformaciones morfológicas. Su fundamento matemático se basa en la teoría de conjunto.

El objetivo de las transformaciones morfológicas es la extracción de estructuras geométricas en los conjuntos sobre los que se opera, mediante la utilización de otro conjunto de forma conocida, al que se le denomina elemento estructurante. El tamaño y forma del elemento estructurante se elige, a priori, de acuerdo con la morfología sobre la que va a intersecar y en función de las formas que se desea extraer. Muchas veces el proceso morfológico es utilizado luego de la segmentación por extracción de bordes. Antes de abordar las transformaciones morfológicas más comunes, es necesario primero tener claro el concepto de *conectividad* entre los píxeles de una imagen.

1.6.2.3.1 Conectividad

La conectividad define la forma en que los píxeles de una imagen están interconectados. Este concepto se aplica generalmente para imágenes binarias. Por otro lado, la vecindad, define una agrupación de píxeles, *blob* o etiqueta, según el autor.

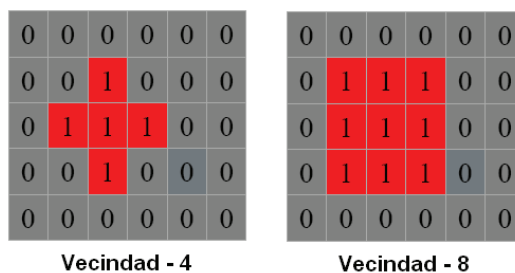


Figura 1.18 Tipos de Conectividad, tomado de [6].

Existen dos tipos de vecindad: 4 y 8. En una malla rectangular de píxeles, vecindad 4 considera a píxeles conectados en direcciones perpendiculares, es decir, arriba, abajo, izquierda y derecha, del píxel central. En vecindad 8, la consideración se extiende también a los píxeles de las diagonales. Si un píxel del vecindario del píxel central, y también el píxel central tienen valor '1', ya sea con vecindad 4 u 8, se puede decir que existe un blob o etiqueta.

La conectividad, establece una conexión o camino entre agrupaciones, BLOBs o etiquetas. Una vez definida esta particularidad, se puede analizar las principales operaciones o transformaciones morfológicas, las cuales afectan la forma de los objetos facilitando su posterior análisis.

1.6.2.3.2 Erosión

Reduce el tamaño de los objetos en la imagen. Una operación de *erosión* remueve capas de píxeles alrededor de los objetos en la imagen. El número de capas removidas depende del tamaño del elemento estructurante utilizado en la operación. Un elemento estructurante de 3x3 remueve una capa de píxeles; uno de 5x5 remueve dos capas, etc. La *erosión* elimina píxeles aislados y remueve penínsulas de pequeño tamaño; es decir, la operación de *erosión* elimina ruido de la imagen.

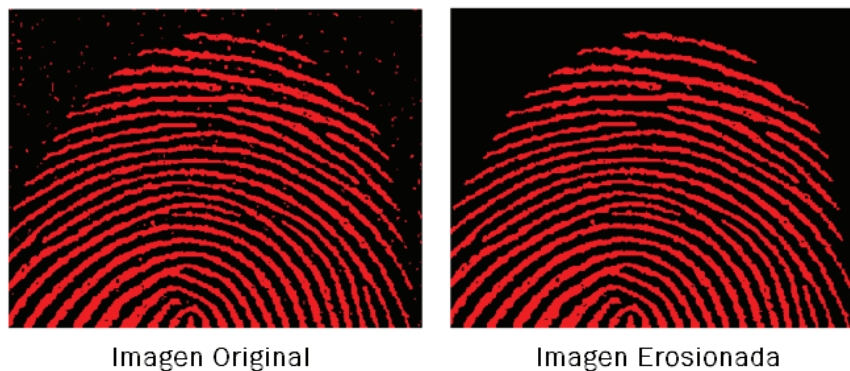


Figura 1.19 Operación de Erosión

1.6.2.3.3 Dilatación

Incrementa el tamaño de los objetos en la imagen. Una operación de *dilatación* aumenta una capa de píxeles alrededor de los objetos, incluyendo objetos que tengan agujeros; es decir, que una operación de dilatación, rellena agujeros presentes en los objetos de la imagen. El número de capas de píxeles añadidos depende del tamaño del elemento estructurante.

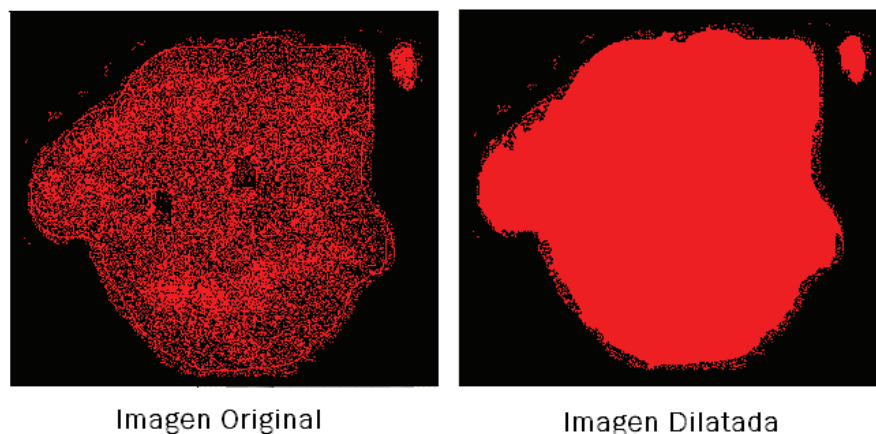


Figura 1.20 Operación de Dilatación

1.6.2.3.4 Apertura y Cierre

Una operación de *apertura*, realiza una *erosión* seguida de una *dilatación*. Una operación de *apertura* suaviza los contornos de los objetos y elimina pequeñas uniones entre vecindarios; es decir, remueve pequeños objetos y franjas, pero no remueve agujeros.

Una operación de *cierre*, realiza una *dilatación* seguida de una *erosión*. Una operación de *cierre* también suaviza contornos, además cierra agujeros y rellena discontinuidades en los contornos; es decir, remueve agujeros pero no remueve pequeños objetos o penínsulas.

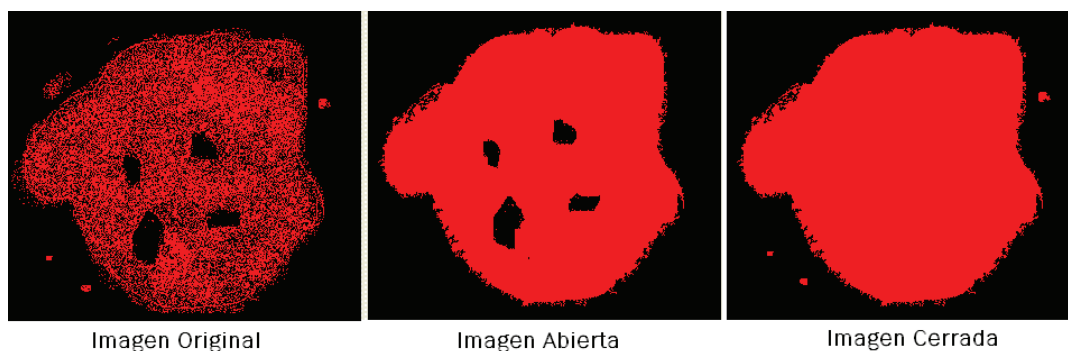


Figura 1.21 Operaciones de Apertura y Cierre

Ambas operaciones *apertura* y *cierre* no necesariamente afectan el tamaño del objeto. Estas operaciones se usan cuando se desea remover pequeños objetos o agujeros, que generalmente son ruido en la imagen. Un mejor resultado se obtiene cuando se usa una secuencia de operaciones de *apertura-cierre*, o *cierre-apertura*.

La segmentación basada en la morfología, normalmente sigue los siguientes pasos, no necesariamente en el orden descritos.

- Adquirir imagen
- Binarizar la imagen aplicando umbrales apropiados
- Remover pequeñas partículas de la imagen
- Remover penínsulas que estén en contacto con el borde de la imagen. (El borde de la imagen es el recuadro, no un contorno de un objeto).
- Eliminar agujeros en los objetos [7]

Luego de estos pasos, se procede a obtener información de medio nivel de la imagen, como área, perímetro, distancia, etc. en los objetos de la imagen segmentada.

1.6.2.4 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. [8]. Dentro de este tipo de segmentación se encuentra la *Transformada de Hough*, que es un método muy difundido que se aplica en identificación de huellas, rasgos faciales, iris, control de calidad, etc.

1.6.2.4.1 Transformada de Hough

Este algoritmo trata de detectar formas geométricas sencillas en una imagen. Es bastante robusta frente al ruido. Para su ejecución requiere como entrada una imagen binaria, en la que previamente se ha realizado detección de bordes. Hough permite extraer formas de alto nivel como son líneas, circunferencias, elipses o cualquier tipo de curva [5]. El único inconveniente de utilizar este método, es el alto coste computacional que requiere su procesamiento.

1.6.2.4.2 Método General para Implementar la Transformada de Hough

Para determinar la transformada de Hough en una imagen, se realizan los siguientes pasos:

- Determinar la ecuación correspondiente al rasgo que se desea buscar, así como los parámetros en función de los que se define.
- Destacar los elementos de la imagen original que dan forma al rasgo buscado, como pueden ser puntos o bordes.
- Discretización del espacio de búsqueda.
- Calcular la transformada de Hough.

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

A continuación se muestra un ejemplo gráfico de la transformada de Hough.

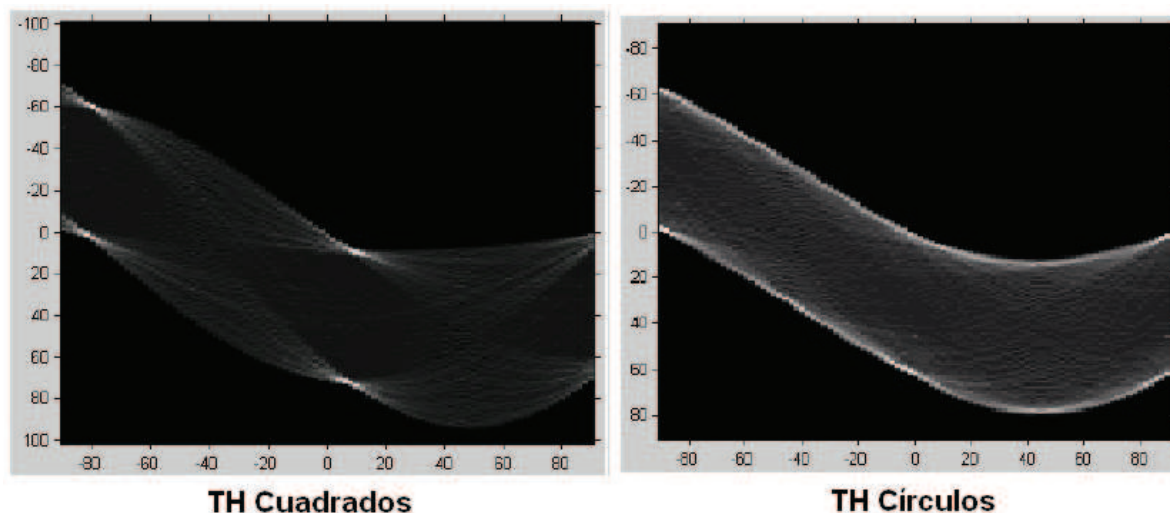


Figura 1.22 Transformada de Hough de figuras comunes, tomado de [8]

1.6.3 ETAPA DE REPRESENTACIÓN Y DESCRIPCIÓN

Luego de haber procesado la imagen y haberla tratado de tal forma que se puedan realizar las características o información de interés que contiene, el siguiente paso es extraer esta información, que generalmente es de tipo morfológico como por ejemplo: área, perímetro, excentricidad, esqueletos; o bien otras características como textura, color, conteo de objetos, etc.

En esta etapa, se analiza la información obtenida de la etapa de segmentación, y los algoritmos de análisis dependen intrínsecamente de la aplicación, por lo que no es posible generalizar un modo de desarrollar esta etapa, sino indicar algunas de las operaciones comúnmente desarrolladas para la extracción de información.

1.6.3.1 Búsqueda de Objetos

Se emplea generalmente en búsqueda de coincidencias o patrones.



Figura 1.23 Búsqueda de Objetos, tomado de [6]

1.6.3.2 Medición de Objetos

Permite obtener información de las dimensiones de los objetos presentes en la imagen.

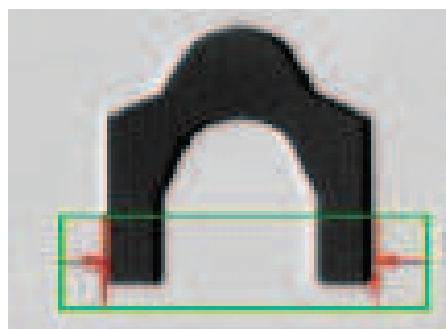


Figura 1.24 Medición de Objetos, tomado de [6]

También se puede medir área, perímetro, posición de centro de masa, distancia; basta con utilizar la herramienta adecuada en el software con el que se trabaje, y utilizar un algoritmo conveniente.

1.6.4 ETAPA DE RECONOCIMIENTO E INTERPRETACIÓN

En esta etapa se utiliza la información obtenida en la etapa anterior, para ser utilizada en el algoritmo principal de aplicación del sistema de visión artificial. Este algoritmo depende de la aplicación.

En capítulos posteriores se abordará nuevamente la etapa de reconocimiento e interpretación para explicar el modo de trabajo del software de visión artificial desarrollado mediante el Toolkit '*Vision and Motion*' de LabVIEW.

CAPÍTULO 2

DISEÑO Y CONSTRUCCIÓN DEL PROTOTIPO

El prototipo de manipulador que se ha construido, tiene fundamentalmente dos etapas de diseño bien diferenciadas: Diseño Mecánico y Diseño Electrónico. Mediante estos dos tipos de diseño, se puede crear un prototipo virtual antes de proceder a la construcción del manipulador real; es decir, se puede diseñar una máquina virtual y probar su desempeño, ventajas, desventajas, errores, antes de proceder a su construcción, lo cual da la oportunidad de solucionar posibles contrariedades volviendo al diseño, en lugar de tener problemas con la máquina ya construida.

Para la obtención del modelo virtual del manipulador, se ha utilizado un software de diseño CAD, y para la simulación de su desempeño se ha utilizado al mismo software de diseño CAD, o alternativamente el Toolkit de Mecatrónica de LabVIEW 8.6.

2.1 DISEÑO MECÁNICO

Para la etapa de diseño mecánico, se ha utilizado el software de diseño CAD: *SOLIDWORKS* y sus complementos *COSMOSWorks* y *COSMOSMotion*. Mediante estas herramientas se puede realizar un modelo virtual del manipulador, analizar su desempeño, resistencia, movimiento, etc. lo cual permite validar el diseño mediante simulación. A continuación se da una breve explicación de las herramientas mencionadas; así como también se presentan los diseños y análisis realizados para el manipulador robótico construido.

2.1.1 SOLIDWORKS

Este software de diseño CAD, como ya se mencionó anteriormente, permite realizar modelos virtuales de sistemas mecánicos, para ello cuenta con herramientas de diseño y croquizado en dos y tres dimensiones. Adicionalmente, es posible realizar diferentes operaciones de conformado mecánico como son: Extrusión, Corte, Redondeos, Chaflanes, Sólidos de Revolución, Vaciado, etc. Una vez dibujado el croquis en el espacio de trabajo de *SOLIDWORKS*, se puede

realizar cualquiera de estas operaciones, teniendo en cuenta que el croquis dibujado debe ser cerrado, y congruente con el tipo de operación a realizar. Luego de realizar una operación, se puede realizar nuevos croquis 2D o 3D sobre la pieza ya diseñada, y realizar nuevas operaciones sobre la misma; por ejemplo es posible realizar una extrusión, y luego sobre la pieza extruida, realizar un corte, o una nueva extrusión. A continuación se muestran ejemplos de croquis y posteriores operaciones.

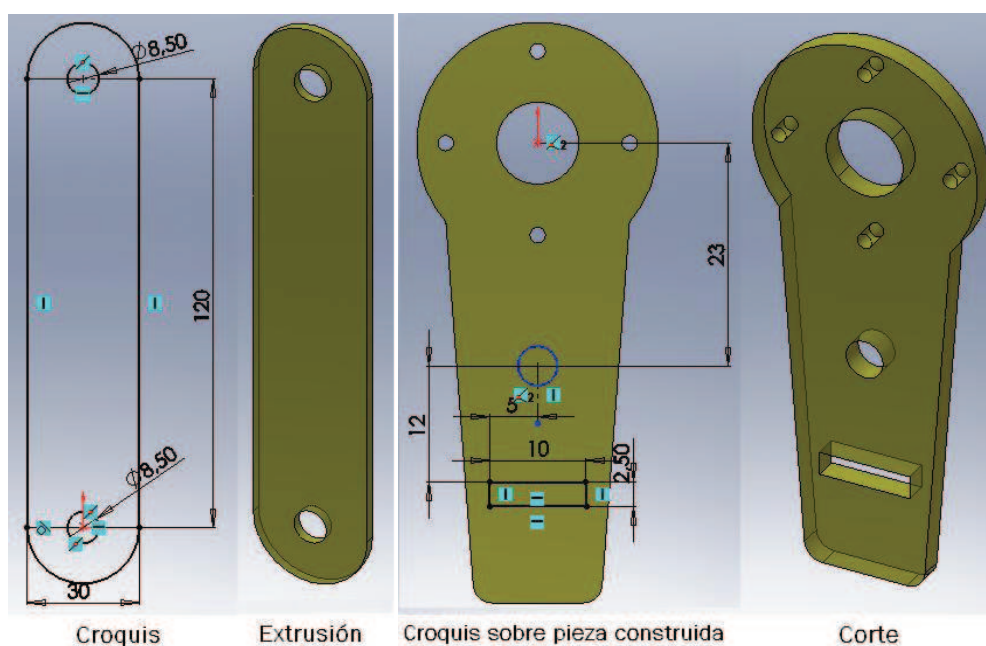


Figura 2.1 Ejemplos de Croquis y Operaciones

En cada una de las piezas que se diseñen, se puede escoger el material del cual estarán formadas, color, textura para brindar a la simulación más herramientas que la acerquen a la realidad.

Las piezas generadas, posteriormente pueden agruparse y formar parte de un *ensamble* que represente finalmente una estructura más elaborada. Para formar un ensamble se debe colocar relaciones de posición entre las piezas que forman el ensamble; por ejemplo, relaciones de perpendicularidad, paralelismo, tangencia, concéntricas, etc.

Mediante la correcta aplicación de relaciones de posición entre las diferentes piezas que forman el ensamble se puede llegar a una estructura final que tenga

las propiedades adecuadas. A continuación se muestra un ejemplo de ensamble en donde se muestran primero todos los componentes del ensamble dispersos, y luego formando la estructura final.



Figura 2.2 Ejemplo de ensamble en SOLIDWORKS

Para construir mecanismos más avanzados, se puede ensamblar no solo piezas sino también ensambles previamente realizados; es decir, se puede realizar ensamblajes de ensambles. A continuación se muestra un ejemplo de ensamble que contiene ensambles previos.

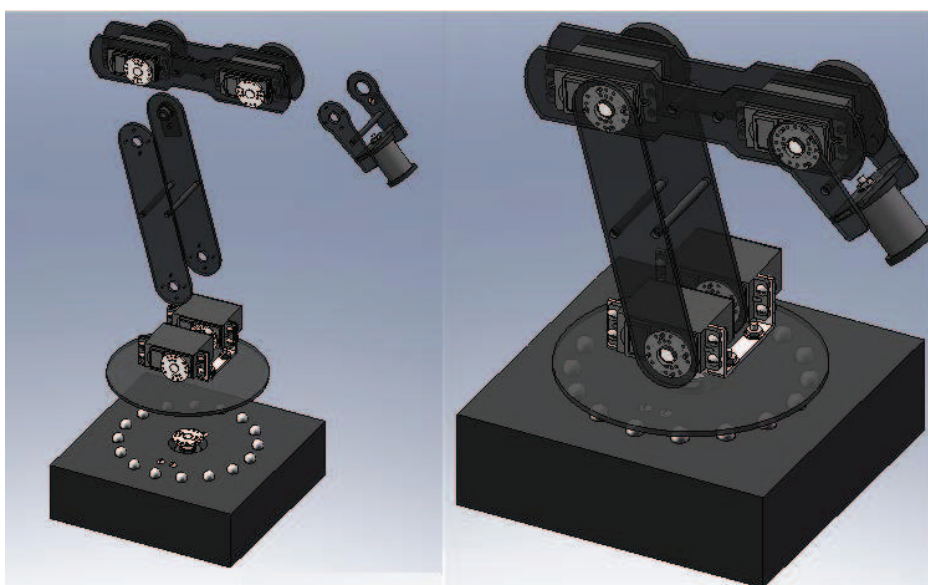


Figura 2.3 Ensamblaje que contiene ensambles previos

2.1.2 DISEÑO REALIZADO EN SOLIDWORKS PARA EL ROBOT

A lo largo del presente trabajo se citarán en varias ocasiones a las diferentes partes que componen al robot, por lo que es conveniente definir las con anterioridad.

- Las partes que forman la estructura del robot, es decir, los ensambles; se denominarán como eslabones.
- Las articulaciones que dan movimiento a cada eslabón, se definirán como: base, hombro, codo y muñeca en función del eslabón al cual den movimiento.

Las diferentes piezas que componen al sistema completo del manipulador industrial, fueron diseñadas en SOLIDWORKS y posteriormente construidas. A continuación se presentan cada una de las piezas diseñadas, así como los ensambles de cada eslabón. Los planos de cada una de las piezas diseñadas, se presentan en el ANEXO B.

El material utilizado para la construcción es Acrílico de 2,5mm de espesor.

En los siguientes análisis y diseños, se han incluido también a los servomotores que dan movimiento a los eslabones, para lo cual se ha considerado el peso máximo que puede tener un servomotor de las dimensiones escogidas (47gr).

2.1.2.1 Piezas que conforman la base

La base del robot está compuesta por una caja metálica construida en platina de hierro y chapa metálica, la cual servirá de contrapeso para brindar equilibrio al robot. Sobre la superficie de la caja metálica se han realizado perforaciones avellanadas para situar esferas metálicas de 5/16" que dan soporte al eje del motor de la base en su movimiento, ya que permiten que el peso del robot se apoye sobre las esferas evitando la deformación del eje del motor. El motor de la base va dentro de la caja metálica, y únicamente su eje sobresale de la misma.

Mediante un ensamblaje se colocan todas las piezas juntas para formar la base, con la ayuda de tornillos, tuercas, arandelas y demás elementos de unión y sujeción.



Figura 2.4 Piezas que conforman la base



Figura 2.5 Ensamble final de la base

2.1.2.2 Piezas que conforman el primer eslabón

El primer eslabón está compuesto por una plataforma que se monta sobre el eje del servomotor de la base, adicionalmente, sobre esta plataforma se montan los dos servomotores que dan movimiento al segundo eslabón. Estos servomotores

representan el hombro, y están fijados a la plataforma mediante dos piezas que sirven de acople.

Mediante un ensamblaje se colocan todas las piezas juntas para formar el primer eslabón, con la ayuda de tornillos, tuercas, arandelas y demás elementos de unión y sujeción.

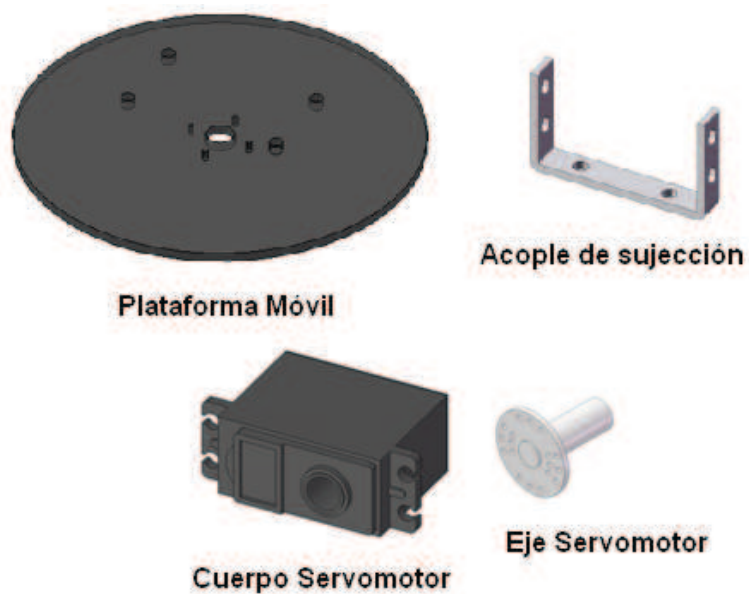


Figura 2.6 Piezas que conforman el primer eslabón

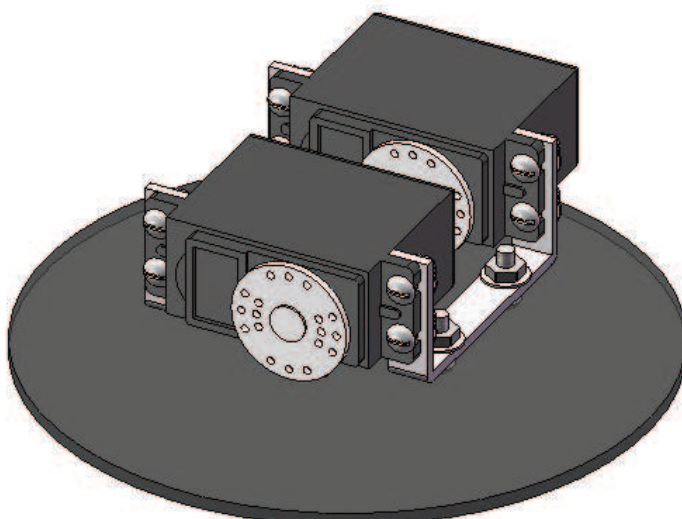


Figura 2.7 Ensamble final del primer eslabón

2.1.2.3 Piezas que conforman el segundo eslabón

El segundo eslabón está compuesto por dos perfiles que dan forma al eslabón. Estos perfiles están unidos mediante dos tornillos de sujeción; adicionalmente, sobre uno de los perfiles va montada la parte fija del acople que facilita el montaje del servomotor para el siguiente eslabón.

Mediante un ensamblaje se colocan todas las piezas juntas para formar el segundo eslabón, con la ayuda de tornillos, tuercas, arandelas y demás elementos de unión y sujeción.



Figura 2.8 Piezas que conforman el segundo eslabón



Figura 2.9 Ensamble final del segundo eslabón

2.1.2.4 Piezas que conforman el tercer eslabón

El tercer eslabón está formado por dos perfiles que dan forma al eslabón; estos perfiles, tienen agujeros para alojar dos servomotores. El servomotor de la parte inferior da movimiento al tercer eslabón; es decir, este motor representa al codo. El servomotor de la parte superior da movimiento al cuarto eslabón; es decir, este motor representa la muñeca. Tanto en la parte inferior como en la superior, hay acoples que facilitan el montaje de los demás eslabones.

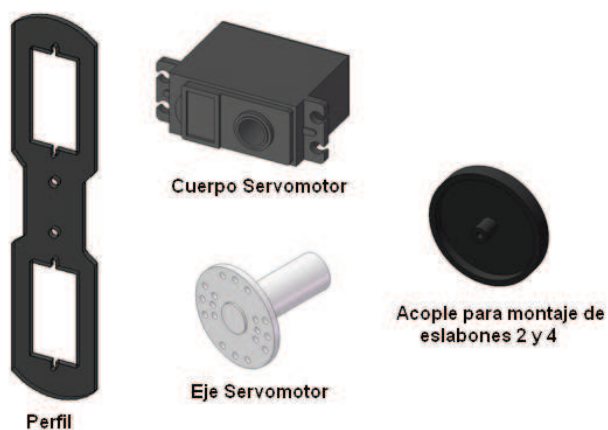


Figura 2.10 Piezas que conforman el tercer eslabón

Mediante un ensamblaje se colocan todas las piezas juntas para formar el tercer eslabón, con la ayuda de tornillos, tuercas, arandelas y demás elementos de unión y sujeción.

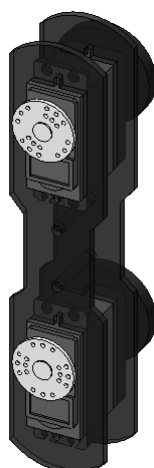


Figura 2.11 Ensamblaje final del tercer eslabón

2.1.2.5 Piezas que conforman el cuarto eslabón

El cuarto eslabón está formado por dos perfiles que dan forma al eslabón; adicionalmente, este eslabón lleva el electroimán que es el efector final del manipulador industrial. Uno de los perfiles lleva un acople que facilita el ensamble de este eslabón con el tercer eslabón.



Figura 2.12 Piezas que conforman el cuarto eslabón

Mediante un ensamblaje se colocan todas las piezas juntas para formar el cuarto eslabón, con la ayuda de tornillos, tuercas, arandelas y demás elementos de unión y sujeción.

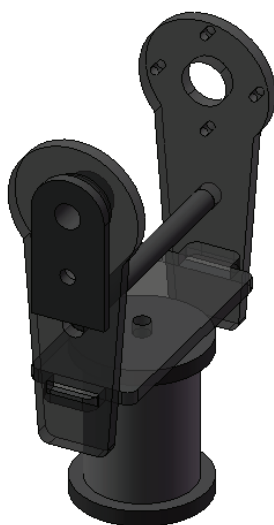


Figura 2.13 Ensamblaje final del cuarto eslabón

Finalmente, luego de haber realizado de forma individual el ensamble de cada eslabón, se procede a desarrollar el ensamble del sistema mecánico completo; es decir, un ensamble de ensamblajes previos.

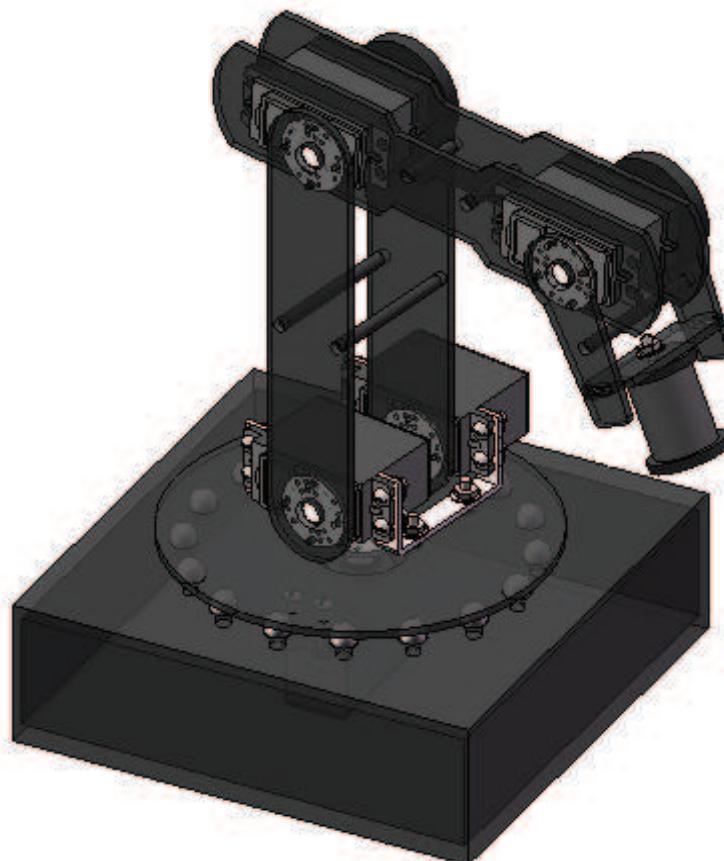


Figura 2.14 Ensamble del robot completo

Una vez concluido este proceso de diseño, se pueden realizar diversos análisis sobre la estructura construida. Estos análisis principalmente son de esfuerzo y movimiento, como se verá brevemente más adelante.

2.1.3 COSMOSWorks

Este complemento de SOLIDWORKS, permite realizar un análisis de esfuerzos mediante la simulación de fuerza, presión, restricciones, torques, etc. aplicados al mecanismo diseñado. Este análisis puede realizarse en una pieza o en un ensamble. Para activar el complemento COSMOSWorks, se accede al menú Herramientas/Complementos y en la ventana que se muestra se activa la caja de verificación correspondiente a COSMOSWorks.

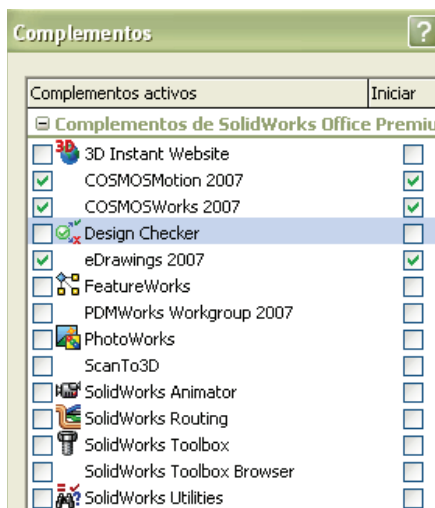


Figura 2.15 Ventana de activación de complementos: COSMOSWorks

Luego de activar COSMOSWorks, dentro del gestor de diseño de SOLIDWORKS, se tiene un nuevo ícono que corresponde al estudio de esfuerzos. Para realizar un estudio de este tipo, se debe hacer clic derecho en la ventana de estudio de esfuerzos sobre el nombre de la pieza o ensamble, y escoger *Estudio*; de esta forma se crea un nuevo estudio que presenta las siguientes opciones.

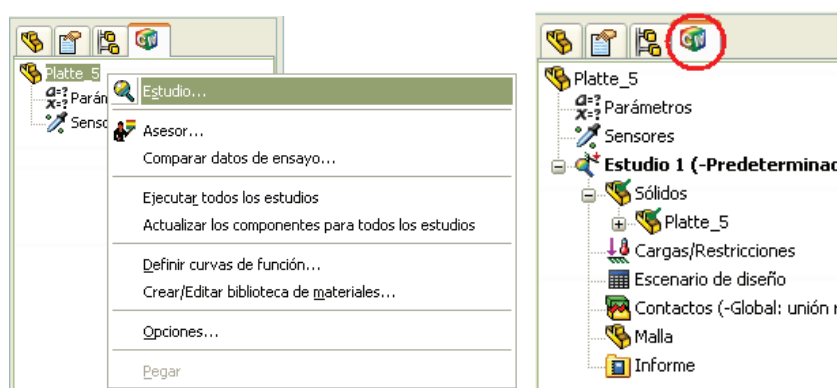


Figura 2.16 Creación de Estudio de Esfuerzos

Sólidos: muestra todos los sólidos que forman parte del archivo (pieza o ensamble), y permite definir el material de cada uno de ellos.

- **Cargas/Restricciones:** define el tipo de cargas a aplicar; sobre qué partes se aplican las cargas; y también qué partes son fijas.



Figura 2.17 Tipos de Cargas/Restricciones en COSMOSWorks

- *Contactos*: permite agregar restricciones de contacto entre piezas con movimiento relativo.
- *Malla*: define la malla o cuadrícula que divide al archivo gráfico, para poder realizar el análisis matemático.

Para poder realizar un análisis de esfuerzos, se deben seguir los siguientes pasos:

- Crear un nuevo estudio estático.
- Definir material de sólidos.
- Definir restricciones de partes fijas.



Figura 2.18 Definición de Restricciones: Partes fijas

Definir tipo, magnitud y ubicación de las cargas aplicadas. COSMOSWorks permite aplicar: fuerza, presión, torque, etc. sobre superficies planas, superficies curvas, puntos o aristas.

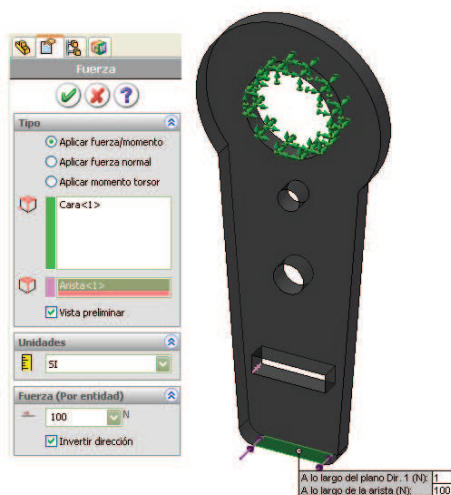


Figura 2.19 Definición de Cargas aplicadas

- Crear malla para el análisis matemático.

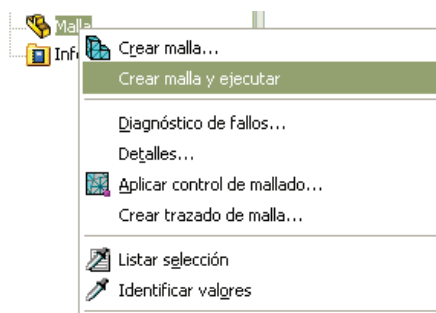


Figura 2.20 Crear malla y ejecutar análisis matemático

- Ejecutar análisis.
- Interpretar resultados.

Luego de ejecutar el estudio de esfuerzos, COSMOSWorks devuelve varios tipos de resultados, que brindan información completa que permite decidir si el diseño realizado soporta o no las condiciones simuladas en el estudio.

2.1.3.1 Criterio de von Mises

El Criterio de Máxima Tensión de von Mises, establece que un material dúctil empieza a ceder o cortarse en cualquier punto donde la tensión de von Mises sea igual al límite elástico del material [9]. Esto permite determinar el Factor de Seguridad (FDS) de la pieza o ensamble, tan solo dividiendo el valor del límite elástico del material para la tensión de von Mises. El límite elástico del material es una propiedad de todo material.

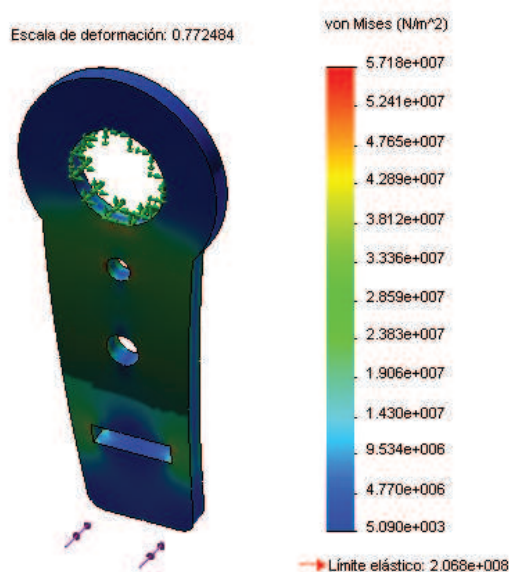


Figura 2.21 Criterio de Tensión Máxima de von Mises

2.1.3.2 Interpretación del Factor de Seguridad (FDS)

Es importante interpretar correctamente la información que COSMOSWorks proporciona.

- Un FDS inferior a 1,0 en una ubicación significa que el material de esa ubicación ha cedido y que el diseño no es seguro [9].

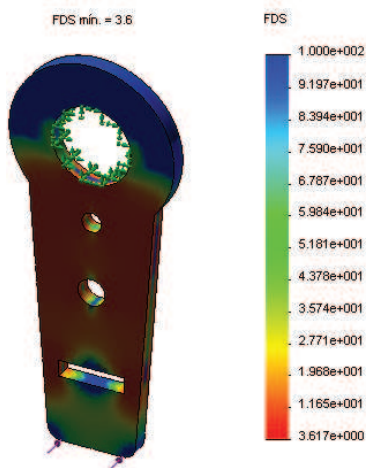


Figura 2.22 Factor de Seguridad (FDS) de un diseño

- Un FDS de 1,0 en una ubicación significa que el material de esa ubicación ha empezado a ceder [9].
- Un FDS superior a 1,0 en una ubicación significa que el material de esa ubicación no ha cedido; es decir, que el diseño es seguro [9].

Hay que garantizar que en toda ubicación del diseño, el FDS mínimo sea mayor a 1,0.

2.1.3.3 Análisis de Deformación Unitaria

La deformación unitaria es la proporción de cambio de longitud δL a la longitud original L . La deformación unitaria es una cantidad sin dimensión [9].

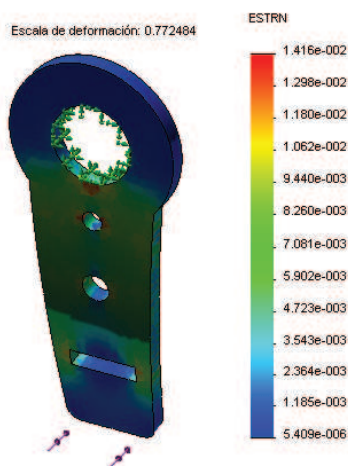


Figura 2.23 Análisis de Deformaciones Unitarias

COSMOSWorks realiza además otros análisis que pueden ser útiles dependiendo del tipo de información que se necesite. Estos tipos de análisis no serán abordados en el presente trabajo.

Para una correcta interpretación de resultados es importante tener en cuenta la escala de la gráfica mostrada, ya que generalmente la gráfica exagera la deformación para que esta pueda ser apreciada.

2.1.4 ANÁLISIS DE ESFUERZOS DEL ROBOT

En base a los ensambles realizados para cada eslabón, se puede realizar un análisis de esfuerzos aplicando una fuerza que simule las piezas que el efector final del robot va a levantar, y considerando el material y peso de cada uno de los componentes del ensamble. La mayoría de piezas que conforman el ensamble están construidas de acrílico, cuyo límite elástico es de $2,068 \text{ e}+008$.

El análisis de esfuerzo considera únicamente a las piezas que conforman la estructura del robot, mas no los motores que lo mueven, debido a que su dimensionamiento y análisis serán realizados en la parte de Diseño Electrónico.

A continuación se presentan los análisis realizados sobre cada uno de los ensambles. También es posible realizar estos análisis sobre cada pieza pero es más tangible un análisis sobre el ensamble, ya que considera las relaciones de posición de cada una de las piezas involucradas.

2.1.4.1 Análisis de Esfuerzos del Tercer Eslabón

La presentación del análisis empieza por el último eslabón, debido a que es el que menor esfuerzo debe realizar.

Para ejecutar este análisis, se define una fuerza en dirección de la gravedad que simula el peso máximo que el manipulador podría levantar, el cual es de 0,5kg. Las partes que soportan el esfuerzo son las caras internas de los perfiles que dan forma al eslabón. Sobre estas caras irán los ejes del movimiento, ya sea rodamientos o ejes de servomotor. Luego de realizar este estudio, los resultados obtenidos son los siguientes:

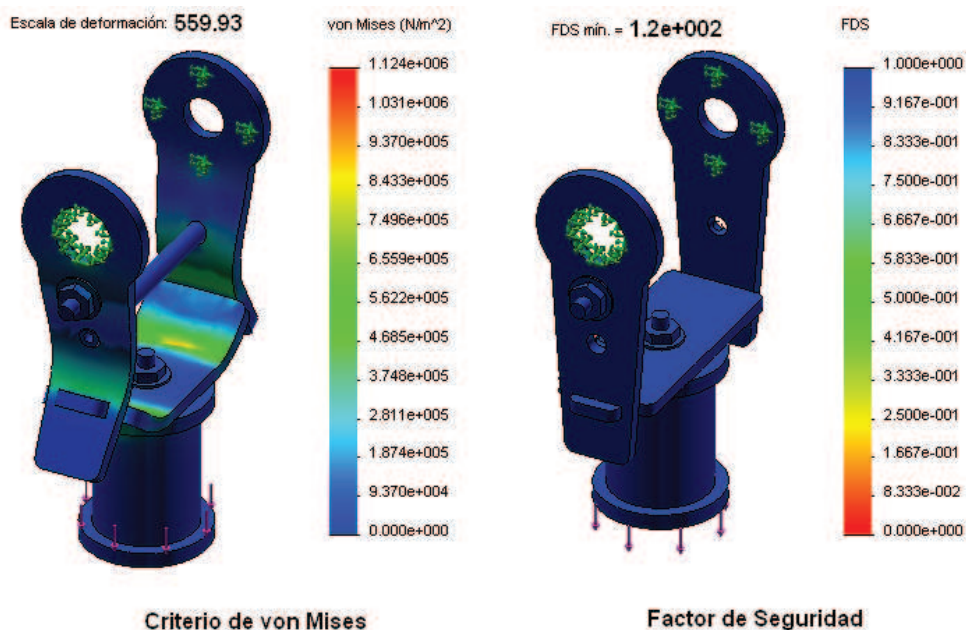


Figura 2.24 Análisis de esfuerzos del tercer eslabón

En la Figura 2.24 se muestra la deformación del análisis de von Mises tiene una escala de 559,93; además se puede notar que el FDS mínimo es de 1,2e+002 lo que indica que el diseño es seguro para el tipo de carga que debe soportar.

2.1.4.2 Análisis de Esfuerzos del Segundo Eslabón

Para realizar el análisis de esfuerzos de este eslabón, se debe considerar además el peso del tercer eslabón. Para obtener el peso del tercer eslabón, se puede acceder a sus propiedades físicas dentro del archivo de ensamblaje de SOLIDWORKS.

El peso del tercer eslabón es de 140,62 gramos, el cual debe ser añadido al peso de 0,5kg de carga aplicada para la simulación. Las partes que soportan el peso, son el eje del servomotor que mueve el codo y el acople que permite el ensamblaje con el tercer eslabón. Luego de realizar el análisis, se han obtenido los siguientes resultados:

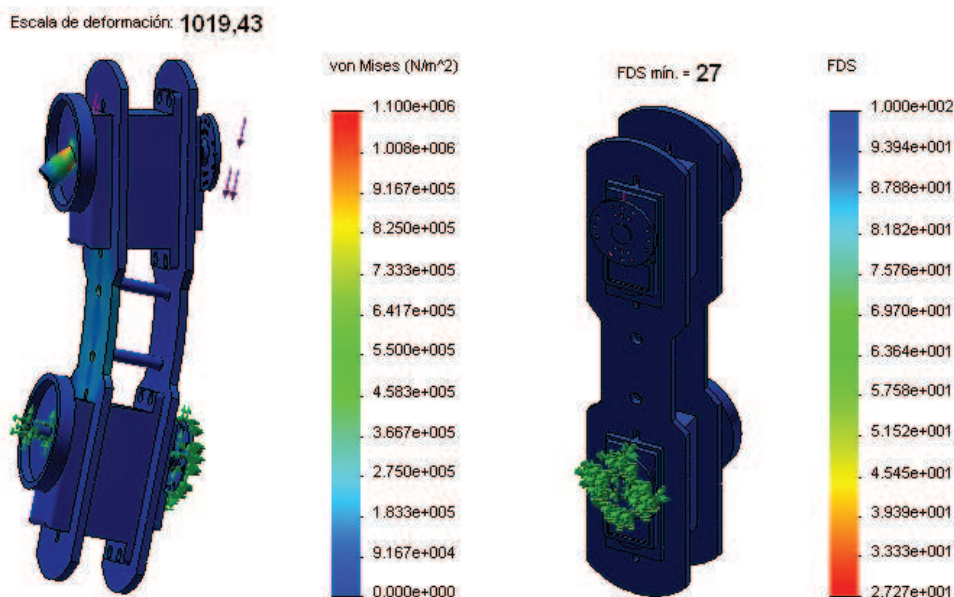


Figura 2.25 Análisis de esfuerzos del segundo eslabón

En la Figura 2.25 se indica la deformación del análisis de von Mises tiene una escala de 1019,43; además se puede notar que el FDS mínimo es de 27, lo que indica que el diseño es seguro para el tipo de carga que debe soportar.

2.1.4.3 Análisis de Esfuerzos del Primer Eslabón

Para realizar el análisis de esfuerzos de este eslabón, se debe considerar además el peso de los eslabones segundo y tercero. Para obtener el peso de los eslabones segundo y tercero, se puede acceder a sus propiedades físicas dentro de los archivos de ensamble de SOLIDWORKS.

El peso del tercer eslabón es de 140,62 gramos, y el peso del segundo eslabón es de 123,48 gramos. Estos pesos deben ser añadidos al peso de 0,5kg de carga aplicada para la simulación. Las partes que soportan el peso, son las caras internas de los perfiles que dan forma al primer eslabón. Luego de realizar el análisis, se han obtenido los siguientes resultados:

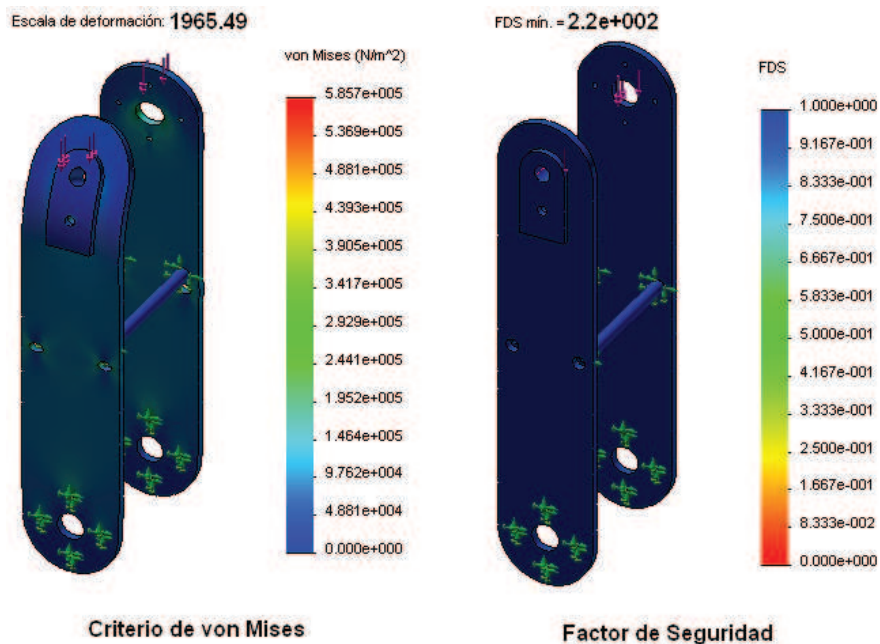


Figura 2.26 Análisis de esfuerzos del primer eslabón

En la Figura 2.26 se presenta la deformación del análisis de von Mises tiene una escala de 1965,49; además se puede notar que el FDS mínimo es de 2.2e+002, lo que indica que el diseño es seguro para el tipo de carga que debe soportar.

2.1.4.4 Análisis de Esfuerzos de la Base

Para el caso de la base, se deben considerar dos tipos de fuerzas; una debida al peso de la estructura total, y otra debida al movimiento propio de la base. La parte que soporta el peso de la estructura es la base auxiliar sobre la cual se montan todas las piezas del robot; por otro lado, las partes que soportan el movimiento de la base son las caras internas de los agujeros para ensamblar la base auxiliar al eje del motor de la base. Para aplicar la fuerza debido al peso de la estructura, hay que considerar el peso de los eslabones anteriores más el peso del actual debido al diseño. Primero: 26,70gr, segundo: 123,48gr, tercero: 140,62gr. Para aplicar la fuerza debida al movimiento de la base, se considerará un torque de prueba máximo de 4,5kg-cm. Luego de realizar el respectivo análisis, se tienen los siguientes resultados:

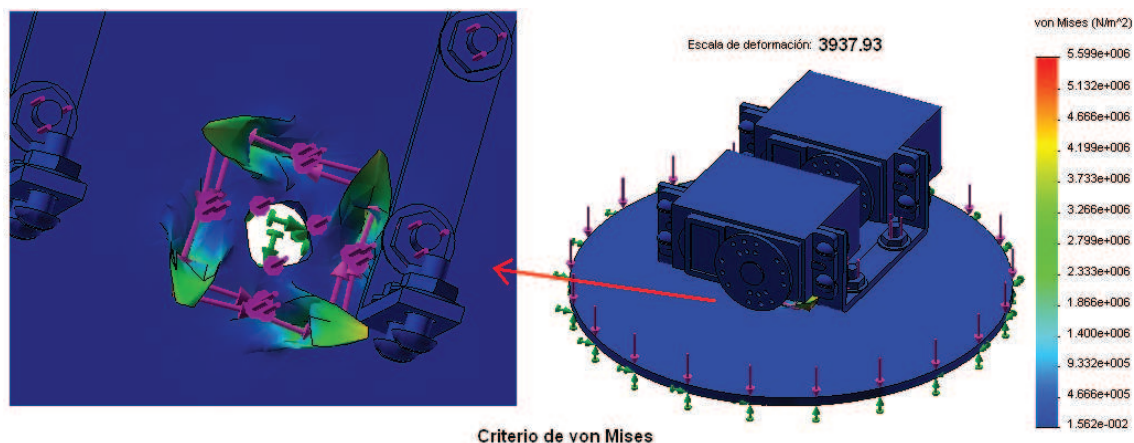


Figura 2.27 Análisis de esfuerzos de la base: Criterio de von Mises

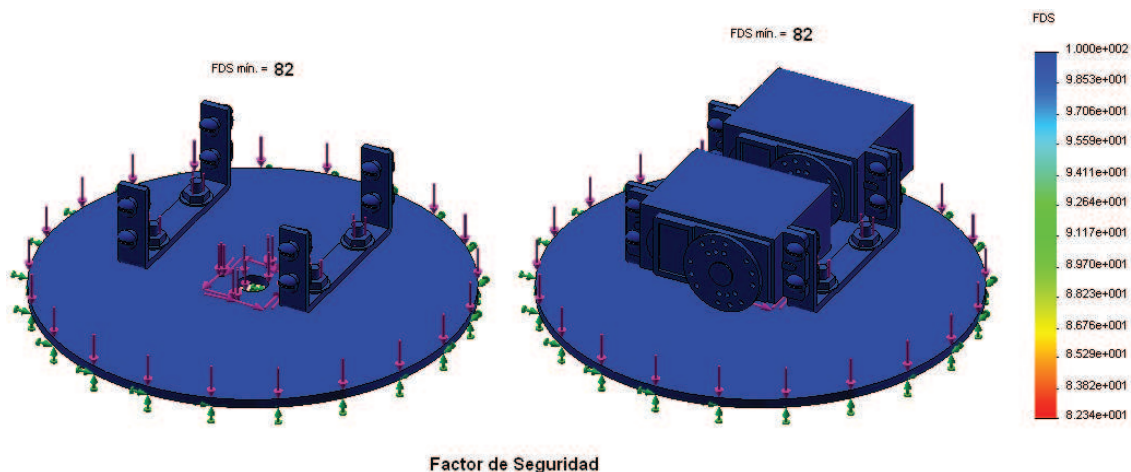


Figura 2.28 Análisis de esfuerzos de la base: Factor de Seguridad

En la Figura 2.28 se indica la deformación del análisis de von Mises tiene una escala de 3937,93; además se puede notar que el FDS mínimo es de 82, lo que indica que el diseño es seguro para el tipo de carga que debe soportar.

2.2 DISEÑO ELECTRÓNICO

En este apartado, se realizará el dimensionamiento de los actuadores que moverán las estructuras anteriormente diseñadas y analizadas mediante SOLIDWORKS y COSMOSWorks, y también se describirán los circuitos y equipos utilizados para realizar el sistema de control.

Para el dimensionamiento de actuadores, se utiliza el complemento de SOLIDWORKS denominado COSMOSMotion. En el ANEXO B, se muestra una

alternativa para dimensionamiento de actuadores mediante Mechatronics Toolkit de LabVIEW 8.6.

2.2.1 COSMOSMotion

COSMOSMotion permite realizar un estudio de movimiento de la estructura diseñada, dando al diseñador una mejor apreciación del desempeño real de la máquina. Para activar el complemento COSMOSMotion, se accede al menú Herramientas/Complementos y en la ventana que se muestra se activa la caja de verificación correspondiente a COSMOSMotion.

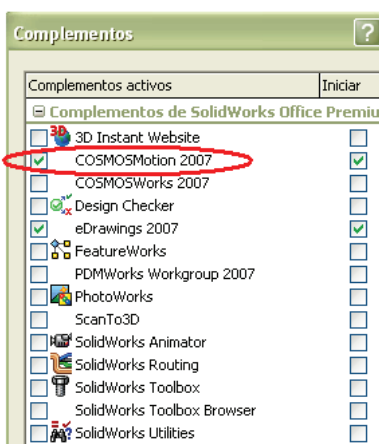


Figura 2.29 Ventana de activación de complementos: COSMOSMotion

Cabe anotar que la ventana mostrada y la siguiente explicación, son relacionadas con SOLIDWORKS 2007, debido a la compatibilidad de esta versión con el paquete de desarrollo LabVIEW Mechatronics Toolkit. Las nuevas versiones de SOLIDWORKS permiten también desarrollar estudios de movimiento y esfuerzos, pero para realizar una interfaz con LabVIEW debe utilizarse el Toolkit NI SoftMotion, cuyo uso es más complejo que el Mechatronics Toolkit, y su explicación está fuera del alcance de este trabajo.

Luego de activar COSMOSMotion, dentro del gestor de diseño de SOLIDWORKS, se tiene un nuevo ícono que corresponde al estudio de movimiento, y que muestra principalmente las siguientes opciones.



- *Moving Parts*: define partes móviles para el estudio de movimiento.
- *Ground Parts*: define partes fijas para el estudio de movimiento.
- *Constraints*: contiene todas las restricciones existentes en el ensamble, como son juntas, contactos, etc.
- *Joints*: son las juntas existentes, que se forman automáticamente en base a las relaciones de posición definidas en el proceso de ensamble.
- *Results*: muestra los resultados del estudio de movimiento, como pueden ser curvas de trayectoria, torque, velocidad, etc.

Figura 2.30 Ventana principal del estudio de movimiento

Tan pronto como se pase a la ventana de estudio de movimiento, el ensamble sobre el cual se está trabajando, muestra las diferentes juntas o articulaciones que posee, mediante un gráfico en color celeste que indica el tipo de articulación, y su posición.

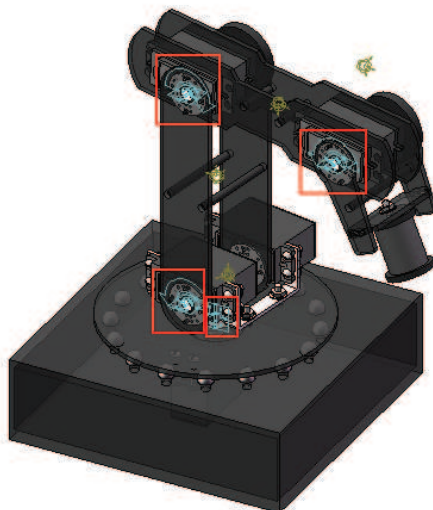


Figura 2.31 Localización automática de articulaciones

Para agregar un tipo de movimiento al sistema, se despliega el menú “*Joints*”, y se da clic derecho sobre la junta a la cual se quiere aplicar movimiento. Luego se abre una ventana que permite configurar el tipo de movimiento a realizar.

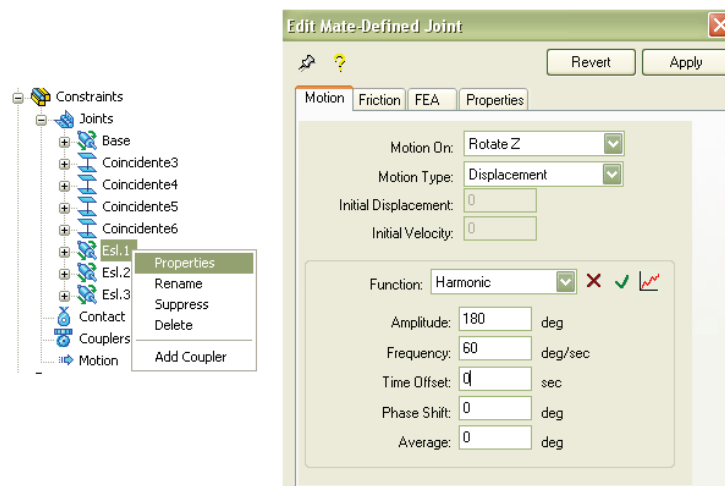


Figura 2.32 Ventana de Propiedades de movimiento

Dentro de esta ventana se puede escoger en donde realizar el movimiento (eje rotacional, traslacional), que tipo de movimiento realizar (desplazamiento, velocidad, aceleración); y se puede definir una función para el movimiento (armónica, paso, constante, ecuación personalizada, secuencia de puntos).

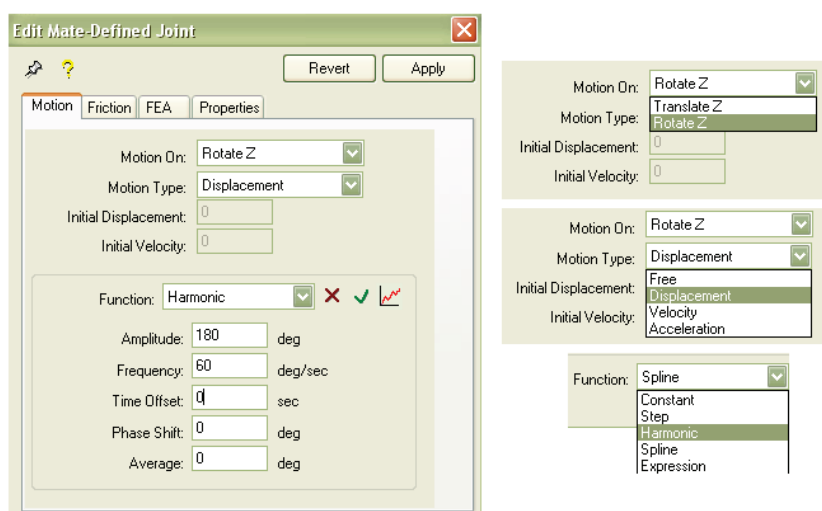


Figura 2.33 Configuración de movimiento

De esta forma se puede asignar movimiento a cada una de las articulaciones del ensamble. Para ejecutar el estudio de movimiento, se debe presionar el ícono en forma de calculadora en la parte inferior izquierda de la ventana de COSMOSMotion.

Adicionalmente, COSMOSMotion, permite obtener curvas de resultado del estudio de movimiento realizado. Estas curvas pueden representar torque, velocidad, posición, etc. y son obtenidas luego de que se haya realizado el estudio de movimiento; es decir, luego de presionar el ícono en forma de calculadora, se selecciona la articulación deseada haciendo clic derecho y se escoge el tipo de resultado deseado.

Mediante las curvas obtenidas, se pueden dimensionar los actuadores que vayan a generar el movimiento simulado. Hay que tener presente las unidades en las que se da la información en las curvas de resultado, ya que es posible que algún tipo de conversión sea necesaria.

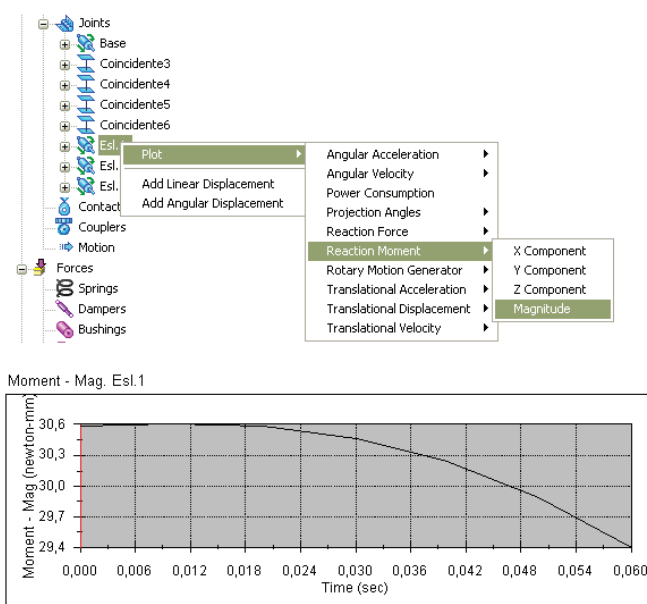


Figura 2.34 Generación de curvas de resultado

2.2.2 ANÁLISIS DE MOVIMIENTO DEL ROBOT

En base a las relaciones de posición definidas en el archivo de ensamblaje del manipulador industrial, se configura un estudio de movimiento que permita dimensionar los actuadores de cada articulación del robot.

El robot diseñado cuenta con cuatro grados de libertad, cada uno movido por servomotores de modelismo, dado que es un prototipo. A continuación se presenta la ventana de configuración de movimiento de cada articulación.

2.2.2.1 Configuración del movimiento de las articulaciones

Todas las articulaciones del robot diseñado, tendrán un movimiento rotacional. En base al análisis cinemático del robot, se ha definido al eje “z” como el eje de rotación de cada articulación, en función de su correspondiente sistema coordenado. Para la simulación, se asignará un movimiento armónico que cubra el rango de movimiento de cada articulación, y se asignará la máxima velocidad permisible para el tipo de servomotores escogido, que es 60°/seg.

A continuación se presentan las ventanas de configuración de movimiento para las diferentes articulaciones.

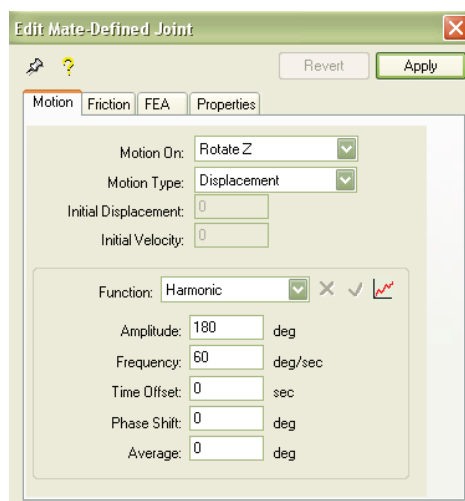


Figura 2.35 Configuración de movimiento de la base

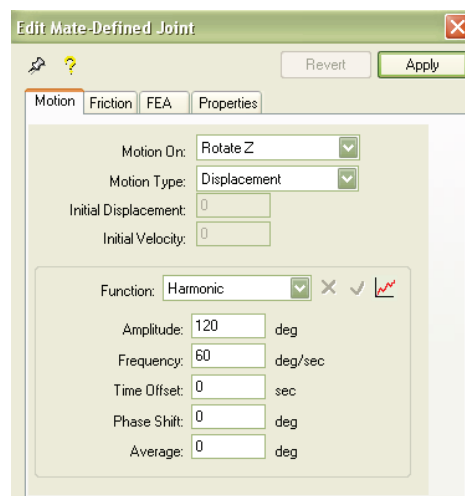


Figura 2.36 Configuración de movimiento del hombro

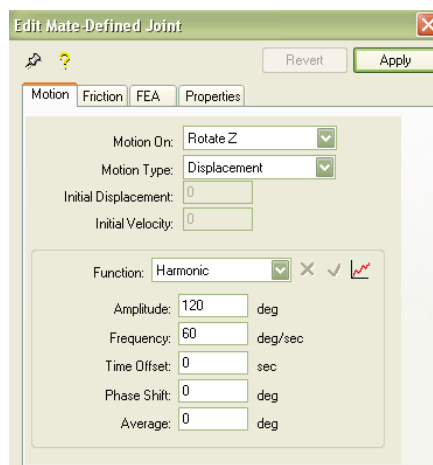


Figura 2.37 Configuración de movimiento del codo

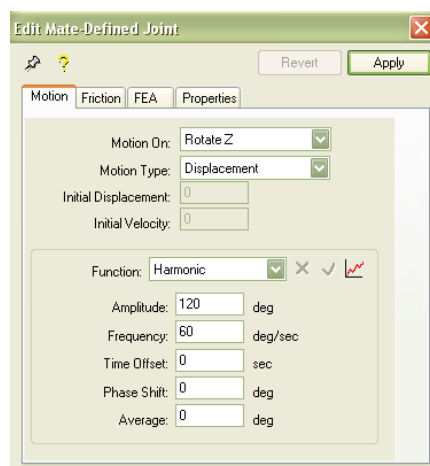


Figura 2.38 Configuración de movimiento de la muñeca

Finalmente se coloca una fuerza en el extremo del manipulador, orientada hacia abajo, para simular un peso levantado de 0,5 kg. De esta forma se pueden aplicar parámetros externos a la estructura del robot, y evaluar su respuesta.

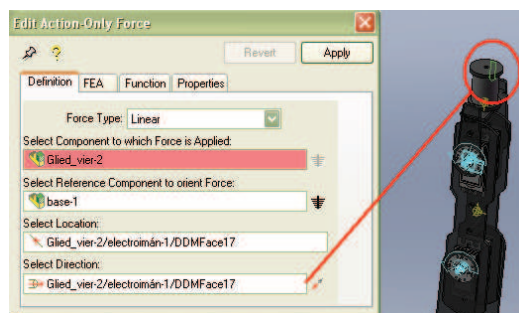


Figura 2.39 Configuración de fuerza aplicada

Luego de ejecutar el estudio de movimiento, con una fuerza aplicada, se han obtenido las siguientes curvas de torque para cada articulación. Estas curvas están expresadas en Newton-mm; mientras que el tipo de motores a usarse, se dimensiona en base al torque expresado en kg-cm, por lo que es necesaria una conversión de unidades.

Para el dimensionamiento de actuadores, se toma el valor pico de la curva de torque mostrada.

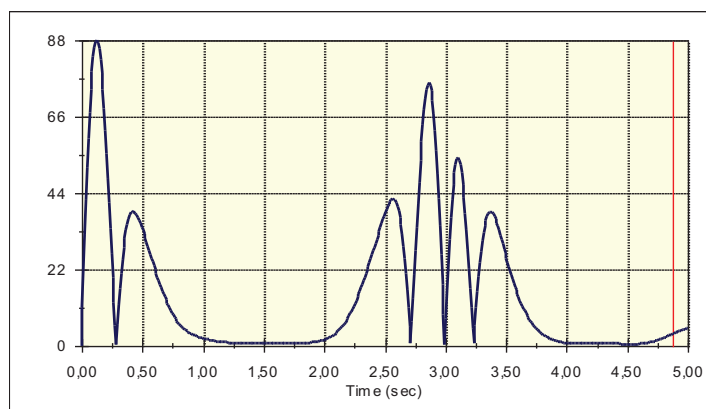


Figura 2.40 Curva de Torque requerido por la base

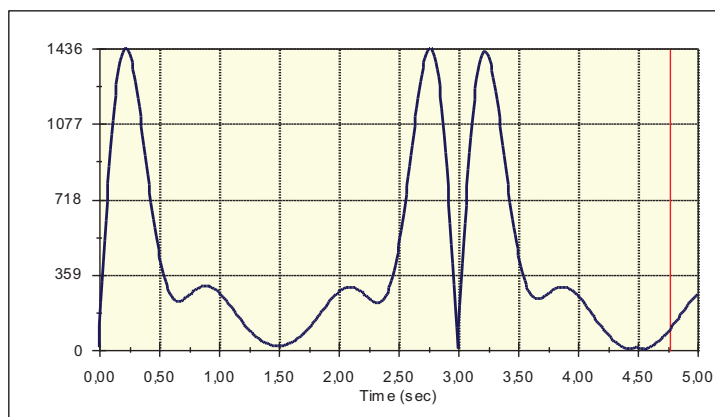


Figura 2.41 Curva de Torque requerido por el hombro

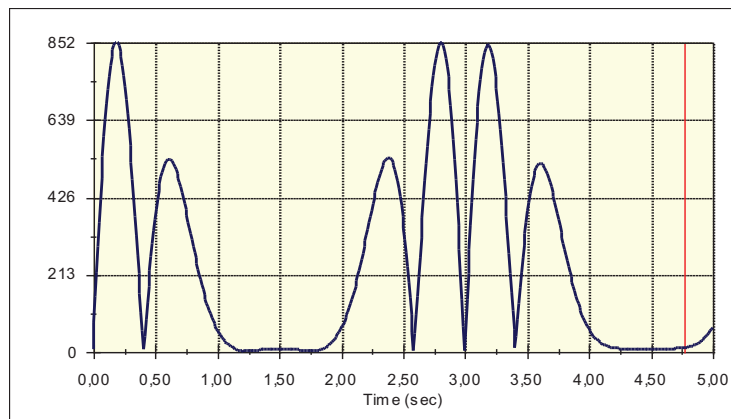


Figura 2.42 Curva de Torque requerido por el codo

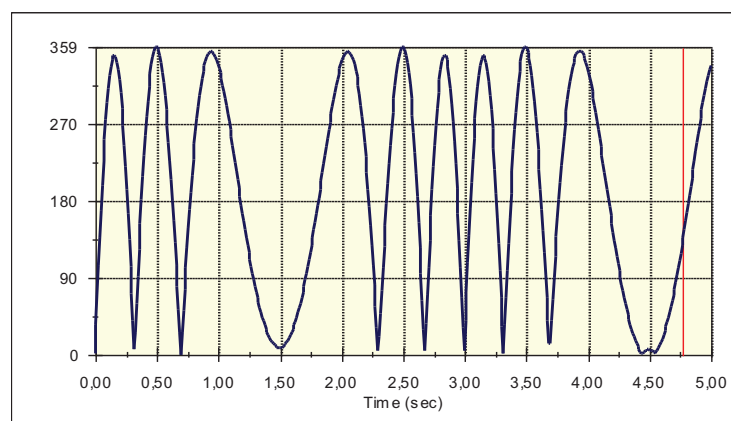


Figura 2.43 Curva de Torque requerido por la muñeca

Luego del análisis y la respectiva conversión de unidades ($1\text{N}\cdot\text{mm}=0,0102\text{kg}\cdot\text{cm}$), se procede a la selección de los actuadores. En la Tabla 2.1 se muestra el requerimiento de torque para cada articulación, y el correspondiente servomotor escogido.

Los servomotores mostrados en la Tabla 2.1, son los escogidos para la construcción del prototipo.

Tabla 2.1 Dimensionamiento de Actuadores

Articulaciones	Torque Requerido	Servomotor Escogido	Torque Servomotor
Base	0,9 kg-cm	HS-311	3,6 kg-cm
Hombro	14,6 kg-cm	2 x HD-2500A	10 kg-cm
Codo	8,7 kg-cm	HD-2500	10 kg-cm
Muñeca	3,6 kg-cm	HS-311	3,6 kg-cm

2.2.3 HARDWARE UTILIZADO PARA EL CONTROL DE POSICIÓN

El sistema de control de posición del efector final del manipulador involucra distintas partes de hardware para cada una de las etapas del sistema. A continuación se describen los equipos utilizados para cada etapa.

2.2.3.1 Realimentación de Posición

La posición del efector final depende del ángulo de giro de cada una de las articulaciones, valor que se obtiene mediante la cinemática inversa. Sin embargo, es necesario constatar la posición del efector final, para lo cual se puede medir el ángulo que se ha movido cada articulación; es decir, implementar un sistema de control en lazo cerrado para cada servomotor.

Para obtener la señal de realimentación de posición, se ha recurrido a medir el valor del potenciómetro interno de cada servomotor, cuyos valores extremos varían dependiendo de cada motor. La salida de voltaje medida es diferencial, por lo que es necesario referirla a tierra para su ingreso a los equipos de adquisición de datos.

Para referir las salidas de voltaje diferenciales que entregan los potenciómetros de los servomotores, se han utilizado amplificadores diferenciales con ganancia $G=1$. No es necesario el uso de un filtro activo, debido a que posteriormente se utilizará una tarjeta de adquisición de datos mediante la cual se pueden implementar filtros digitales por software.

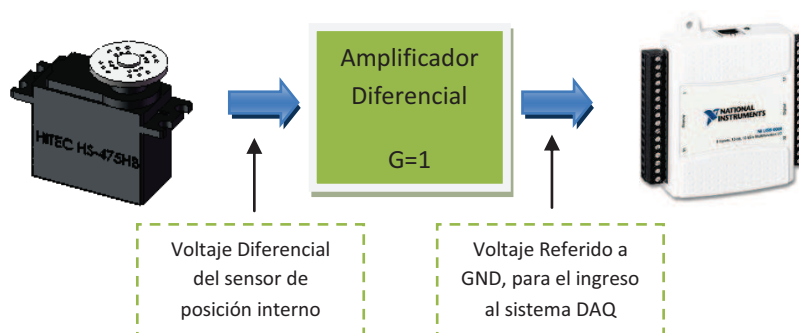
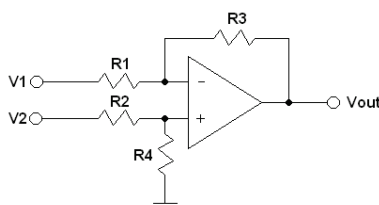


Figura 2.44 Etapas del Acondicionamiento de Señal de Realimentación

Los circuitos referenciadores implementados son de fácil diseño, por lo que únicamente se mostrará su esquema electrónico final.



$$V_{out} = \frac{R_3}{R_1}(V_2 - V_1); \quad R_1 = R_2 = 10k\Omega, R_3 = R_4 = 10k\Omega$$

Figura 2.45 Amplificador Diferencial con Ganancia Unitaria

La diferencia V_2-V_1 es la entrada diferencial de la señal de realimentación de posición; mientras que V_{out} es la salida de voltaje referida a tierra. Este circuito es utilizado para la señal de realimentación de cada uno de los motores. Luego de referir a una misma tierra todas las señales, estas son ingresadas a la tarjeta de adquisición de datos.

2.2.3.2 Adquisición de Datos

Para la adquisición de datos de las señales de posición se ha utilizado una tarjeta NI USB-6009, la cual posee 8 entradas análogas SE de 14 bits, 48kmuestras/s (AI0 – AI7), 2 salidas análogas de 12 bits (AO0 y AO1), un puerto digital TTL/CMOS de 12 bits y una entrada tipo contador de 32bits.

La NI USB-6009 permite también adquirir señales diferenciales únicamente usando las entradas AI0-AI3 como terminales positivos, y las entradas AI4-AI7 como negativos.



Figura 2.46 NI USB-6009, tomado de [11].

La NI USB-6009 fue utilizada para adquisición de señal de posición obtenida a la salida del amplificador diferencial. Se utilizaron 5 canales análogos de entrada a 8k muestras cada uno, para la adquisición de cada señal de realimentación de posición. Adicionalmente se utilizó una salida digital para el control ON-OFF del electroimán del efector final. El circuito de potencia que acciona el electroimán será analizado más adelante.

Se puede cambiar la configuración y programar la adquisición de datos de la tarjeta NI USB-6009 en el entorno de LabVIEW.

2.2.3.3 Circuito de Potencia para el Electroimán

El circuito utilizado para el accionamiento del electroimán, es del tipo ON-OFF basado en un MOSFET IRF640, que trabaja en corte-saturación. El circuito implementado se muestra a continuación.

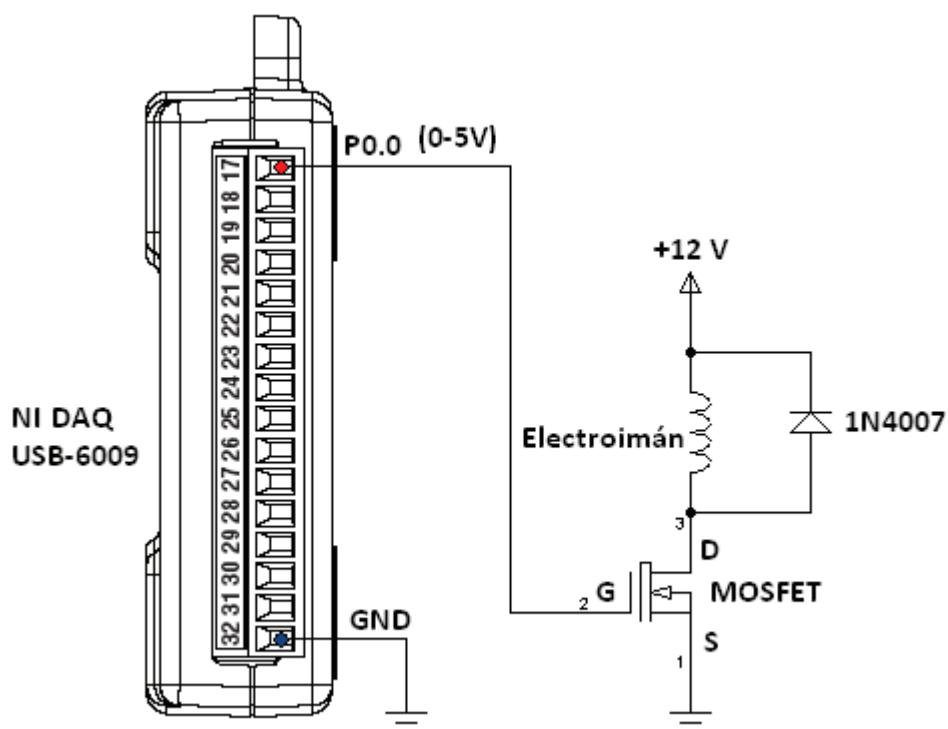


Figura 2.47 Circuito de Potencia para Electroimán.

2.2.3.4 Señales de Control

Todos los algoritmos de control están implementados por software, pero el elemento que aplica estas señales es el driver de servos Pololu, el cual permite controlar hasta 16 servomotores simultáneamente generando 16 señales de control independientes. El servo controlador puede generar pulsos desde 0.25ms hasta 2.75ms; rango que es mayor que el de la mayoría de servos, permitiendo operarlos sobre los 180° [10].

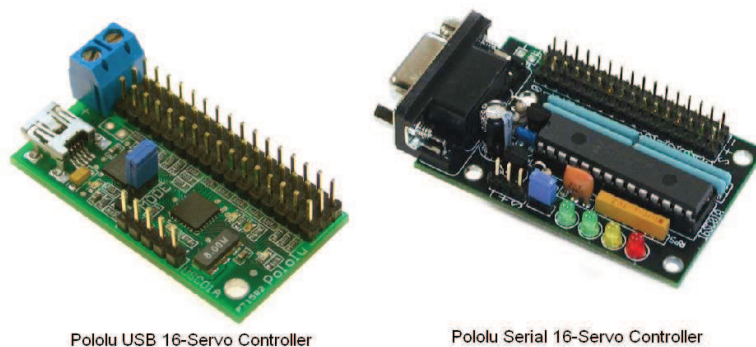


Figura 2.48 Pololu Servo Controller, tomado de [10].

El controlador de servos recibe datos en forma serial. Cada byte debe tener 8 bits de datos, sin paridad y 1 bit de stop. Existen drivers que tienen puerto USB en lugar de Serial, pero el modo de funcionamiento es el mismo, ya que únicamente realizan una conversión USB-Serial dentro del circuito.

Existen dos formas de comunicarse con el driver de servos: *Pololu Mode* y *Mini SSC II Mode*.

- *Pololu Mode*: El primer modo permite controlar no solo posición sino también velocidad, sentido de giro, baud rate y otras opciones. En este modo de trabajo los datos seriales enviados al controlador deben tener la siguiente trama:

start byte = 0x80	device ID = 0x01	command	servo num	data 1	data 2
-------------------	------------------	---------	-----------	--------	--------

Donde el primer byte es de sincronización (0x80); el segundo byte es la identificación del driver (0x01); el tercer byte es el comando a ejecutar (set position, set speed, baud rate, etc.); el cuarto byte es el número de servo; y los dos últimos bytes son los datos correspondientes al comando a ejecutar [10].

- *Mini SSC II Mode*: En este modo de control el driver de servos puede funcionar únicamente a 2400 o 9600 bps, y permite controlar posición pero no velocidad. Para dar el setpoint de posición el controlador recibe la siguiente trama:

start byte = 0xFF	servo number, 0x00-0xFE	Servo position, 0x00-0xFE
-------------------	-------------------------	---------------------------

Donde el primer byte es de sincronización (0xFF); el segundo byte es el número de servo; y el tercer byte es el setpoint de posición (0-254).

En este modo de funcionamiento existen dos rangos de trabajo, diferenciados por el número de servomotor especificado; si se coloca un número de servo entre 0 y 15, el rango de giro es de 90°; mientras que si coloca un número de servo entre 16 y 31 el rango es de 180° [10].

En el desarrollo de este proyecto se utilizó el driver de servos en el modo Pololu, dado que ofrece una resolución de 0,05 grados para el control. La comunicación serial se realiza a 28800 bps.

De forma general, el sistema de control puede expresarse mediante el siguiente diagrama de bloques.

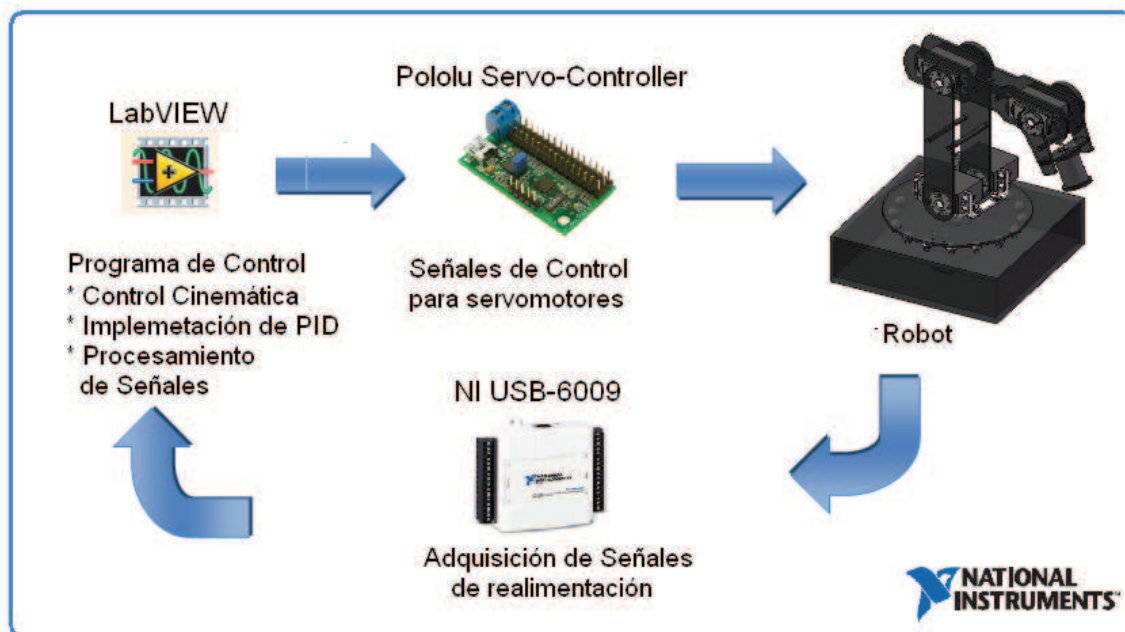


Figura 2.49 Equipo Utilizado para el Sistema de Control en Lazo Cerrado

2.2.4 HARDWARE UTILIZADO PARA EL SISTEMA DE VISIÓN ARTIFICIAL

Para el sistema de adquisición de imagen, se utilizó una cámara web que internamente tiene una matriz de sensores CMOS de 300k con una resolución de adquisición de hasta 8M.



Figura 2.50 Cámara utilizada para la adquisición de Imagen

A más del sistema óptico, otra parte fundamental en el sistema de visión artificial, es el diseño de iluminación. Para el presente proyecto se utilizó iluminación difusa para eliminar sombras. Como fuentes luminosas se utilizaron lámparas fluorescentes de 8W distribuidas uniformemente en la parte superior del soporte para cámaras construido. Adicionalmente se utilizó fórmica blanca como elemento de reflexión en algunas paredes del soporte, para distribuir mejor la iluminación dentro del área de trabajo del manipulador. En la Figura 2.52 se presenta el soporte sin paredes de fórmica.

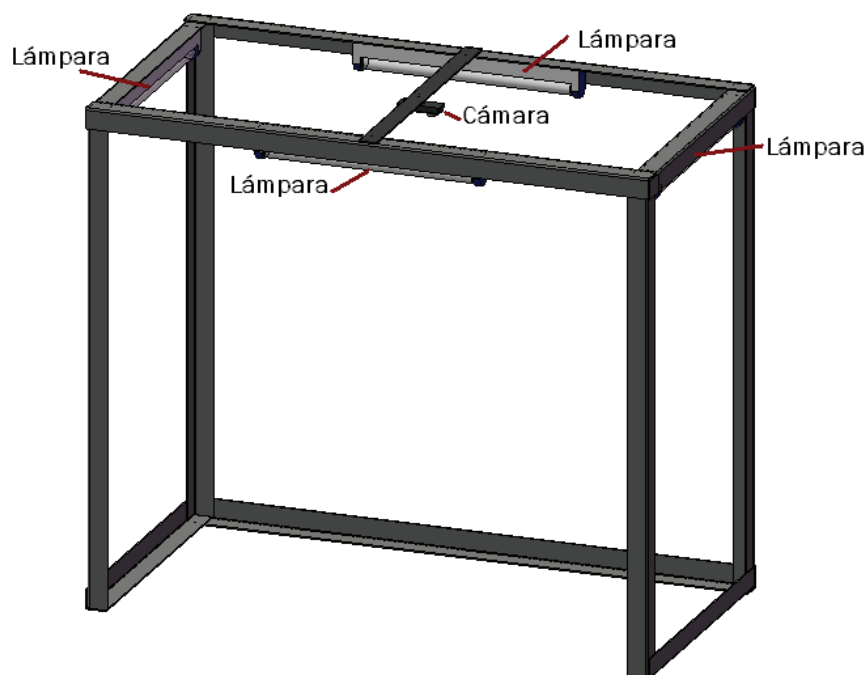


Figura 2.51 Soporte para Cámaras y Sistema de Iluminación

Se coloca fórmica en las dos paredes laterales y en la superior para proteger al sistema en cierta medida contra la entrada de luz desde el exterior.



Figura 2.52 Parte interna superior del soporte para Cámara

2.2.5 SISTEMA DE ALIMENTACIÓN

El sistema de control implementado requiere de fuentes de alimentación de la siguiente denominación:

- 5V / 4A – Alimentación a Servomotores
- +-12V / 2A – Alimentación Amplificadores Diferenciales y Electroimán

Para suplir estos requerimientos se ha incorporado al sistema general de control a una fuente de computador que cumple con las especificaciones mencionadas.

2.2.6 MONTAJE DEL HARDWARE PARA EL SISTEMA DE CONTROL

Una vez descritos los elementos utilizados para implementar el sistema de control, se procederá a explicar el proceso de montaje.

En primer lugar se retiró el 'case' original de la fuente de computador, para alojarla dentro uno más amplio en donde también se puedan colocar los demás elementos utilizados.

Luego se colocó la placa de amplificadores diferenciales; posteriormente se colocó el controlador de servos dejando accesible el puerto serial del mismo; y

finalmente se realizó el montaje de la tarjeta de adquisición de datos, y se deja en el case una apertura para el cable de conexión de la misma.



Figura 2.53 Montaje de hardware de control



Figura 2.54 Montaje de circuito referenciador.

El case que encierra a todo el sistema físico de control, tiene una abertura adicional para permitir el paso de las señales de realimentación y control desde y hacia los servomotores del robot.

CAPÍTULO 3

DESARROLLO DEL PROGRAMA DE CONTROL

3.1 CINEMÁTICA DEL ROBOT

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia sin considerar las fuerzas que intervienen.

Así, la cinemática se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo y en particular por las relaciones entre la posición y la orientación del extremo final del robot con los valores que toman sus coordenadas articulares.

Los problemas fundamentales a resolver en la cinemática del robot son, el **problema cinemático directo**, que 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 del robot [1].

El otro problema a resolverse es el **problema cinemático inverso**, que consiste en determinar la configuración que deben adoptar cada una de las articulaciones del robot para ubicarlo en un punto con una respectiva orientación conociendo la posición y orientación del extremo del robot [1].

Denavit y Hartenberg propusieron un método sistemático para describir y representar la geometría espacial de los elementos de la cadena cinemática, y en particular de un robot, con respecto a un sistema de referencia fijo. Este método utiliza una matriz de transformación homogénea para describir la relación espacial entre dos elementos rígidos adyacentes, reduciendo el problema cinemático directo a encontrar una matriz de transformación homogénea 4 x 4 que relacione la localización espacial del robot con respecto al sistema de coordenadas de su base.

Por otra parte, la cinemática del robot trata también de encontrar las relaciones entre las velocidades del movimiento de las articulaciones y las del extremo. Este modelo viene dado por el **modelo diferencial** expresado mediante la **matriz Jacobiana** [1].

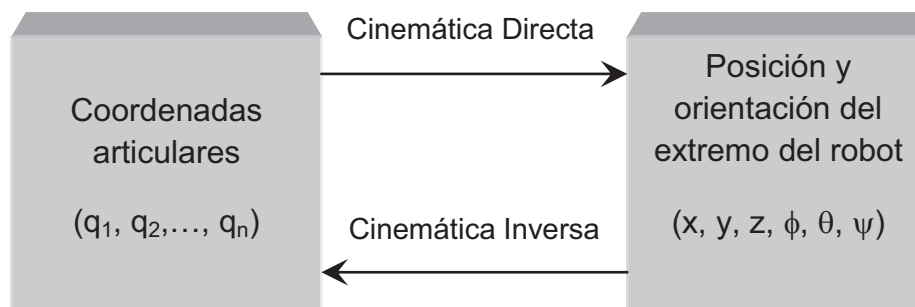


Figura 3.1 Diagrama de relación entre Cinemática Directa e Inversa

3.2 EL PROBLEMA CINEMÁTICO DIRECTO

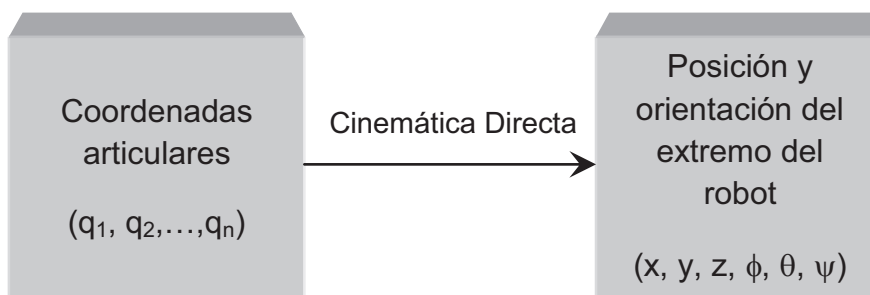


Figura 3.2 Representación de la Cinemática Directa

A través de la resolución del problema cinemático directo, es posible determinar cuál es la posición y orientación que adopta el extremo del robot cuando cada una de las variables que fijan la posición u orientación de sus articulaciones toman valores determinados.

Dado que las variables articulares pueden ser leídas directamente de los correspondientes sensores por unidad de control del robot, el modelo cinemático directo será utilizado por éste, entre otros fines, para presentar al usuario información relativa a la localización espacial del extremo del robot.

Así, se han escogido coordenadas cartesianas y ángulos de Euler para representar la posición y orientación del extremo del robot de cuatro grados de libertad [1].

$$x = f_x(q_1, q_2, q_3, q_4)$$

$$y = f_y(q_1, q_2, q_3, q_4)$$

$$z = f_z(q_1, q_2, q_3, q_4)$$

$$\theta = f_\beta(q_1, q_2, q_3, q_4)$$

El modelo cinemático directo, puede ser obtenido mediante dos enfoques diferentes, denominados métodos geométricos y métodos basados en cambios de sistemas de coordenadas.

Los primeros son adecuados para casos simples, ya que su aplicación está limitada para robots de pocos grados de libertad debido a que no son sistemáticos.

Los métodos basados en cambios de sistemas de referencia, permiten resolver el problema cinemático directo de una manera más sistemática, por lo cual es posible abordar configuraciones de robots de "n" grados de libertad, por lo cual son los sistemas más usados, en particular los que usan matrices de transformación homogénea.

3.2.1 RESOLUCIÓN DEL PROBLEMA CINEMÁTICO DIRECTO MEDIANTE MÉTODOS GEOMÉTRICOS

Para resolver el problema cinemático directo del robot antropomórfico con cuatro grados de libertad, es necesario encontrar las relaciones que permiten conocer la localización espacial del extremo del robot a partir de sus coordenadas articulares.

La obtención de estas relaciones se logra fácilmente en ciertos casos, donde el número de grados de libertad es bajo, a través de consideraciones geométricas simples.

A continuación se presenta la resolución del problema cinemático directo, cuyos resultados se contrastarán con el método mediante matrices homogéneas, para comprobar resultados y procedimientos.

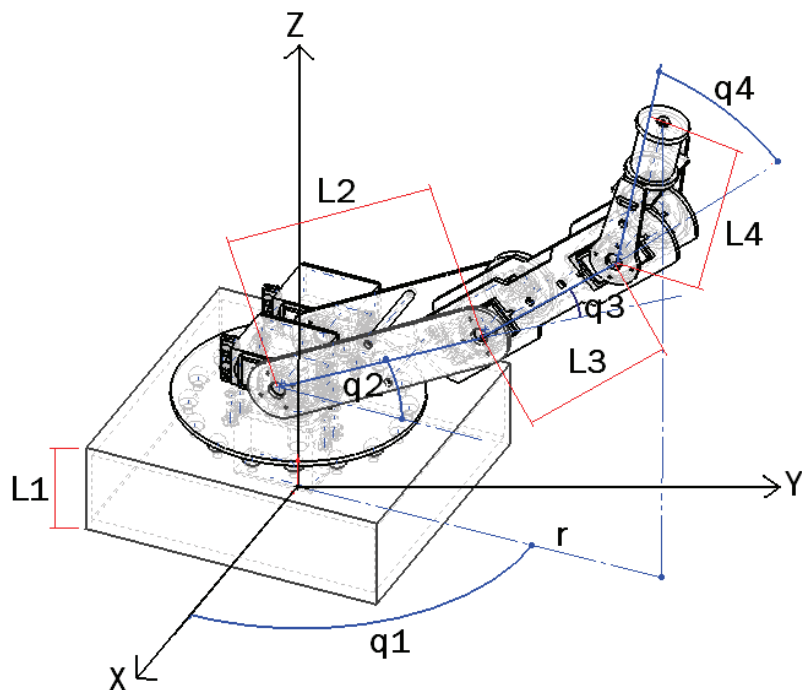


Figura 3.3 Esquema general del brazo antropomórfico representado mediante el sistema de coordenadas rectangulares (x, y, z)

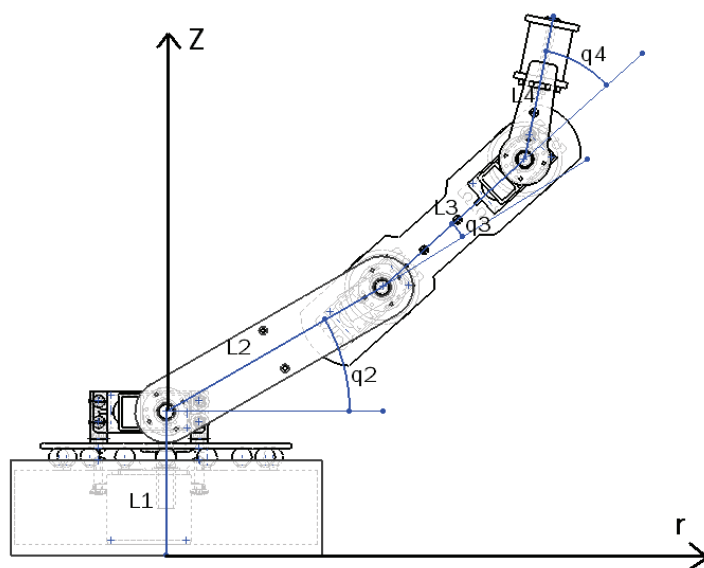


Figura 3.4 Eslabones y articulaciones cuyos movimientos se realizan solamente en el plano z-r.

Por medio de la Figura 3.4 se puede determinar el valor de r :

$$r = l_2 \cos(q_2) + l_3 \cos(q_2 + q_3) + l_4 \cos(q_2 + q_3 + q_4) \quad (3.1)$$

De la misma manera es posible determinar z :

$$z = l_2 \sin(q_2) + l_3 \sin(q_2 + q_3) + l_4 \sin(q_2 + q_3 + q_4) \quad (3.2)$$

Tomando como referencia el sistema de coordenadas de la Figura 3.3 se puede determinar fácilmente las ecuaciones correspondientes a x e y :

$$x = r \cos(q_1) \quad (3.3)$$

$$y = r \sin(q_1) \quad (3.4)$$

Reemplazando la ecuación (3.1) en (3.3) y (3.2) en (3.4), se tiene:

$$x = l_2 \cos(q_1) \cos(q_2) + l_3 \cos(q_1) \cos(q_2 + q_3) + l_4 \cos(q_1) \cos(q_2 + q_3 + q_4)$$

$$y = l_2 \sin(q_1) \cos(q_2) + l_3 \sin(q_1) \cos(q_2 + q_3) + l_4 \sin(q_1) \cos(q_2 + q_3 + q_4)$$

$$z = l_2 \sin(q_2) + l_3 \sin(q_2 + q_3) + l_4 \sin(q_2 + q_3 + q_4)$$

Para simplificar las ecuaciones se las escribirá de la siguiente manera:

$$x = l_2 C_1 C_2 + l_3 C_1 C_{23} + l_4 C_1 C_{234} \quad (3.5)$$

$$y = l_2 S_1 C_2 + l_3 S_1 C_{23} + l_4 S_1 C_{234} \quad (3.6)$$

$$z = l_2 S_2 + l_3 S_{23} + l_4 S_{234} \quad (3.7)$$

$$[n \ o \ a] = Rotz(q_1) \cdot Roty(-q_2 - q_3 - q_4) \cdot Rotx\left(\frac{\pi}{2}\right) \quad (3.8)$$

Ésta ecuación representa la cadena de rotaciones para la orientación del extremo del robot.

De ésta manera, se ha determinado las ecuaciones correspondientes a la cinemática directa, mediante las cuales se podrá determinar la posición y orientación del extremo del robot con sólo saber la posición angular de cada una de sus articulaciones.

3.2.2 ALGORITMO DE DENAVIT-HARTENBERG PARA LA OBTENCIÓN DEL MODELO CINEMÁTICO DIRECTO

Éste método permite describir la relación que existe entre dos elementos contiguos, donde se puede hacer uso de cualquier sistema de referencia. Denavit-Hartenberg establecieron un método matricial para establecer la localización que debe tomar cada sistema de coordenadas $\{S_i\}$ ligado a cada eslabón "i" de una cadena articulada, para poder sistematizar la obtención de las ecuaciones cinemáticas de la cadena completa.

Escogiendo los sistemas de coordenadas asociados a cada eslabón según la representación propuesta por Denavit-Hartenberg, será posible pasar de uno al siguiente mediante 4 transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón.

Estas 4 transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia del elemento $i-1$ con el sistema del elemento i . Las transformaciones en cuestión son las siguientes [1]:

1. Rotaciones alrededor del eje z_{i-1} un ángulo θ_i .
2. Traslación a lo largo de z_{i-1} una distancia d_i ; vector d_i (0, 0, d_i).
3. Traslación a lo largo de x_i una distancia a_i ; vector a_i (a_i , 0, 0).
4. Rotación alrededor del eje x_i un ángulo α_i .

Estas transformaciones se refieren al sistema móvil. Dado que el producto de matrices no es conmutativo, las transformaciones se han de realizar en el orden indicado. De este modo se tiene que:

$${}^{i-1}A_i = \mathbf{Rot}_z(\theta_i) \mathbf{T}(0,0,d_i) \mathbf{T}(a_i,0,0) \mathbf{Rot}_x(\alpha_i) \quad (3.9)$$

Y realizando el producto entre las matrices se obtiene lo siguiente:

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} =$$

$${}^{i-1}A_i = \begin{bmatrix} C\theta_i & -C\alpha_i S\theta_i & S\alpha_i S\theta_i & a_i C\theta_i \\ S\theta_i & C\alpha_i C\theta_i & -S\alpha_i C\theta_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

Donde θ_i , d_i , a_i , α_i son parámetros D-H del eslabón ' i '. De este modo, basta con identificar los parámetros para obtener las matrices ${}^{i-1}A_i$ y relacionar así todos y cada uno de los eslabones del robot.

Es necesario además escoger correctamente los ejes del movimiento, y conjuntamente con la definición de los 4 parámetros de D-H, se forma el siguiente algoritmo para la resolución del problema cinemático directo [1]:

- DH1.** Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
- DH2.** Numerar cada articulación comenzando por 1 (correspondiente al primer grado de libertad) y acabando en n .
- DH3.** Localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
- DH4.** Para i de 0 a $n-1$ situar el eje z_i sobre el eje de la articulación $i+1$.
- DH5.** Situar el origen del sistema de la base $\{S_0\}$ en cualquier punto del eje z_0 . Los ejes x_0 y y_0 se situarán de modo que formen un sistema ortogonal a derechas con z_0 .
- DH6.** Para ' i ' de 1 a $n-1$, situar el origen del sistema $\{S_i\}$ (solidario al eslabón i) en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i . Si ambos ejes se cortasen se situaría $\{S_i\}$ en el punto de corte. Si fueren paralelos $\{S_i\}$ se situaría en la articulación $i+1$.
- DH7.** Situar x_i en la línea normal común a z_{i-1} y z_i .
- DH8.** Situar y_i de modo que forme un sistema ortogonal a derechas con x_i y z_i .

- DH9.** Situar el sistema $\{S_n\}$ en el extremo del robot de modo que z_n coincida con la dirección de z_{n-1} y x_n sea normal a z_{n-1} y z_n .
- DH10.** Obtener θ_i como el ángulo que hay que girar en torno a z_{i-1} para que x_{i-1} y x_i queden paralelos.
- DH11.** Obtener d_i como la distancia, medida a lo largo de z_{i-1} , que habría que desplazar $\{S_{i-1}\}$ para que x_i y x_{i-1} quedasen alineados.
- DH12.** Obtener a_i como la distancia medida a lo largo de x_i (que ahora coincidirá con x_{i-1}) que habría que desplazar el nuevo $\{S_{i-1}\}$ para que su origen coincida con $\{S_i\}$.
- DH13.** Obtener α_i como el ángulo que habría que girar entorno a x_i , para que el nuevo $\{S_{i-1}\}$ coincida totalmente con $\{S_i\}$.
- DH14.** Obtener las matrices de transformación ${}^{i-1}A_i$ definidas en la ecuación matricial (3.9).
- DH15.** Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot $T = {}^0A_1 \cdot {}^1A_2 \cdot \dots \cdot {}^{n-1}A_n$.
- DH16.** La matriz T define la orientación (submatriz de rotación) y posición (submatriz de traslación) del extremo referido a la base, en función de las 'n' coordenadas articulares.

Los cuatro parámetros de D-H dependen únicamente de las características geométricas de cada eslabón y de las articulaciones que lo unen con el anterior y siguiente. En concreto estos representan:

- θ_i Es el ángulo que forman los ejes x_{i-1} y x_i medido en un plano perpendicular al eje z_{i-1} , utilizando la regla de la mano derecha. Se trata de un parámetro variable en articulaciones giratorias.
- d_i Es la distancia a lo largo del eje z_{i-1} desde el origen del sistema de coordenadas (i-1) -ésimo hasta la intersección de eje z_{i-1} con el eje x_i . Se trata de un parámetro variable en articulaciones prismáticas.

- a_i Es la distancia a lo largo del eje x_i que va desde la intersección del eje z_{i-1} con el eje x_i hasta el origen del sistema i -ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes z_{i-1} y z_i .
- α_i Es el ángulo de separación del eje z_{i-1} y el eje z_i , medido en un plano perpendicular al eje x_i , utilizando la regla de la mano derecha.

Una vez obtenidos los parámetros D-H, el cálculo de las relaciones entre los eslabones consecutivos del robot es inmediato, ya que vienen dadas por las matrices ${}^{i-1}A_i$, que se calculan según la expresión general (3.10).

Las relaciones entre varios eslabones consecutivos dos a dos vienen dadas por las matrices \mathbf{T} que se obtienen como producto de un conjunto de matrices \mathbf{A} .

Obtenida la matriz \mathbf{T} , ésta expresará la orientación (submatriz (3x3) de rotación) y posición (submatriz (3x1) de traslación) del extremo del robot en función de sus coordenadas articulares, con lo que quedará resuelto el problema cinemático directo.

Siguiendo el algoritmo propuesto por Denavit-Hartenberg, se procede a seguir cada uno de los ítems del algoritmo para el robot antropomórfico de 4 GDL:

En primer lugar, se localizan los sistemas de referencia de cada una de las articulaciones del robot (Figura 3.5).

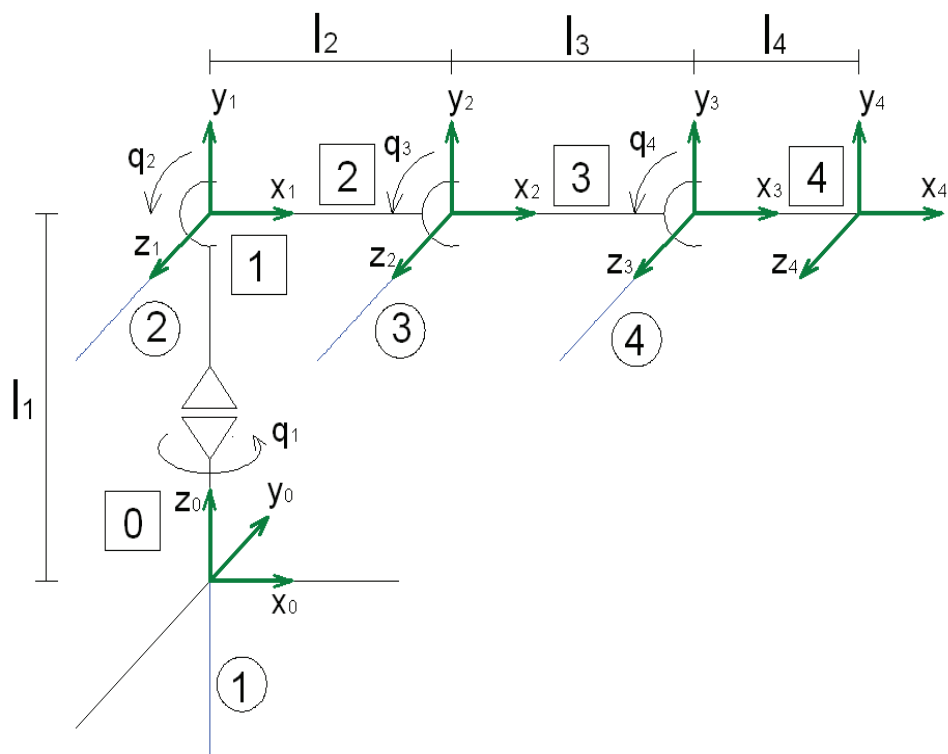


Figura 3.5 Esquema simplificado del Robot Antropomórfico de 4GDL, parámetros Denavit-Hartenberg.

Posteriormente, se determinan los parámetros de Denavit-Hartenberg del robot, con los que se construye la Tabla 3.1.

Tabla 3.1 Parámetros de D-H del robot

ARTICULACIÓN	θ	d	a	α
1	q_1	l_1	0	$\pi/2$
2	q_2	0	l_2	0
3	q_3	0	l_3	0
4	q_4	0	l_4	0

Una vez determinados los parámetros de Denavit-Hartenberg del robot, se calculan las matrices ${}^{i-1}A_i$, sustituyendo la ecuación general de la siguiente manera:

$$\begin{aligned}
 {}^0A_1 &= \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^1A_2 &= \begin{bmatrix} C_2 & -S_2 & 0 & l_2C_2 \\ S_2 & C_2 & 0 & l_2S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^2A_3 &= \begin{bmatrix} C_3 & -S_3 & 0 & l_3C_3 \\ S_3 & C_3 & 0 & l_3S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} & {}^3A_4 &= \begin{bmatrix} C_4 & -S_4 & 0 & l_4C_4 \\ S_4 & C_4 & 0 & l_4S_4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

$$T = {}^0A_1 \cdot {}^1A_2 \cdot {}^2A_3 \cdot {}^3A_4 \quad (3.11)$$

El resultado que se debe alcanzar, es la obtención de la matriz **T** (ecuación (3.11)) que indica la localización del sistema final con respecto al sistema de referencia de la base del robot. Realizando las respectivas multiplicaciones de las matrices **A** que se determinaron anteriormente, se logrará llegar a la siguiente expresión:

$$T = \begin{bmatrix} C_1C_{234} & -C_1S_{234} & S_1 & l_4C_1C_{234} + l_3C_1C_{23} + l_2C_1C_2 \\ S_1C_{234} & -S_1S_{234} & -C_1 & l_4S_1C_{234} + l_3S_1C_{23} + l_2S_1C_2 \\ S_{234} & C_{234} & 0 & l_4S_{234} + l_3S_{23} + l_2S_2 + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

Las ecuaciones que se encuentran resaltadas en la ecuación (3.12), representan la posición del extremo del robot respecto a la base en un sistema de coordenadas rectangulares (x_0, y_0, z_0). Si se las contrasta con las ecuaciones (3.5), (3.6) y (3.7) obtenidas a través de métodos geométricos, se puede comprobar que son exactamente iguales. La diferencia más significativa es el grado de dificultad en la resolución, la cual será más evidente cuando el número de GDL aumente, pero para el caso del robot antropomórfico de 4 GDL del cual se está realizando el estudio, puede realizarse con cualquier método, pero es evidente que el método geométrico es el más fácil de realizar. En éste caso se realizaron los dos procedimientos para validar las ecuaciones, puesto que serán un punto de partida para la resolución del problema cinemático inverso.

3.3 EL PROBLEMA CINEMÁTICO INVERSO

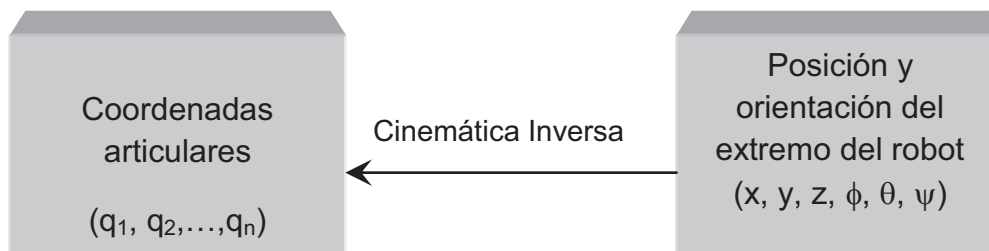


Figura 3.6 Representación de la Cinemática Inversa

El objetivo del problema cinemático inverso consiste en encontrar los valores que deben adoptar las coordenadas articulares del robot $q = [q_1, q_2, \dots, q_n]^T$ para que su extremo se posicione y oriente según una determinada localización espacial (\mathbf{p} , $[\mathbf{n}, \mathbf{o}, \mathbf{a}]$).

A diferencia del problema cinemático directo, el cual se puede abordar de una manera sistemática a partir de la utilización de matrices de transformación homogénea, e independientemente de la configuración del robot; en el problema cinemático inverso el procedimiento de obtención de las ecuaciones es fuertemente dependiente de la configuración del robot.

Existen algunos métodos genéricos susceptibles de ser programados, de modo que un computador pueda, a partir de la cinemática directa del robot (con sus parámetros de Denavit-Hartenberg, por ejemplo) obtener el conjunto de 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 en sí no está siempre garantizada.

A la hora de resolver el problema cinemático inverso es mucho más adecuado encontrar una solución cerrada. Esto es, encontrar una relación matemática explícita de la forma:

$$q_k = f_k(x, y, z, \phi, \theta, \psi) \quad (3.13)$$

$$K = 1 \dots n \text{ (GDL)}$$

Este tipo de solución presenta, entre otras, las siguientes ventajas:

1. En muchas aplicaciones, el problema cinemático inverso ha de resolverse en tiempo real (por ejemplo, en el seguimiento de una determinada trayectoria). Una solución de tipo iterativo no garantiza tener la solución en el momento adecuado.
2. Al contrario de lo que ocurría en el problema cinemático directo, con cierta frecuencia la solución del problema cinemático inverso no es única; existiendo diferentes conjuntos $[q_1, \dots, q_n]^T$ que posicionan y orientan el extremo del robot del mismo modo. Para evitar esto se debe usar una solución cerrada, puesto que permite incluir determinadas reglas o restricciones que aseguren que la solución obtenida sea la más adecuada de entre las posibles (por ejemplo, límites en los recorridos articulares).

No obstante, a pesar de las dificultades comentadas, la mayor parte de los robots poseen cinemáticas relativamente simples como es el caso de robot antropomórfico que facilitan en cierta medida la resolución del problema cinemático inverso.

El robot del que trata el presente trabajo es del tipo antropomórfico con 4 GDL, donde tres GDL forman parte de la posición del extremo, mientras que el último es parte la orientación. Para facilitar el análisis cinemático inverso de este robot, se pueden considerar solamente los tres GDL correspondientes a la posición, dejando de lado al GDL correspondiente a la orientación, ya que este será considerado más adelante como un vector de offset para el posicionamiento.

Por lo tanto para el caso citado, es posible considerar las pautas generales antes descritas que permitan plantear y resolver el problema cinemático inverso de una manera sistemática.

Los métodos geométricos permiten normalmente, obtener los valores de las primeras variables articulares, que son las que consiguen posicionar el robot (prescindiendo de la orientación). Para ello se utilizan relaciones trigonométricas y geométricas sobre los elementos del robot. Se suele recurrir a la relación de triángulos formados por elementos y articulaciones.

Como alternativa para resolver el mismo problema se puede recurrir a manipular directamente las ecuaciones correspondientes al problema cinemático directo, puesto que éste establece la relación:

$$\begin{bmatrix} \mathbf{n} & \mathbf{o} & \mathbf{a} & \mathbf{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = [t_{ij}] \quad (3.14)$$

Donde los elementos t_{ij} son función de las coordenadas articulares $[q_1, \dots, q_n]^T$. Es posible pensar que mediante ciertas combinaciones de las 12 ecuaciones planteadas se puedan despejar las 'n' variables articulares q_i en función de las componentes de los vectores \mathbf{n} , \mathbf{o} , \mathbf{a} y \mathbf{p} . Debe considerarse en este caso que en general las 12 ecuaciones responden a ecuaciones trigonométricas acopladas cuya resolución no es trivial.

Otra forma de resolver el problema cinemático inverso es utilizar el método de desacoplo cinemático, el cual permite, para determinados robots, resolver los primeros grados de libertad, dedicados al posicionamiento, de manera independiente a la resolución de los últimos grados de libertad, dedicados a la orientación. Cada uno de estos dos problemas más simples, podrán ser tratados por cualquier tipo de procedimiento.

Para el desarrollo de un sistema de control en tiempo real, debe resolverse el problema cinemático inverso por métodos que permitan llegar a una solución cerrada, ya que un tipo de solución numérica-iterativa, no es aplicable para este tipo de control.

Para obtener una solución cerrada se debe encontrar una relación matemática explícita con la forma de la ecuación (3.13).

Para encontrar una solución cerrada, se pueden tomar dos caminos, uno mediante métodos geométricos y otro mediante métodos algebraicos.

La resolución por métodos algebraicos se logra a través de la matriz de transformación homogénea y de la resolución de ecuaciones no lineales trigonométricas, lo cual implica un significativo grado de dificultad respecto al geométrico, el cual utiliza relaciones geométricas de triángulos formados por la

propia estructura del robot, el cual es mucho más sencillo de resolver en muchas morfologías de robots de pocos grados de libertad.

3.3.1 RESOLUCIÓN DEL PROBLEMA CINEMÁTICO INVERSO POR MÉTODOS GEOMÉTRICOS

Como se ha indicado, este procedimiento es adecuado para robots de pocos grados de libertad o para el caso de que se consideren sólo los primeros grados de libertad, dedicados a posicionar el extremo.

El procedimiento en sí se basa en encontrar un suficiente número de relaciones geométricas en las que intervendrán las coordenadas del extremo del robot, sus coordenadas articulares y las dimensiones físicas de sus elementos.

Lo que se desea obtener es el extremo del robot en un punto y orientación específica del espacio. El número de ecuaciones será igual al número de grados de libertad que tenga el robot, ya que se busca conocer los valores que deben adoptar las variables articulares para que el robot tome una determinada forma para lograr posicionar y orientar su extremo [1].

$$q = [q_1, q_2, q_3, q_4]^T \quad (3.15)$$

Lo primero que se va a obtener es q_1 , a partir de la Figura 3.7.

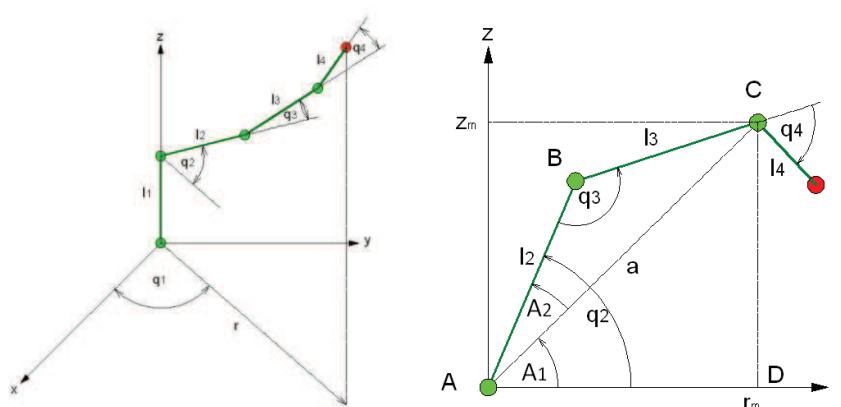


Figura 3.7 Disposición de las coordenadas articulares del robot antropomórfico

$$q_1 = \text{arctag} \left(\frac{y}{x} \right) \quad (3.16)$$

Donde q_1 puede tomar cualquier valor, pero estaría restringido físicamente al rango de valores que puede tomar el servomotor, el cual es 180° . Es decir:

$$0^\circ < q_1 < 180^\circ$$

La orientación del último eslabón es la suma de las variables articulares de los eslabones precedentes, que se encuentran agrupados en un solo plano de movimientos (planar), como se puede apreciar en la Figura 3.8.

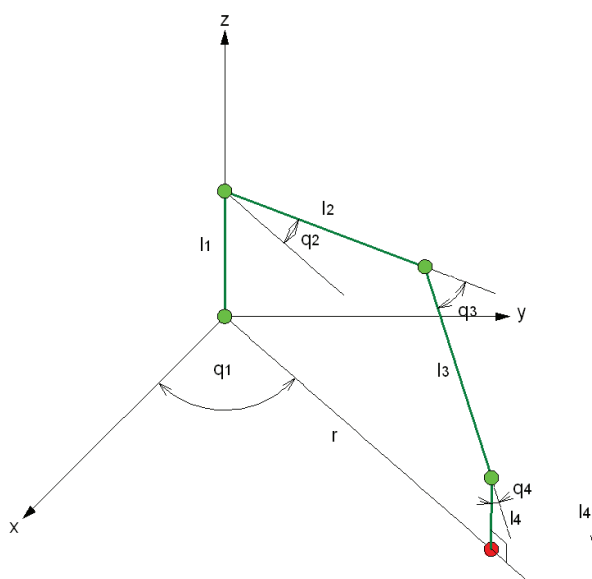


Figura 3.8 Orientación de la última extremidad del robot antropomórfico

$$\theta = q_2 + q_3 + q_4 \quad (3.17)$$

Para empezar a plantear el problema cinemático inverso, lo primero es encontrar todas las relaciones geométricas posibles.

Antes de empezar con el planteamiento de las ecuaciones necesarias, se debe tener en cuenta que el último eslabón cumple la única función de orientar la herramienta final, que en este caso es el electroimán, el cual se desea que se aproxime a su destino de manera perpendicular, lo cual implica que la orientación que se desea obtener es constante, lo que va a facilitar de manera importante la resolución.

Para simplificar la resolución del problema cinemático, se va a reducir el número de GDL, a través de la eliminación matemática del último eslabón por medio de la resta de coordenadas mediante offsets previamente calculados:

$$z_{offset} = l_4 \cdot \text{sen}(\theta) \quad (3.18)$$

$$r_{offset} = l_4 \cdot \text{cos}(\theta) \quad (3.19)$$

Donde θ representa el ángulo que forma la muñeca con el plano de trabajo, que para éste caso será de -90° .

Tomando en cuenta éstos valores de offset, se puede determinar la posición de la muñeca, la cual será:

$$r = \sqrt{x^2 + y^2} \quad (3.20)$$

$$z_m = z - z_{offset} - l_1 \quad (3.21)$$

$$r_m = r - r_{offset} \quad (3.22)$$

Tomando como referencia la Figura 3.7, se puede tomar el triángulo ΔACD , del cual se puede obtener lo siguiente:

$$a = \sqrt{r_m^2 + z_m^2} \quad (3.23)$$

$$A1 = \text{atan}\left(\frac{z_m}{r_m}\right) \quad (3.24)$$

Para determinar A2, se toma el triángulo ΔABC :

$$A2 = \text{acos}\left(\frac{l_2^2 - l_3^2 + a^2}{2l_2a}\right) \quad (3.25)$$

Finalmente, con las ecuaciones anteriores es posible determinar las ecuaciones de cada una de las articulaciones restantes para el robot:

$$q_2 = A1 + A2 \quad (3.26)$$

Sustituyendo las expresiones (3.24) y (3.25), en la ecuación (3.26)

$$q_2 = \operatorname{atan}\left(\frac{z_m}{r_m}\right) + \operatorname{acos}\left(\frac{l_2^2 - l_3^2 + a^2}{2l_2 a}\right) \quad (3.27)$$

Reemplazando las ecuaciones (3.20), (3.21) y (3.22), en (3.27):

$$q_2 = \operatorname{atan}\left(\frac{z - z_{\text{offset}} - l_1}{r - r_{\text{offset}}}\right) + \operatorname{acos}\left(\frac{l_2^2 - l_3^2 + r_m^2 + z_m^2}{2l_2 \sqrt{r_m^2 + z_m^2}}\right) \quad (3.28)$$

Con las ecuaciones (3.18), (3.21) y (3.22), y reemplazando en la ecuación (3.28) se tiene lo siguiente:

$$q_2 = \operatorname{atan}\left(\frac{z - l_4 \cdot \operatorname{sen}(\theta) - l_1}{r - l_4 \cdot \operatorname{cos}(\theta)}\right) + \operatorname{acos}\left(\frac{l_2^2 - l_3^2 + (r - r_{\text{offset}})^2 + (z - z_{\text{offset}} - l_1)^2}{2l_2 \sqrt{(r - r_{\text{offset}})^2 + (z - z_{\text{offset}} - l_1)^2}}\right) \quad (3.29)$$

Y finalmente sustituyendo la expresión (3.29) por las ecuaciones (3.18) y (3.19):

$$q_2 = \operatorname{atan}\left(\frac{z - l_4 \cdot \operatorname{sen}(\theta) - l_1}{r - l_4 \cdot \operatorname{cos}(\theta)}\right) + \operatorname{acos}\left(\frac{l_2^2 - l_3^2 + (r - l_4 \cdot \operatorname{cos}(\theta))^2 + (z - l_4 \cdot \operatorname{sen}(\theta) - l_1)^2}{2l_2 \sqrt{(r - l_4 \cdot \operatorname{cos}(\theta))^2 + (z - l_4 \cdot \operatorname{sen}(\theta) - l_1)^2}}\right) \quad (3.30)$$

$$0^\circ < q_2 < 180^\circ$$

La ecuación que corresponde a q_3 , se puede determinar mediante el triángulo $\triangle ABC$:

$$q_3 = \operatorname{acos}\left(\frac{l_2^2 + l_3^2 - a^2}{2l_2 l_3}\right) \quad (3.31)$$

La expresión (3.31) se reemplaza por la ecuación (3.23):

$$q_3 = \operatorname{acos}\left(\frac{l_2^2 + l_3^2 - (r - r_{\text{offset}})^2 - (z - z_{\text{offset}} - l_1)^2}{2l_2 l_3}\right) \quad (3.32)$$

Y finalmente, para dejar la expresión con términos conocidos se reemplazan las ecuaciones (3.18) y (3.19):

$$q_3 = \operatorname{acos}\left(\frac{l_2^2 + l_3^2 - (r - l_4 \cdot \operatorname{cos}(\theta))^2 - (z - l_4 \cdot \operatorname{sen}(\theta) - l_1)^2}{2l_2 l_3}\right) \quad (3.33)$$

$$0^\circ < q_3 < 180^\circ$$

Para el último grado de libertad, únicamente se considera el ángulo para orientar al extremo de forma perpendicular al área de trabajo, por lo tanto se tiene:

$$q_4 = -90 - q_2 - q_3 \quad (3.34)$$

Debido al montaje de los servomotores, es necesario hacer un ajuste cinemático, que se basará a las referencias expuestas en la Figura 3.9.

$$q_1 = q_1 \quad (3.35)$$

$$q_2 = q_2 \quad (3.36)$$

$$q_3 = 180 + q_3 \quad (3.37)$$

$$q_4 = 90 + q_4 \quad (3.38)$$

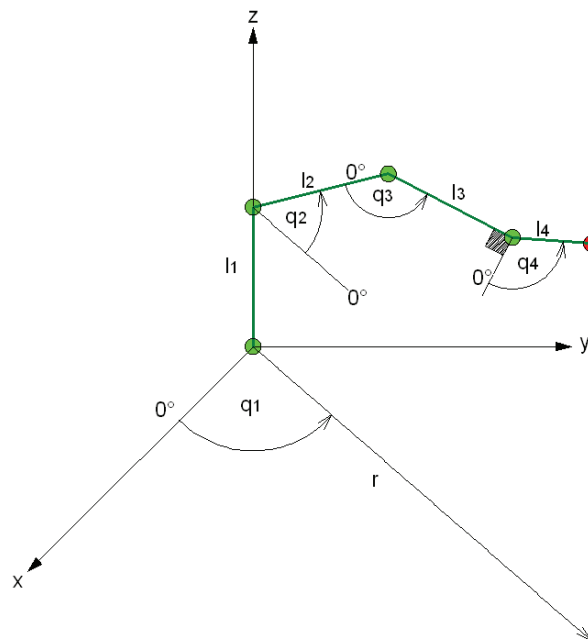


Figura 3.9 Referencias angulares para cada una de las articulaciones del robot

Con esto se finaliza el cálculo de las expresiones correspondientes a la cinemática inversa (q_1, q_2, q_3, q_4) en función de la posición y de los parámetros físicos del robot.

Mediante la utilización de una hoja de cálculo de Microsoft EXCEL, se pudieron ingresar las ecuaciones de cinemática inversa y su cálculo para diferentes valores

de (x, y, z); mediante el cual también se logró graficar la cadena cinemática del robot, a través del cálculo de cada una de las posiciones de las articulaciones del robot.

También se consideraron algunas restricciones como las siguientes:

$$Distancia_{Hombro - Muñeca} < l_1 + l_2$$

Además se debe tomar en cuenta que 'z' y 'y' solamente pueden tomar valores positivos o iguales a cero.

$$Distancia_{Hombro - Muñeca} < l_1 + l_2 \quad \wedge \quad z \geq 0 \quad \wedge \quad y \geq 0$$

Ejemplo, para valores de (x, y, z) = (140; 0; 10) mm

		rad	sin	cos	Altura	Profundidad
Base	0,00	0,0000	0,0000	1,0000		
Altura Base	76				76,0000	0,0000
Hombro	49,19	0,8586	0,7569	0,6535		
Antebrazo	122				92,3464	79,7254
Codo	-119,87	-2,0922	-0,9437	0,3309		
Brazo	91,5				-86,3464	30,2746
Muñeca	-19,32	-0,3372	-1,0000	0,0000		
Mano	72				-72,0000	0,0000
					10,0000	110,0000
Los valores en AMARILLO deben ser ingresados						
Valores calculados					Ubicación de cada punto en el espacio	
Constantes					z	r
					Base	0
					Hombro	76,0000
					Codo	168,3464
					Muñeca	82,0000
					Extremo	10,0000

BASE	HOMBRO	CODO	MUÑECA
0,00	49,19	60,13	70,68

Figura 3.10 Ejemplo Cinemática Inversa

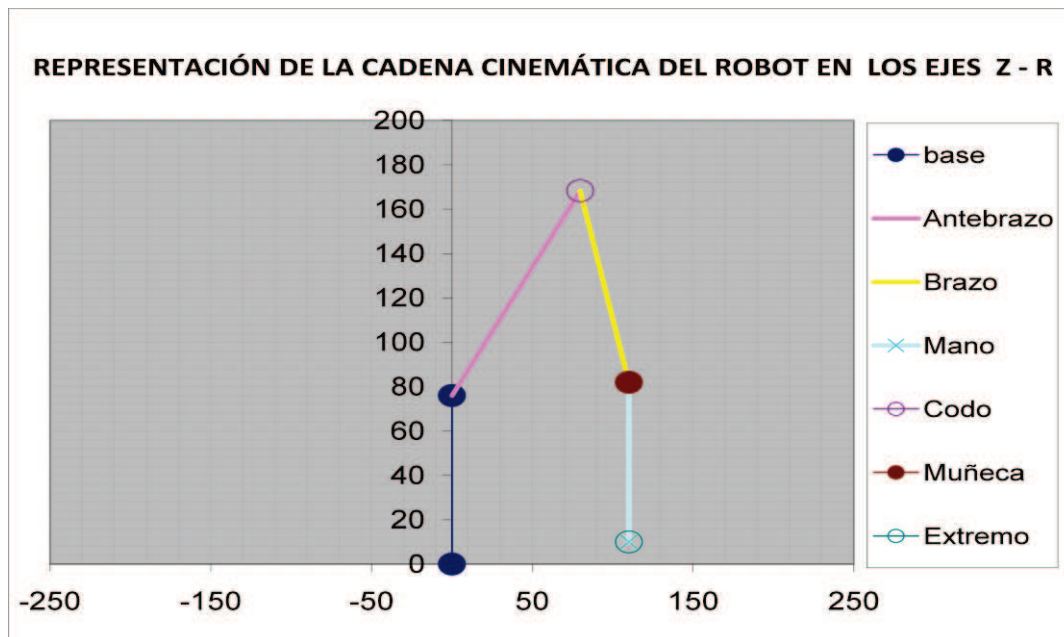


Figura 3.11 Gráfica obtenida en base a los cálculos de cinemática inversa realizados en Microsoft EXCEL.

3.4 DESARROLLO DEL SOFTWARE DE CONTROL EN LabVIEW

Para el control del robot se ha desarrollado un software dedicado dividido en dos partes; la primera con el fin de realizar el control de posición del robot, y la segunda dedicada a la adquisición, procesamiento de imagen e interpretación de la información contenida en ella.

Ambas partes del software comparten información que permite sincronizar el funcionamiento del sistema en conjunto. Esta información principalmente consiste en setpoints, alarmas, y selectores de funcionamiento.

Se expondrá cada parte del software por separado debido a la inherente diferencia existente entre cada una de ellas

A continuación se muestra el diagrama de flujo general de todo el sistema de control.

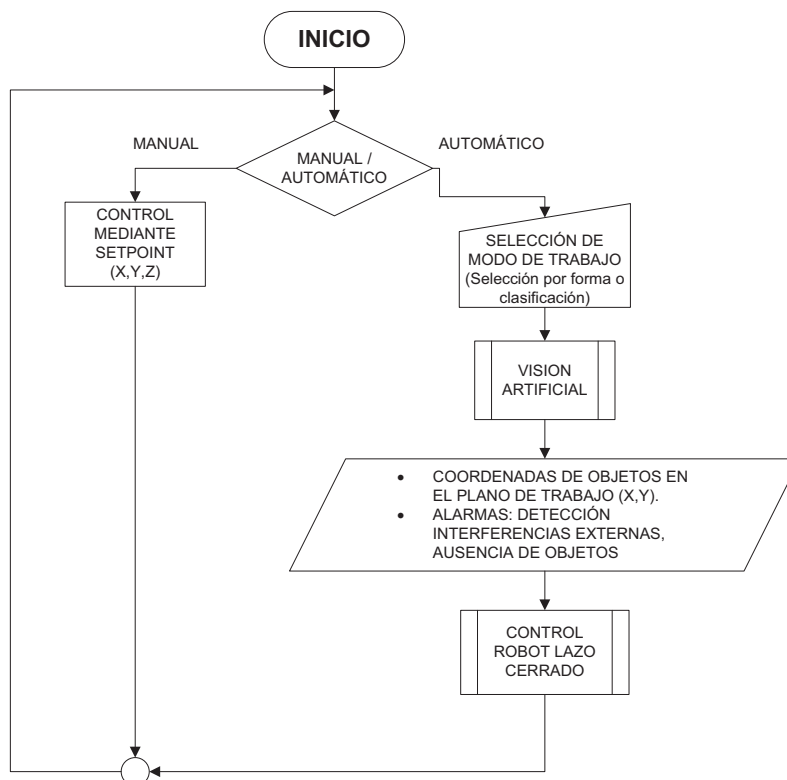


Figura 3.12 Diagrama de Flujo General

A continuación se explicará cada una de las partes constitutivas del software de control.

3.4.1 SOFTWARE PARA PROCESAMIENTO DE IMAGEN

El software para procesamiento de imagen fue desarrollado completamente mediante LabVIEW con ayuda del Toolkit de Vision and Motion. A continuación se presenta el diagrama de flujo del software de visión artificial desarrollado.

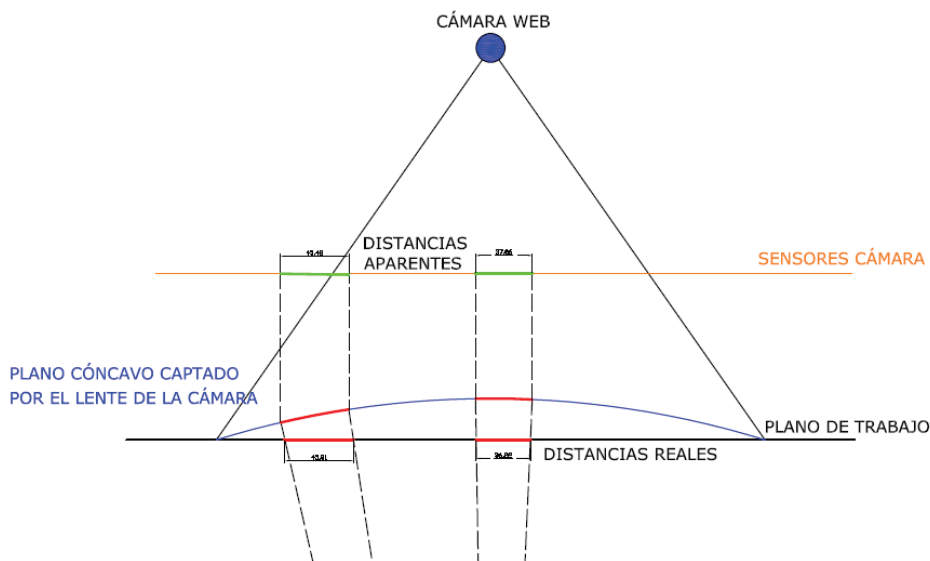


Figura 3.14 Esquema de deformación esférica de la imagen

Para la compensación de la deformación esférica, se utilizan funciones de calibración de LabVIEW; para ello, primero se debe tener una imagen de referencia que indique el grado de deformación existente.

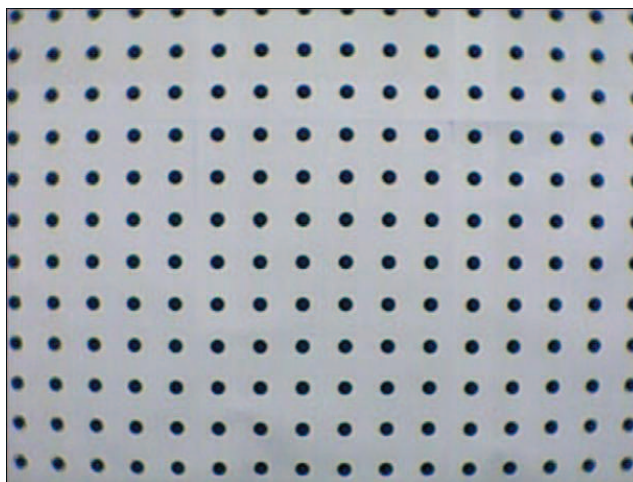


Figura 3.15 Imagen de referencia para corregir la deformación esférica

Se puede usar una imagen de una matriz de puntos, e ingresar a la función de calibración los valores correspondientes a las distancias horizontal y vertical reales entre los centros de los puntos, de tal forma que la función de LabVIEW pueda obtener el grado de deformación existente en base a la comparación de la imagen de referencia y la imagen real a ser capturada.

Luego de realizar esta corrección, se procede a desarrollar un algoritmo de conversión píxel-cm; para ello primero se colocan dos objetos en el área de trabajo que tienen posiciones xy conocidas (-100,200 y 100,200). Luego al capturar la imagen, el software identifica los centroides de los objetos en unidades de píxeles y mediante la información de la posición xy real de los objetos, el software desarrolla el algoritmo de conversión píxel-cm.

Posteriormente se realiza un proceso de auto-calibración de iluminación, el cual analiza durante un cierto tiempo los niveles de iluminación existentes a lo largo de líneas de perfil ubicadas en los bordes de la imagen captada.

El análisis de línea de perfil evalúa el nivel de gris a lo largo de la línea, de tal forma que puede mostrar una curva de respuesta de la siguiente forma.

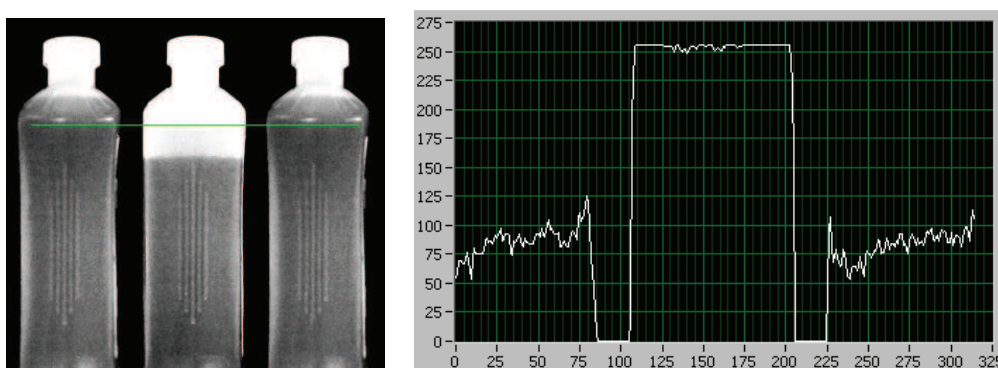


Figura 3.16 Análisis de Línea de Perfil, tomado de [6]

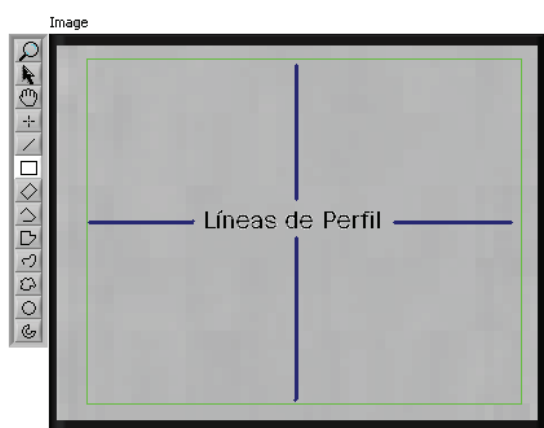


Figura 3.17 Líneas de perfil ubicadas sobre imagen adquirida

Al final de este proceso, se saca el promedio del nivel de iluminación detectado sobre cada línea de perfil, de tal forma que estos valores sirvan de referencia para

la posterior detección de interferencias o cambios bruscos de iluminación en la escena.

Los procesos antes descritos se realizan una sola vez en el software de visión artificial, dado que únicamente contribuyen a establecer parámetros o algoritmos de referencia.

La siguiente etapa del software de visión artificial, requiere de participación del usuario, el cual selecciona el programa a usar y su respectivo modo de trabajo. Se han desarrollado dos programas para Visión Artificial, uno para 'Selección de Forma' y otro para 'Clasificación de Objetos'.

En el primer programa hay dos modos de trabajo, el primero obtiene coordenadas de todos los objetos presentes en el área de trabajo, mientras que el segundo obtiene coordenadas únicamente de la forma seleccionada por el usuario. El segundo programa tiene solo un modo de funcionamiento en donde el software organiza las coordenadas de cada objeto en grupos correspondientes a las diferentes formas (círculo, cuadrado, triángulo, etc.).

Posteriormente, el software realiza la adquisición de imagen, mejoramiento espacial de la imagen adquirida mediante filtros Laplacianos, y posteriormente su binarización y segmentación basada en morfología.

La segmentación aplicada consiste de los siguientes pasos.

- Dilatación de la imagen para llenar agujeros.
- Rechazar objetos que estén en contacto con el borde de la imagen.
- Erosión para eliminar ruido en la imagen.
- Remoción de pequeñas partículas en base a filtros de morfología.

Estas operaciones permiten distinguir mejor las formas y posición de los objetos presentes en la escena captada.

Una vez realizado el proceso de segmentación, se obtiene información específica de la imagen procesada, como por ejemplo los centroides de cada uno de los objetos presentes en la escena.

Adicionalmente se puede realizar una búsqueda dentro de los objetos detectados; esta búsqueda consiste en identificar o clasificar a los objetos en función de su forma, para lo cual se compara cada objeto detectado con imágenes de referencia con las diferentes formas a utilizar. Para ello, se captura la imagen de un objeto con una determinada forma, la cual posteriormente será utilizada como referencia.

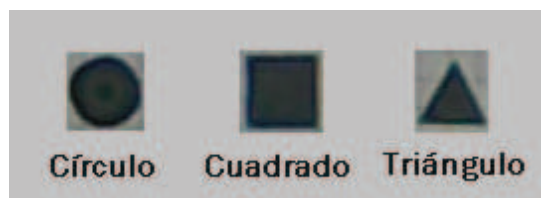


Figura 3.18 Figuras de Referencia

Una vez que la imagen ha sido procesada teniendo en cuenta el modo de operación seleccionado por el usuario, este debe presionar el botón 'Ejecutar' para enviar las coordenadas de los objetos detectados al sistema de control de posición.

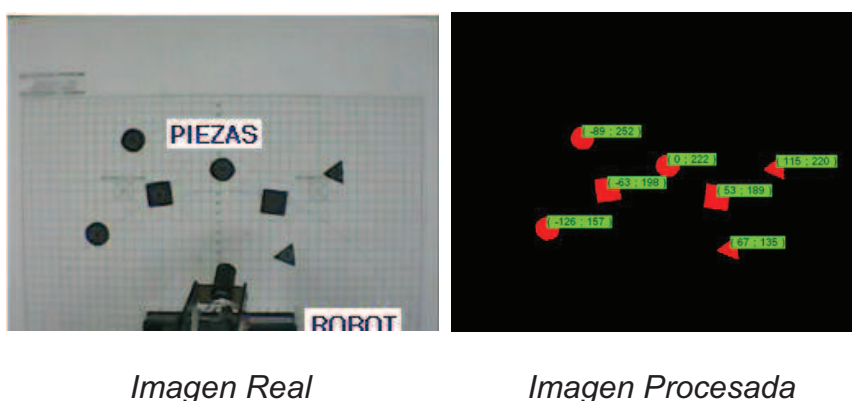


Figura 3.19 Imagen procesada

En caso de que no haya ningún objeto presente en el área de trabajo, el sistema activa una alarma, que es enviada al software de control posición y adicionalmente muestra en pantalla el mensaje de la Figura 3.20.

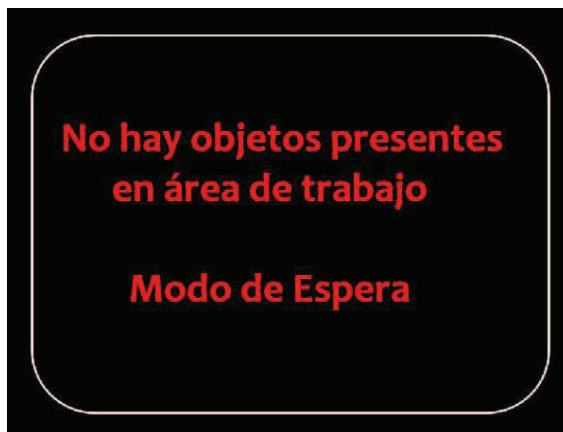


Figura 3.20 Mensaje de ausencia de objetos

Finalmente, el software realiza un escaneo de la escena para determinar si hay objetos extraños dentro de la misma. Para ello primero se realiza una nueva adquisición de imagen, se la convierte a escala de grises y luego se realiza un análisis de niveles de iluminación sobre las líneas de perfil ubicadas en los bordes de la imagen.

Mediante este análisis se puede determinar si algún objeto extraño ingresa al área de trabajo, debido a que perturba la distribución de luz a lo largo de una o más líneas de perfil en los bordes de la imagen. Para realizar este efecto, se toman como referencia los niveles de iluminación promedio obtenidos en el proceso de auto-calibración de iluminación que se realiza al principio de la ejecución del software de visión artificial.

En caso de que algún objeto extraño o interferencia sea detectada se envía una alarma al sistema de control de posición para detener la operación del robot, y además se muestra el mensaje de la Figura 3.21.

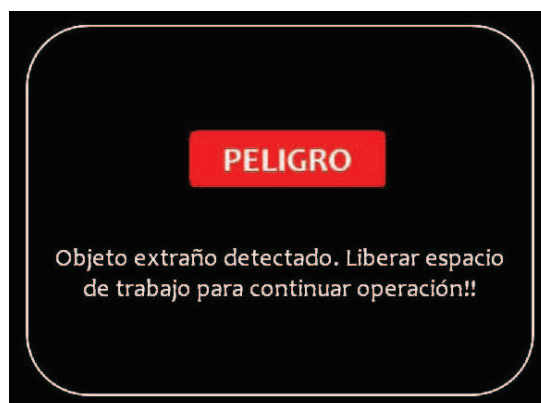


Figura 3.21 Mensaje de objetos extraños detectados

Luego de cumplir con este análisis, el software repite todos los procesos antes descritos secuencialmente, excepto los referentes a corrección de deformación esférica, conversión píxel-cm y auto-calibración de iluminación.

Como ya se mencionó anteriormente, en este proyecto se han desarrollado dos programas para la parte de visión artificial; a continuación se muestran las ventanas que el usuario percibe mientras el software se ejecuta en cada uno de los programas diseñados.

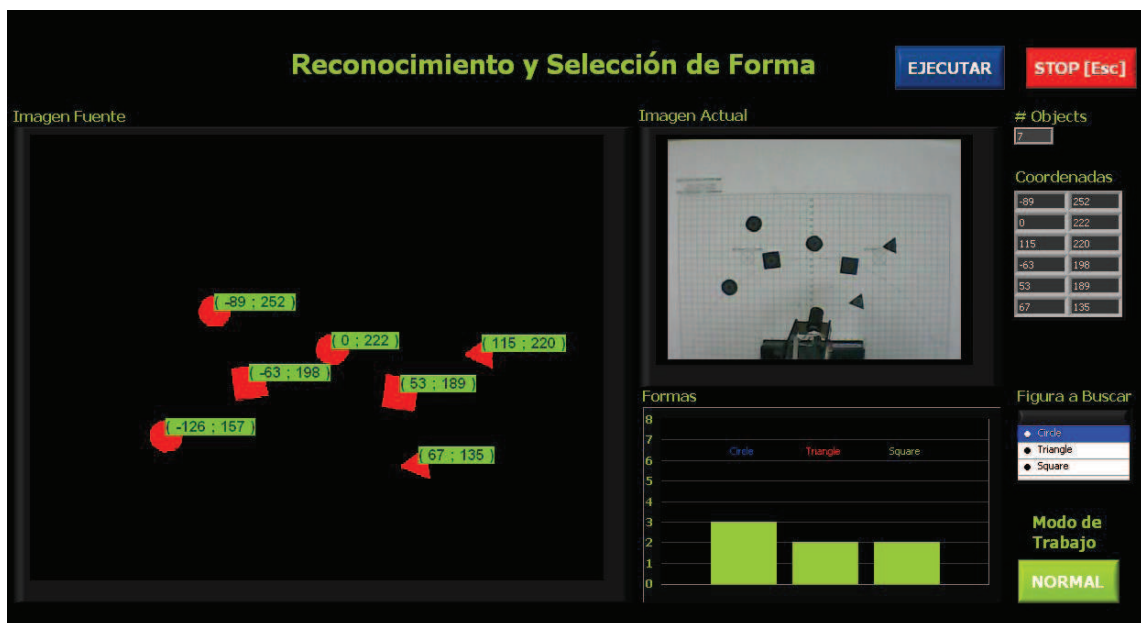


Figura 3.22 HMI del Primer Software de Visión Artificial, modo NORMAL

En la Figura 3.22 se muestra el primer software en uno de sus dos modos de trabajo (NORMAL). En este modo el software identifica todos los objetos presentes en el área de trabajo, calcula sus coordenadas y las coloca en el arreglo llamado “Coordenadas”, que posteriormente envía estos datos al software de control de posición.

Este HMI tiene la siguiente distribución:

- En la parte izquierda se muestra la imagen procesada con las coordenadas de los objetos detectados.
- En la parte derecha se muestra la imagen real adquirida.

- En la extrema derecha se muestra el arreglo de coordenadas de los objetos detectados que están dentro del alcance físico del robot (cumplen con restricciones de cinemática).
- En la parte inferior se muestra una distribución estadística de los objetos presentes en función de su forma.
- En la parte extrema inferior derecha, se encuentra un botón que permite seleccionar el modo de trabajo entre NORMAL y FORMAS.
- En la parte superior derecha se encuentran los botones “EJECUTAR” y “STOP”.

El segundo modo de trabajo del primer software de visión se muestra en la siguiente figura. En este modo el usuario puede escoger que piezas trasladar; es decir, el usuario selecciona la forma del objeto y luego el software coloca en el arreglo de “Coordenadas” los datos correspondientes únicamente a los objetos que cumplen con la forma seleccionada, los cuales se muestran dentro de un recuadro.

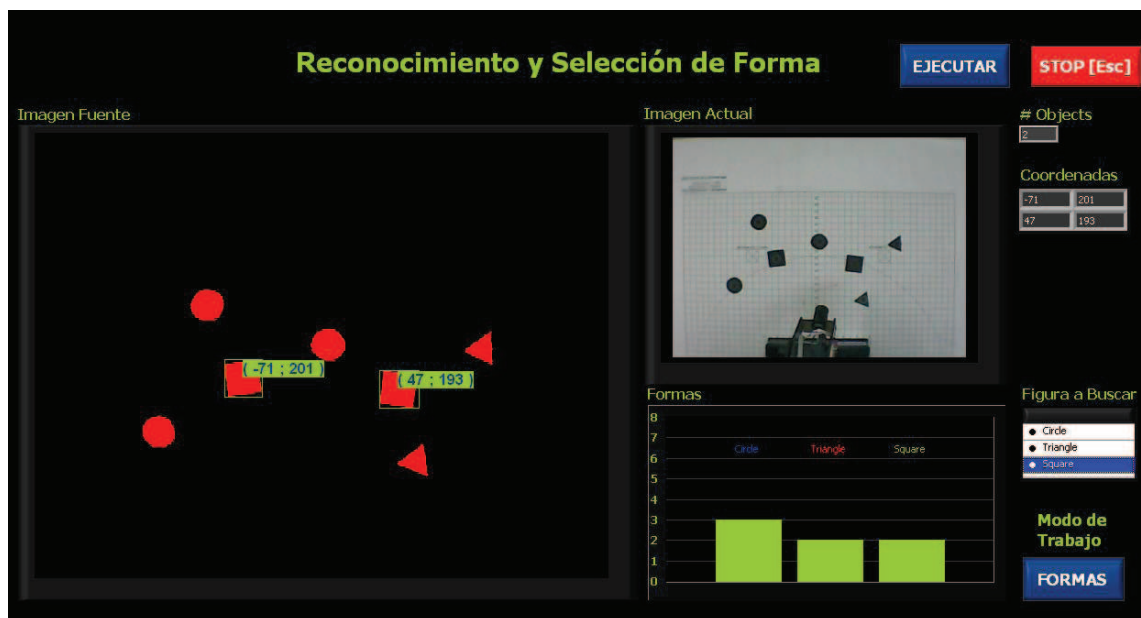


Figura 3.23 HMI de Primer Software de Visión Artificial, modo FORMAS

El segundo software de visión artificial permite clasificar los objetos presentes en el área de trabajo en función de su forma. El software muestra un recuadro de

colores diferentes alrededor de los objetos, en función de la figura a la que corresponden.

En este programa se muestran tres arreglos de coordenadas en la parte derecha; uno correspondiente a cada una de las formas preestablecidas. Los datos de las coordenadas en cada uno de los arreglos, son enviados al software de control de posición, el cual ejecuta el proceso de clasificación.

En general el HMI de este programa es similar al anterior, diferenciándose únicamente en que no posee un botón para selección de modo debido a que solo trabaja en el modo de clasificación.



Figura 3.24 HMI del Segundo Software de Visión Artificial

3.4.2 SOFTWARE PARA CONTROL DE POSICIÓN

El software de control de posición fue desarrollado en su totalidad mediante LabVIEW, con ayuda del Toolkit *Control, Design & Simulation*. En la Figura 3.25 se presenta el diagrama de flujo del software de control de posición desarrollado.

Como parte inicial, el software de control de posición configura parámetros esenciales como por ejemplo la comunicación serial con el controlador de servos, y además parámetros del sistema mecánico como son velocidades de los servomotores y referencias para ubicar al robot en una posición inicial.

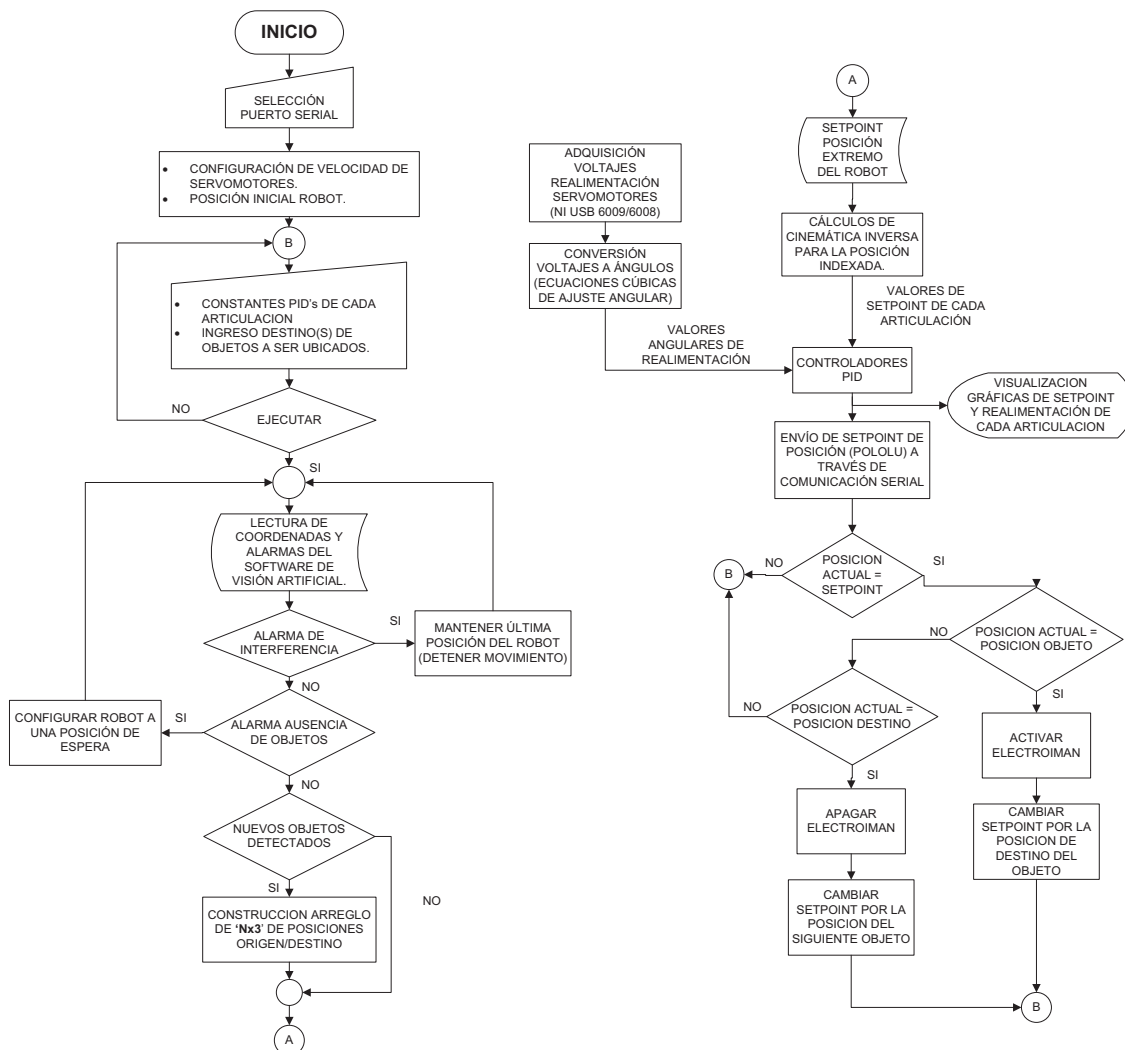


Figura 3.25 Diagrama de Flujo del Software de Control de Posición

Tabla 3.2 Configuración de parámetros para el control de posición

Parámetro	Valor
Velocidad Comunicación Serial	28800 bps
Bits de Datos	8
Bits de Parada	1
Paridad	Ninguna
Velocidad Servo Base	15*
Velocidad Servos Hombro	10*
Velocidad Servo Codo	10*
Velocidad Servo Muñeca	20*

* La velocidad de los servomotores se configura mediante el envío serial de un valor entre 0 y 127 al controlador de servomotores. El valor de '0' implica que el servomotor no se mueve; mientras que el valor de '127' representa la velocidad máxima. Para variar la velocidad, el controlador cambia el periodo del tren de pulsos aplicado al servomotor, manteniendo constante la relación de trabajo.

Para ubicar al robot en la posición inicial se envían referencias angulares en lazo abierto a las articulaciones. Estos valores angulares se muestran en la Tabla 3.3.

Tabla 3.3 Valores angulares para la posición inicial del robot

Articulación	Ángulo
Base	90°
Hombro	90°
Codo	60°
Muñeca	30°

Una vez realizada esta configuración, el software de control de posición espera hasta que el software de visión artificial termine de realizar la auto-calibración de iluminación. Una vez terminado este proceso, y cuando el usuario haya presionado el botón de 'Ejecutar', el software de control de posición pasa a la etapa de control en lazo cerrado para cada uno de los servomotores que componen las articulaciones del robot.

Inicialmente el software analiza el estado de las alarmas enviadas por el programa de visión artificial, referentes a 'Ausencia de Objetos' y a 'Detección de Interferencias'.

En caso de no haber objetos en el plano de trabajo, el robot se dirige a la posición establecida como inicial o por defecto (que ya fue mencionada anteriormente), hasta que el estado de la alarma cambie, indicando que ya hay objetos detectados.

En caso de que existan interferencias en el plano de trabajo, el control de posición bloquea el movimiento del robot manteniendo su posición actual hasta que la alarma cambie de estado, indicando que ya no hay objetos extraños dentro del área de trabajo, con lo cual el robot continúa su movimiento para cumplir con la última referencia dada.

Además de la lectura de alarmas, el programa lee las coordenadas de los objetos detectados, y posteriormente pasa a la lectura de las coordenadas de destino ingresadas por el usuario. Una vez que el programa cuenta con las coordenadas de los objetos y de el/los destino(s), se realiza una reorganización de los puntos

en el espacio hacia los cuales se dirigirá el extremo del robot. El proceso de reorganización se realiza mediante un SubVI, y se muestra a continuación.

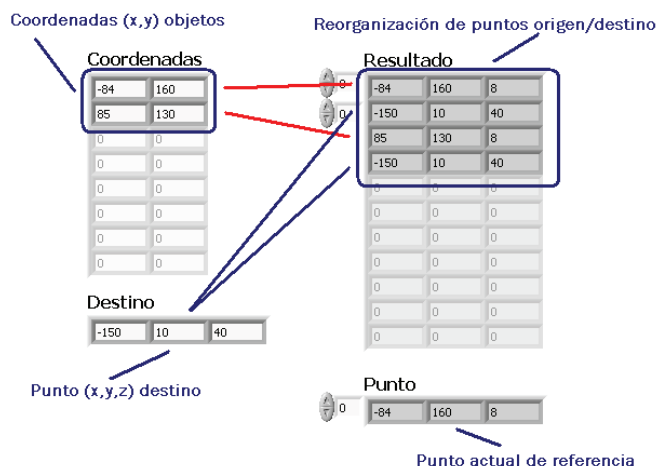


Figura 3.26 Arreglo de Posiciones Origen/Destino

En la Figura 3.26, el arreglo 'Punto' representa la referencia de posición (x, y, z) a la cual se debe dirigir el extremo del robot; este arreglo toma uno por uno los puntos que forman el arreglo 'Resultados' que contiene la reorganización de coordenadas.

El valor de 'Punto' se actualiza o cambia cada vez que el controlador de posición indique que se ha cumplido con la referencia dada; es decir cuando el robot posicione su extremo en el lugar adecuado. Para establecer que el robot ya ha cumplido con la referencia de posición dada, se cuenta con un algoritmo que analiza si cada una de las articulaciones se ha posicionado en el ángulo correcto con un margen de error máximo de $0,5^\circ$.

Como ya se mencionó anteriormente, el arreglo 'Resultados' contiene los puntos que serán cumplidos secuencialmente por el robot. El contenido de este arreglo se actualizará solo cuando el robot haya cumplido con los puntos anteriores.

Las coordenadas entregadas por el arreglo 'Punto' son la referencia (x, y, z) hacia donde debe dirigirse el extremo del robot; pero para realizar el control de posición se deben convertir estas coordenadas espaciales a valores angulares para cada articulación, lo cual se consigue resolviendo el problema cinemático inverso.

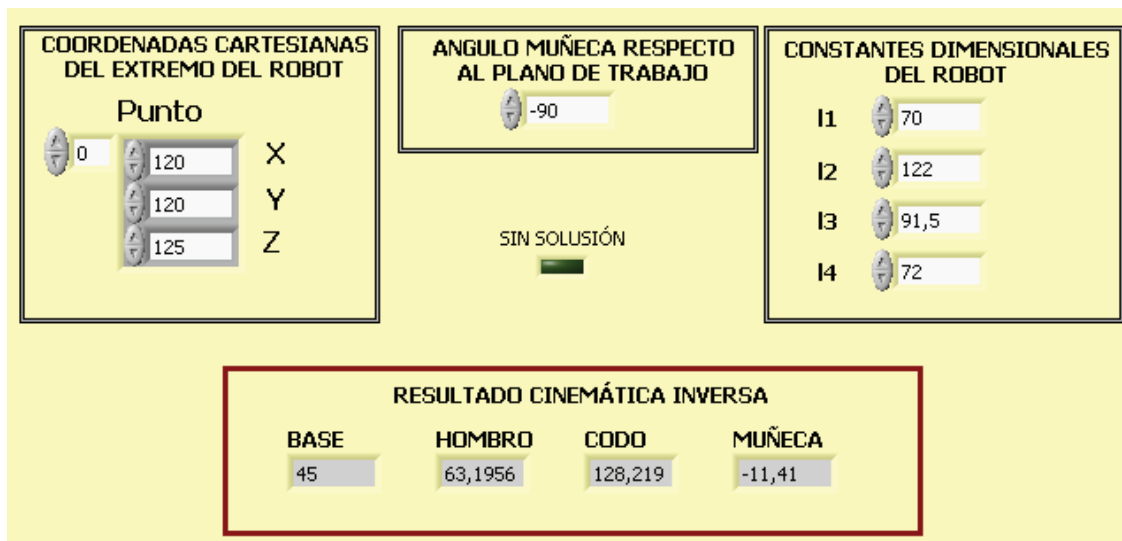


Figura 3.27 Resolución del Problema Cinemático Inverso mediante LabVIEW

Para la resolución de la cinemática inversa, se ha desarrollado un SubVI en LabVIEW, que aplica las respectivas ecuaciones y además evalúa restricciones físicas de posicionamiento, de tal forma que se pueda determinar si la posición (x, y, z) dada, puede ser alcanzada por el robot. En caso de que no pueda cumplirse con la posición deseada, el robot no tomará en cuenta este punto de referencia, y pasará a cumplir con la siguiente coordenada válida. Es decir que, si en un grupo de objetos, uno de ellos no cumple con las restricciones de la cinemática, el robot trabajará solo con los objetos dentro de su rango de trabajo, dejando de lado únicamente al objeto que está fuera de él.

Posteriormente, una vez que se han convertido las coordenadas espaciales a valores angulares, el software realiza control en lazo cerrado mediante el Toolkit *Control, Design & Simulation*.

El tipo de control implementado es 'Control en Cascada', cuyo esquema general se muestra a continuación.

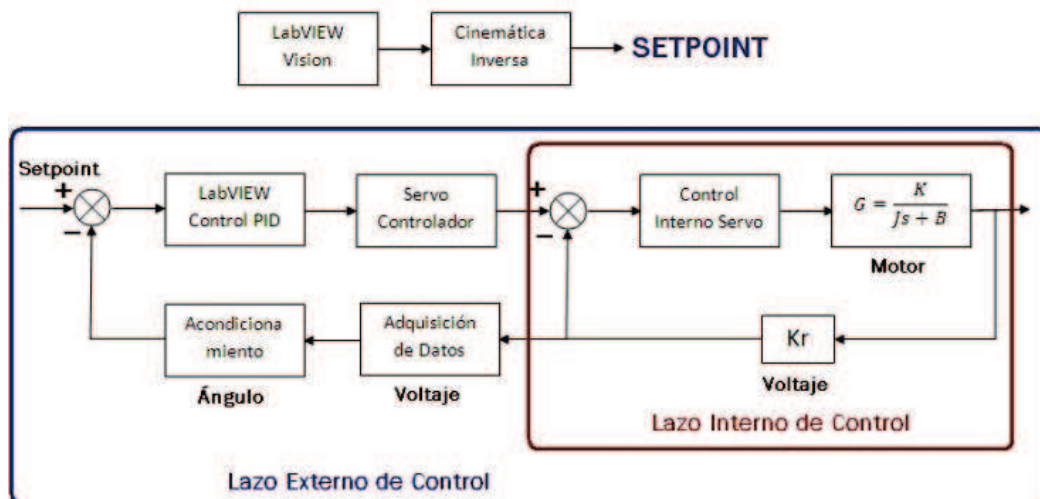


Figura 3.28 Sistema de Control Implementado

En este control, el lazo interno está previamente implementado por la placa de control interna del servomotor. Esta placa realiza un control proporcional para el posicionamiento del eje del motor, y únicamente requiere un dato de *ancho de pulso* como setpoint para operar. La señal de realimentación que utiliza es la del potenciómetro interno del servomotor.

El lazo de control externo, que es dónde propiamente se realiza el control, trabaja dando el setpoint para el lazo interno; para ello se toman las coordenadas angulares a la salida del SubVI que realiza el proceso cinemático inverso, y se ingresan estas coordenadas como referencia. La señal de realimentación que se usa es la proporcionada por el potenciómetro interno del servomotor, la cual mediante un divisor de voltaje es ingresada a la tarjeta de adquisición de datos NI USB-6009. El voltaje recibido por la tarjeta es tratado con un filtro digital en LabVIEW y acondicionado también dentro de LabVIEW mediante una ecuación de tercer grado, para convertir el voltaje en coordenada angular.

El controlador implementado en el lazo externo es del tipo PI, el cual envía serialmente los datos correspondientes a su acción de control hacia el driver de servomotores POLOLU, el cual en función del dato serial recibido, genera un tren de pulsos con un determinado ancho de pulso, que sirve de setpoint para el lazo interno de control.

El usuario puede sintonizar el sistema de control mediante la modificación de la ganancia crítica K_c y el tiempo integral T_i (en minutos) del controlador PI de LabVIEW. Debido a que la planta es un robot que debe posicionarse exactamente en la referencia dada, se debe tratar de que la respuesta del sistema sea sobre amortiguada para que el extremo del robot no exhiba sobrepasos al momento de posicionarse.

Se ha desarrollado un sistema de control como el explicado anteriormente para cada uno de los motores que constituyen las articulaciones del robot. A continuación se muestra parte del HMI de control de posición en donde se pueden ajustar las constantes del controlador.



Figura 3.29 Ajuste de ganancias de los controladores PI de cada articulación

Para el ajuste de los controladores PI, se ha tomado como punto de partida el criterio de Zieger-Nichols para sistemas en los que no se conoce la función de transferencia, y luego se ha realizado el ajuste mediante prueba y error.

Tabla 3.4 Ajuste de parámetros de Zieger-Nichols

Motor	Kc	Pc (seg)
Base	18	1,17466
Hombro Externo	18	1,1953
Hombro Interno	18	1,1673
Codo	18	1,1953
Muñeca	20	1,1816

Tipo de Controlador	Kc	Ti	Td
P	$0.5 K_{cr}$	inf	0
PI	$0.45 K_{cr}$	$1/1.2 P_{cr}$	0
PID	$0.6 K_{cr}$	$0.5 P_{cr}$	$0.125 P_{cr}$

El ajuste de ganancias puede realizarse en todo momento durante la ejecución del programa. Para el monitoreo de la respuesta de cada uno de los servomotores, el software muestra curvas de respuesta con actualización en tiempo real para cada motor.

La Figura 3.30 muestra una de las gráficas mencionadas. En la parte superior de la gráfica se muestran los valores numéricos tanto del setpoint como de la respuesta, para una mejor apreciación.

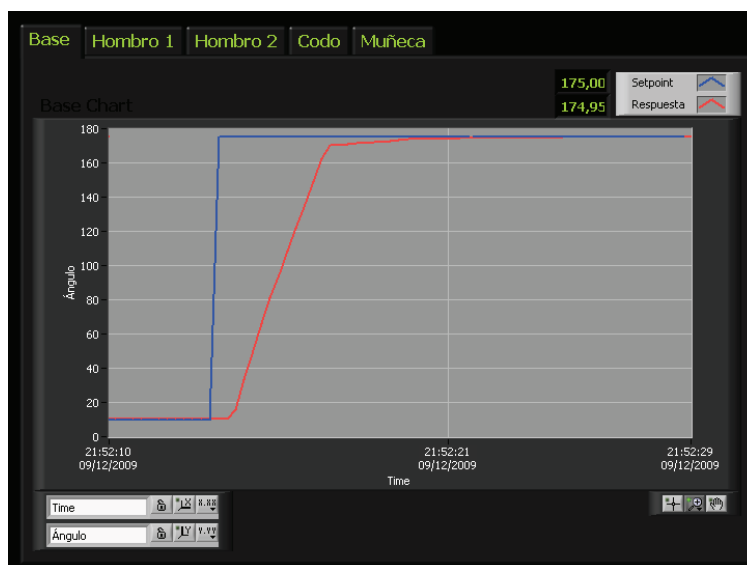


Figura 3.30 Gráfica Setpoint – Respuesta del control del servomotor de la Base

Para la correcta operación de la función PID de LabVIEW, también se debe colocar un bloque de saturación para limitar las señales de control dentro del rango de valores que permiten posicionar a cada motor entre 0 y 180°. Este bloque de saturación contiene los datos seriales que enviados al driver de servos, posiciona a los motores en los extremos mencionados. Existe un bloque de saturación diferente para cada motor.

Finalmente, una vez que se cumple con el posicionamiento del extremo del robot en el espacio, resta la parte de la actualización de las coordenadas x, y, z que posteriormente serán convertidas a coordenadas angulares que sirvan como referencia. Es necesaria una actualización debido a que cuando el extremo del robot se posicione sobre una de las piezas a trasladar, luego debe dirigirse a un lugar de destino con coordenadas diferentes, y luego debe dirigirse hacia una

nueva pieza. Un SubVI que fue explicado anteriormente realiza la reorganización de todas las coordenadas con las que ha de cumplir el robot, y también la actualización de coordenadas, una vez que el extremo del robot se haya posicionado con un error angular máximo de 0,5 grados en cada articulación.

El funcionamiento del electroimán también depende de la señal que indica que el extremo del robot ya se ha posicionado, ya que cuando el robot se dirige a una coordenada que es de una pieza, el electroimán se enciende cuando el extremo ya ha llegado; y se mantiene encendido hasta que el extremo del robot haya cumplido con unas coordenadas que correspondan a una dirección de destino.

Dado que existen dos tipos de programas de visión artificial, también existen dos programas de control de posición, uno para el modo de reconocimiento y selección de qué objeto se desea transportar, y otro para clasificar todos y cada uno de los objetos presentes en el área de trabajo en función de su forma.

Adicionalmente, cabe anotarse que ambos software de control de posición tienen dos modos de trabajo: MANUAL y AUTOMÁTICO. El modo AUTOMÁTICO funciona de la forma anteriormente descrita, pero el modo MANUAL omite la lectura de coordenadas de los objetos, reorganización de coordenadas, actualización de coordenada actual, debido a que el setpoint (x, y, z) es dado por un grupo de controles ubicados en el HMI. El análisis cinemático inverso se realiza en ambos modos de trabajo, dado que el robot trabajará únicamente con los objetos que estén dentro de los límites físicos impuestos por la cinemática. A continuación se muestra el HMI diseñado para el primer programa de control. El segundo programa es similar, y únicamente difiere en que contiene tres controles para indicar tres coordenadas de destino, debido a que se desea clasificar objetos de tres diferentes formas.

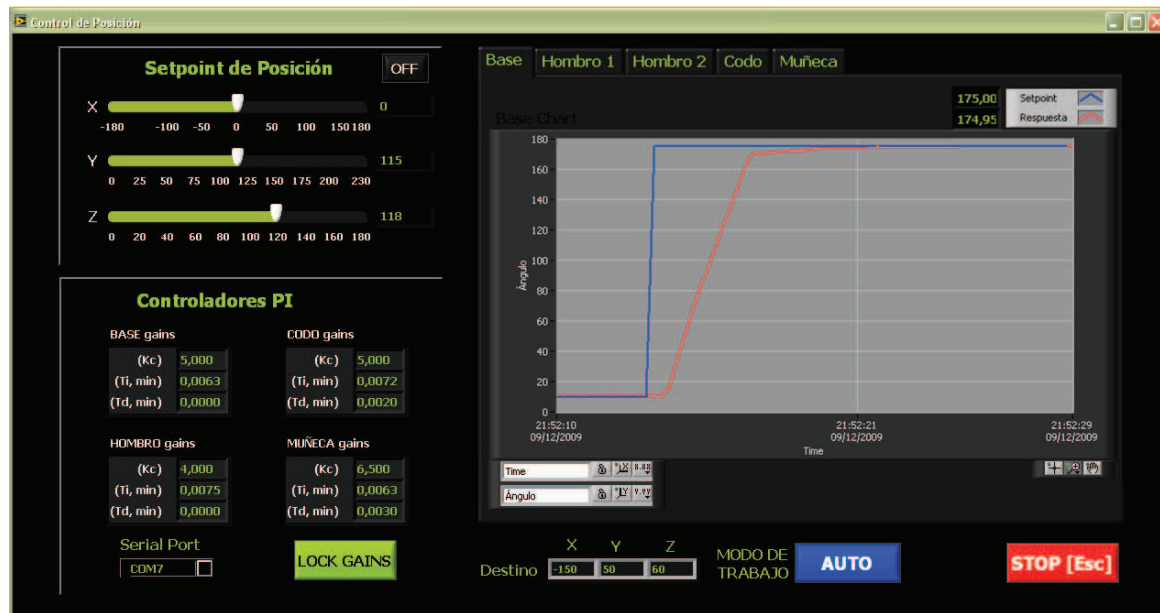


Figura 3.31 HMI del Software de Control de Posición

- En la parte superior izquierda se tienen controles que modifican las coordenadas x, y, z del setpoint para el caso de operación manual.
- En la parte inferior izquierda se tienen controles que modifican las constantes de cada uno de los controladores PI. Adicionalmente hay un control que permite seleccionar el puerto serial para la comunicación, y un botón que permite bloquear los valores actuales de las constantes de los controladores PI.
- En la parte derecha se muestran las gráficas del comportamiento setpoint-respuesta de cada uno de los motores. Para seleccionar que gráfica ver, existen pestañas con los nombres de los motores utilizados (Base, Hombro 1, Hombro 2, Codo, Muñeca).
- En la parte inferior central se tiene un control que permite ingresar las coordenadas x, y, z del destino. Junto a este control está un botón que permite seleccionar el modo de operación entre MANUAL, y AUTOMÁTICO, en el cual el sistema trabaja en función de las coordenadas recibidas desde el software de visión artificial.
- Finalmente en la parte inferior derecha está el botón de STOP que detiene el software.

CAPÍTULO 4

PRUEBAS Y RESULTADOS

Para el correcto funcionamiento del sistema de control, hay que realizar varias pruebas que permiten acondicionar, calibrar, etc. varios parámetros que son necesarios tanto para el software de visión artificial como para el de control de posición. Por facilidad de explicación se dividirá el presente capítulo en dos partes, la primera dedicada a las pruebas para visión artificial, y la segunda para el control de posición.

4.1 PRUEBAS DEL SISTEMA DE VISIÓN ARTIFICIAL

Para obtener un correcto funcionamiento del sistema de visión artificial, son necesarias algunas pruebas que permiten validar el software diseñado para la adquisición y procesamiento de imagen. A continuación se describen las pruebas más relevantes.

4.1.1 CORRECCIÓN DE DEFORMACIÓN ESFÉRICA Y CONVERSIÓN PIXELES - CENTÍMETROS

Cuando la imagen es adquirida, la óptica de la cámara la muestra como si fuera una parte de la superficie de una esfera. Esta deformación esférica en la adquisición genera posteriores inconvenientes al momento de detectar la posición exacta de los objetos presentes en la escena, debido a que una vez que la imagen es adquirida, el posterior procesamiento se basa en la distribución de píxeles, la cual no es un reflejo fiel de la escena real; por lo tanto los cálculos realizados tendrán errores significativos que obstaculizan el correcto funcionamiento del sistema en conjunto.

Para corregir esta deformación, se han utilizado funciones específicas de LabVIEW Vision and Motion, que permiten corregir deformación esférica en la adquisición. En primer lugar se debe dar información al conjunto de funciones de corrección, acerca del grado de deformación existente. Para ello se utiliza una plantilla (imagen previamente adquirida con la cámara usada) en forma de matriz de puntos, en la que el usuario indica las distancias vertical y horizontal entre los

centros de los puntos de la matriz, y esta información servirá de referencia para la corrección. Luego, en base a esta información las funciones 'aprenden' el nivel de deformación y generan un algoritmo de corrección, el cual posteriormente debe ser aplicado a la imagen adquirida para corregir su deformación.

El algoritmo de corrección será único para cada cámara que se utilice, debido a las inherentes diferencias físicas y de construcción existentes. También se deberá desarrollar un nuevo algoritmo si se cambia la distancia entre el lente de la cámara y la escena que se quiere captar.

Adicionalmente al algoritmo de corrección se ha implementado una rutina de conversión píxel-cm la cual permite traducir la información en píxeles de la imagen a centímetros del área de trabajo. En caso de que la cámara se mueva un poco, las coordenadas de los objetos serán desplazadas ocasionando errores en la detección; para atenuar este efecto, el software de visión pide al usuario colocar dos objetos en coordenadas conocidas (-100;200 y 100;200) al principio de la ejecución, con lo cual realiza la conversión píxel-cm en función de la posición actual del objeto y no de la posición que debería tener dentro de la imagen captada; es decir, se compensa el movimiento de la cámara.

Para comprobar la validez de los algoritmos implementados, se han colocado en el área de trabajo cinco objetos con posiciones conocidas, luego se realizó la adquisición y procesamiento de imagen con el algoritmo de corrección y también sin él, para evaluar la diferencia. En la prueba realizada la cámara fue movida un poco para apreciar las diferencias que existen. Luego de realizar la prueba se tienen los siguientes resultados.

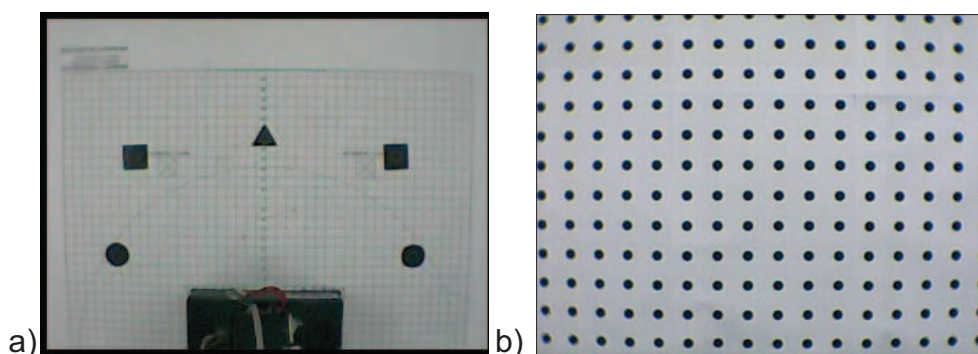


Figura 4.1 Corrección de deformación esférica, a) Original, b) Plantilla para corrección

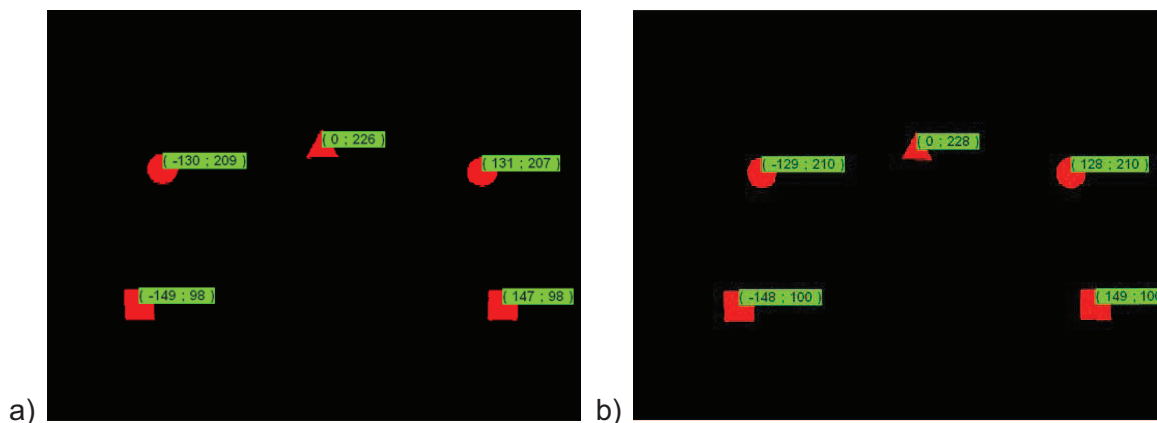


Figura 4.2 Imágenes procesadas, a) Sin corrección, b) Con corrección

La Tabla 4.1 muestra las diferencias entre coordenadas reales, coordenadas obtenidas con algoritmo de corrección y sin él.

Tabla 4.1 Resultados de prueba de corrección de deformación esférica

Coordenadas Reales en mm (x ; y)	Coordenadas en mm sin Algoritmo de Corrección	Error relativo	Coordenadas en mm con Algoritmo de Corrección	Error relativo
(-150 ; 100)	(-149 ; 98)	1,33%	(-148 ; 100)	0,67%
(-130 ; 210)	(-130 ; 209)	0,24%	(-129 ; 210)	0,38%
(0 ; 230)	(0 ; 226)	0,87%	(0 ; 228)	0,43%
(130 ; 210)	(131 ; 207)	1,10%	(128 ; 210)	0,77%
(150 ; 100)	(147 ; 98)	2%	(149 ; 100)	0,33%

* Nota: Para el análisis de los errores, se sacó un promedio entre el error relativo de la coordenada 'x' y el error relativo de la coordenada 'y'.

Como indica la Tabla 4.1 los errores en la identificación de coordenadas de los objetos en el plano de trabajo, son menores al aplicar la corrección a la deformación esférica. Esto se debe a que al no aplicar la corrección de deformación, las posiciones captadas de los objetos corresponden a la proyección de su posición real sobre la superficie de la esfera que se forma en la óptica de la cámara.

En la aplicación en la que está trabajando las diferencias existentes entre la detección con el algoritmo de corrección y sin él, son pequeñas, pero para aplicaciones en las que la cámara esté más alejada del plano o la precisión requerida sea mayor, se verá con mayor claridad la necesidad de corregir la deformación que existe en la imagen adquirida.

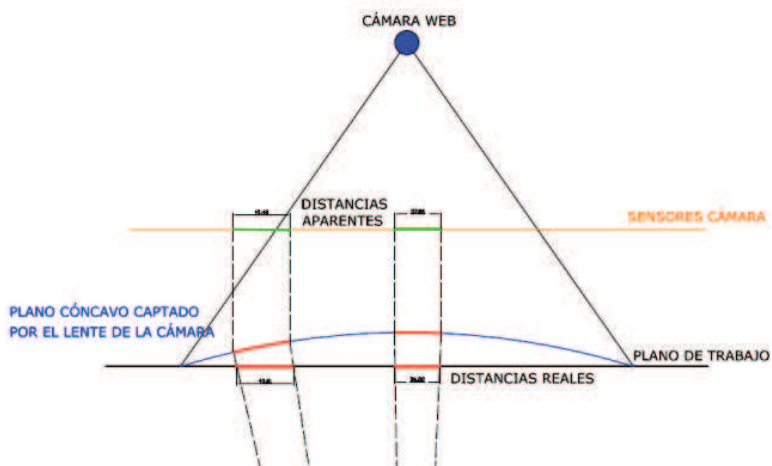


Figura 4.3 Deformación esférica en la óptica de la cámara

Al aplicar la subrutina de conversión píxel-cm a una imagen sin corrección de deformación esférica, se aumenta el error existente, debido a que la imagen capturada mostrará los objetos presentes en zonas en las que no se encuentran en realidad, brindando información errónea, en píxeles, a la rutina de conversión, y este error se amplifica al momento de expresar la posición en cm. En la aplicación probada, los errores mostrados luego de la conversión son bastante pequeños.

La Tabla 4.1 sirve de soporte para poder afirmar que mediante el procesamiento de imagen implementado considerando la corrección a la deformación esférica de la imagen y la conversión automática de unidades píxel-cm, los errores obtenidos son pequeños (máximo de 0,77%), lo que indica que el procesamiento de imagen es eficiente. Los errores se deben en parte al tipo de cámara utilizado, una Webcam, la cual tiene una resolución baja y una velocidad de captura máxima de 30 cuadros por segundo; estas características no son aptas para aplicaciones industriales, en donde las cámaras más comúnmente usadas por su resolución y velocidad son del tipo Firewire o SMART Camera.

4.1.2 DETECCIÓN DE OBJETOS EXTRAÑOS EN LA ESCENA

Cuando el robot está operando, es posible que las personas que están cerca puedan introducir sus manos dentro del espacio de trabajo. En este caso el robot podría colisionar con ellas y ocasionar un eventual daño a la persona. Para el

caso del prototipo construido, esto no se aplica dado el tamaño y potencia del mismo, pero para robots de mayor tamaño, este aspecto es muy importante ya que hay que garantizar la seguridad del operador y demás personas que puedan estar en contacto o cerca al área de trabajo del robot.

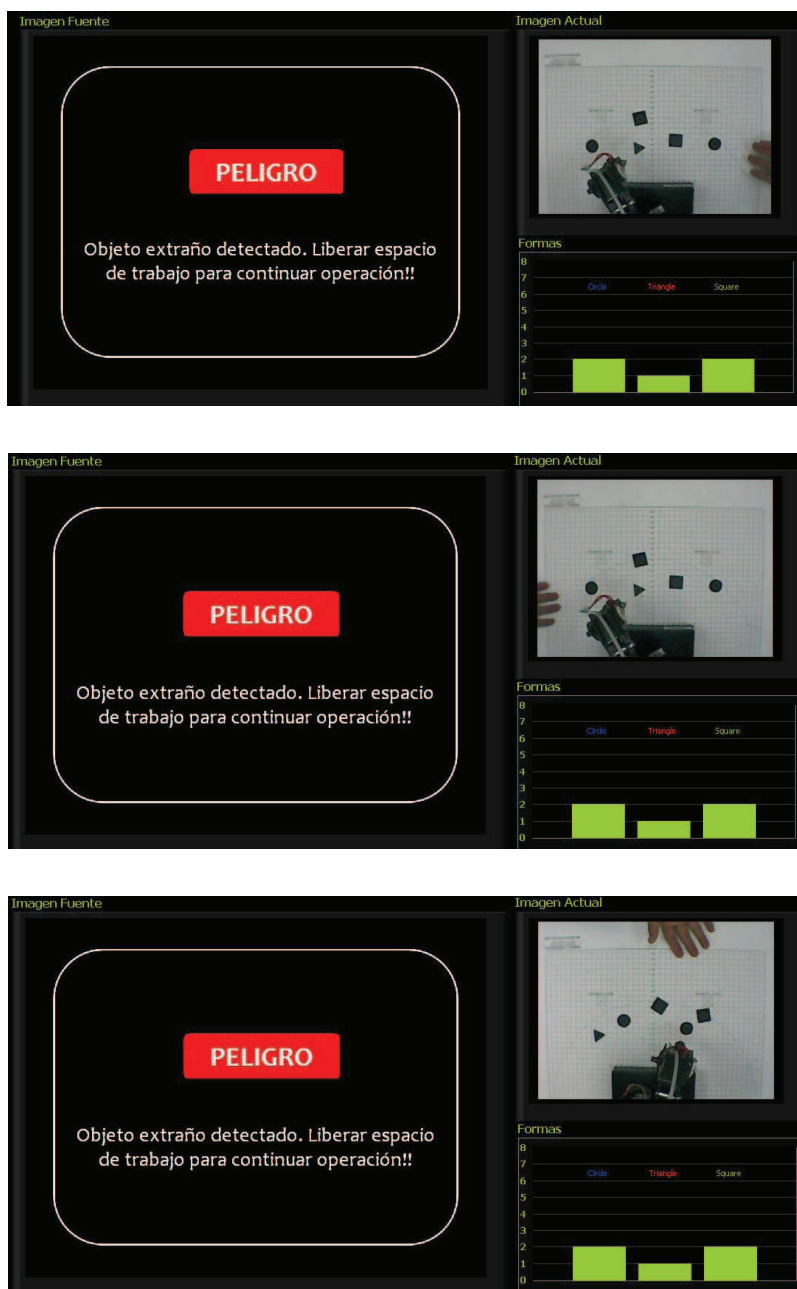


Figura 4.4 Detección de obstáculos

Para detectar objetos extraños se ha utilizado la técnica de análisis de perfil, mediante la cual se dibujan líneas en los bordes de la imagen, excepto en el borde sobre el que está colocado el robot para no interrumpir su movimiento, y

luego se analiza constantemente la distribución luminosa a lo largo de las líneas. En caso de que algún objeto ingrese al área de trabajo, se alterará la distribución luminosa a lo largo de una o más líneas de perfil, lo que activa un mensaje de alarma en el software de visión, el cual envía la alarma al software de control de posición para detener el movimiento del robot.

Luego de implementar este algoritmo dentro del software de visión artificial, se comprobó que efectivamente cuando se introduce la mano dentro de la escena desde cualquiera de los tres bordes que contienen línea de perfil, se activa la alarma. El algoritmo dio los mismos resultados para diferentes posiciones del robot dentro de la escena.

4.1.3 EXTRACCIÓN DE COORDENADAS Y FILTRADO DE FORMA

Finalmente, es necesario constatar el funcionamiento del software completo del sistema de visión artificial. Dado que se desarrollaron dos programas distintos se procederá a probar cada uno de ellos. La diferencia entre estos dos programas, es que uno permite seleccionar el tipo de objetos a trasladar en función de su forma; mientras que el otro permite trasladar y clasificar a todos los objetos presentes.

Ambos programas también deben evaluar las restricciones impuestas por la cinemática del robot. Si alguno de los objetos detectados está fuera del espacio definido como 'área de trabajo', las coordenadas de este objeto no se van a considerar como un punto con el que el robot deba cumplir.

4.1.3.1 Programa de Selección de Forma

Este programa tiene dos modos de funcionamiento. En el primer modo, el software debe obtener las coordenadas de todos los objetos, y mostrar una distribución estadística en función de la forma de los objetos presentes. En el segundo modo, el usuario selecciona una forma de objeto, y el software debe obtener las coordenadas únicamente de los objetos que cumplen con la forma seleccionada. En este modo también se muestra una distribución estadística.

Para probar la validez de ambos modos se realizaron las siguientes acciones:

- Colocar en la escena 3 objetos con forma de 'círculo', tres con forma de 'cuadrado' y dos con forma de 'triángulo'.
- Asegurarse de que un 'cuadrado', un 'círculo' y un 'triángulo' estén fuera del área de trabajo, para evaluar las restricciones cinemáticas.
- Asegurarse de que los objetos estén distribuidos a lo largo y ancho de toda la escena, para evaluar validez de la detección en todas las áreas.
- Ejecutar programa en el modo de detección total (NORMAL)
- Ejecutar programa en modo FORMAS, y seleccionar la forma 'Círculo'.
- Luego ubicar un objeto con forma cuadrada, sobre un objeto con forma de círculo, y comprobar que el software identifique al nuevo objeto como 'OTROS', en ambos modos de trabajo.

La distribución de objetos con la que se trabajó se muestra en la Figura 4.5, que es la imagen capturada por la cámara.

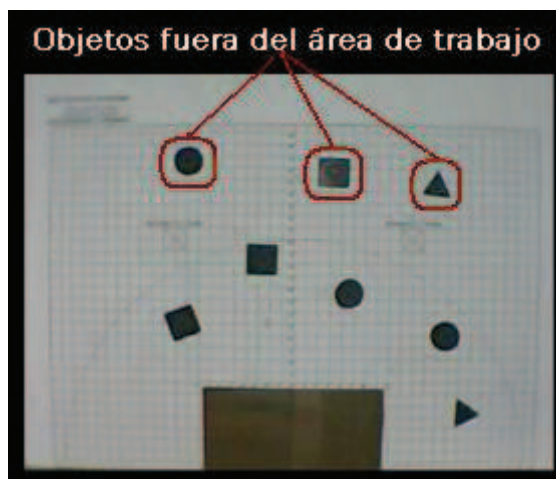


Figura 4.5 Imagen de prueba

En la imagen, los objetos que han sido encerrados con un recuadro son los que están fuera del área de trabajo del robot. Luego de realizar el procedimiento citado, se tuvieron los siguientes resultados.

Al ejecutar el programa en el modo NORMAL, se puede apreciar que se obtienen las coordenadas de todos y cada uno de los objetos, y se las muestra sobre la imagen. Adicionalmente, se puede notar que en el arreglo denominado 'Coordenadas' se han colocado los valores (x, y) correspondientes únicamente a los objetos que están dentro del área de trabajo.

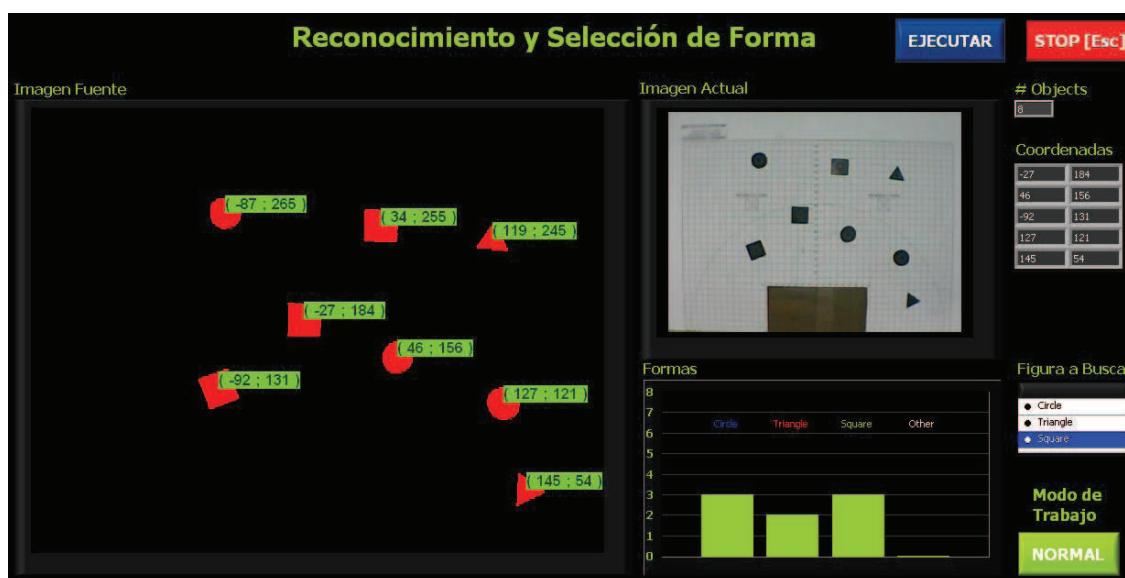


Figura 4.6 Trabajo en modo NORMAL

Luego, al ejecutar el programa en modo FORMAS, y seleccionando la opción 'Círculo', puede notarse que en la imagen se encierran con un recuadro a todos los objetos que corresponden con la forma seleccionada y se muestra sus coordenadas sobre la imagen. Adicionalmente, puede notarse que en el arreglo 'Coordenadas' se han colocado los valores (x, y) correspondientes únicamente a los dos 'Círculos' que están dentro del área de trabajo del robot.

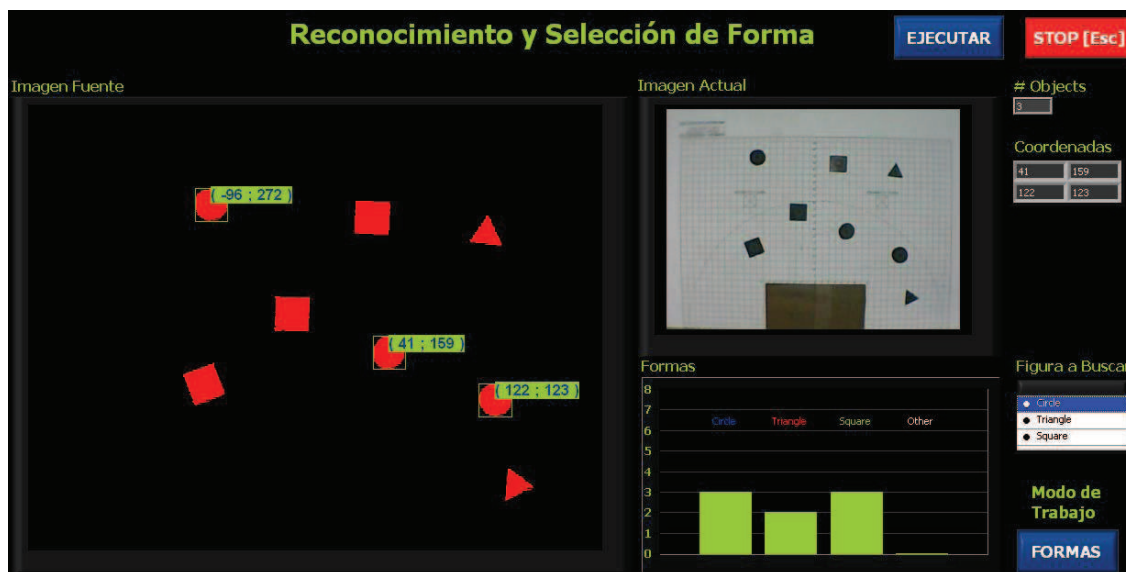


Figura 4.7 Trabajo en modo FORMAS

Luego, se colocó un objeto 'Cuadrado' sobre un 'Círculo', para evaluar la detección de objetos que no corresponden con las formas predefinidas. Se probó el software en ambos modos de trabajo, y se puede notar que en la distribución estadística de objetos se detecta un objeto como 'OTROS'.

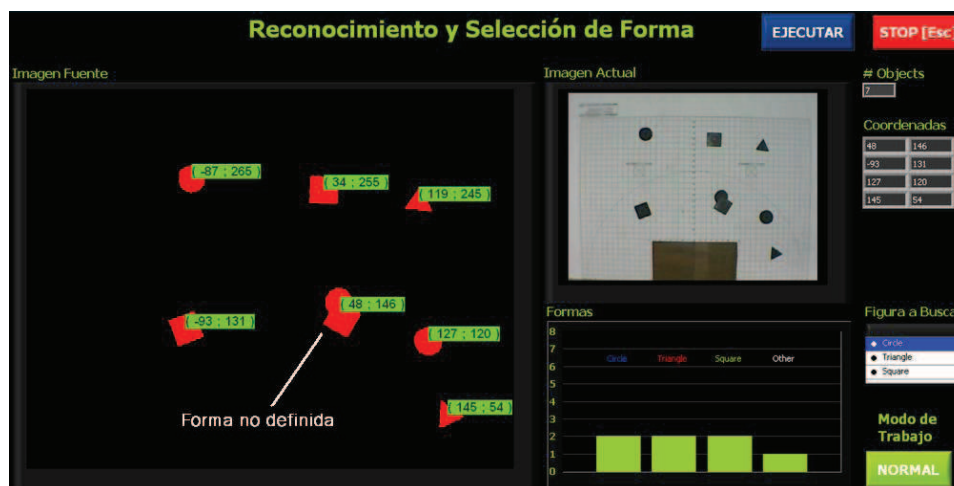


Figura 4.8 Detección de otros objetos en modo NORMAL.

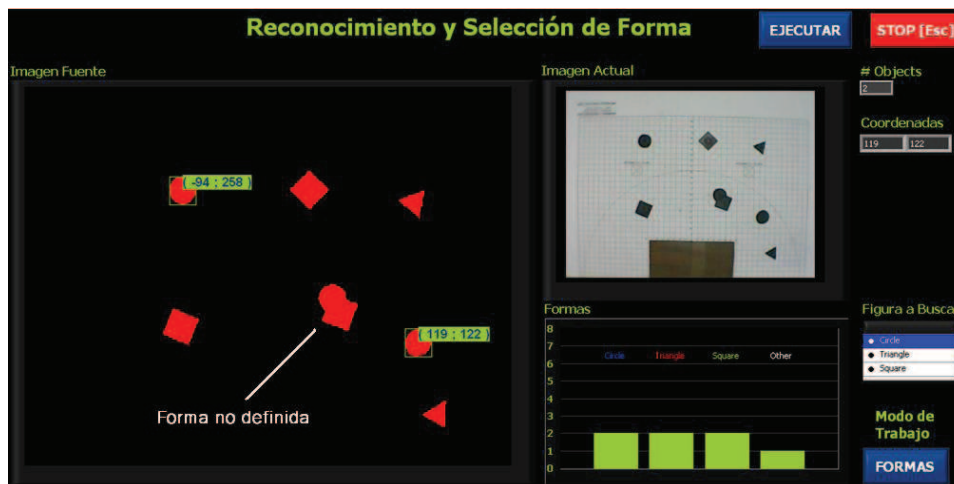


Figura 4.9 Detección de otros objetos en modo FORMAS.

Luego de las pruebas realizadas, se puede concluir que el programa diseñado cumple con los requerimientos planteados, pues ofrece excelentes resultados y desempeño.

4.1.3.2 Programa de clasificación

Este programa obtiene las coordenadas de todos los objetos presentes en la escena, y las agrupa en función de la forma del objeto al que pertenecen. De esta manera, si por ejemplo se consideran tres formas o geometrías, entonces el programa da como salida tres arreglos de coordenadas, uno para cada forma de los objetos. El programa a probar fue diseñado para reconocimiento de tres diferentes formas, en caso de que se requiera el reconocimiento de más formas, hay que guardar una imagen de referencia de las nuevas formas y modificar una subrutina del programa.

Para diferenciar a cada objeto, el HMI del programa encierra con recuadros de diferentes colores a los objetos en función de su forma. Este programa también muestra una distribución estadística de los objetos.

Para probar la validez de este software se colocaron en la escena los mismos objetos que se usaron para probar el primer programa, teniendo en cuenta lo siguiente:

- Dejar fuera del área de trabajo por lo menos a uno de los objetos.

- Verificar que el software identifique y clasifique los objetos encerrándolos en recuadros con diferentes colores, y colocando sus coordenadas en el arreglo adecuado en función de la forma del objeto.

Luego de realizar esta prueba se tuvieron los siguientes resultados.



Figura 4.10 Pantalla de software de clasificación

Puede notarse que los objetos con forma de 'Círculo' han sido encerrados en recuadros color verde, los objetos con forma de 'Cuadrado' en recuadros color amarillo, y los objetos con forma de 'Triángulo' en recuadros color blanco.

Adicionalmente se puede notar que en los arreglos de coordenadas correspondientes a cada forma, se han colocado únicamente los valores (x, y) de los objetos que están dentro del área de trabajo del robot.

Luego de las pruebas realizadas, se puede concluir que el software diseñado cumple con las funciones y requerimientos planteados, y ofrece excelentes resultados.

4.2 PRUEBAS DEL SISTEMA DE CONTROL DE POSICIÓN

El software de control de posición se encarga de comandar los motores que forman las articulaciones del robot para posicionar su extremo en el espacio. Para realizar este control, primero es necesario realizar algunas pruebas previas que

permitan configurar el subsistema de adquisición y acondicionamiento de señales, verificar funcionamiento del sistema de control de posición angular y espacial, etc.

4.2.1 ACONDICIONAMIENTO DE SEÑAL DE POSICIÓN

Para la realimentación de la señal de posición se adquiere la señal dada por el potenciómetro interno de cada servomotor; el inconveniente es que esta señal tiene diferentes escalas y valores límite dependiendo del servomotor usado. Aún si se usa el mismo tipo de servomotor, su señal de realimentación puede variar.

Para solventar este problema, se siguieron estos pasos:

1. Definir valores seriales, que enviados al driver de servos POLOLU, ubican el eje de cada uno de los motores en 0 y 180 grados; es decir, definir límites.
2. Una vez definidos los límites de la señal de control a enviar, se procedió a variar la misma dentro del rango establecido de tal forma que el eje del motor empiece en 0 grados, y llegue hasta 180 grados, en pasos de 10 grados. Cada 10 grados de movimiento, se procedió a leer la señal de realimentación de posición mediante la DAQ USB-6009, de tal forma que se pueda obtener una tabla de valores Voltaje vs Ángulo, para construir una ecuación de acondicionamiento de la señal de posición.

Para llevar a cabo los pasos descritos, se utilizó un archivo de calibración desarrollado en LabVIEW, el cual permite enviar señales seriales de control que posicionan el eje de cada motor, y adicionalmente permite guardar el valor leído por la tarjeta de adquisición datos, en el momento en que el usuario lo desee. El panel frontal de este VI se muestra en la Figura 4.11.



Figura 4.11 VI de calibración de motores

En la parte izquierda están los controles que permiten variar la posición del eje de cada motor; en la parte derecha están las instrucciones de uso del VI, y los controles que permiten guardar los datos *Ángulo* y *Voltaje Adquirido*, del motor con el que se esté trabajando. El usuario debe ingresar el ángulo real del eje del motor en la entrada 'Ángulo' del VI, y luego presionar el botón correspondiente a la articulación que representa el motor que se está probando (por ejemplo base), para guardar los datos.

Para la medición del ángulo real del eje del motor, se utilizó un círculo graduado impreso en acetato, que se coloca bajo el eje del motor, de tal forma que se pueda observar el ángulo exacto en el que se encuentra el eje.

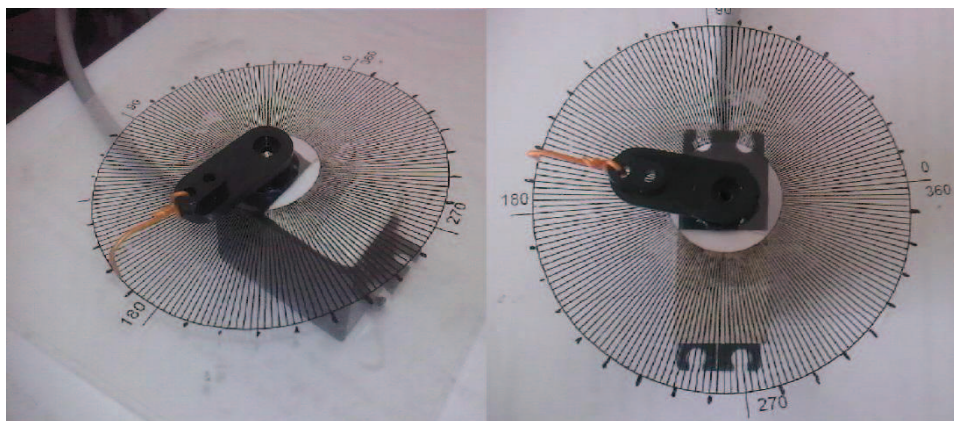


Figura 4.12 Medición del Ángulo Real del eje de cada motor

Luego de realizar este proceso cada 10 grados, se tienen suficientes datos para elaborar una ecuación de acondicionamiento, mediante EXCEL. En la Figura 4.12 se presentan los datos tomados para el motor de la Muñeca, y su respectiva ecuación generada.

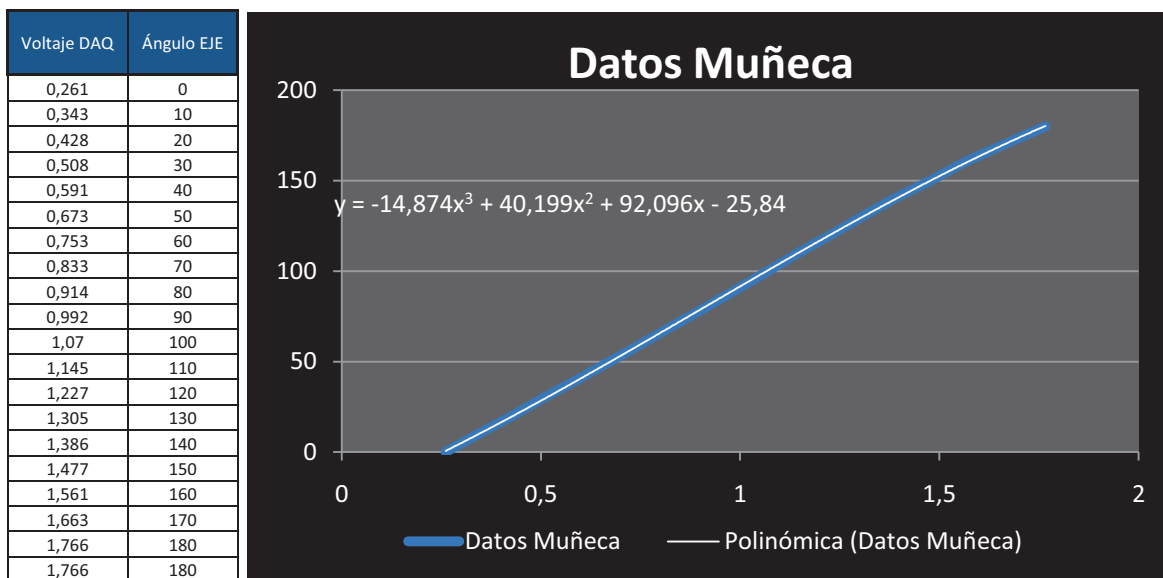


Figura 4.13 Generación de Ecuación de Acondicionamiento

Se realizó este proceso para cada uno de los motores, y luego se implementaron las ecuaciones obtenidas dentro de un SubVI.

4.2.2 SISTEMA DE CONTROL DE POSICIÓN ANGULAR

Antes de trabajar con las coordenadas (x,y) de los objetos detectados, primero hay que garantizar el funcionamiento del sistema de control de posición angular de cada motor. Para ello se utiliza una plataforma de pruebas construida en LabVIEW, que permite variar el setpoint de ángulo para cada articulación, obtener la curva de respuesta, y finalmente graficar resultados.

Este VI construido, permite calibrar constantes de los controles PI para fijarlas como valores por defecto para el programa final, evaluar curvas de respuesta, determinar error persistente, etc.

Para la realización de la prueba se han dado diferentes ángulos como referencia a cada articulación, y se evalúa la respuesta del robot. A continuación se presentan los resultados obtenidos.

Tabla 4.2 Resultados primera prueba

Articulación	Posición Inicial	Setpoint	Respuesta	Error relativo	ts (s)
Base	10°	150°	150,11°	0,07%	3,75
Hombro	20°	150°	150°	0%	3,97
Codo	150°	35°	35,90°	2,57%	6,63
Muñeca	10°	150°	150,16°	0,11%	4,20

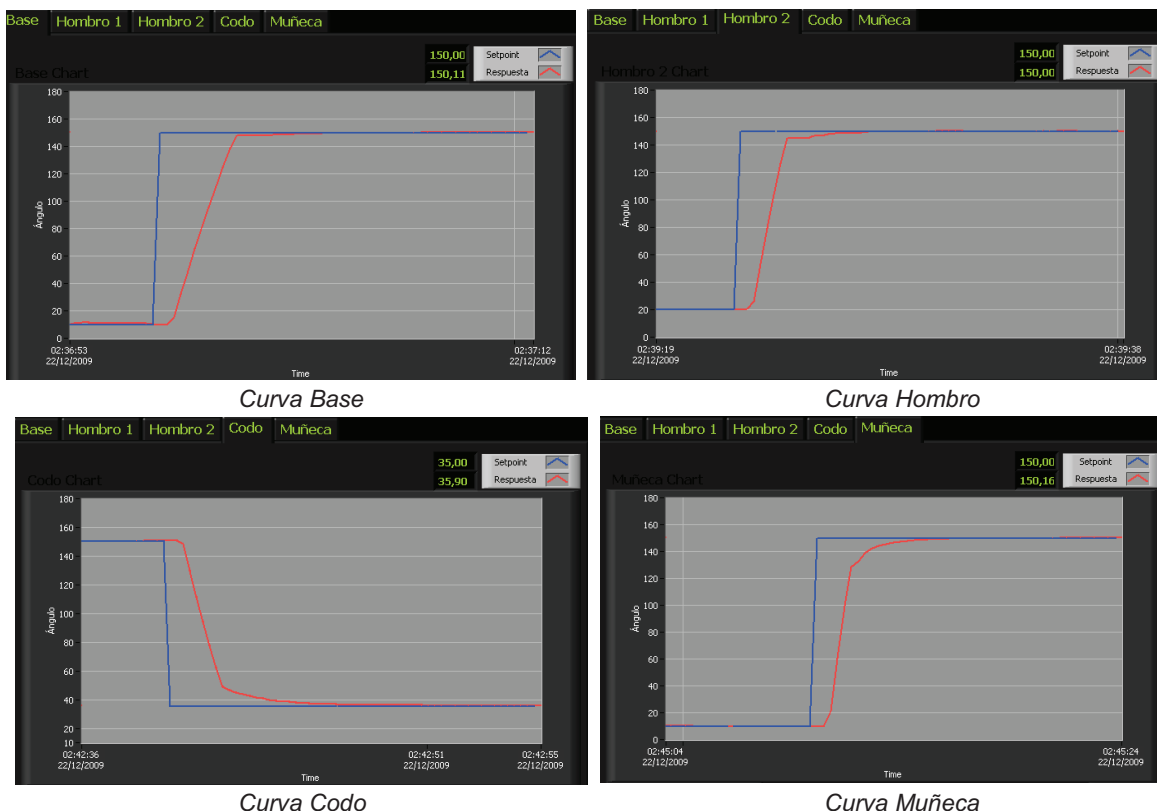


Figura 4.14 Prueba 1 de Control Angular

Se puede notar que cada una de las articulaciones del robot cumple con el valor de referencia dado con un error máximo de 2,57%, por lo que se puede validar el sistema de control implementado. Como se puede observar, el sistema es sobre amortiguado, puesto en el posicionamiento del extremo del robot, no deben exhibirse oscilaciones.

Por otro lado se observa que el tiempo que tarda el sistema de control en posicionar cada articulación es de un máximo de 6,63 segundos; esto se debe a la velocidad reducida a la que trabajan los servomotores para evitar daños en la caja de engranes; y también al sistema de control que es sobre amortiguado.

En la siguiente prueba se cubre un desplazamiento angular menor.

Tabla 4.3 Resultados segunda prueba

Articulación	Posición inicial	Setpoint	Respuesta	Error relativo	ts (s)
Base	120	60	60,56	0,93%	6,63
Hombro	90	30	30,31	1,03%	4,7
Codo	120	60	60,71	1,18%	6,97
Muñeca	100	50	49,86	0,28%	3,75

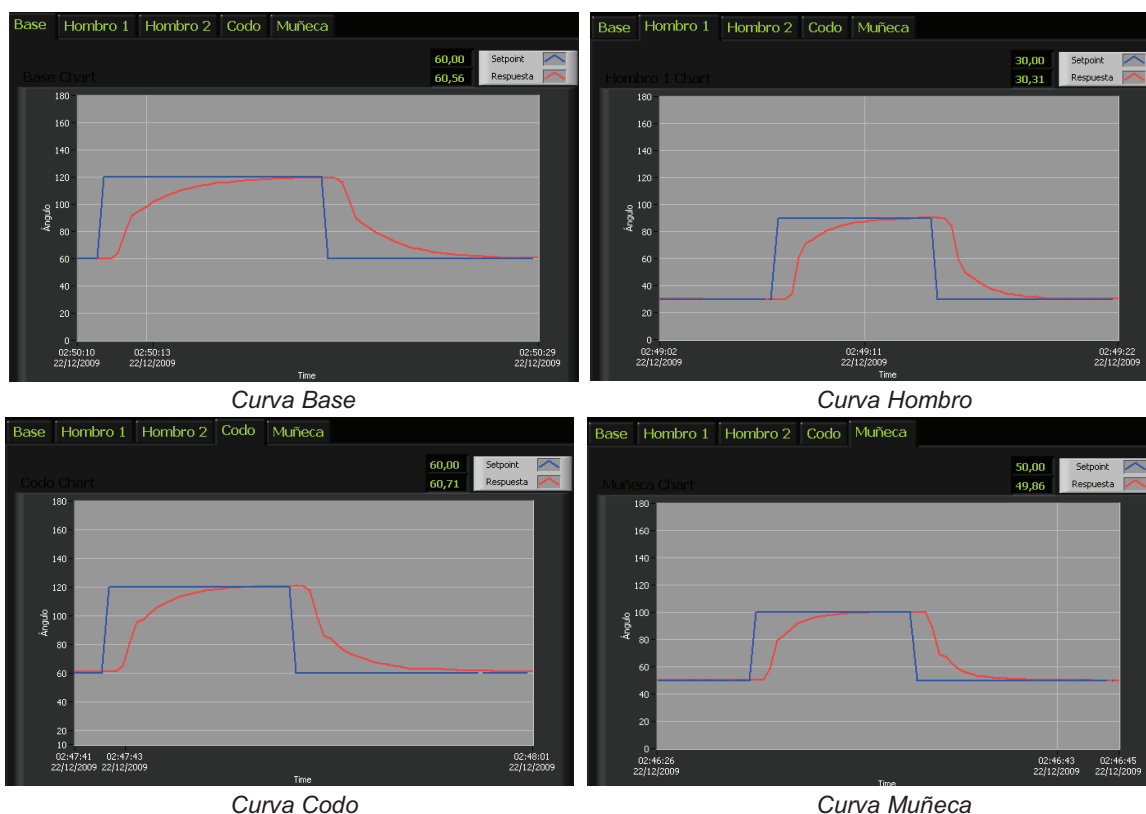


Figura 4.15 Prueba 2 de Control Angular

En la segunda prueba se puede notar que cada articulación cumple con las referencias angulares dadas con un error máximo de 1,18%, lo cual permite validar el sistema de control angular sobre amortiguado implementado. Por otro lado el tiempo de establecimiento es un tanto alto (máximo de 6,97 seg.); esto se explica en el hecho de que el valor de la variable de control tienen incrementos de menor tamaño que pueden caer dentro del rango de valores que no ocasionan movimiento en el servomotor debido a la baja sensibilidad de la tarjeta interna de control de cada motor. Esta prueba muestra que para diferencias entre setpoint y

posición inicial grandes, el extremo del robot se posiciona con mayor rapidez debido a que la variable de control tiene incrementos mayores que permiten contrarrestar la baja sensibilidad de la tarjeta interna de control de los servomotores.

En conclusión el sistema de control en lazo cerrado sobre amortiguado implementado, permite al robot cumplir con las posiciones de referencia dadas.

4.2.3 SISTEMA DE CONTROL DE POSICIÓN ESPACIAL

Una vez que se ha validado el sistema de control angular para el robot, es necesario comprobar que el SubVI que resuelve el problema cinemático inverso funciona adecuadamente, para ello se usa el programa final para el sistema de control de posición, en los modos MANUAL y AUTOMÁTICO.

4.2.3.1 Modo Manual

En este modo, el usuario da los valores de setpoint x,y,z hacia los cuales debe dirigirse el extremo del robot. Para evaluar el trabajo del robot, se han dado cinco valores en el espacio (x,y,z) , y se verifica que el extremo del robot se dirija hacia dichas coordenadas.

En la Tabla 4.4 se presentan las coordenadas de referencia dadas, y las coordenadas en las que se posiciona el extremo del robot.

Tabla 4.4 Resultados prueba cinemática

Coordenadas Reales en mm (x ,y, z)	Posición Extremo Robot	Error relativo (%)
(-150 ,100, 60)	(-148, 101, 58)	1,88%
(-130, 180, 15)	(-125, 178, 15)	1,65%
(0 ,220 , 10)	(0, 215,10)	0,76%
(130, 180, 15)	(125, 180, 15)	1,28%
(150, 100, 60)	(147, 99, 60)	1,00%

* Nota: Para el análisis de los errores, se sacó un promedio entre el error relativo de la coordenada 'x', error relativo de la coordenada 'y' e de la coordenada 'z'.

En la Figura 4.16 se muestran las mediciones tomadas para el primer punto de prueba.

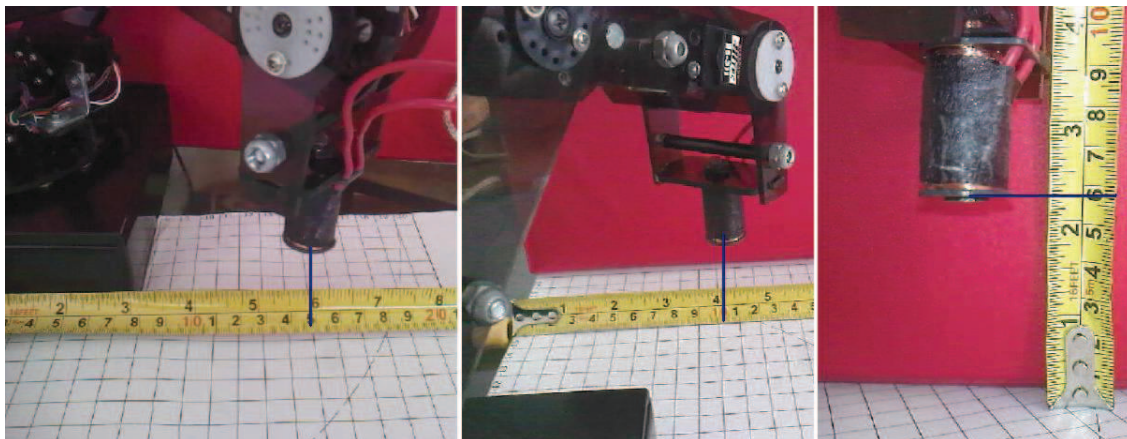


Figura 4.16 Prueba cinemática en Modo MANUAL, punto (-150, 100, 60)

Como puede apreciarse en los resultados, los errores de posicionamiento son menores al 1,88% lo que permite validar el algoritmo de cinemática inversa implementado puesto que el extremo del robot se posiciona bastante bien en la referencia espacial dada.

4.2.3.2 Modo Automático

En este modo, el sistema responde a los valores de setpoint que da el software de visión artificial, y adicionalmente a una posición denominada como destino, en donde se deben colocar todos los objetos detectados. Para evaluar el trabajo del robot, se han colocado tres objetos en el espacio de trabajo, de los cuales el software de visión artificial proporcionará sus coordenadas, y se ha colocado una posición de destino $(x,y,z) = (-150,50,40)$. El robot debe dirigirse a cada objeto, sujetarlo mediante el electroimán y luego trasladarlo a la posición de destino. En la Figura 4.17 se muestra la distribución inicial de los objetos en el plano, y la forma como el extremo del robot se dirige y posiciona sobre el objeto a trasladar.

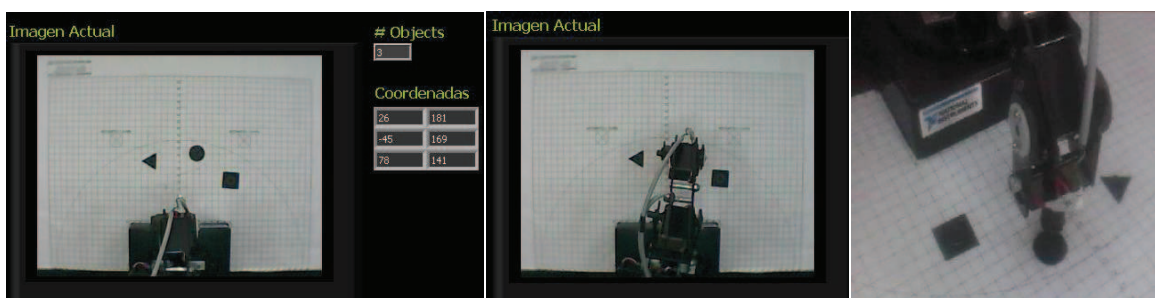


Figura 4.17 Prueba cinemática en Modo AUTOMÁTICO

Luego de realizar esta prueba se pudo comprobar que el robot se dirige a cada uno de los objetos presentes en el plano de trabajo y los traslada hacia la posición de destino especificada. El robot traslada a los objetos en función del valor de su coordenada 'y' empezando por el que tiene la coordenada de mayor valor.

Al finalizar esta prueba se pudo comprobar que el sistema trabaja del modo esperado, puesto que detecta a los objetos dentro del área de trabajo y los traslada a la posición de destino.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

Luego del proceso de diseño, construcción y control de un robot antropomórfico de 4 grados de libertad dotado con un sistema de visión artificial, se pueden citar las siguientes conclusiones y recomendaciones.

5.1 CONCLUSIONES

Tras haber desarrollado el prototipo en base a servomotores de modelismo, un driver de servomotores POLOLU, una tarjeta de adquisición de datos NI USB-6009, una cámara Web y software completamente implementado en base a LabVIEW; se puede anotar las siguientes conclusiones.

- Para la construcción de un sistema mecánico es conveniente realizar primero un diseño completo del sistema mediante software de diseño CAD, como por ejemplo SOLIDWORKS, de tal forma que se obtenga una “Máquina Virtual”. Una vez obtenido el modelo virtual, se pueden realizar las diferentes pruebas de esfuerzo, movimiento, restricciones, etc. que permitan validar el modelo, dimensionar actuadores y corregir errores mientras todavía se está en la etapa de diseño, en lugar de tener que corregir posibles equivocaciones cuando se tiene la máquina ya construida.
- El uso conjunto de LabVIEW y SOLIDWORKS permite ahorrar tiempo en el dimensionamiento de los actuadores necesarios para el robot. Además los resultados obtenidos son muy confiables puesto que en el proceso se consideran todas las características, relaciones y restricciones físicas propias del movimiento del robot.
- El uso de LabVIEW para el desarrollo completo del software de control permitió implementar un programa de altas prestaciones, con una interfaz de usuario amigable y funcional. Al usar LabVIEW se pudo desarrollar programas independientes para procesamiento de imagen y para control de posición, pero a su vez juntar ambos programas con todas sus

características individuales y compartidas, dentro de una aplicación que funciona de forma sincronizada y eficiente en tiempo real.

- Para posicionar el eje de un servomotor de modelismo se debe aplicar un tren de pulsos con una determinada relación de trabajo. Para mantener la posición del eje, el tren de pulsos debe estar siempre presente y con la misma relación de trabajo, ya que si se suprime la señal de control se tiene el equivalente a apagar el motor. Para variar la posición del eje se debe aplicar un tren de pulsos con una nueva relación de trabajo.
- El uso de un driver de servomotores POLOLU, facilitó el control de los servomotores de modelismo puesto que recibe una señal serial y en base a ésta genera un tren de pulsos con determinada relación de trabajo que permite posicionar el eje del motor. Además permitió enviar las señales de control desde LabVIEW vía comunicación serial con velocidades de 28800bps, lo que permite aplicar con mayor rapidez la señal del sistema de control.
- El uso de una tarjeta de adquisición de datos NI USB-6009 para la lectura de las señales de posición de cada servomotor, facilita en gran medida la implementación de circuitos de acondicionamiento, puesto que la tarjeta ofrece entradas análogas de 14 bits y frecuencias de muestreo de hasta 48k muestras en total, lo que permite en cierta medida garantizar la calidad de la señal adquirida. Adicionalmente, al usar una DAQ USB-6009, facilita el proceso de filtrado, ya que se puede implementar filtros digitales mediante la configuración propia de la tarjeta de adquisición o mediante software (LabVIEW).
- Para el desarrollo de un sistema de control de posición en tiempo real para un manipulador industrial, se debe resolver el problema cinemático inverso mediante una solución cerrada en lugar de una iterativa, ya que es más rápida y proporciona valores exactos que facilitan la implementación del control y mejoran su desempeño.

- Los servomotores que cumplen la función de actuadores han sido acoplados directamente a las articulaciones del robot; esto permite un posicionamiento preciso, rápido y con una simplificación sobre todo del sistema mecánico; con la desventaja de que se necesita un mayor torque que se ve reflejado en los costos.
-
- Los servomotores de modelismo tienen una relación no tan lineal entre la señal de control aplicada y la respuesta o posición angular que adopta el eje del motor. Esto se debe a que hay rangos de la señal de control que no provocan movimiento en el motor, es decir, el sistema de control interno de los servomotores tienen poca sensibilidad y además la caja reductora tiene un 'juego' que no permite mover el eje en pequeños incrementos. Esto hace necesario el desarrollo de una ecuación de acondicionamiento cuadrática o cúbica, la cual fácilmente puede ser implementada mediante LabVIEW.
- Otro problema de los servomotores de modelismo es que aún si se está usando motores del mismo tipo, denominación, características, los rangos de la señal de control que posicionan al eje en un determinado ángulo, son diferentes. Este inconveniente obliga a usar lazos de control independientes para cada motor; es decir, se trata a un sistema multi-variable como múltiples lazos simples.
- Para la implementación de un sistema robótico en lazo cerrado en base a servomotores de modelismo es necesaria la implementación de un sistema de medición angular externo, o bien se debe recurrir a leer la señal del potenciómetro interno del servomotor, el cual es diferente para cada servomotor, lo que hace indispensable el uso de un sistema de adquisición y acondicionamiento de señal independiente para cada motor. En el presente proyecto se utilizó una DAQ USB-6009.
- Dado que los servomotores de modelismo tienen internamente un control proporcional, para lograr un control en lazo cerrado se ha implementado un

control en cascada en donde un controlador PI da el set-point para el sistema interno de control.

- Cuando un servomotor se aproxima al valor de referencia dado, el controlador PI envía señales de control que están dentro del rango de valores que no causan movimiento en el eje del motor, lo cual genera un error persistente, que provoca una variación mayor de la señal de control, hasta que se consiga mover el eje del motor. Este comportamiento ocasiona que el movimiento del motor hasta llegar al valor de referencia, sea un tanto discontinuo. Este efecto refleja la baja sensibilidad del sistema de control interno de los servomotores de modelismo.
- El uso de una cámara Web para un sistema de detección de objetos en un plano, no es adecuado cuando se desea obtener una posición exacta. Esto se debe a que los fotosensores y la óptica que componen a una Webcam, están orientados a una aplicación diferente, por lo que no brindan buena resolución, ni velocidad en la captura de imágenes para una aplicación industrial. Para este efecto, se recomienda el uso de cámaras Firewire, o SMARTCAMERA que dan resoluciones bastante altas y una velocidad de hasta 60 cuadros por segundo.
- En un sistema de visión artificial uno de los aspectos principales a cubrir es un buen sistema de iluminación, que facilite la adquisición y posterior procesamiento de la imagen. Si se implementa un sistema de iluminación adecuado, se facilita en gran medida el procesamiento e interpretación de la información de la imagen adquirida.
- El uso de los algoritmos de corrección de deformación esférica de la imagen adquirida y de conversión píxel-cm, permiten mejorar la calidad de la imagen, y por ende localizar con mayor exactitud a los objetos y atenuar los efectos del movimiento de la cámara.
- El uso de filtros Laplacianos, umbralización, segmentación basada en morfología y demás procesamiento de imágenes binarias, permitió cumplir la tarea fundamental del software de visión artificial, que es detectar la

ubicación de los objetos en el plano, y reconocer su forma, con excelentes resultados.

- El uso de la técnica de análisis de líneas de perfil para detectar la intrusión de objetos extraños dentro del área de trabajo del robot, da muy buenos resultados. Esto permite brindar mayor seguridad a los operadores del sistema y a las personas que estén cerca del mismo.
- El sistema de control implementado es eficiente y se puede adaptar a cualquier tipo de robot industrial construido en base al mismo tipo de motores utilizado.

5.2 RECOMENDACIONES

Luego del desarrollo del presente proyecto, se pueden anotar las siguientes recomendaciones.

- Cuando se utilizan cámaras comunes que no están orientadas a una aplicación específica, se recomienda realizar un subproceso de corrección o acondicionamiento de la imagen adquirida, ya que generalmente la imagen se muestra como la superficie de una esfera, debido a la óptica de la cámara; más específicamente al lente. La necesidad de corregir la deformación de la imagen es más evidente cuando la distancia entre la cámara y el plano de trabajo es mayor.
- Para el ensamble y control de robots de pequeñas dimensiones o prototipo, es factible el uso de servomotores de modelismo, siempre y cuando se tengan en cuenta las restricciones y características especiales que poseen. Para la implementación de sistemas de mayor tamaño se recomienda optar por motores de otro tipo, como por ejemplo DC con caja reductora y encoder, servomotores industriales, steppers, etc.
- Para adaptar el sistema de control diseñado en LabVIEW a aplicaciones de mayor potencia y tamaño, se puede trabajar con sistemas de control embebido como un COMPACT RIO y sus módulos de entradas análogas y salidas digitales. Esto se puede conseguir aún si se trabaja con otro tipo de

motores, ya que únicamente es necesario modificar la forma en la que se generan las señales de control, y se adquieren las señales de realimentación. Ambos cambios se realizan mediante dos subrutinas del programa diseñado.

- Para modificar la forma en que se trasladan las piezas, se puede optar por cambiar el efector final del robot con solo modificar una de las tantas piezas que conforman al manipulador industrial construido.
- El sistema de visión artificial puede ser orientado a otros tipos de detecciones como por ejemplo color, reconocimiento de detalles superficiales, etc. mediante el uso de las funciones adecuadas del Toolkit Vision and Motion de LabVIEW.
- Antes de iniciar con el desarrollo de un proyecto similar, se recomienda investigar y analizar las diferentes herramientas computacionales que existen, puesto que el buen uso de ellas facilita todo el proceso de diseño, validación, construcción y control.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Barrientos A.; Peñin L.; y otros, “Fundamentos de Robótica”. Segunda Edición. Editorial McGraw-Hill. España. 2007.
- [2] Comité Español de Automática, “Tipos de Robots”, España, Octubre 2009, www.cea-ifac.es.
- [3] Universidad del Valle de México, “Configuraciones más frecuentes de robots”, México, 2005, <http://www.uvmnet.edu/investigacion/episteme/numero6-06/reportes/imgs/controlF1.gif>
- [4] Wikipedia, “Servomotor de modelismo”, http://es.wikipedia.org/wiki/Servomotor_de_modelismo
- [5] Platero, Carlos, “Apuntes de Visión Artificial”. Dpto. Electrónica, Automática e Informática Industrial. Archivo PDF. GNU Free Documentation License. España.
- [6] Minchala, Ismael, Presentación curso “Visión Artificial”. National Instruments. Ecuador. 2009
- [7] National Instruments, LabVIEW Machine Vision And Image Processing Course Manual, May 1998 Edition.
- [8] Basantes, J.; TORRES, F., “Desarrollo de un Sistema de Control para un Brazo Robótico mediante Adquisición y Procesamiento de Imágenes”. Facultad de Ingeniería Eléctrica y Electrónica. Escuela Politécnica Nacional. Ecuador. Quito. Febrero 2009.
- [9] Ayuda SolidWorks 2008
- [10] Pololu 16-Servo Controller User’s Guide
- [11] National Instruments, “Manual NI USB-6009”, Austin-Texas-USA, 2009, <http://sine.ni.com/nips/cds/view/p/lang/es/nid/14605>

ANEXO A

MANUAL DE USUARIO

Los programas o VI's para el Control en lazo cerrado del Robot, así como para la detección de interferencias y ubicación de objetos mediante Visión Artificial; se encuentran conjuntamente con los SubVI's necesarios para su ejecución, dentro de un Proyecto de LabVIEW.

Dentro de la carpeta **CONTROL PID ROBOT-Pololu**, se encuentran todos los VI's necesarios para el funcionamiento del Robot.

A.1 REQUERIMIENTOS BÁSICOS

Antes de abrir los archivos es necesario contar con la versión LabVIEW 2009, además deberán estar instalados los toolkits de *Control Design & Simulation* y *Machine Vision & Motion*, para evitar errores durante la ejecución de las aplicaciones. Para poder ejecutar los VI's de diseño mediante SolidWorks, es necesario tener instalado SolidWorks 2007, con las herramientas de COSMOSWorks y COSMOSMotion. Además se recomienda tener los siguientes requerimientos mínimos del computador donde se ejecuten los programas:

Sistema Operativo: Windows 2000

Procesador: 1.6 GHz

Memoria RAM: 1GB

Para obtener un mejor rendimiento de las aplicaciones, se recomienda utilizar un computador con características superiores a los requerimientos mínimos indicados.

A.2 DESCRIPCIÓN DEL ARCHIVO DE PROYECTO

Para ingresar al proyecto y a todas las aplicaciones de control del robot, es necesario abrir el archivo: **Control RA4-V1.lvproj**.

NOTA: Todos los archivos de la carpeta **CONTROL PID ROBOT-Pololu**, deben mantenerse juntos sin importar la ubicación de la carpeta dentro de los directorios del computador.

Abrir el archivo: **CONTROL PID ROBOT-Pololu>Control RA4-V1.lvproj**

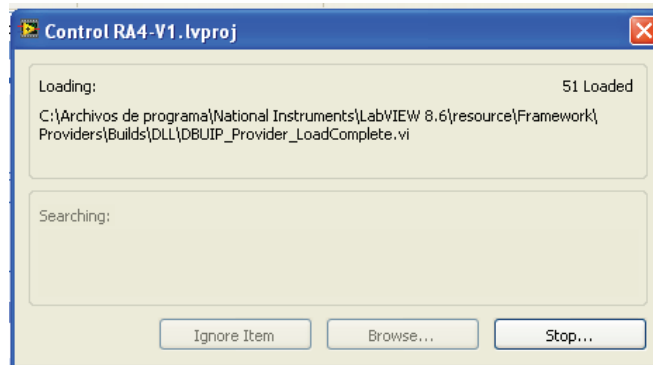


Figura A.1 Ventana que muestra la carga de los archivos del proyecto

En ese momento se cargan todos los archivos que forman parte del proyecto como se muestra en la Figura A.1. Una vez cargados todos los archivos se abre una ventana como la que se muestra a continuación:

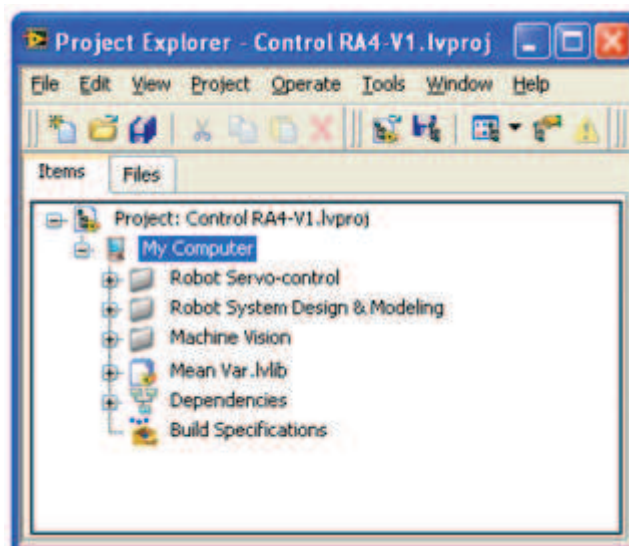


Figura A.2 Ventana de proyecto

Como se puede ver en la Figura A.2, dentro de **Project: Control RA4-V1.lvproj**, se tiene **My Computer**, que indica el equipo donde se van a ejecutar los programas.

Dentro de **My Computer** se tiene las siguientes subcarpetas:

- Robot Servo-control
- Robot System Design & Modeling
- Machine Vision

Dentro de **Robot Servo-control**, se presentan los archivos distribuidos como se muestra en la Figura A.3.

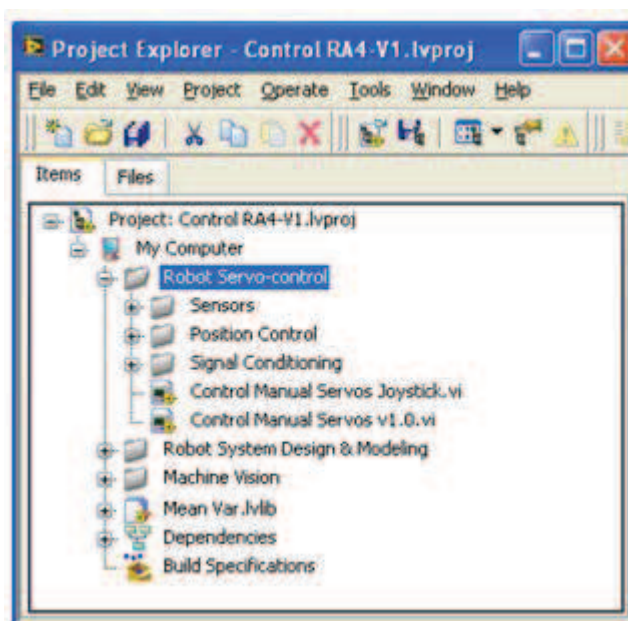


Figura A.3 Despliegue de la carpeta Robot Servo-control dentro del proyecto

En ésta carpeta se encuentra todo lo referente al Control del Robot en lazo abierto y en lazo cerrado. A continuación se da una breve explicación de cada una de las subcarpetas.

A.2.1 Sensors

En ésta carpeta se encuentran los archivos que permiten la calibración y linealización de los servomotores; entre los cuales está el VI llamado **Calibración Sensores-Pololu Mode-Angle.vi**, que permite calibrar y determinar los límites de operación de cada servomotor, como también su linealización. La descripción de su operación se explica con mayor detalle en el capítulo 4 apartado 4.2.1. Cabe

anotarse que en caso de reemplazar un servomotor sin importar que sea de la misma marca o modelo, debe realizarse el proceso de calibración para este nuevo motor, antes de ensamblar el robot y cambiar todos los valores asociados a éste.

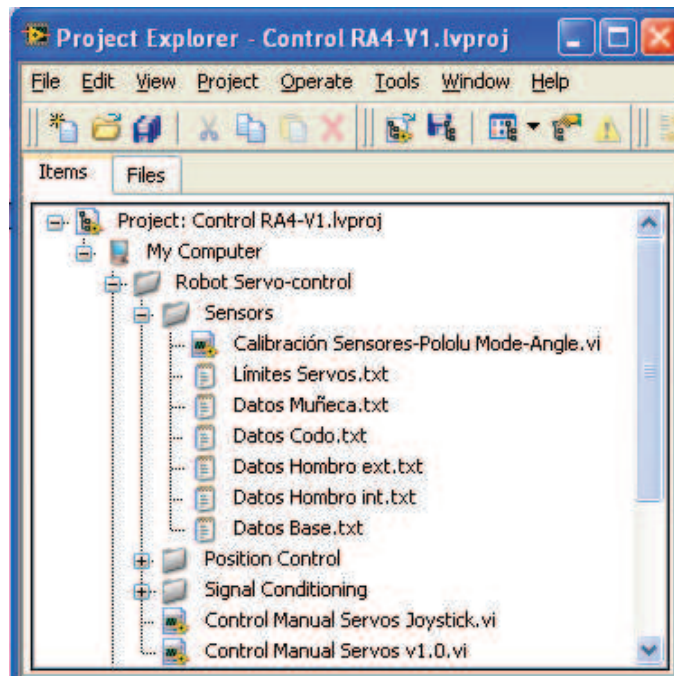
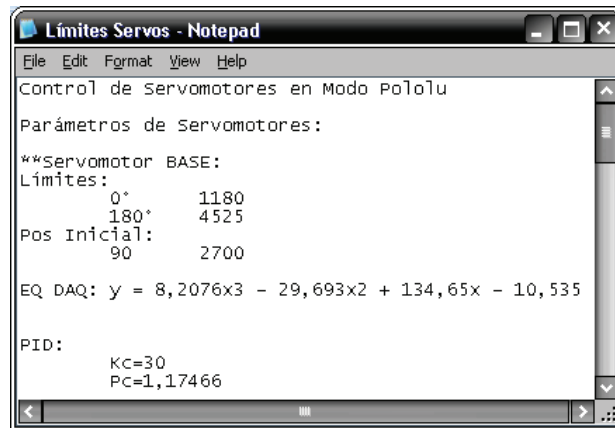


Figura A.3 Despliegue del directorio **Robot Servo-control / Sensors**

Además se cuenta con archivos .txt donde se guardan los respectivos valores de la calibración de cada servomotor durante la ejecución del programa, y además se cuenta con un archivo llamado *Límites servos.txt* que se usa para registrar todas las calibraciones para posteriormente ingresarlos de forma manual a los diferentes SubVI's.

Luego de realizar todas las calibraciones se debe guardar toda la información necesaria respecto a los límites de los valores enviados al controlador de servos Pololu, para obtener un rango de valores de 0 a 180°, en el archivo **Límites servos.txt**. También debe guardarse el valor de la posición inicial del servo para ubicar al robot en una ubicación específica antes de que el control en lazo cerrado entre en funcionamiento. Debe ingresarse además la ecuación del ajuste realizado en Microsoft Excel, con la ayuda de los datos obtenidos para cada uno de los servos.

Finalmente se pueden hacer pruebas mediante el VI: **Control PID Angular.vi**, ubicado dentro de la carpeta **Position Control** para hacer principalmente los ajustes de los parámetros de PID para cada articulación utilizando cualquier método, para luego registrar estos valores para configurar otros VI's donde interviene el control en lazo cerrado.



```

Límites Servos - Notepad
File Edit Format View Help
Control de Servomotores en Modo Pololu

Parámetros de Servomotores:

**Servomotor BASE:
Límites:
0°      1180
180°    4525
Pos Inicial:
90      2700

EQ DAQ:  $y = 8,2076x^3 - 29,693x^2 + 134,65x - 10,535$ 

PID:
      Kc=30
      Pc=1,17466
  
```

Figura A.4 Archivo .txt para el registro de datos

A.2.2 Position Control

Dentro de ésta carpeta se tiene los VI's y SubVI's del control de posición del Robot.

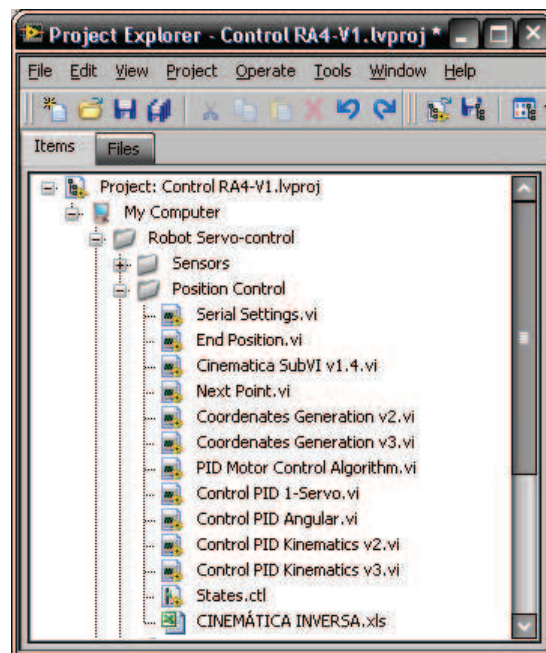


Figura A.5 Despliegue del directorio **Robot Servo-control / Position Control**

A continuación se describen brevemente los archivos que forman parte del control:



Serial Settings.vi: Éste SubVI, permite configurar la velocidad de actualización del valor de setpoint a cada servomotor a través del controlador de Servos Pololu, es decir, permite especificar un valor de velocidad fija para el giro de cada uno de los servomotores; y además permite ubicarlos en un ángulo inicial fijo.



End Position.vi: Éste SubVI, ubica al robot en una posición final o de reposo, que es igual a la posición inicial. A diferencia del anterior SubVI, éste no configura velocidad, ya que ya se encuentra configurada en el controlador de servos Pololu, por este motivo sólo es necesario enviar los setpoints correspondientes a la posición final, cuyos valores se tienen registrados en el archivo Límites servos.txt.



Cinemática SubVI v1.4.vi: en éste SubVI se calculan los ángulos correspondientes a cada articulación en base a los valores (x, y, z) de la posición que se desea obtener en el extremo del brazo robótico. Además éste SubVI tiene una alarma que indica si la posición (x, y, z) deseada puede ser alcanzada por el robot; es decir, analiza las restricciones físicas del sistema.



Next Point.vi: A través de éste SubVI se logra determinar el instante en que el valor de la realimentación y del setpoint tienen una diferencia menor a 1° . Cuando esto se cumple, se actualiza el valor de setpoint de posición, el cual debe ser cumplido por el robot.



Coordenates Generation v2.vi: Este SubVI permite generar la secuencia de posiciones que debe adoptar el Robot para cumplir con las tareas propuestas. En base a los datos de posición de los objetos detectados por los VI's de Visión Artificial y a la posición de destino ingresada por el usuario en el VI de control utilizado (ambos programas se describirán más adelante), este SubVI organiza en una tabla la secuencia de funcionamiento que ha de cumplirse para trasladar todos los objetos detectados a su correspondiente destino.



Coordenates Generation v3.vi: La función de éste SubVI es similar al anterior, la diferencia radica en que éste permite generar una secuencia de posiciones para trasladar los objetos en función de su forma y ubicarlos en destinos diferentes.



PID Motor Control Algorithm.vi: en éste SubVI se ejecutan los algoritmos de control PID para cada uno de los motores, teniendo en cuenta ángulos como valores de setpoint y de realimentación. Los límites de saturación de cada uno de los PID, corresponden a los valores límites obtenidos en la calibración de cada uno de los servomotores; éstos valores son los enviados al controlador de servos Pololu y deberán ser registrados en el proceso de calibración dentro del archivo **Límites Servos.txt**.

Control PID 1-Servo.vi: este VI fue programado básicamente para hacer pruebas de control para servomotores independientes, y así comprobar o calibrar las constantes de cada PID.

Control PID Angular.vi: este VI permite el control del Robot en lazo cerrado en base a los ángulos de referencia para cada articulación. Se debe tener precaución de usar éste VI, debido a que no considera las restricciones físicas impuestas por la cinemática para los ángulos ingresados, y podría causar averías en el Robot, por ese motivo deben ingresarse valores angulares que no causen daño a éste.

Control PID Kinematics v2.vi: con éste VI se puede lograr el control del efector final del Robot (Electroimán), a través de la cinemática inversa del Robot y del control en lazo cerrado. Conjuntamente con éste VI, puede funcionar el VI de Visión Artificial llamado **Vision Shapes Selector.vi**, el cual permite la selección de una forma específica de objeto, o bien detectar a todo objeto presente. Cuando se escoge un modo de funcionamiento automático se logra la ubicación del efector final en la ubicación exacta de los objetos encontrados. En modo manual lo único que se puede hacer es dar un setpoint de posición espacial (x, y, z) para el efector final desde el PC.

Control PID Kinematics v3.vi: éste VI es similar al anterior, aunque éste permite ubicar los diferentes tipos de objetos encontrados de acuerdo a su forma

y ubicarlos en posiciones de destino específicas ingresadas por el usuario. Para el funcionamiento en Modo Automático, es decir, en conjunto con Visión Artificial, deberá utilizarse con el VI: **Vision Clasifier.vi**, el cual de manera diferenciada entrega las posiciones de acuerdo a la forma del objeto, formando tres grupos que corresponden a las posiciones de Círculos, Cuadrados y Triángulos; los cuales son procesados por el VI de Control PID Kinematics v3, para ubicar cada uno de los objetos previamente diferenciados por su forma en posiciones anteriormente ingresadas.

States.ctl: es un control tipo Strict Type Definition, el cual consta de un control tipo Enum, que sirve para adicionar o quitar estados a la estructura de programación denominada máquina de estados de los diferentes SubVI's.

Cinemática Inversa.xls: en éste archivo de Microsoft Excel, se incorporan las ecuaciones de cinemática inversa y directa del robot, el cual permite básicamente comprobar los resultados obtenidos de manera rápida y confiable.

A.2.3 Signal Conditioning

Dentro de ésta carpeta se encuentran todos los SubVI's que intervienen en el acondicionamiento de las señales en las diferentes etapas del control.

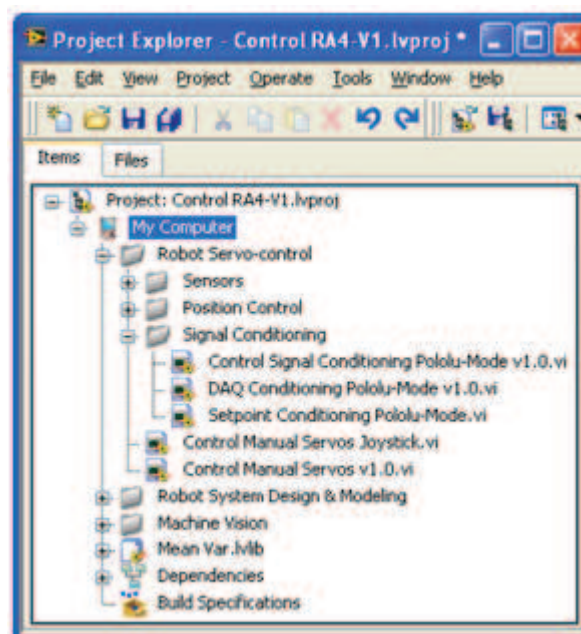


Figura A.6 Despliegue del directorio **Robot Servo-control / Signal Conditioning**

Los SubVI's se describen a continuación:



Control Signal Conditioning Pololu Mode v1.0.vi: a través de éste SubVI se logra limitar el rango de valores que se envían al controlador de servomotores Pololu, para evitar que valores arrojados por el control PID puedan superar las limitaciones físicas de los servomotores para cada articulación.



DAQ Conditioning Pololu Mode v1.0.vi: en éste SubVI, se ingresan las ecuaciones de ajuste que permiten linealizar los voltajes de realimentación obtenidos a través de la NI DAQ 6009, y pasarlos a valores angulares para que puedan ingresar al algoritmo de control, y ser comparadas con los setpoints que también se expresan como ángulos.

Control Manual Servos Joystick.vi: este VI permite el control angular del robot ya sea mediante la PC o a través de un Joystick, éste programa permite un control en lazo abierto de una manera rápida y sencilla.

A.2.4 Robot System Design & Modeling

En ésta carpeta se encuentran las partes del modelado CAD 3D del robot realizado en SolidWorks, además del VI que permite la comunicación LabVIEW – SolidWorks, que se detalla de mejor manera en el ANEXO C. El **VI Motor Sizing.vi** permite dimensionar los servomotores de cada articulación del Robot, a través del diseño en SolidWorks, el cual incluye todos los parámetros físicos necesarios, para de ésta manera obtener resultados aproximados a la realidad.

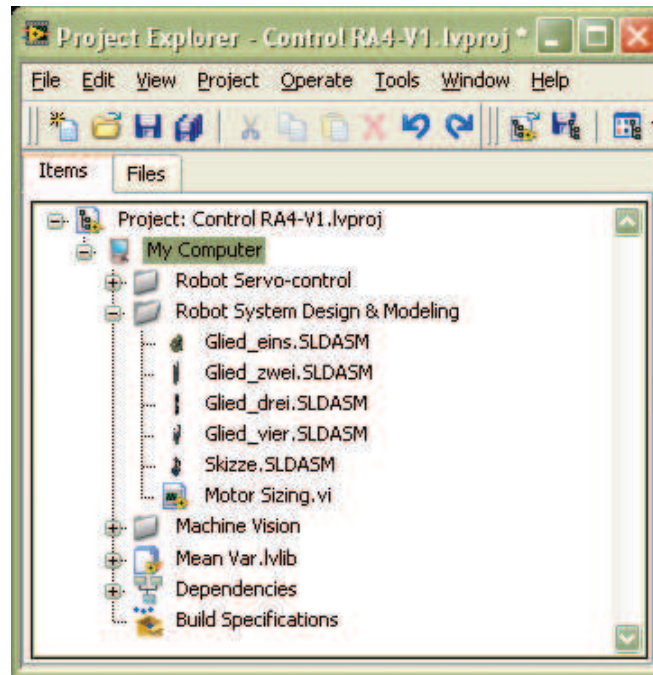


Figura A.7 Despliegue del directorio *Robot System Design & Modeling*

A.2.5 Machine Vision

Esta carpeta contiene todos los VI's necesarios para la aplicación de Visión Artificial dentro del proyecto.

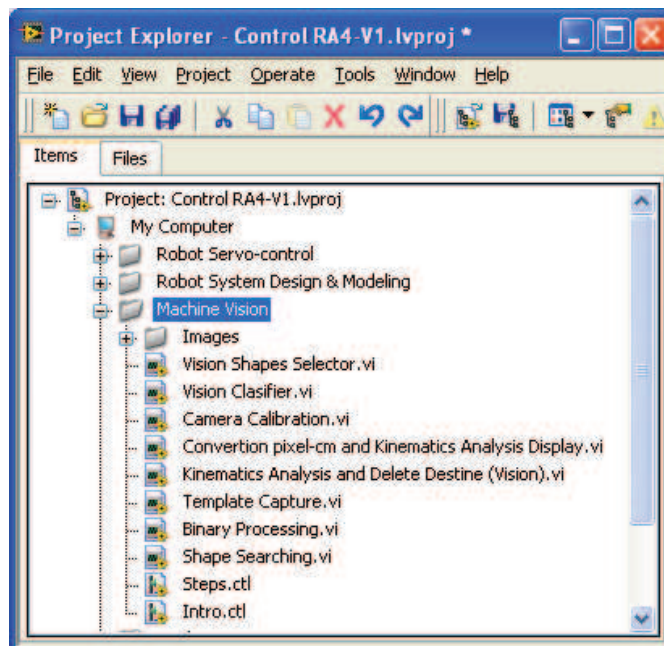


Figura A.8 Despliegue del directorio *Machine Vision*

Images: Subcarpeta que contiene todas las presentaciones del programa para el inicio de la aplicación de Visión Artificial, así como también los mensajes de interferencia, calibración, etc. Además de los patrones de cuadrado, círculo y triángulo para la búsqueda de éstos objetos en los diferentes VI's. En la Figura A.9 se pueden ver todos los archivos mencionados.

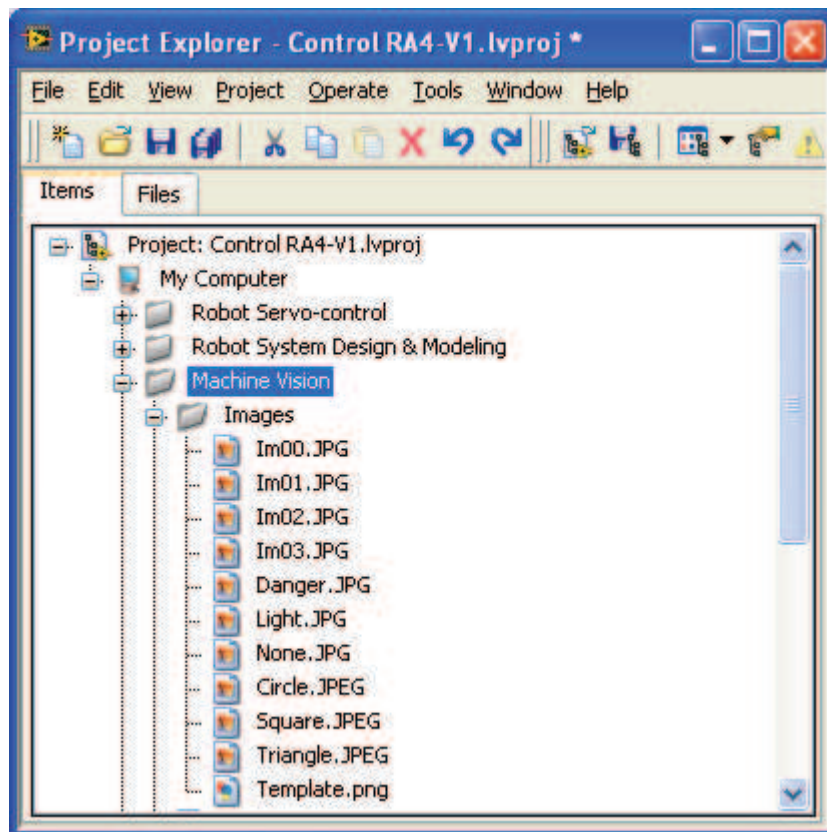


Figura A.9 Despliegue del directorio *Machine Vision / Images*

A continuación se describen cada uno de los VI's y SubVI's utilizados en Visión Artificial:

Vision Shapes Selector.vi: éste VI puede detectar las posiciones de todos los objetos presentes, o bien únicamente de objetos de una forma específica que puede ser escogida por el usuario. En cualquier caso el resultado es una lista de posiciones de los objetos detectados.

Vision Clasifier.vi: Por medio de éste VI se pueden clasificar tres tipos de formas (cuadrados, triángulos y círculos), es decir, generar una lista de posiciones para

cada grupo de formas predefinidas de manera independiente, ya sean éstas cuadrados, triángulos o círculos.

Camera Calibration.vi: éste VI permite la calibración básica de la imagen capturada mediante la WebCam, realizando una medición en píxeles de un objeto plano de dimensiones conocidas. A través de los píxeles obtenidos en la medición y de la dimensión conocida del objeto, el VI nos entrega la respectiva conversión píxeles-mm, que será utilizada en SubVI's de calibración más avanzada.



Conversion pixel-mm and Kinematics Analysis Display.vi: Éste

es un SubVI que permite la conversión de píxeles a milímetros de las coordenadas x-y de los objetos detectados mediante la cámara, y además muestra un mensaje en pantalla que indica si alguno de los objetos está fuera del área de trabajo del robot.



Kinematics Analysis and Delete Destine (Vision).vi: Éste SubVI,

permite determinar si las coordenadas de los objetos encontrados son alcanzables por el brazo, en caso contrario descarta éstas posiciones y genera un arreglo con las posiciones que pueden ser alcanzadas. Adicionalmente, este SubVI permite aplicar una máscara a la imagen para no considerar los objetos que ya han sido trasladados. El área asignada para la máscara es para $x < -110$ mm.

Template Capture.vi: Éste VI permite guardar una imagen que servirá como referencia para la búsqueda de objetos y la calibración de la cámara.



Binary Procesing.vi: Éste SubVI permite realizar todo el

procesamiento binario a las imágenes: Resaltar bordes, umbralización, rellenar huecos, rechazar objetos en contacto con borde de imagen y remover partículas aisladas.



Shape Searching.vi: Permite la búsqueda de geometrías previamente grabadas. Las geometrías que puede buscar pueden ser de diferente tipo,

basta con capturar una imagen de referencia para su búsqueda mediante el VI **Template Capture.vi**.

Steps.ctl: es un control Enum modificado tipo Strict Type Definition, el cual permite el funcionamiento de la Máquina de Estados, que sirve básicamente para el control de las diferentes etapas del sistema de Visión Artificial.

Intro.ctl: Al igual que el anterior, es un control Enum modificado tipo Strict Type Definition; que permite realizar la presentación inicial del programa de Visión Artificial.

A.2.6 Mean Var.lvlib

Dentro de ésta librería se tiene todas las variables globales utilizadas en el proyecto, como se indica en la Figura A.10.

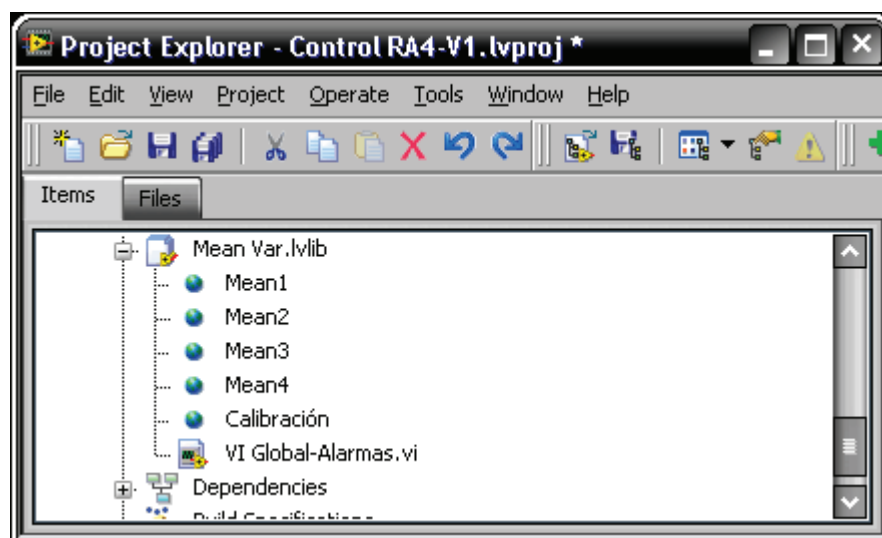


Figura A.10 Despliegue de las variables globales utilizadas

- *Mean1, Mean2, Mean3 y Mean4:* son las variables compartidas que indican el nivel medio de iluminación que existe en las líneas de perfil que se encuentran alrededor del plano de trabajo y sirven para la detección de interferencias.
- *Calibración:* variable compartida que entrega el ajuste necesario para las coordenadas de posición de los objetos en el plano de trabajo.

- **VI Global-Alarmas.vi:** VI global que contiene todas las variables globales utilizadas en el proyecto.

A.3 PASOS PARA EJECUTAR LOS VI's PARA CONTROL DEL ROBOT

Antes de ejecutar los VI's es necesario realizar las siguientes revisiones:

1. Verificar que las conexiones entre el robot y la caja de control estén correctas.
2. Comprobar que el cable de la fuente de poder esté conectado y posteriormente encender el sistema. Una vez encendido verificar de que no haya ningún problema de tipo eléctrico. ***En caso de cortocircuito o funcionamiento anormal, apague inmediatamente la caja de control.***
3. Verificar mediante el MAX (Measurement and Automation Explorer) de National Instruments, si la tarjeta de adquisición de datos y la WebCam han sido reconocidas y verificar que el nombre de la NI DAQ 6009 sea Dev2, caso contrario cambiarlo. Una vez identificados tanto la DAQ como la cámara, se puede comprobar si la realimentación de las señales es correcta moviendo los ejes de los servomotores a través del TestPanel de la DAQ y verificar que las señales de control estén entre los rangos de 0 a 2,5V, como se muestra en la Figura A.11.

Para la cámara, ésta debe estar lo más centrada posible de manera que capture correctamente la mayor parte del área de trabajo, y tener un modo de video de 640x480 RGB24 30.00fps. Esta configuración se muestra en la Figura A.12.

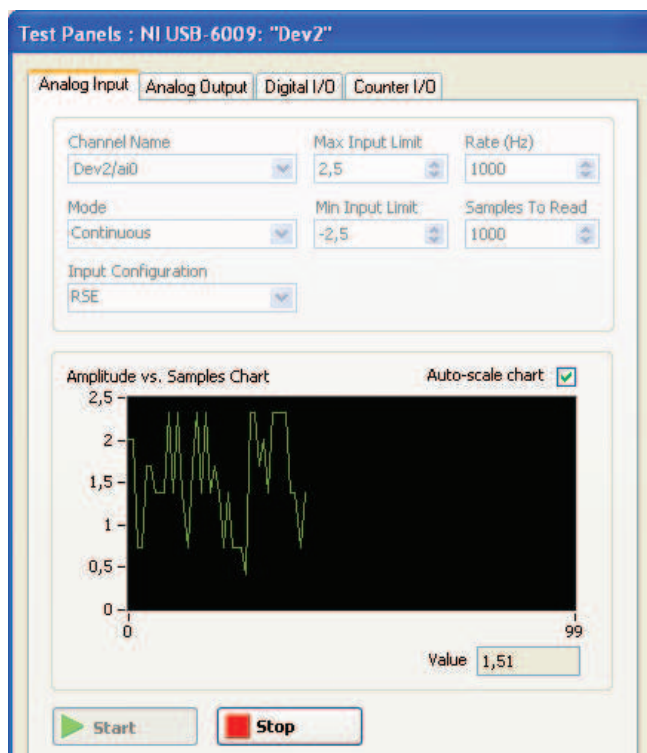


Figura A.11 Adquisición de señal de realimentación mediante Test Panels (MAX)

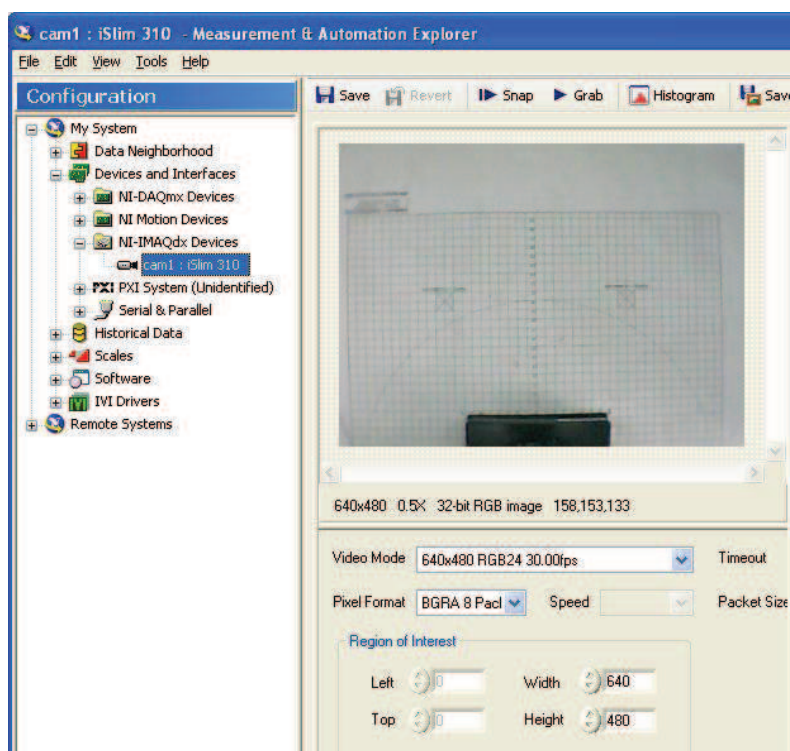


Figura A.12 Configuración de Cámara en MAX

Una vez realizados estos tres pasos, se puede seguir con la ejecución de los programas.

A.4 CONJUNTO DE VI's DEL SISTEMA DE CONTROL

Para el robot fueron creados VI's con diferentes tipos de funcionamiento, las cuales depende del programa utilizado de Visión Artificial y de Control.

A continuación se describen los más importantes.

A.4.1 SELECCIÓN DE FORMA

Para la detección de cualquier tipo de objeto o bien reconocer un objeto de forma determinada y la posterior recolección y reubicación del mismo pueden usarse los siguientes VI's:

- Control PID Kinematics v2.vi
- Vision Shapes Selector.vi



Figura A.13 Panel Frontal VI: Control PID Kinematics v2

Si se quiere realizar un funcionamiento manual sin Visión Artificial, el primer botón de la parte inferior derecha de la pantalla llamado WITHOUT VISION debe estar en NO VISION y el botón siguiente de MODO DE TRABAJO debe estar en

MANUAL. Para de ésta manera dar paso a los controles ubicados en la parte superior izquierda dentro del bloque SETPOINT DE POSICIÓN.

Antes de ejecutar el programa debe asegurarse de haber escogido correctamente el puerto de comunicación Serial y verificar que los valores de los parámetros de los PID estén correctos (compararlos con los registrados en el archivo **Límites Servos.txt**).

Para la ejecución con el programa de Visión Artificial, deben estar los botones de control de la parte inferior derecha de la pantalla en VISION y en modo AUTO. Además debe ingresar la posición de destino de los objetos detectados. Una vez configurados estos parámetros se puede ejecutar el programa y pasar a la ventana de Visión Artificial que se muestra en la Figura A.14.



Figura A.14 Panel Frontal VI: Vision Shapes Selector

En la parte izquierda del Panel Frontal está el indicador llamado Imagen Fuente de los objetos binarizados con las respectivas posiciones, en la Parte derecha en Imagen Actual, se muestra en tiempo real, la captura de video del plano de trabajo, y debajo de éste está una estadística de la cantidad de objetos de una determinada forma. En la parte derecha está un selector de modo de trabajo que permite escoger entre modo NORMAL y SHAPES. El modo normal detecta a todos los objetos sin importar su forma, y el modo shapes detecta únicamente los

objetos que cumplen con la forma especificada en el selector llamado 'Figura a Buscar'.

Una vez que se corre el VI, en primer lugar debe calibrarse los ejes de coordenadas, para eso deben seguirse los pasos que indique el programa. Posteriormente, se realiza la calibración de iluminación, durante la cual no debe ingresarse objetos extraños de ninguna manera al plano de trabajo o interferir con la iluminación del mismo, ya que se interferirá con los cálculos del nivel medio de iluminación de las líneas de perfil.

Una vez terminadas estas etapas se puede ingresar los objetos que se deseen en cualquier lugar, y paralelamente se podrá visualizar en el Panel Frontal lo que sucede en el plano de trabajo, además de indicarnos las posiciones de cada uno de los objetos. Una vez ubicados los objetos debe presionarse el botón llamado 'EJECUTAR' ubicado en la parte superior derecha para compartir los datos de las posiciones de los objetos, los cuales serán leídos por el VI de Control y automáticamente el extremo del robot se ubicará en las posiciones correspondientes empezando por la primera posición dentro del arreglo de posiciones creado. Al finalizar, el robot se situará en su posición de reposo hasta que nuevamente se ubiquen nuevos objetos y se presione EJECUTAR. Dentro del programa de Visión Artificial, se encuentra el algoritmo que permite la detección de interferencia, es decir, en el momento en que un objeto extraño ingrese al plano de trabajo se generará una alerta que detendrá el Robot en caso de que se encuentre en movimiento hasta que el peligro desaparezca.

Mientras se ejecutan los programas, se tiene la posibilidad de pasar de una ventana a otra, por ejemplo para visualizar las curvas de respuesta de cada servomotor en el Panel Frontal de control, inclusive para cambiar parámetros de control o pasar a un modo de funcionamiento MANUAL de ser el caso.

Finalmente, cuando se quiera detener el programa, basta con presionar ESC o hacer clic el botón STOP dentro del Panel Frontal de Vision Artificial o del programa de Control.

A.4.2 CLASIFICACIÓN POR FORMA

Para la clasificación de objetos de acuerdo a su forma, debe utilizarse los siguientes VI's:

- Control PID Kinematics v3.vi
- Vision Classifier.vi



Figura A.15 Panel Frontal VI: Control PID Kinematics v3

A diferencia del VI **Control PID Kinematics v2**, consta de tres controles para ingresar las posiciones de destino de cada uno de los objetos de acuerdo a la forma. El modo de operación del VI, es igual al anterior.

Para la parte de visión Artificial se tiene el VI **Vision Classifier**, en la Figura A.16 se indica el Panel Frontal de dicho VI.



Figura A.16 Panel Frontal VI: Vision Clasifier

Como se puede ver en la Figura A.16, a diferencia del anterior VI de Visión Artificial, éste consta de tres indicadores de las posiciones para cada forma. Este programa no da la opción de cual se quiere reubicar, ya que envía tres grupos de posiciones para cada forma; de ésta manera el VI de Control los ubica de acuerdo a las posiciones ingresadas.

A.4.3 CONTROL MANUAL DEL SISTEMA

Como ya se indicó anteriormente, se pueden utilizar los programas de control en forma independiente para realizar un control manual sobre el robot, pero adicionalmente se tienen VI's que permiten el control manual en lazo abierto del robot mediante joystick. En la Figura A.17 se indica el Panel Frontal del VI utilizado.

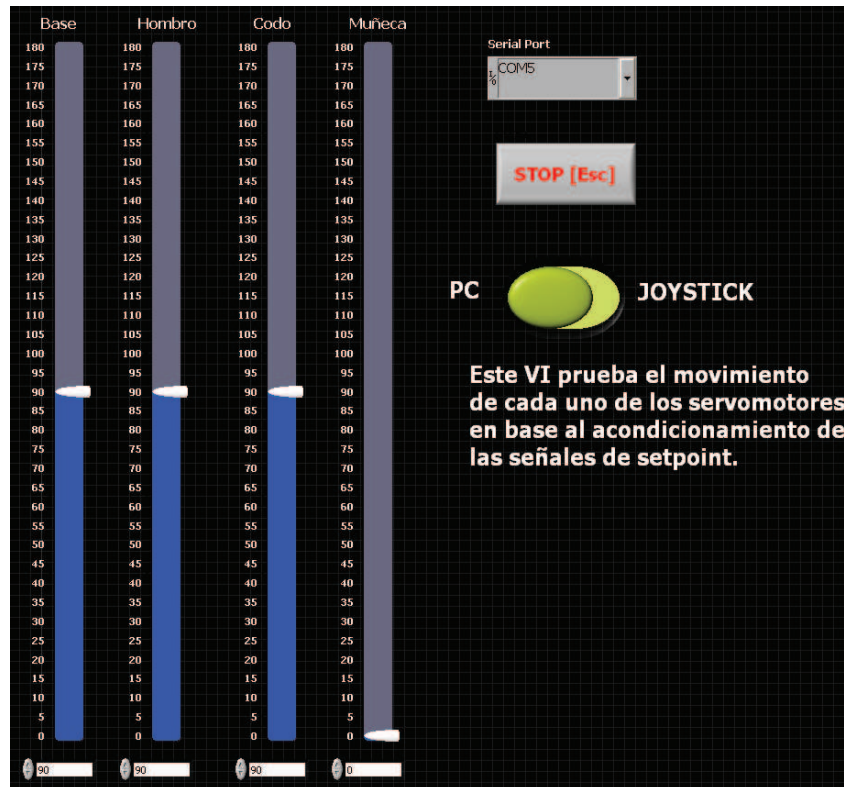


Figura A.17 Panel Frontal VI: Control Manual Servos Joystick

Este VI se encuentra dentro de la carpeta **Robot Servo-control**. Al igual que todos los VI's de control, debe configurarse el Puerto Serial que se esté utilizando para comunicarse con el Controlador de Servomotores Pololu.

Este VI permite la selección del interfaz de control que se desea utilizar, ya sea mediante la PC o mediante un Joystick.

Debe tenerse cuidado en cuanto al ángulo que se ingrese, ya que puede causar averías en el Robot, por ese motivo debe moverse con prudencia los ángulos de las articulaciones.

Para detener el programa basta con presionar ESC o hacer clic en STOP.

ANEXO B

PLANOS DE LAS PIEZAS QUE COMPONEN AL ROBOT

ANEXO C

USO DE LabVIEW MECHATRONICS TOOLKIT

LabVIEW permite al usuario desarrollar una interfaz entre un archivo de ensamblaje de SOLIDWORKS y un VI (archivo de LabVIEW), mediante el Toolkit de Mecatrónica, el cual permite cargar perfiles de movimiento al archivo de ensamblaje. Para que la interfaz tenga éxito deben cumplirse las siguientes condiciones:

- El archivo de ensamblaje debe estar abierto.
- COSMOSMotion debe estar habilitado.
- Deben estar definidas partes fijas, partes móviles, juntas, etc. dentro del análisis de movimiento. La configuración de estos parámetros ya fueron vistos en el CAPÍTULO 2.

Para cargar perfiles de movimiento en el archivo de ensamblaje, LabVIEW ofrece varias opciones, las cuales se citan a continuación:

- Cargar perfil de movimiento definido por un contorno que el usuario ingresa en forma de pares ordenados tiempo vs posición.
- Construir perfil de movimiento en base a VI's existentes: Straight Line Move y Arc Move.
- Construir perfil de movimiento en base a VI's Express que generan señales simuladas como por ejemplo una onda sinusoidal.

Para construir con éxito el perfil de movimiento deseado, hace falta configurar completamente los VI's antes citados. Para esta configuración debe cumplirse lo siguiente:

- Escoger el tipo de movimiento a realizarse.
- Escoger la articulación que se va a mover mediante su nombre usado en COSMOSMotion.

- Verificar que el tipo de movimiento sea congruente con el tipo de articulación; por ejemplo, si se tiene una articulación rotacional, el tipo de movimiento asignado también debe ser rotacional y no traslacional.

Una vez configurados estos parámetros, se puede ejecutar el archivo de LabVIEW y observar el movimiento del ensamble, en el archivo de SOLIDWORKS. A continuación se detallan las funciones usadas y el procedimiento a seguir, para usar el Mechatronics Toolkit de LabVIEW como generador de perfiles de movimiento.

C.1 MECHATRONICS TOOLKIT

El Toolkit de Mecatrónica de LabVIEW, tiene funciones o VI's especialmente diseñados para crear, administrar y ejecutar perfiles de movimiento, que serán aplicados a un archivo de ensamblaje de SOLIDWORKS.

La compatibilidad entre los dos paquetes de software soporta versiones desde LabVIEW 8.2, y únicamente SOLIDWORKS 2007. Las versiones siguientes de SOLIDWORKS también son compatibles con LabVIEW mediante el Toolkit NI SoftMotion, cuyo uso es más complejo. Para usar el Mechatronics Toolkit, debe usarse únicamente SOLIDWORKS 2007.

C.1.1 FUNCIONES PARA CREAR MOVIMIENTO

Las funciones del Toolkit de Mecatrónica que permiten crear movimiento, han sido desarrolladas en base a las funciones que forman parte de la paleta Motion Assistant. Principalmente existen tres tipos de funciones, que serán vistas a continuación.

C.1.1.1 1D or 2D Contour Move.vi

Este VI carga un arreglo 2D de puntos desde un archivo y genera un perfil de movimiento basado en los parámetros configurados por el usuario. Las entradas necesarias para que este VI se ejecute, son las siguientes:

- *Contour File Path*: ruta del archivo que contiene el arreglo 2D de puntos.

- *Profile Calculations Constraints*: define el límite del perfil de movimiento (tiempo de ejecución, o bien número de puntos en un arreglo a ser ejecutados)
- *Move Constraints*: define parámetros que sirven de referencia o límite para desarrollar el movimiento. Estos parámetros son:
 - Tipo de perfil de velocidad (Trapezoidal: aceleración y deceleración súbita; S Curve: aceleración y deceleración suavizada)
 - Máxima velocidad de la trayectoria en rev/segundo.
 - Máxima Aceleración del movimiento en rev/seg^2
 - Máxima Deceleración del movimiento en rev/seg^2
 - Máxima variación de la aceleración (derivada de la aceleración): Jerk
 - Velocidad Inicial
 - Velocidad Final
- *Motion Profile In*: en caso de que exista un perfil de movimiento previamente creado, se lo ingresa en esta entrada, y se agrega a dicho perfil, el movimiento a ser creado por este VI.
- *Joint Name*: arreglo que contiene el nombre de las articulaciones sobre las cuales se va a aplicar el movimiento. Los nombres escritos deben ser los mismos usados en el análisis de movimiento realizado en COSMOSMotion.
- *Motion On*: especifica el tipo de movimiento a realizar (rotacional o traslacional) en las articulaciones escogidas, y el eje sobre el cual se realizará (x, y, z); es decir, selecciona el grado de libertad.
- *Profile Scaling Factor*: permite modificar la escala de los puntos que conforman el perfil de movimiento.

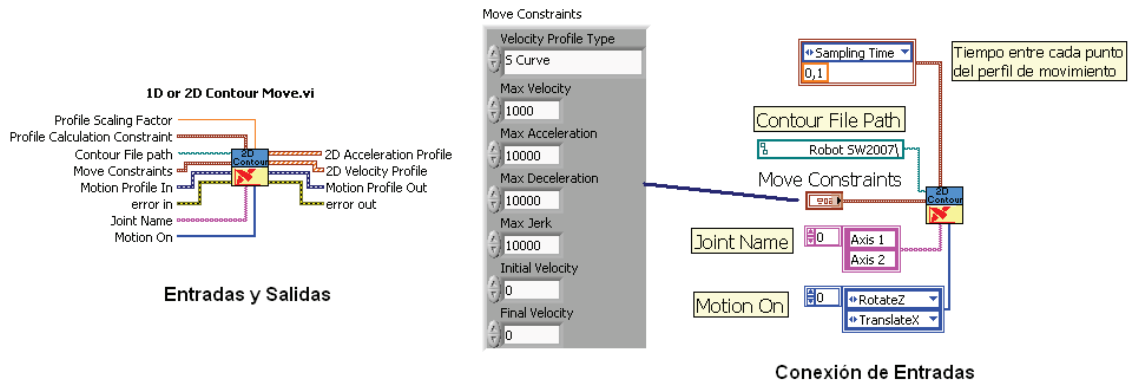


Figura C.1 1D or 2D Contour Move.vi Símbolo y principales conexiones

Este VI proporciona información de velocidad y aceleración vs tiempo en 2D como salidas (*2D Acceleration/Velocity Profile*), los cuales pueden ser graficados mediante una Waveform Graph. La salida *Motion Profile Out*, es el perfil que será cargado en COSMOSMotion.

C.1.1.2 1D or 2D Straight Line Move.vi

Este VI genera movimiento rectilíneo en 2 o 3 dimensiones, basado en las propiedades y restricciones definidas por el usuario. A continuación se detalla el significado de cada entrada del VI.

- *Straight Line Move Properties*: define las propiedades del movimiento rectilíneo.
 - *Dimension*: especifica el número de ejes del movimiento (2 o 3).
 - *Move Mode*: determina si el movimiento será realizado en función de un control de posición o de velocidad (Position Mode, Velocity Mode).
 - *Position Mode – End Point*: punto final del movimiento para cada eje (x, y, z). Es válido solo en caso de escoger *Position Mode* en la entrada *Move Mode*.
 - *Velocity Mode – Travel Time*: determina la velocidad a la que se moverá el ensamble sobre cada eje. Es válido solo en caso de escoger *Velocity Mode* en la entrada *Move Mode*.

Las demás entradas son similares a las utilizadas anteriormente para el VI *1D or 2D Contour Move.vi*.

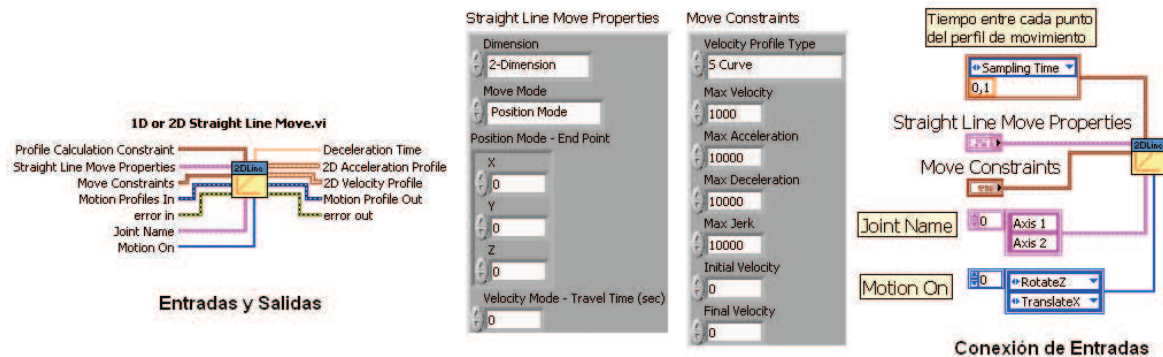


Figura C.2 1D or 2D Straight Line Move.vi Símbolo y principales conexiones

Este VI también proporciona información sobre perfiles de velocidad y aceleración como salidas. *Motion Profile Out*, es el perfil de movimiento que será cargado en COSMOSMotion.

C.1.1.3 2D Arc Move.vi

Este VI genera un movimiento en forma de arco en dos dimensiones, basado en las propiedades y restricciones definidas por el usuario. A continuación se detalla el significado de cada entrada del VI.

- *Arc Move Properties*: define las propiedades del movimiento en forma de arco.
 - *Arc Type*: define el tipo de arco a realizarse (2D: circular; 3D: spherical, helical).
 - *Start Degrees*: especifica el ángulo de partida desde el cual se formará el arco.
 - *Travel Degrees*: especifica la distancia angular total que recorrerá el arco. Para múltiples giros se debe ingresar un ángulo mayor a 360°.

- *Yaw, Pitch & Roll*: parámetros válidos solo para arcos 3D. Rotan al arco alrededor de los ejes 'z', 'x' y 'y' respectivamente, a través de su punto de partida en grados.
- *Z Travel*: válido para arcos 3D en forma helicoidal. Define la distancia lineal entre los extremos de la helicoidal.
- *Radius*: especifica el radio del arco en grados.

Las demás entradas son similares a las utilizadas anteriormente para los VI's *1D or 2D Contour Move.vi* y *1D or 2D Straight Line Move.vi*.

Este VI también proporciona información sobre perfiles de velocidad y aceleración como salidas. *Motion Profile Out*, es el perfil de movimiento que será cargado en COSMOSMotion.

Es posible crear perfiles de movimiento más complejos, conectando en serie varios VI que generan líneas, arcos o perfiles personalizados, uniendo la salida *Motion Profile Out* de un VI con la entrada *Motion Profile In* del siguiente VI.

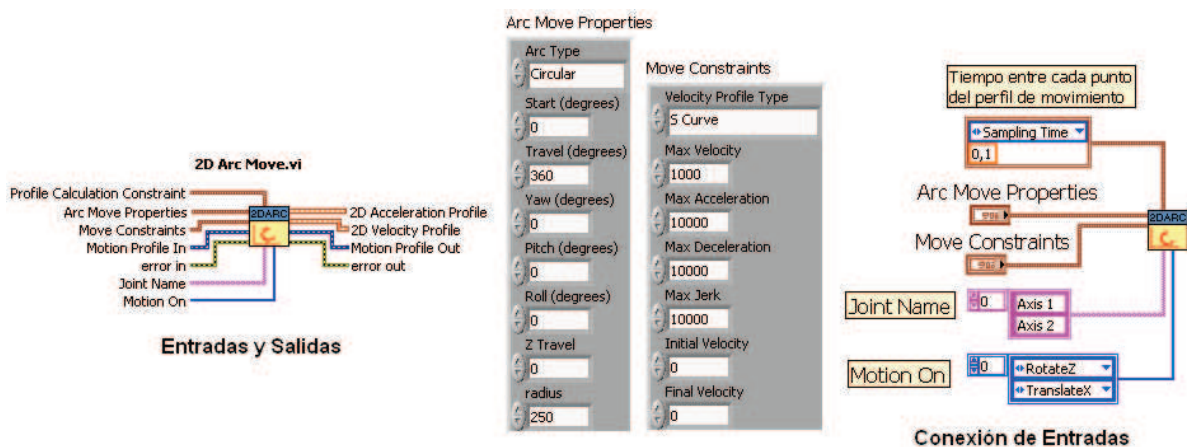


Figura C.3 2D Arc Move.vi Símbolo y principales conexiones

C.1.1.4 Funciones Simuladas

Es posible asignar también funciones de movimiento conocidas, como por ejemplo armónicas, constantes, paso, etc.; para ello se usan VI's Express que

permiten simular estas funciones. Por ejemplo para simular una función armónica (sinusoide) se usa el VI *Simulate Signal*, en donde se configuran amplitud y frecuencia, que en parámetros de movimiento son traducidas a distancia angular y velocidad angular respectivamente.

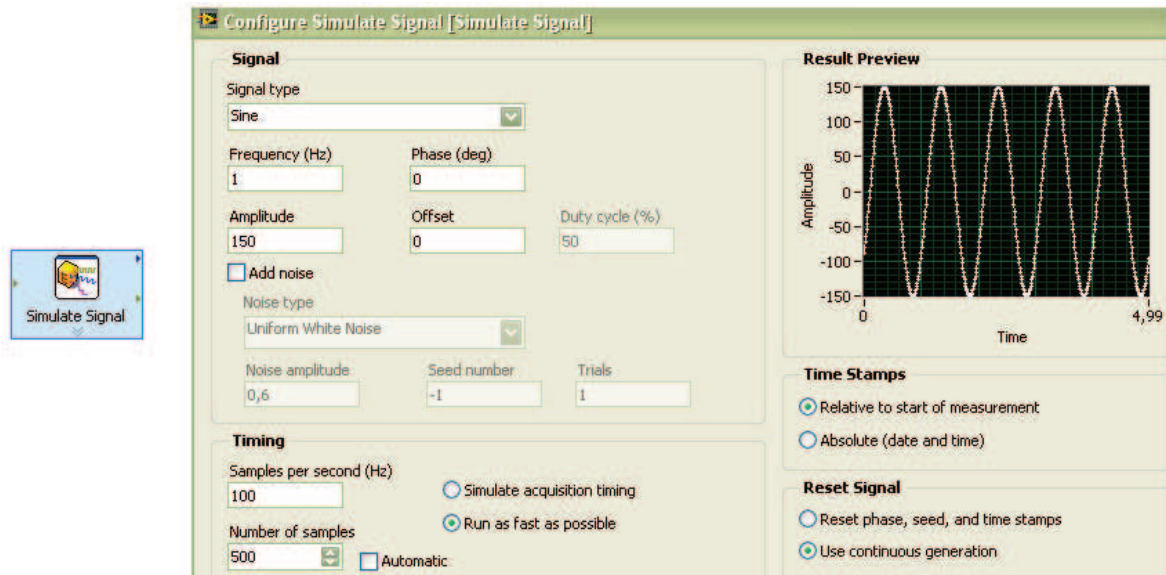


Figura C.4 VI Simulate Signal y su configuración

Mediante las opciones *Samples per second* y *Number of Samples*, se define el tiempo de simulación, como la relación:

$$time = \frac{Number\ of\ Samples}{Samples\ per\ Second}$$

A diferencia de los VI's propios del Toolkit de Mecatrónica, para que los datos de salida de este VI sean compatibles con el perfil de movimiento que usa SOLIDWORKS, se necesita de un convertor. Este convertor también forma parte del Toolkit de Mecatrónica, y se describe a continuación.

C.1.1.1.1 Convertor Dynamic Data – SOLIDWORKS Attributes

Este convertor recibe como entrada el dato dinámico que entrega el VI Simulate Signal y lo transforma en un perfil de movimiento, para lo cual requiere de otras entradas detalladas a continuación.

- *Motion On*: especifica el tipo de movimiento a realizar (rotacional o traslacional) en las articulaciones escogidas, y el eje sobre el cual se realizará (x, y, z); es decir, selecciona el grado de libertad.
- *Motion Type*: especifica el tipo de movimiento a realizarse sobre la articulación escogida (Libre, Desplazamiento, Velocidad, Aceleración).
- *Joint Name*: arreglo que contiene el nombre de las articulaciones sobre las cuales se va a aplicar el movimiento. Los nombres escritos deben ser los mismos usados en el análisis de movimiento realizado en COSMOSMotion.

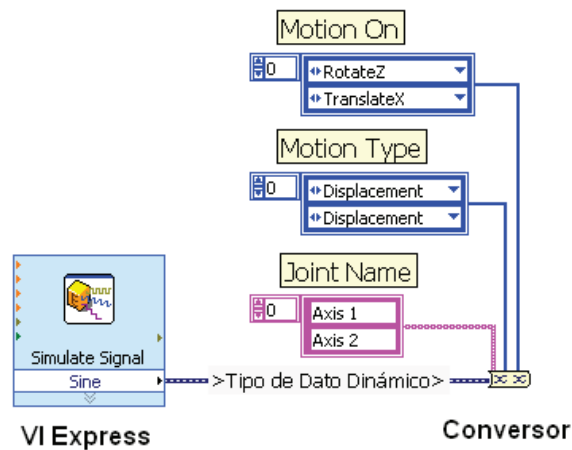


Figura C.5 Conversor Dynamic Data – SOLIDWORKS Atributes y su conexión

La salida del conversor es compatible con la configuración del análisis de movimiento desarrollado en COSMOSMotion.

Luego de haber creado el perfil de movimiento, es necesaria la utilización de una función que sirve de nexo entre el perfil creado y el ensamble de SOLIDWORKS.

C.1.2 FUNCIONES PARA EXPORTAR PERFIL DE MOVIMIENTO

Luego de haber creado y configurado el perfil de movimiento, es necesario exportarlo hacia el archivo de ensamblaje de SOLIDWORKS; para ello se utiliza una función del Toolkit de Mecatrónica denominada '*Run COSMOSMotion Simulation*', la cual se describe a continuación.

C.1.2.1 Run COSMOSMotion Simulation.vi

Este VI borra los datos de simulación existentes en COSMOSMotion, carga los datos del perfil de movimiento desarrollado en LabVIEW dentro de la configuración de las articulaciones; y finalmente ejecuta la simulación del mecanismo. Hay que tener claro que al ejecutar este VI se borra el perfil de movimiento anterior y no la configuración de las articulaciones.

Los atributos incluidos en el perfil de movimiento desarrollado en LabVIEW deben coincidir con la configuración de articulaciones y restricciones desarrolladas en COSMOSMotion, en cuanto a nombres, tipos de articulación, y tipo de movimiento.

A continuación se describen las entradas y salidas que posee este VI.

- *Motion Profile In*: recibe los datos dinámicos del perfil de movimiento que contiene formas de onda del desplazamiento, velocidad y aceleración; y además recibe la configuración de las articulaciones y sus atributos a ejecutarse en COSMOSMotion.
- *Simulation Time Increment*: especifica el tiempo de actualización de los puntos para el perfil de movimiento; es decir, configura el tiempo que transcurre entre la ejecución de un punto y otro, de tal forma que permite controlar el tiempo total de la simulación.
- *Motion Profile Out*: contiene el perfil de movimiento que fue cargado en la simulación de COSMOSMotion.

Un ejemplo de uso del VI '*Run COSMOSMotion Simulation*' se presenta en la siguiente figura.

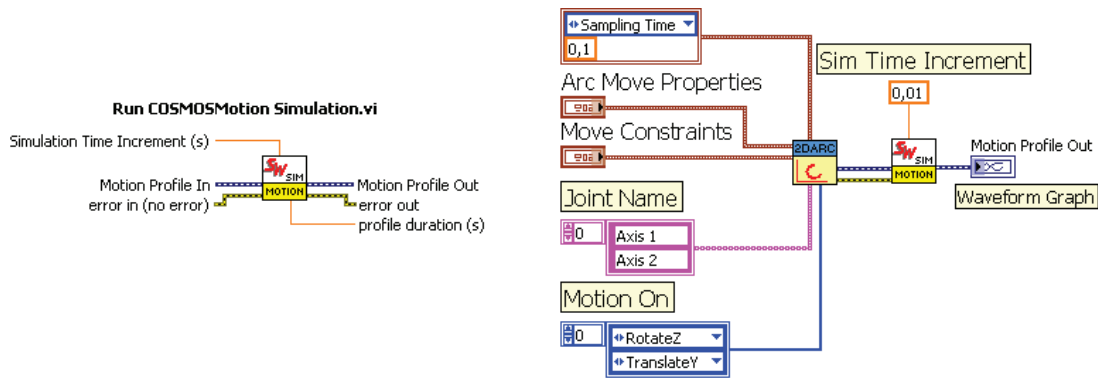


Figura C.6 Uso y configuración del VI Run COSMOSMotion Simulation

C.1.3 FUNCIONES ADICIONALES

Además de las funciones antes mostradas, el Toolkit de Mecatrónica contiene funciones que permiten visualizar los perfiles de movimiento, administrar la simulación en COSMOSMotion de forma personalizada; es decir, no usar el VI Run COSMOSMotion Simulation, sino construir la interfaz para exportar el perfil de movimiento. En este trabajo únicamente se abordarán funciones que permiten visualizar los perfiles de movimiento creados.

C.1.3.1 Plot XY moves.vi

Este VI permite transformar el perfil de movimiento creado en un arreglo de dos dimensiones que contienen puntos (x, y) que pueden ser graficados en una Waveform Graph. Esta función tiene solo una entrada para el perfil de movimiento, y una salida para la gráfica.

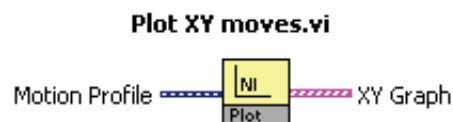
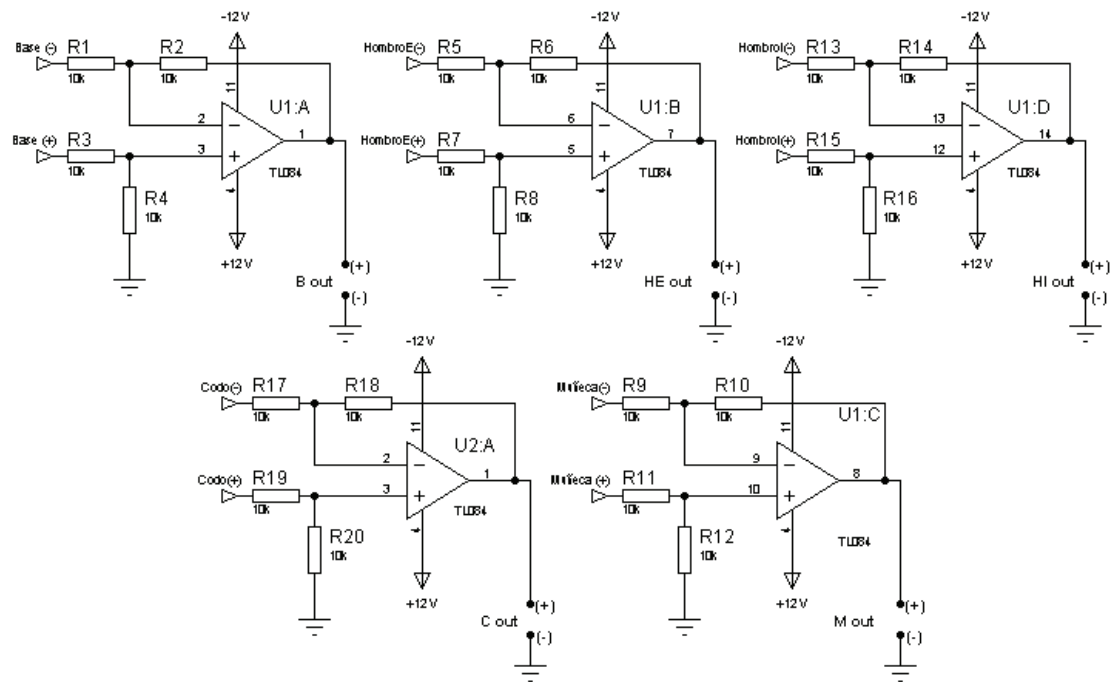


Figura C.7 Plot XY moves.vi

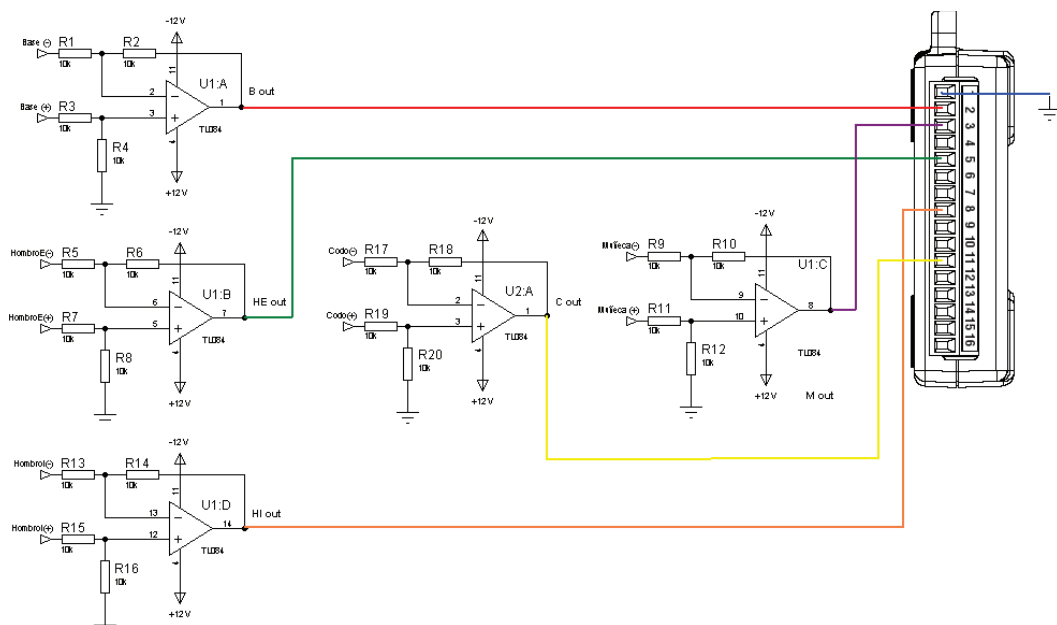
ANEXO D

CIRCUITO REFERENCIADOR Y DE CONTROL DEL ELECTROIMÁN

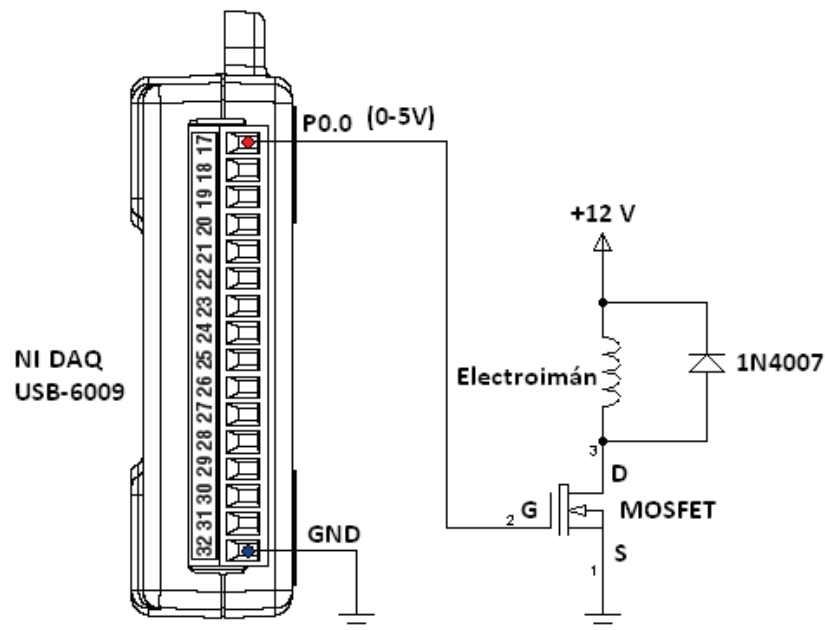
A.1 ESQUEMÁTICO CIRCUITO REFERENCIADOR



A.2 CONEXIONES ENTRE CIRCUITO REFERENCIADOR Y NI DAQ 6009



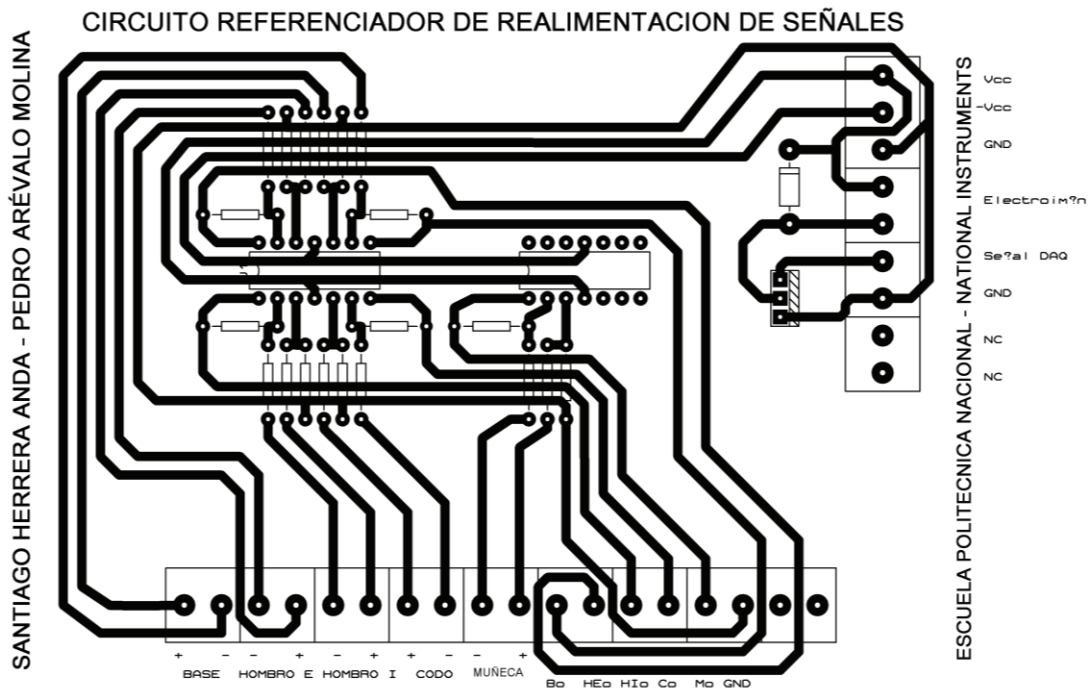
A.3 CONEXIÓN ENTRE CIRCUITO DE CONTROL ELECTROIMÁN Y NI DAQ 6009



A.4 RESUMEN DE CONEXIONES ENTRE LOS CIRCUITOS Y NI DAQ 6009

TERMINAL NI DAQ 6009	SEÑAL DAQ MODO: TIERRA COMÚN	BORNERA CORRESPONDIENTE DEL CIRCUITO REFERENCIADOR
1	GND	GND
2	AI 0	Bo
3	AI 4	Mo
4	GND	-
5	AI 1	HEo
6	AI 5	-
7	GND	-
8	AI 2	Hlo
9	AI 6	-
10	GND	-
11	AI 3	Co
12	AI 7	-
13	GND	-
14	AO 0	-
15	AO 1	-
16	GND	-
17	PO.0	SEÑAL DAQ
32	GND	GND

A.5 PCB CIRCUITO REFERENCIADOR A TIERRA Y CONTROL ELECTROIMÁN



A.5.1 DESCRIPCIÓN ETIQUETAS

Base: indica las borneras para conectar la señal de realimentación de la Base del Robot, con su respectiva polaridad.

Hombro E: indica las borneras para conectar la señal de realimentación del Hombro Externo (Servomotor que sobresale en la estructura), con su respectiva polaridad.

Hombro I: indica las borneras para conectar la señal de realimentación del Hombro Interno (Servomotor que se encuentra entre los perfiles de acrílico), con su respectiva polaridad.

Codo: indica las borneras para conectar la señal de realimentación del servomotor de la articulación denominada Codo, con su respectiva polaridad.

Muñeca: indica las borneras para conectar la señal de realimentación del servomotor de la última articulación del robot, con su respectiva polaridad.

Bo, Meo, Mlo, Co, Mo: borneras para los voltajes de salida de los circuitos para referenciar las señales de realimentación de los servomotores de la Base, Hombro Externo, Hombro Interno, Codo y Muñeca, respectivamente. Estas borneras deben ser conectadas directamente a la NI DAQ 6009 como muestra la Figura D.6.

GND: Tierra común para todas las señales de realimentación de los servomotores; ésta debe conectarse a la tierra de las entradas análogas de la tarjeta NI DAQ 6009, mostrada en la Figura D.6.

En la parte superior derecha de la Figura D.5 se indican las siguientes etiquetas:

Vcc: Corresponde al voltaje de alimentación de +12 V de los amplificadores operacionales TL084 para referenciar las señales a tierra y además sirve para la alimentación del electroimán.

-Vcc: Es el voltaje de -12 V para alimentación exclusiva de los amplificadores operacionales TL084.

GND: Tierra común de todo el Circuito. Es la misma tierra usada para el circuito referenciador y las entradas análogas de la NI DA 6009.

Electroimán: Son las dos borneras utilizadas para conectar el electroimán, en éste caso la polaridad es indiferente.

Señal DAQ: en ésta bornera se debe conectar la salida digital P0.0 de la NI DAQ 6009 para activar el Electroimán. Como se indica en la Figura D.3.

GND: en ésta bornera debe conectarse la tierra de la tarjeta NI DAQ 6009, correspondiente a las salidas digitales de la misma.

