

Programación de trayectorias para un brazo robótico puma empleando un PLC

Ivon Yiraudi Beltran Niño y José David Alba Morán

Trabajo de Grado para optar al título de Ingeniero Electrónico

Director

Ricardo Alzate Castaño

Ph.D. en Ingeniería Informática y Automática

Universidad Industrial de Santander

Facultad de Ingenierías fisicomecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2021

Contenido

	Pág.
Introducción	7
1.1 Objetivos	12
1.1.1 Objetivo general	12
1.1.2 Objetivos específicos	12
2. Programación visual de trayectorias	12
2.1 Cinemática de un robot	13
2.1.1 Análisis cinemático del manipulador PUMA	13
2.2 Visión artificial	19
2.2.1 Captura de imágenes	20
2.2.2 Preproceso	22
2.2.3 Proceso: cálculo del centroide	23
2.3 Manipulador robótico experimental	26
2.4 Sistema de Servo-Visión propuesto	33
3. Verificación del movimiento	38
3.1 Simulación del sistema de Servo-Visión	39
3.2 Análisis de resultados	42
4. Conclusiones	47
5. Recomendaciones	49
5.1 Trabajo futuro	49
Referencias Bibliográficas	50

Lista de Figuras

	Pág.
Figura 1. Manipulador robótico PUMA.....	9
Figura 2. Comparación robot PUMA y brazo humano.....	14
Figura 3. Proyección planar del movimiento del manipulador PUMA	15
Figura 4. Ilustración para movimiento de un vector en el plano	15
Figura 5. Movimiento en el plano para dos articulaciones	18
Figura 6. Etapas de un sistema de visión artificial.....	20
Figura 7. Ilustración para escenario de captura	21
Figura 8. Resultados etapa de preproceso.....	23
Figura 9. Determinación geométrica del centroide.....	24
Figura 10. Ilustración cálculo numérico del centroide.....	25
Figura 11. Prototipo experimental E3T-UIS para manipulador PUMA MA2000.....	27
Figura 12. Motores y servomotores empleados en manipulador experimental	28
Figura 13. Esquema de conexiones para el manipulador robótico experimental	30
Figura 14. Maquina de estado finito para operación del manipulador	35
Figura 15. Esquema para sistema de Servo-Visión en manipulador experimental.....	37
Figura 16. Programación LADDER en TIA Portal para emular movimiento de M3.....	41
Figura 17. Esquema de simulación para sistema de Servo-Visión	42
Figura 18. Interfaz de simulación en panel HMI	42
Figura 19. Base de datos del registro de imágenes	44

Lista de Tablas

	Pág.
Tabla 1. Convenciones empleadas en máquina de estados.....	33
Tabla 2. Pruebas para localización y seguimiento automático de coordenadas	45

Resumen

Título: Programación de trayectorias para un brazo robótico puma empleando un PLC*

Autor: Ivon Yiraudi Beltran Niño; José David Alba Morán**

Palabras Clave: Cálculo cinemático, manipulador robótico puma, servo-visión, visión artificial.

Descripción:

El presente trabajo de grado aborda la programación automática de trayectorias en un manipulador robótico tipo PUMA a partir de herramientas de visión artificial. Inicialmente, se establecen las bases conceptuales para el cálculo cinemático en un manipulador robótico y se justifica la operación del mismo en un plano de movimiento. Posteriormente, se define el problema cinemático inverso en términos de los ángulos para sus articulaciones de brazo y antebrazo. A partir de ello, se propone un sistema de visión artificial basado en el algoritmo de cálculo del centroide para detectar las coordenadas de una esfera en el plano de movimiento, complementado por la apropiada configuración del escenario de captura de imagen y rutinas de preproceso basadas en filtrado digital. De esta manera, el sistema de visión artificial proporciona en modo automático las coordenadas del objeto al algoritmo de cálculo cinemático para transformarlas en ángulos, cuyo posicionamiento en el manipulador robótico experimental se realiza a través de rutinas programadas en lenguaje escalera en un PLC SIEMENS, constituyendo un esquema de Servo-Visión. Para verificar la efectividad de este último, se realizaron simulaciones numéricas en TIA Portal para las rutinas de movimiento del manipulador, complementadas por la ejecución de los algoritmos de visión y cálculo cinemático en MATLAB. Como resultado, se obtuvo un apropiado desempeño de los algoritmos de localización del objeto en el plano de movimiento con errores inferiores al 4 %, al igual que una precisión de posicionamiento angular de ± 0.1 [°].

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones.
Director: Ricardo Alzate Castaño, Doctorado .D. en Ingeniería Informática y Automática

Abstract

Title: Trajectory planning of puma arm in a PLC*

Autor: Ivon Yiraudi Beltran Niño; José David Alba Morán **

Keywords: Computer vision, puma arm, robot kinematic, visual-servoing.

Description:

In this work the automatic planning for trajectories of a PUMA robotic arm employing computer vision tools is addressed. First of all, basic concepts of robot kinematics are defined to understand the general operation of the robotic arm and to justify the limitation of its motion into a plane. Then, the inverse kinematic problem is proposed and solved in terms of the angles for its lower and upper arms. In addition, a computer vision system is proposed to locate the coordinates of an object in the motion plane by using a centroid-based algorithm, complemented by camera calibration routines and digital filtering of images for preprocessing. Accordingly, the computer vision system provides the coordinates of the target point required to solve the inverse kinematics in terms of the angles for joints of the robotic arm, constituting a visual-servoing scheme. By doing so, LADDER routines were programmed in a SIEMENS PLC in the TIA Portal suite to simulate the motion control of DC motors of an experimental PUMA arm, further linked to MATLAB algorithms performing computer vision calculations. As a result, an acceptable performance was achieved on the automatic detection of coordinates for the target object in the motion plane, showing localization errors lower than 4 %, as well as an angular precision of around ± 0.1 [°].

* Project of grade

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones.
Director: Ricardo Alzate Castaño, Doctorado .D. en Ingeniería Informática y Automática

Introducción

La robótica es una rama interdisciplinaria que se desprende de las ingenierías mecánica, electrónica, eléctrica, de control y de las ciencias de la computación. Estudia el análisis, diseño, manufactura y aplicación de máquinas automáticas con cierto grado de inteligencia, capaces de realizar tareas que pueden reemplazar las actividades de un ser humano (Siciliano, Sciavicco, Villani, & Oriolo, 2010). De otro lado, la palabra robot deriva del *eslavo* “*robota*” que a su vez refiere a la “*servidumbre*”.

El inicio de la robótica moderna se puede ubicar en el tiempo hacia finales del siglo XVIII y principios del siglo XIX, con la creación de una maquina textil programable mediante el uso de tarjetas perforadas. En 1920 el dramaturgo checo *Karel Capek* emplea por primera vez la palabra robot en una obra de teatro llamada “*Robots Universales Rossum*”. El desarrollo de los microprocesadores en el Siglo XX fue uno de los factores más influyentes para masificar las aplicaciones de robótica en el ámbito industrial (Siciliano, Sciavicco, Villani, & Oriolo, 2010).

En la robótica existen tres leyes o principios fundamentales, que fueron enunciados por *Isaac Asimov* en el relato titulado “*Círculo vicioso*”, publicado en 1942: 1) Un robot no hará daño a un ser humano o, por inacción, permitirá que un ser humano sufra daño; 2) Un robot debe hacer o realizar las órdenes dadas por los seres humanos, excepto si estas órdenes entran en conflicto con la primera ley; 3) Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con las dos leyes anteriores. Otras versiones para dichos principios han sido propuestas. Por ejemplo, la Unión Europea planteo 6 leyes que debe satisfacer todo sistema robótico (Reyes, 2011).

En modo general, los robots pueden clasificarse como: *de servicio*, cuando se emplean para realizar asistencia en tareas no industriales; *androides* y *zoomórficos*, cuando adquieren formas de seres vivos ya sean humanos o animales; *móviles*, cuando se desplazan por algún medio en el espacio de movimiento; e *industriales*, cuando se emplean para desarrollar actividades en el contexto del sector productivo (Reyes, 2011). En particular, la *Federación Internacional de Robótica* (<https://ifr.org/>) define a los robots industriales en el modo siguiente: “*máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento*”. A partir de lo anterior, un robot se concibe mediante partes rígidas (eslabones) unidas por articulaciones, para producir movimientos de desplazamiento, giro o la combinación de ambos. Cada uno de los movimientos independientes que puede realizar una articulación se denomina *grado de libertad*. El número de grados de libertad de un robot se encuentra definido por el número de articulaciones o ejes que posee (Siciliano, Sciavicco, Villani, & Oriolo, 2010).

Dependiendo de la configuración de sus ejes, un robot puede clasificarse como: *polar o esférico*, cuando dos ejes de rotación son perpendiculares entre sí y el tercero es prismático; *cilíndrico*, cuando posee una base rotacional y otros dos ejes perpendiculares entre sí; *cartesiano*, cuando sus tres articulaciones son prismáticas con ejes ortogonales entre sí; *SCARA* (Selective Compliant Articulated Robot Arm) cuando posee dos ejes rotacionales controlando la posición en un plano de movimiento y el tercero prismático para controlar profundidad; y *angular o antropomórfico*, cuando sus tres principales articulaciones son rotacionales, emulando el movimiento de extremidades humanas (Reyes, 2011). De este último, el *PUMA* (Programmable

Universal Manipulación Arm) ha sido uno de amplio uso en el ámbito industrial desde su aparición en 1978 a partir de los desarrollos realizados por *Víctor Scheinman* para General Motors. La Fig. 1 ilustra un manipulador *PUMA* de seis grados de libertad, lo cual le permite posicionarse de manera simple y flexible en cualquier punto del espacio de movimiento. El posicionamiento de un robot se logra a través del estudio de su cinemática directa e inversa (Siciliano, Sciavicco, Villani, & Oriolo, 2010).

Figura 1.

Manipulador robótico PUMA



Nota Tomado de Wikipedia.org

De otro lado, es posible utilizar técnicas de visión artificial para agregar ojos a un robot a partir del concepto de *Visual-Servoing* o *Servo-Visión*, en el cual la información del entorno es capturada y procesada en tiempo real para facilitar el cálculo de trayectorias del robot y tomar decisiones, por ejemplo, en aplicaciones de seguridad y de control de calidad de productos (Gutierrez & Nogales, 2011).

En la Universidad Industrial de Santander, en el año de 1995 se adquirió por parte de la Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones (E3T) un equipo de control

numérico computarizado (CNC) a la empresa TecQuipment Ltd. Este equipo contaba con un robot *PUMA* de referencia MA2000 para realizar la movilización de piezas a ser maquinadas. Posterior a la obsolescencia del equipo, varios trabajos de grado se han realizado al interior de la E3T para recuperar la funcionalidad del manipulador robótico con fines didácticos. Inicialmente, *Peña et al.* realizaron en (Peña, Pérez, & Suarez, 2005) el diseño y construcción de un prototipo para robot *PUMA* de 6 grados de libertad, con adecuaciones y mejoras en su parte eléctrica realizados por *Gutiérrez* en (Gutiérrez, 2010). Posteriormente, *Beltrán y Medina* desarrollaron en (Beltrán, 2011) el primer esfuerzo por acondicionar el mecanismo de la maquina CNC antes mencionada a partir de la implementación de una nueva tarjeta de control, siendo complementado por los trabajos de *Ortiz* en (Ortiz, 2012) y *Poveda* en (Poveda, 2014), orientados al modelado, identificación de parámetros y síntesis de controladores para el movimiento del mecanismo, como apoyo a la tesis de maestría de *Carreño* en (Carreño, 2015). De manera más reciente, *Monsalve* en (Monsalve, 2019) desarrollo adecuaciones adicionales al prototipo *PUMA* MA2000 para efectuar su accionamiento desde un PLC. Sin embargo, la operabilidad del sistema se limita a movimientos manuales imprecisos en lazo abierto; es decir, sin la posibilidad de verificar el correcto posicionamiento de las articulaciones del robot y, por ende, limitando la posibilidad de programar trayectorias automáticas. Por su parte, la Escuela de Ingeniería Mecánica documenta un seminario de grado en robótica industrial (Ardila, Dulcey, Mendoza, & Mendoza, 2012) y el desarrollo de un robot cartesiano de tipo neumático (Buitrago & Cediell, 2010). Otros trabajos incluyen el estudio de vehículos aéreos no tripulados como en (Álvarez, 2014) (Herrera & Rincón, 2012) (Jauregui & Vianchada, 2010). Finalmente, se resalta el trabajo realizado por *Amaya y Ruíz* en (Amaya & Ruiz, 2005) utilizando técnicas de visión artificial para aplicaciones en robótica.

En otras Universidades de Colombia e Iberoamérica, se destacan los trabajos realizados en la Universidad Nacional de Colombia sede Manizales con el manipulador robótico SCORBOT (Arango, 2014); en la Universidad Libre de Bogotá sobre la simulación de manipuladores robóticos con varios grados de libertad (Leguizamón, 2016); en el Tecnológico de Monterrey en México a través del desarrollo de interfaces para manipuladores *PUMA* (Olivares, 2016); en la Universidad Politécnica de Madrid sobre el desarrollo de robots para asistencia en movilidad (Carrera, 2011); en la Universidad de las Fuerzas Armadas de Ecuador mediante la simulación en tiempo real de manipuladores *PUMA* (Allauca & Estévez, 2019) y en la Escuela Politécnica de Ecuador a partir del diseño y simulación de robots para manipulación de grandes cargas mecánicas (Vaca, 2012).

El presente proyecto de grado retomara el trabajo desarrollado en (Monsalve, 2019), para adicionar la ejecución de trayectorias programadas (o calculadas a partir de análisis cinemático), junto con la incorporación de un sistema de sensado basado en visión artificial.

A partir de lo anterior se plantean las siguientes inquietudes: ¿Como configurar un sistema de sensado para un robot manipulador *PUMA* empleando visión artificial?, ¿Como acoplar una rutina para el cálculo de trayectorias de la cinemática del robot a través de un PLC?, ¿Como automatizar el movimiento del manipulador robótico *PUMA* MA2000?

1.1 Objetivos

1.1.1 Objetivo general

Realizar el cálculo y la programación para las trayectorias de movimiento de un *brazo robótico PUMA* en el laboratorio de control e instrumentación de la E3T-UIS

1.1.2 Objetivos específicos

Desarrollar una rutina computacional para calcular las coordenadas de un objeto en el espacio, a partir de técnicas de *visión artificial*;

Determinar las trayectorias de movimiento para un *brazo robótico PUMA* aplicando análisis cinemático sobre la información suministrada por el algoritmo de *visión artificial*;

Ejecutar las trayectorias en una interfaz que simula el movimiento del *brazo robótico PUMA MA2000* a través de instrucciones programadas en un PLC.

2. Programación visual de trayectorias

La forma clásica de determinar las trayectorias de movimiento de un brazo robótico es empleando el *análisis cinemático* (directo e inverso), a través del cual se realiza un plan de traslación en el espacio desde un punto de inicio hasta el punto final objetivo.

El presente *Capítulo* aborda los conceptos básicos del planeamiento de trayectorias en un brazo robótico y los complementa con técnicas de visión artificial para ejecutar cálculo de movimiento en un manipulador tipo PUMA.

2.1 Cinemática de un robot

En física, la *cinemática* hace referencia al estudio del movimiento. En un *análisis cinemático* las variables del movimiento se calculan sin considerar las fuerzas causantes del mismo, siendo una situación diferente al caso de la *dinámica* que precisamente estudia la relación entre un movimiento y sus fuerzas asociadas (Siciliano, Sciavicco, Villani, & Oriolo, 2010).

A su vez, existen dos tipos principales de problemas cinemáticos: el *directo* y el *inverso*. La *cinemática directa* aplica las ecuaciones del movimiento (traslación en el espacio) para calcular una trayectoria que evoluciona hacia una posición final. Por su parte, la *cinemática inversa* determina todas las posibles trayectorias a partir de esa posición final.

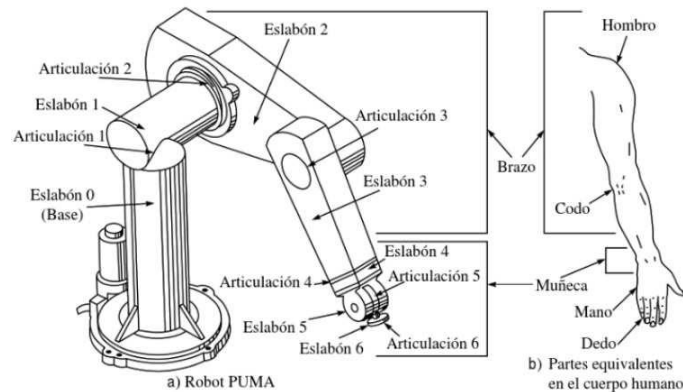
Para el caso de manipuladores robóticos el cálculo de trayectorias a partir de cinemática directa e inversa suele ser complejo, principalmente cuando se consideran varios grados de libertad.

2.1.1 Análisis cinemático del manipulador PUMA

Un manipulador robótico PUMA es similar a un brazo humano, con 6 articulaciones rotatorias e independientes (*grados de libertad*) para posicionarse y orientarse. La Fig. 2 presenta una comparación por contraste.

Figura 2.

Comparación robot PUMA y brazo humano



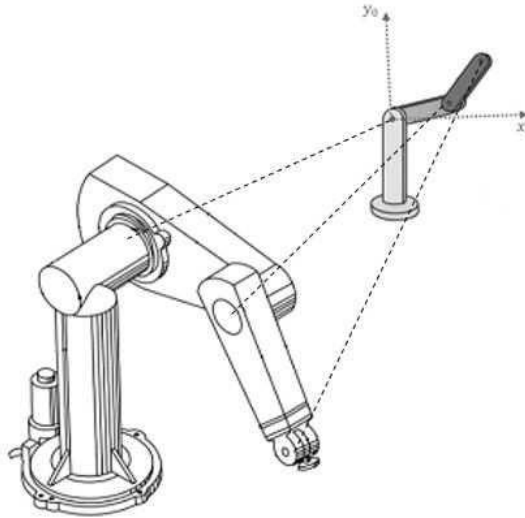
Lo anterior permite realizar un planeamiento de trayectorias eficiente, pues el manipulador PUMA brinda diferentes posibilidades para optimizar el movimiento de su estructura en el espacio. Sin embargo, esta versatilidad se paga con una alta complejidad asociada a la formulación del problema matemático y al correspondiente cálculo de su solución.

Por tanto, en el presente trabajo de grado se realizará una simplificación que permita reducir la complejidad matemática del planteamiento y facilite concentrar el esfuerzo en la integración de técnicas de visión artificial para el cálculo automático de las trayectorias.

Dicha simplificación, consiste básicamente en considerar que el manipulador PUMA se desplaza al interior de un plano de movimiento. A partir de ello, se considera la proyección mostrada en la Fig. 3 donde se asumen inmóviles los eslabones correspondientes a la base, al hombro y a la muñeca, dejando el movimiento únicamente para las articulaciones del brazo, el antebrazo y la apertura y cierre de la pinza.

Figura 3.

Proyección planar del movimiento del manipulador PUMA



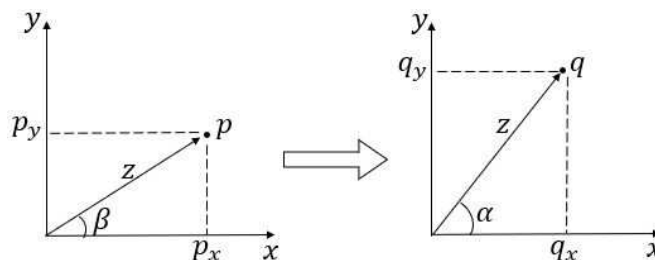
De esta manera, es posible considerar un modelo equivalente en dos dimensiones para el manipulador PUMA, cuyos cálculos cinemáticos en modos directo e inverso se ilustran a continuación empleando un método basado en representaciones geométricas.

Calculo cinemático en \mathbb{R}^2

Para abordar el cálculo cinemático considérese inicialmente la situación planteada en la Fig. 4, en la cual se muestra el cambio de posición (i.e. movimiento) de un vector en el plano.

Figura 4.

Ilustración para movimiento de un vector en el plano



Las relaciones geométricas del problema permiten obtener:

$$\cos \alpha = \frac{q_x}{|z|} \rightarrow q_x = |z| \cos \alpha$$

$$\sin \alpha = \frac{q_y}{|z|} \rightarrow q_y = |z| \sin \alpha$$

$$\cos \beta = \frac{p_x}{|z|} \rightarrow p_x = |z| \cos \beta$$

$$\sin \beta = \frac{p_y}{|z|} \rightarrow p_y = |z| \sin \beta$$

Ahora bien, considerando que $\theta = (\alpha - \beta)$ es la diferencia angular entre las dos posiciones del vector (i.e. el movimiento angular neto) es posible reescribir las coordenadas para el punto q en los términos siguientes:

$$\begin{aligned} q_x &= |z| \cos (\theta + \beta) \\ &= |z| (\cos \theta \cos \beta - \sin \theta \sin \beta) \\ &= (|z| \cos \beta) \cos \theta - (|z| \sin \beta) \sin \theta \\ &= p_x \cos \theta - p_y \sin \theta \end{aligned}$$

$$\begin{aligned} q_y &= |z| \sin (\theta + \beta) \\ &= |z| (\sin \theta \cos \beta + \cos \theta \sin \beta) \\ &= (|z| \cos \beta) \sin \theta + (|z| \sin \beta) \cos \theta \\ &= p_x \sin \theta + p_y \cos \theta \end{aligned}$$

y en modo matricial:

$$\begin{bmatrix} q_x \\ q_y \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

$$q = \mathbf{R}\{\theta, p\},$$

siendo $\mathbf{R}\{.\}$ un operador lineal que permite la traslación o movimiento en el plano para el vector \vec{Z} , desde el punto inicial p hasta el punto final q a través de una rotación con ángulo θ .

El resultado anterior permite diferenciar dos situaciones particulares:

Conozco el ángulo θ y calculo el punto de destino q aplicando \mathbf{R} desde el punto de partida p . A este caso lo denominaremos *problema cinemático directo*;

Conozco el punto de destino q y calculo el ángulo de rotación θ que a través de \mathbf{R} me permite alcanzar dicha coordenada desde el punto de partida p . A este caso lo denominaremos *problema cinemático inverso*.

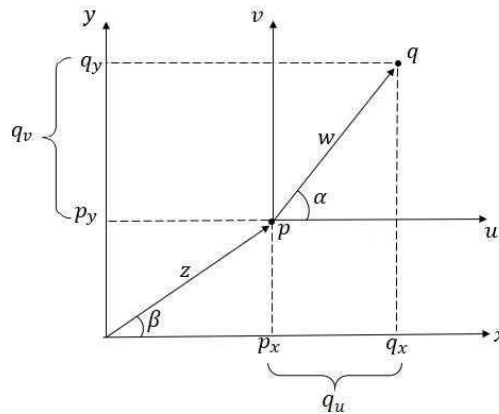
Como se observa, en cualquiera de los casos el elemento relevante es el ángulo de rotación θ . A su vez, es importante resaltar que cualquier punto del plano no necesariamente es alcanzable. Por ejemplo, para el caso de una sola articulación como el ilustrado en la Fig. 4, los únicos puntos que podrán ser alcanzados por el movimiento serán aquellos descritos por una circunferencia de radio $|z|$. Asimismo, el movimiento tendrá solo dos posibles trayectorias: en sentido horario o antihorario.

Por otro lado, agregando una segunda articulación (en adelante denominada \vec{w}) será posible combinar el movimiento de dos vectores en el plano en modo tal que los movimientos de la primera constituyan un marco de referencia para la segunda (i.e. movimiento relativo). Dicha situación se ilustra en la Fig. 5, a partir de la cual es posible construir las siguientes relaciones geométricas:

$$\begin{aligned} q_x &= p_x + q_u = |z|\cos\beta + |w|\cos\alpha \\ q_y &= p_y + q_u = |z|\sin\beta + |w|\sin\alpha \end{aligned} \quad (2.1)$$

Figura 5.

Movimiento en el plano para dos articulaciones



Definiendo las coordenadas de la posición final del vector \vec{w} en términos de los ángulos β y α . De esta manera, el *problema cinemático inverso* en el caso de dos articulaciones (i.e. brazo y antebrazo), se plantea y resuelve mediante un cálculo combinado ante el cual:

Dada una coordenada deseada q , será posible dibujar circunferencias de radio $|w|$ en el plano que la incluyan como parte de su contorno, con lo cual se delimitará un primer conjunto de puntos o región del plano del movimiento;

El centro de dichas circunferencias corresponde con los valores posibles de p ; es decir, vectores de radio $|z|$ trazados desde el origen. Lo anterior, impone una limitación adicional que reduce la cantidad de puntos posibles en el plano;

Empleando criterios de optimalidad como la reducción de distancias y restricciones o límites prácticos de la naturaleza del movimiento, se puede seleccionar un valor apropiado para p y su correspondiente ángulo β empleando **R**;

Teniendo β, p y q , además de las magnitudes de los vectores \vec{z} y \vec{w} (i.e. las dimensiones del brazo y antebrazo), será posible determinar el valor del ángulo α a partir de (2).

Dicha idea fue codificada en MATLAB empleando un algoritmo de fuerza bruta, en el cual se consideraron restricciones para ambas articulaciones en el rango $\beta \in [0, 180^\circ]$ y $\alpha \in [0, 180^\circ]$ además de dimensiones $|z| = 32 [cm]$ y $|w| = 22 [cm]$ tomando en cuenta las características del brazo robótico PUMA del *Laboratorio de Control e Instrumentación* de la E3T-UIS (Monsalve, 2019). A partir de ello, en un ciclo anidado se calcularon todos los posibles puntos p ante valores de β y, para cada uno, todos los posibles puntos q ante valores de α . Finalmente, empleando la función `knnsearch(.)` se determinó la coordenada más cercana al q deseado de entre los valores calculados, y, en consecuencia, los valores de α y β que resuelven (numéricamente) el *problema cinemático inverso*.

2.2 Visión artificial

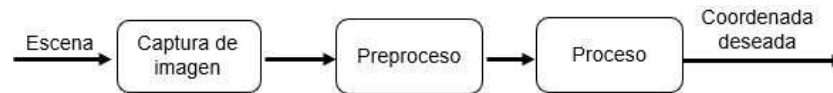
La *visión artificial* o *visión por computadora*, se define como todos aquellos métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real y extraer de ellas información numérica o simbólica para ser procesada digitalmente en un microprocesador (Cuevas, Zaldivar, & Pérez, 2010). Dichas herramientas de visión artificial se emplean actualmente para resolver diferentes problemas en el ámbito industrial, debido principalmente al avance tecnológico en la capacidad de procesamiento y al abaratamiento de costos para acceder a las herramientas de cómputo y a los sistemas de información (i.e. conectividad), incluyendo aplicaciones en: seguridad, inspección de productos, control de calidad y mantenimiento preventivo, entre otras.

A nivel general, un sistema para el procesamiento digital de imágenes se constituye de las etapas ilustradas en la Fig. 6. Este esquema será empleado en el presente proyecto para determinar

(en tiempo de ejecución) la coordenada de destino del manipulador robótico PUMA según se explica a continuación.

Figura 6.

Etapas de un sistema de visión artificial

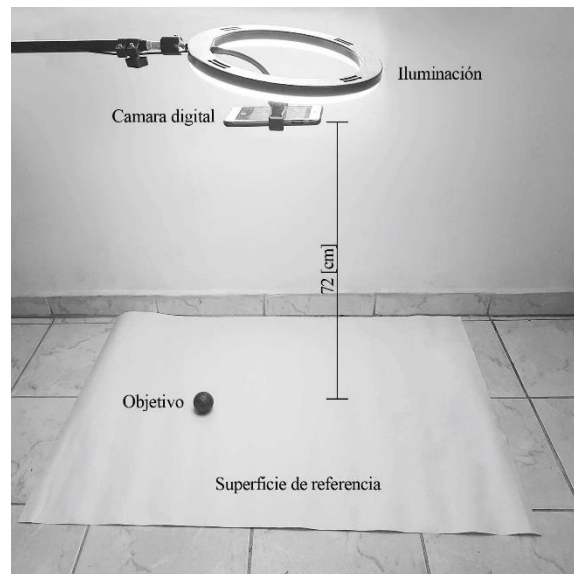


2.2.1 Captura de imágenes

Un primer elemento esencial en el esquema de visión artificial corresponde con el protocolo de *captura de imágenes*. Para ello se creó el escenario ilustrado en la Fig. 7, siendo configurado como sigue:

Figura 7.

Ilustración para escenario de captura



Se empleó como elemento de captura una cámara digital de marca Apple, con resolución de 12 [megapíxeles] correspondientes a imágenes en formato RGB de dimensiones 3024 x 4032;

La cámara se ubicó en posición *cenital* (es decir, vertical superior al objeto) a una distancia de 72 [cm] de una superficie horizontal y plana, correspondiente con un rectángulo blanco de dimensiones 60 x 80 [cm^2], en un recinto cerrado con abundante iluminación artificial;

Sobre la superficie se ubicó una esfera de icopor, de color rojo y 2.5 [cm] de radio a manera de coordenada de destino del manipulador en el plano de movimiento. Lo anterior representa la aplicación típica de capturar un objeto con ciertas características morfológicas al interior de una región de trabajo determinada. Tomando en cuenta que dicha esfera corresponde con el objeto único de la escena, no se requieren tareas de clasificación adicionales como parte del proceso;

Empleando la información anterior se realizó la calibración de los algoritmos (calibración de cámara) al escenario de captura, encontrando la razón entre [píxeles] y [cm] para las medidas

reales de la escena. Como resultado se obtuvo un error de aproximación de alrededor 0.07 [cm], aceptable para las dimensiones del objeto que se desea localizar.

2.2.2 Preproceso

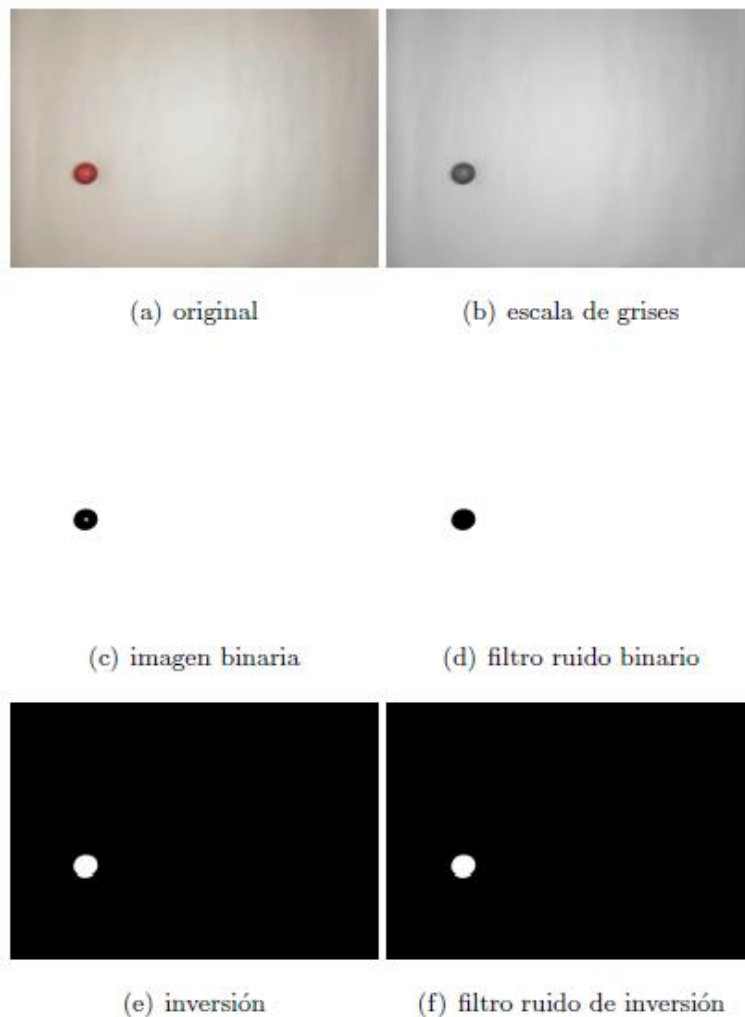
La etapa de *preproceso* o de *acondicionamiento de señal*, es la encargada de preparar la información obtenida en la captura de imágenes para facilitar la tarea posterior de los algoritmos de proceso. Esencialmente, en un *preproceso* se realizan procedimientos de filtrado y amplificación.

Para el caso particular considerado, el primer paso del *preproceso* consistió en obtener una versión en escala de grises para la imagen original en RGB. Posteriormente, la imagen en escala de grises fue binarizada para crear una diferenciación entre el fondo y el objeto. A su vez, a la señal binarizada es filtrada para eliminar componentes de ruido del tipo sal y pimienta (Cuevas, Zaldivar, & Pérez, 2010). La señal binaria filtrada es invertida, para lograr que el objeto se visualice en color blanco y el fondo en color negro. Finalmente, la señal binaria invertida es filtrada nuevamente para eliminar el ruido sal y pimienta.

La cadena de procedimientos descrita para el *preproceso* de la imagen, se desarrolló empleando funciones del Image Processing Toolbox de *MATLAB*. En particular, se hizo uso de las rutinas: `rgb2gray(.)` para conversión a escala de grises, `im2bw(.)` para convertir a imagen binaria y `bwareaopen(.)` para filtrar la imagen binaria. Por su parte se desarrolló una rutina simple a través de ciclos `for` para realizar la inversión de la imagen binaria. Como resultado, la Fig. 8 ilustra los procedimientos de la etapa de *preproceso*.

Figura 8.

Resultados etapa de preproceso



2.2.3 Proceso: cálculo del centroide

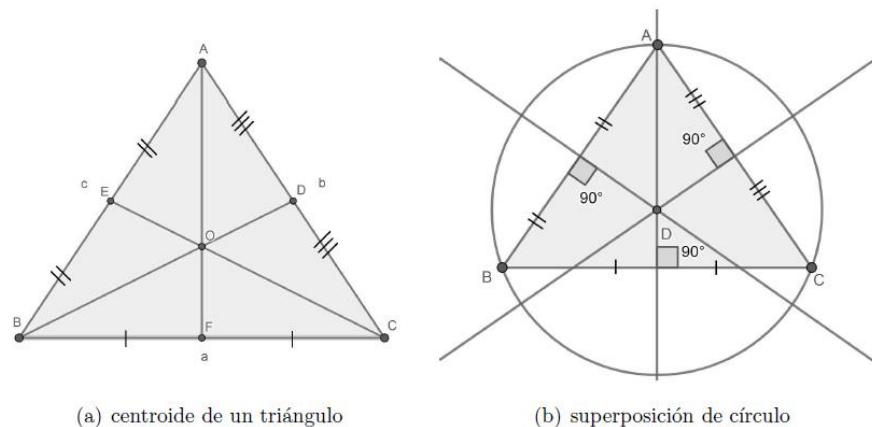
El *preproceso* apenas presentado entrega una imagen que resalta el objeto (en *color blanco*) del fondo de la escena (en *color negro*); es decir, facilita la localización de la esfera en el plano de movimiento. En complemento a lo anterior, el *proceso* consiste en determinar la posición exacta de dicho objeto y para ello se aprovechará un principio geométrico importante: el *centroide*.

Por definición, el *centroide* o *baricentro* de un cuerpo es su centro geométrico, que bajo ciertas condiciones de simetría puede coincidir también con su centro de gravedad (Cuevas, Zaldivar, & Pérez, 2010).

Tomando en cuenta la naturaleza geométrica del problema analizado, se asumirá al *centroide* como el punto donde se produce la intersección de las medianas (rectas trazadas desde un vértice hasta el punto medio del lado opuesto) de un triángulo (Cuevas, Zaldivar, & Pérez, 2010). Dicha representación geométrica se ilustra en la Fig. 9(a) para el caso de un triángulo *equilátero* y a partir de ello, cualquier figura o representación geométrica que contenga (de manera simétrica) un triángulo de tales características permitirá definir su *centroide* en modo equivalente.

Figura 9.

Determinación geométrica del centroide



Más aún, considerando el hecho que la proyección planar de una esfera es un círculo, es claro que dicha figura cumple con las características anteriores y, por tanto, permite calcular de manera simple el punto que representara la coordenada objetivo del manipulador robótico. En estos

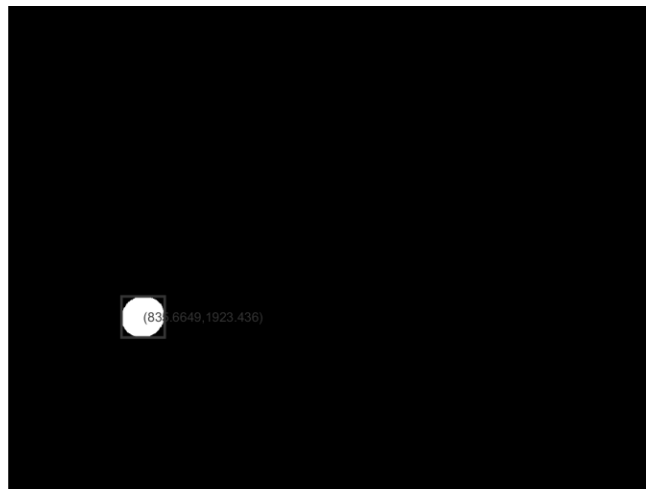
términos, la determinación geométrica del *centroide* para un círculo se representa gráficamente en la Fig. 9(b).

Numéricamente es posible aproximar el cálculo del *centroide* para un objeto empleando la función `regionprops(.)` del Image Processing Toolbox de *MATLAB*, la cual realiza un procedimiento similar al anterior, pero ajustado al caso de una aproximación geométrica a través de un rectángulo.

Detalles particulares al respecto de la configuración y asignación de parámetros para dicha función pueden ser consultados en el manual de usuario provisto por *The Mathworks Inc.* Por su parte, la Fig. 10 ilustra el resultado del cálculo numérico para el *centroide* entregado en forma de coordenada cartesiana.

Figura 10.

Ilustración cálculo numérico del centroide



2.3 Manipulador robótico experimental

Tal y como fue introducido en el *Capítulo 1*, la E3T-UIS dispone de un manipulador robótico *PUMA MA2000* con 6 grados de libertad.

Diferentes trabajos de grado han intentado reacondicionar la operabilidad del prototipo tomando en cuenta el nivel de obsolescencia de su etapa de control. De ellos, el más reciente fue realizado por *J. A. Monsalve* en (Monsalve, 2019) facilitando la ejecución del movimiento del manipulador en modo automático desde un PLC.

En su forma actual, el prototipo experimental cuenta con 4 servomotores y 4 motores de corriente continua para realizar la operación de sus 6 grados de libertad siguiendo la distribución de actuadores mostrada en la Fig. 11. Como *servomotores* se emplean dispositivos de referencia MG 995 posicionados a partir de una señal PWM (i.e. modulada en ancho de pulso). Por su parte, los *motores DC* utilizados son MAXON 2332.908-10.151-051 de 18 [VDC], 11 [W] y 8440 [RPM]. La Fig. 12 ilustra la apariencia física de los dispositivos.

Evidentemente, los motores DC son más robustos que los servomotores para soportar el peso de las piezas con mayor tamaño en la estructura del brazo, siendo complementados por elementos de piñonería (para ajustar su movimiento a restricciones mecánicas del sistema) y dispositivos electrónicos de actuación. En particular, se hace uso de *puentes H* de referencia L298N activados desde el PLC mediante señales PWM. Adicionalmente, tanto los motores como los servomotores tienen acoplado a su eje un potenciómetro lineal para realizar lectura permanente de su posición a partir de una diferencia de potencial analógico. El movimiento de rotación de los motores está limitado al rango $[0^\circ, 180^\circ]$ y el de los servomotores al rango $[0^\circ, 45^\circ]$. Por su parte, el *Motor 1* realiza una traslación mecánica en el rango $[0 \text{ cm}, 100 \text{ cm}]$.

A partir de lo anterior, se constituyó (en (Monsalve, 2019)) el montaje circuital de la Fig. 13 permitiendo identificar el acoplamiento óptico entre las salidas del PLC (digitales PWM) y las conexiones de actuación hacia motores y servomotores. Asimismo, se observa la realimentación de cada potenciómetro para la posición de su respectiva articulación. En el esquema se adoptan las convenciones: M_i para el i -ésimo motor y S_i para el i -ésimo servomotor, en concordancia con la distribución de actuadores de la Fig. 11.

Figura 11.

Prototipo experimental E3T-UIS para manipulador PUMA MA2000

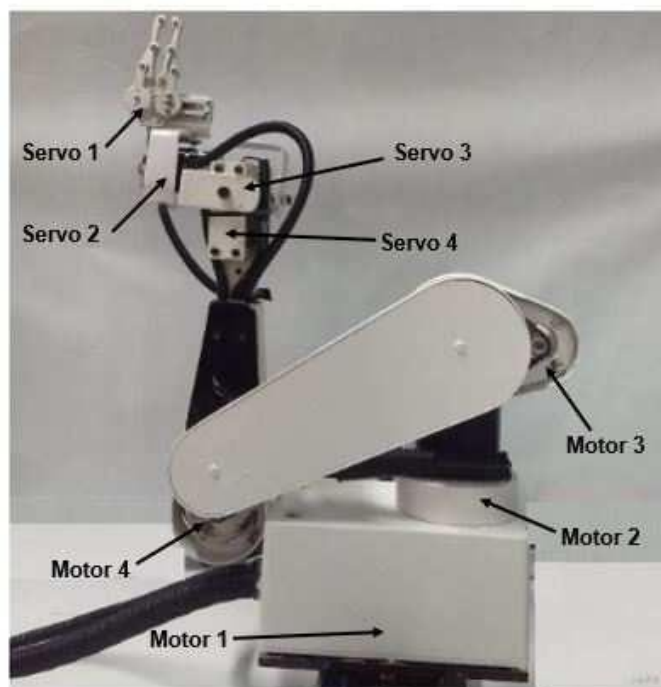


Figura 12.

Motores y servomotores empleados en manipulador experimental



(a) servomotor



(b) motor DC

De manera complementaria al montaje circuital, la operación automática del movimiento del manipulador se ejecuta con base en los estados y transiciones definidos en la Tabla 1 e ilustrados gráficamente a través del esquema de la Fig. 14. La operación de la máquina de estado finito puede resumirse en el modo siguiente:

Inicio: el estado inicial es la condición de HOME, para la cual se tiene: $X = 0$ [cm], $P1 = 0$ [cm],

$$P2 = 0^\circ, P3 = 0^\circ, P4 = 0^\circ, PS1 = 0^\circ, PS2 = 0^\circ, PS3 = 0^\circ \text{ y } PS4 = 0^\circ;$$

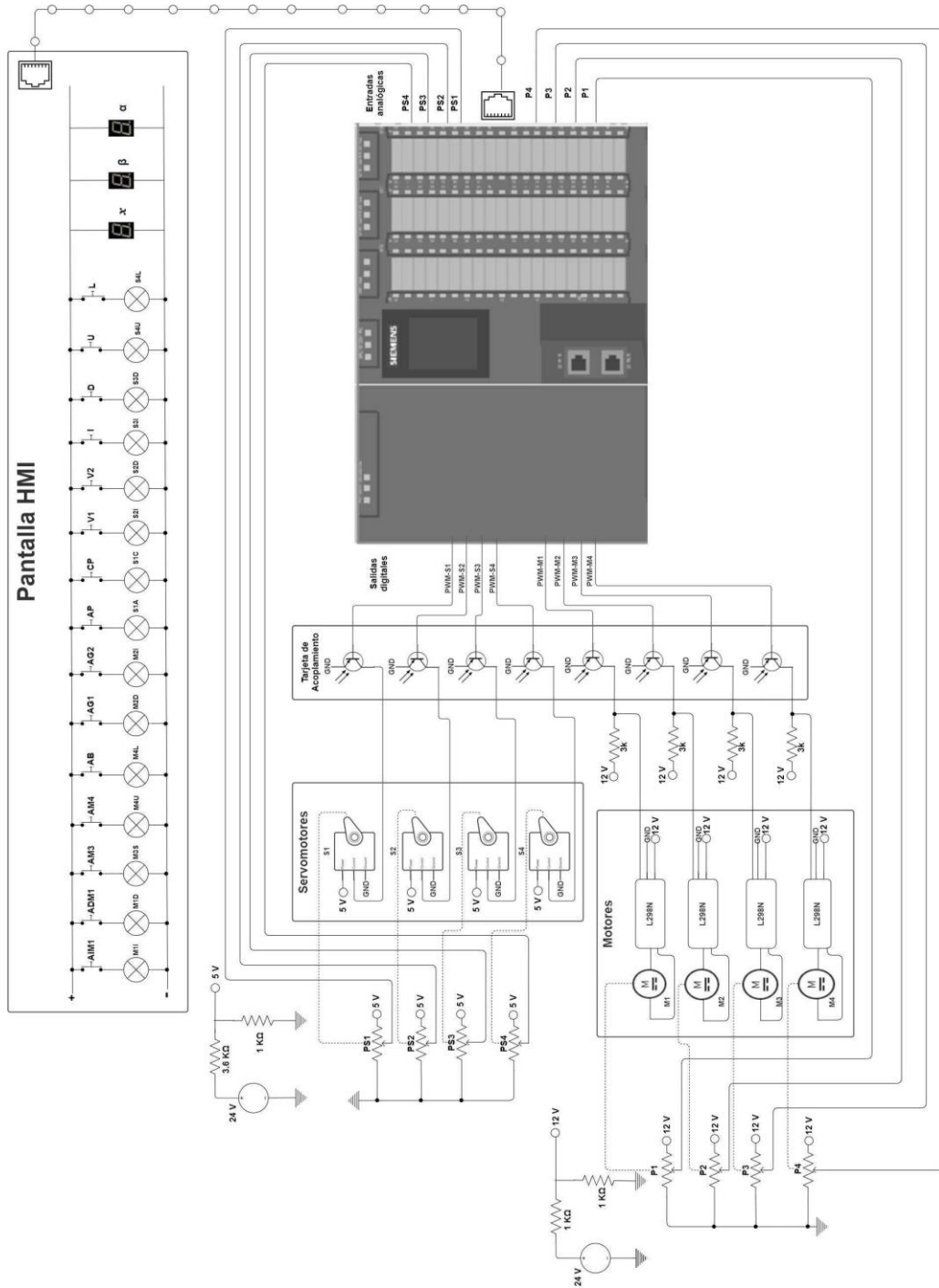
Motor 1: introduciendo un valor deseado X para la posición en [cm] de la base del brazo, se realizará un movimiento de M1 hacia la izquierda (estado M1I) tras pulsar AIM1 y hasta que la lectura del potenciómetro P1 iguale el valor de X , instante en el cual M1 se detendrá (estado M1O). A partir de esta condición, será posible realizar un nuevo movimiento hacia la izquierda ingresando un nuevo valor de $X > P1$. De otro lado, será posible realizar un movimiento de M1 hacia la derecha (estado M1D) tras pulsar AI M1 y siempre que la lectura del potenciómetro P1 sea un valor superior a X . Este movimiento a la derecha se mantendrá hasta que la lectura del potenciómetro P1 iguale el valor de X , instante en el cual M1 se detendrá (estado M1O);

Motor 2 : se realizara un movimiento de M2 hacia la izquierda (estado M1I) tras pulsar AG1 y mientras la lectura del potenciómetro P2 sea inferior al voltaje equivalente a 180° , instante en el cual M2 se detendrá (estado M2O). A partir de esta condición, será posible realizar un movimiento de M2 hacia la derecha (estado M2D) tras pulsar AG2 y siempre que la lectura del potenciómetro P2 sea el voltaje equivalente a 180° . Este movimiento a la derecha se mantendrá hasta que P2 se reduzca al voltaje equivalente a 0° , instante en el cual M2 se detendrá (estado M2O). En este punto será posible reiniciar el movimiento hacia la izquierda pulsando AG1 cuando P2 corresponda con el voltaje equivalente a 0° ;

Motor 3 : introduciendo un valor deseado β para la posición angular del *brazo*, se realizará un movimiento de subida por parte de M3 (estado M3S) tras pulsar AM3 y hasta que la lectura del potenciómetro P3 iguale el valor de β , instante en el cual M3 se detendrá (estado M3O). A partir de esta condición, será posible realizar un nuevo movimiento de subida ingresando un nuevo valor de $\beta > P3$;

Figura 13.

Esquema de conexiones para el manipulador robótico experimental



Motor 4: introduciendo un valor deseado α para la posición angular del *antebrazo*, se realizará un movimiento de subida por parte de M4 (estado M4U) tras pulsar AM4 y hasta que la lectura del potenciómetro P4 iguale el valor de α , instante en el cual M4 se detendrá (estado M4O). A partir de esta condición, será posible realizar un nuevo movimiento de subida ingresando un nuevo valor de $\alpha > P4$. De otro lado, será posible realizar un movimiento de bajada para M4 (estado M4L) tras pulsar AB y siempre que la lectura del potenciómetro P4 sea un valor superior a α . Este movimiento de bajada se mantendrá hasta que la lectura del potenciómetro P4 iguale el valor de α , instante en el cual M4 se detendrá (estado M4O);

Servomotor 1: se realizará un movimiento de apertura por parte de S1 (estado S1A) tras pulsar AP y mientras la lectura del potenciómetro PS1 sea inferior al voltaje equivalente a 45° , instante en el cual S1 se detendrá (estado S1O). A partir de esta condición, será posible realizar un movimiento de cierre por parte de S1 (estado S1C) tras pulsar CP y siempre que la lectura del potenciómetro PS1 sea el voltaje equivalente a 45° . Este movimiento de cierre se mantendrá hasta que PS1 se reduzca al voltaje equivalente a 0° , instante en el cual S1 se detendrá (estado S1O). En este punto será posible reiniciar el movimiento de apertura pulsando AP cuando PS1 corresponda con el voltaje equivalente a 0° ;

Servomotor 2 : se realizara un movimiento hacia la izquierda por parte de S2 (estado S2I) tras pulsar V1 y mientras la lectura del potenciómetro PS2 sea inferior al voltaje equivalente a 45° , instante en el cual S2 se detendrá (estado S2O). A partir de esta condición, será posible realizar un movimiento hacia la derecha por parte de S2 (estado S2D) tras pulsar V2 y siempre que la lectura del potenciómetro PS2 sea el voltaje equivalente a 45° . Este movimiento hacia la derecha se mantendrá hasta que PS2 se reduzca al voltaje equivalente a 0° , instante en el cual S2 se detendrá

(estado S2O). En este punto será posible reiniciar el movimiento hacia la izquierda pulsando V1 cuando PS2 corresponda con el voltaje equivalente a 0° ;

Servomotor 3: se realizará un movimiento hacia la derecha por parte de S3 (estado S3D) tras pulsar D y mientras la lectura del potenciómetro PS3 sea inferior al voltaje equivalente a 45° , instante en el cual S3 se detendrá (estado S3O). A partir de esta condición, será posible realizar un movimiento hacia la izquierda por parte de S3 (estado S3I) tras pulsar I y siempre que la lectura del potenciómetro PS3 sea el voltaje equivalente a 45° . Este movimiento hacia la derecha se mantendrá hasta que PS3 se reduzca al voltaje equivalente a 0° , instante en el cual S3 se detendrá (estado S3O). En este punto será posible reiniciar el movimiento hacia la derecha pulsando D cuando PS3 corresponda con el voltaje equivalente a 0° ;

Servomotor 4 : se realizara un movimiento de subida por parte de S4 (estado S4U) tras pulsar U y mientras la lectura del potenciómetro PS4 sea inferior al voltaje equivalente a 45° , instante en el cual S4 se detendrá (estado S4O). A partir de esta condición, será posible realizar un movimiento de bajada por parte de S4 (estado S4L) tras pulsar L y siempre que la lectura del potenciómetro PS4 sea el voltaje equivalente a 45° . Este movimiento de bajada se mantendrá hasta que PS4 se reduzca al voltaje equivalente a 0° , instante en el cual S4 se detendrá (estado S4O). En este punto será posible reiniciar el movimiento de subida pulsando U cuando PS4 corresponda con el voltaje equivalente a 0° .

La implementación del automatismo se realizó a través de programación en lenguaje escalera (LADDER) empleando la herramienta STEP7 del entorno TIA Portal de *SIEMENS*, para rutinas ejecutadas en un PLC S7-1500 y mando remoto mediante un panel HMI KTP700. Asimismo, la configuración de las señales de actuación PWM se realizó a partir de caracterizaciones experimentales relacionando la duración de los pulsos de activación con la

cantidad de grados de movimiento en cada motor y servomotor. Para detalles específicos sobre la programación realizada y los procedimientos de caracterización, se recomienda al lector interesado consultar (Monsalve, 2019).

2.4 Sistema de *Servo-Visión* propuesto

Tal y como fue mencionado en el *Capítulo 1*, un sistema de *Servo-Visión* (o *Visual Servoing*) es un esquema de actuación para manipuladores robóticos en el cual el cálculo de trayectorias se realiza a partir de la información proporcionada por un sistema de visión artificial.

Tabla 1.

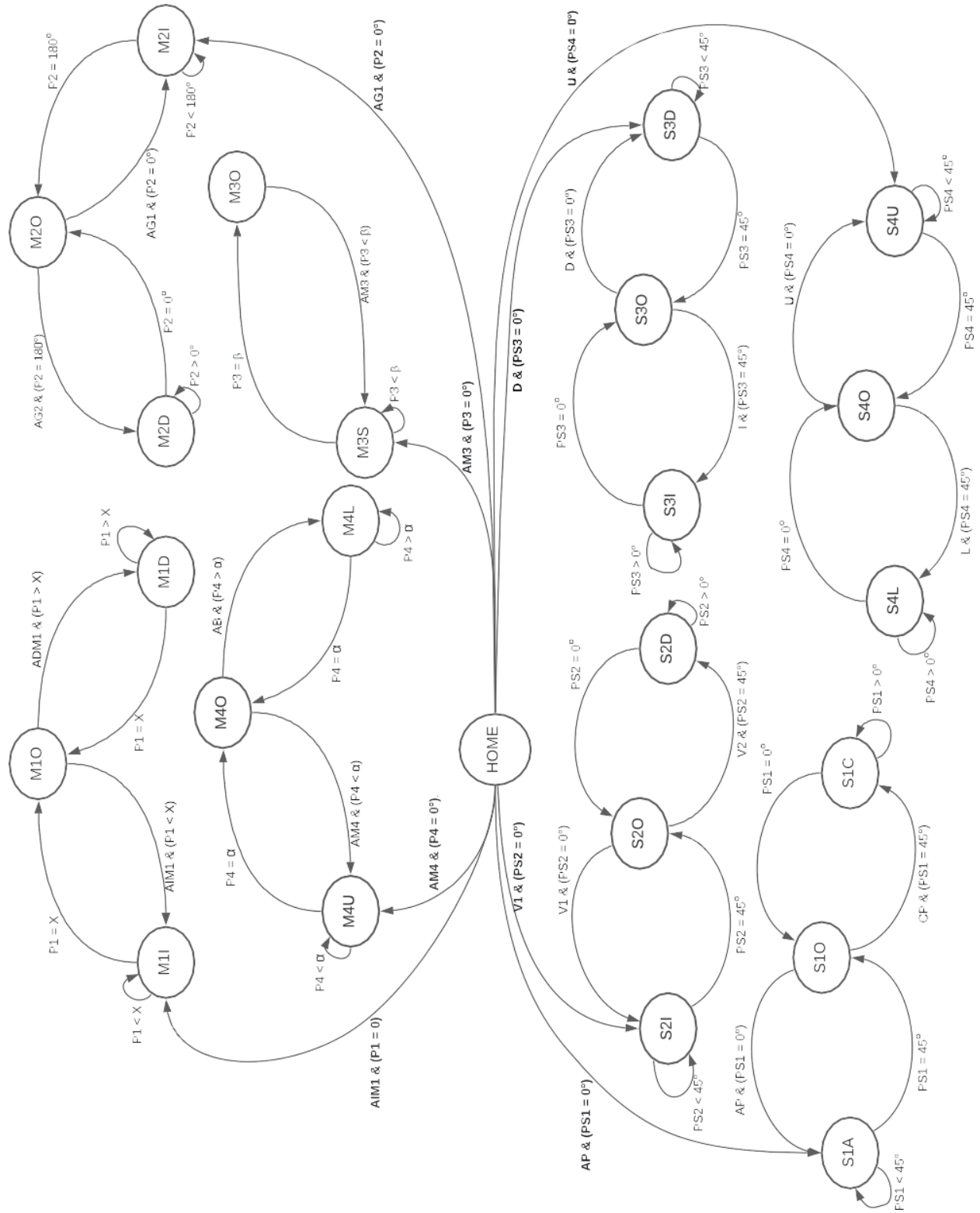
Convenciones empleadas en máquina de estados

Convención	Descripción
HOME	Estado de posiciones iniciales
X	Valor de posición en [cm] para la base del brazo
β	Valor en grados para ángulo de brazo
α	Valor en grados para ángulo de antebrazo
Pi	Valor medido en potenciómetro del <i>i</i> -ésimo motor
PSi	Valor medido en potenciómetro del <i>i</i> -ésimo servomotor
M1I	Movimiento hacia la izquierda en M1, activado por botón AIM1
M1D	Movimiento hacia la derecha en M1, activado por botón ADM1
M2I	Giro hacia la izquierda en M2, activado por botón AG1
M2D	Giro hacia la derecha en M2, activado por botón AG2
M3S	Giro hacia arriba en M3, activado por botón AM3
M4U	Giro hacia arriba en M4, activado por botón AM4

Convención	Descripción
M4L	Giro hacia abajo en M4, activado por botón AB
S1A	Apertura de pinza en S1, activado por botón AP
S1C	Cierre de pinza en S1, activado por botón CP
S2I	Giro hacia la izquierda en S2, activado por botón V1
S2D	Giro hacia la derecha en S2, activado por botón V2
S3I	Giro hacia la izquierda en S3, activado por botón I
S3D	Giro hacia la derecha en S3, activado por botón D
S4U	Giro hacia arriba en S4, activado por botón U
S4L	Giro hacia abajo en S4, activado por botón L
MiO	Estado de paro para i -ésimo motor
SiO	Estado de paro para i -ésimo servomotor

Figura 14.

Máquina de estado finito para operación del manipulador



De esta manera, empleando los desarrollos descritos en las *Secciones* anteriores será posible combinar el cálculo en tiempo real para la coordenada objetivo del escenario de captura con el manipulador experimental, para proporcionar los parámetros β y α del movimiento. En particular, se propone el sistema de *Servo-Visión* ilustrado en la Fig. 15 a partir de los siguientes pasos:

Inicialmente, se asignan en el *prototipo experimental* valores fijos para los motores M1 (i.e. distancia X), M2 ($P2 = 0^\circ$ para posicionar el brazo paralelo al movimiento de M1) y los servomotores S1 ($PS1 = 45^\circ$ para abrir pinza), S2 ($PS2 = 0^\circ$), S3 ($PS3 = 0^\circ$) y S4 ($PS4 = 0^\circ$), de manera tal que el manipulador se desplace al interior de un plano de movimiento similar al propuesto en la Fig. 3, con posicionamiento definido únicamente a partir de los ángulos β y α para las articulaciones de los motores M3 y M4, respectivamente;

Una vez determinada la escena de captura, el sistema de visión artificial descrito en la *Sección 2.2* permitirá determinar (mediante cálculos ejecutados en MATLAB) los valores de β y α para la coordenada deseada;

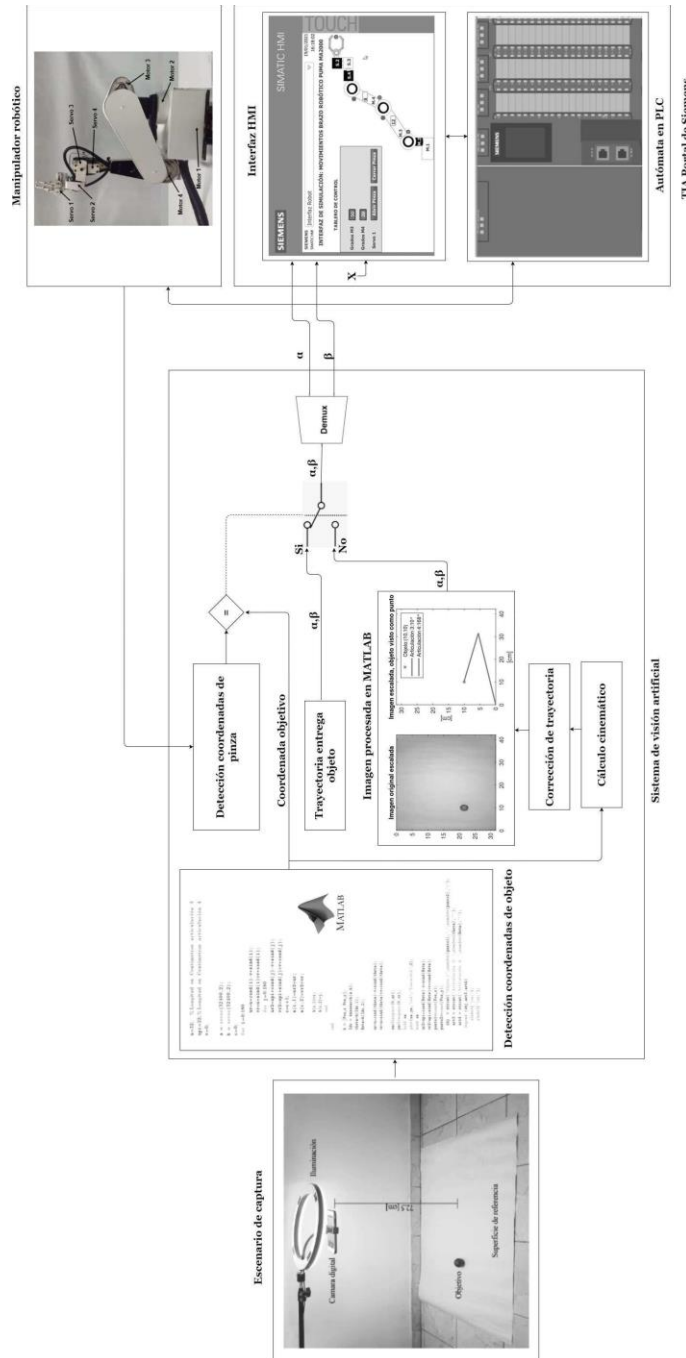
Se realizará un enlace entre la interfaz de cálculo cinemático en MATLAB y el TIA Portal, para transferir los valores calculados de β y α como parámetros de entrada al manipulador experimental, ejecutando así su movimiento;

Una vez ejecutada la trayectoria, el posicionamiento de la pinza (abierta) en la coordenada objetivo podrá ser verificado visualmente a partir de una nueva captura del sistema de visión artificial (ajustado para detectar la posición de la pinza en el espacio de movimiento, más que del objeto). En caso de identificarse algún error de posicionamiento, deberán recalcularse los valores correspondientes de β y α para proceder a ejecutar la corrección de trayectorias. Lo anterior se repetirá de manera cíclica hasta satisfacer algún criterio de parada;

Con la pinza posicionada en la coordenada objetivo, se realizará el cierre de pinza a través de S1 para recuperar el objeto;

Figura 15.

Esquema para sistema de Servo-Visión en manipulador experimental



Desde esa posición el manipulador se desplazará hacia las coordenadas de una posición preprogramada correspondiente con el punto de entrega del objeto, liberándolo tras la apertura de S1;

Finalmente, el manipulador regresara a la posición inicial indicada en el *ítem 1)* quedando listo para recomenzar el proceso.

3. Verificación del movimiento

Inicialmente, el presente trabajo de grado fue formulado pensando en la implementación real en laboratorio para el sistema de *Servo-Visión* sobre el manipulador robótico PUMA MA 2000.

Sin embargo, como consecuencia de la emergencia sanitaria causada por el virus SARSCoV-2 (COVID 19) fue necesario redireccionar los alcances del mismo hacia la configuración de un esquema general para el sistema, el diseño de los algoritmos de proceso de imagen y cálculo de trayectorias, adicional a la validación por simulación de su operación en el autómeta programable.

De esta manera, el presente *Capítulo* aborda el desarrollo y ejecución para un conjunto de rutinas de simulación en la herramienta *PLC Sim* del entorno integrado *TIA Portal V.14* de *SIEMENS*, como una manera de ilustrar la operación del sistema de *Servo-Visión* en el manipulador robótico experimental.

3.1 Simulación del sistema de *Servo-Visión*

Para adecuar el sistema experimental mostrado previamente en la Fig. 13 hacia un entorno de simulación en TIA Portal, fue necesario realizar el siguiente conjunto de adaptaciones al montaje desarrollado por *J. A. Monsalve* en (Monsalve, 2019):

Del esquema original solamente se conservó la rutina de movimiento para el servomotor S1 correspondiente con las maniobras de apertura y cierre de pinza. Su activación se realizó de forma manual a través de los botones AP y CP en el panel HMI;

Por su parte, el movimiento de las articulaciones del brazo y antebrazo del manipulador (i.e. motores M3 y M4) fue emulado mediante un contador cuya salida representa el valor en grados sexagesimales que se desplaza la articulación; es decir, la señal de su respectivo potenciómetro;

La activación del conteo se realizó empleando en TIA Portal un bloque del tipo PULSEGEN, de manera tal que fuese posible manipular la base de tiempo y su sentido entre ascendente y descendente, a partir de un contador bidireccional y una serie de elementos lógicos (de comparación) para determinar el inicio y fin de los conteos dependiendo del valor de coordenada ingresado para las variables α y β en el panel HMI. La Fig. 16 ilustra la programación realizada en TIA Portal para emular el movimiento de M3. De manera similar se realizó la simulación para el movimiento de M4. Ambos diagramas respetan las condiciones generales de operación definidas en la máquina de estados de la Fig. 14;

Los demás elementos del esquema original fueron removidos por no considerarse necesarios para efectos de simulación.

Posteriormente, el sistema modificado para simulación fue adaptado a las características del sistema de *Servo-Visión* presentado en la Fig. 15. Para ello, la coordenada objetivo fue

transferida desde la interfaz de cálculo cinemático en MATLAB hacia el panel HMI mediante la actualización de un archivo de texto tipo CSV (*comma separated values*) leído periódicamente en TIA Portal para ajustar (cada 5 [s]) los valores de parámetro para las coordenadas α y β de operación del manipulador robótico.

A partir de lo anterior, se presenta en la Fig. 17 el esquema de simulación para el sistema de *Servo-Visión* y en la Fig. 18 la interfaz desarrollada para visualizar en el panel HMI la operación simulada del manipulador, mediante la actualización del valor de sus coordenadas de movimiento.

Figura 16.

Programación LADDER en TIA Portal para emular movimiento de M3

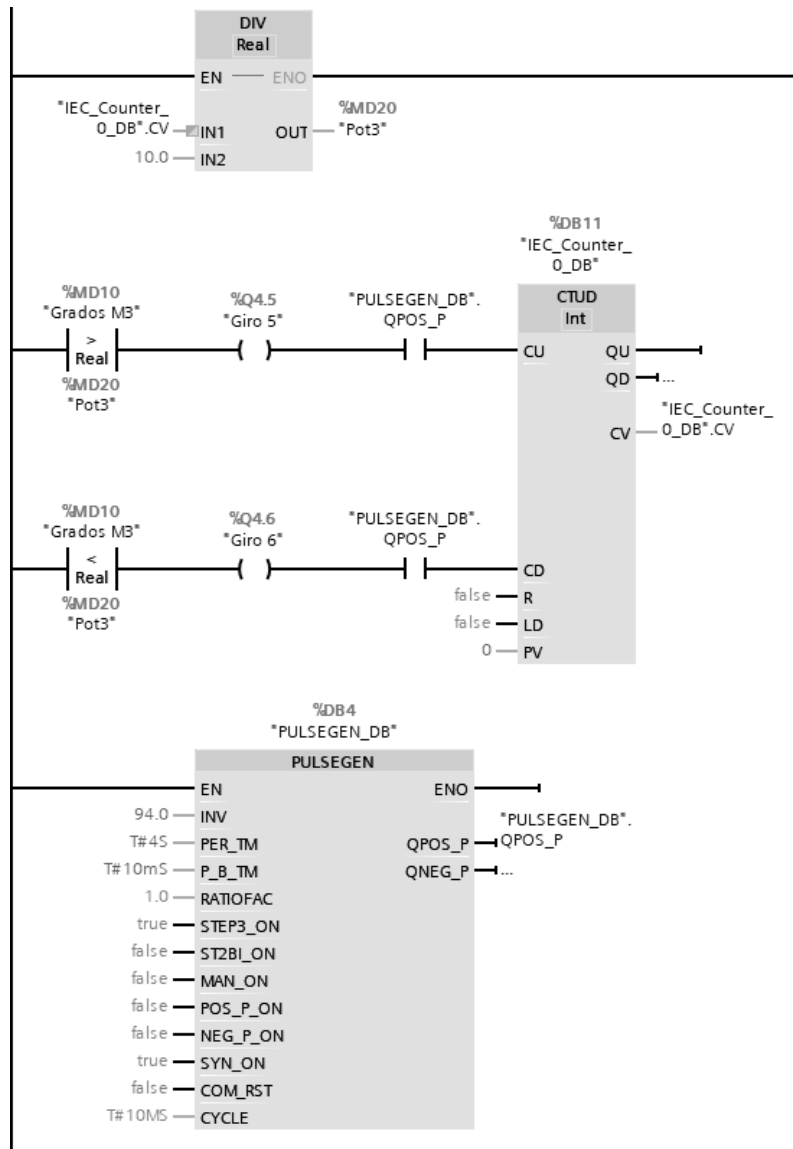


Figura 17.

Esquema de simulación para sistema de Servo-Visión

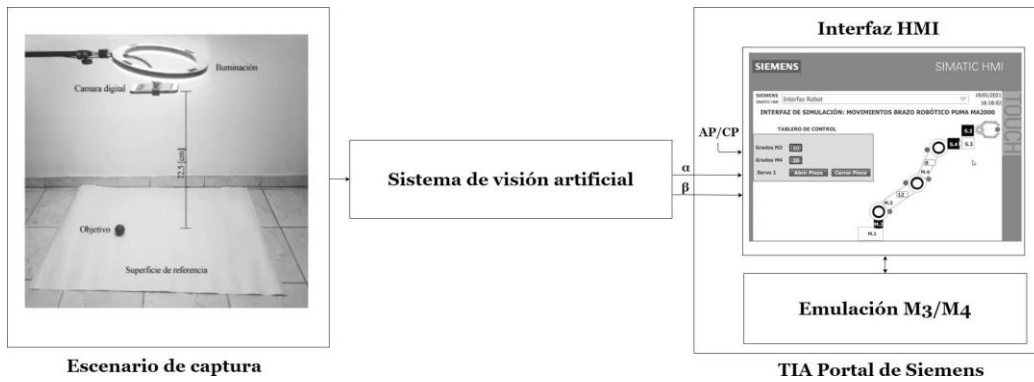
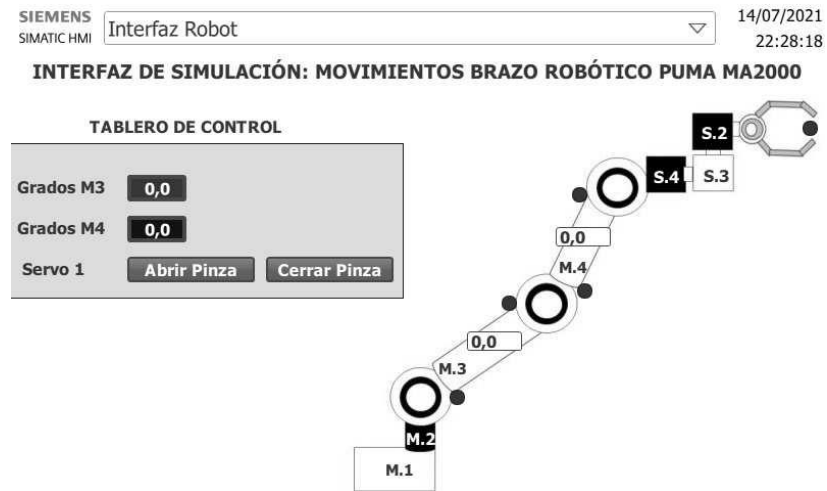


Figura 18.

Interfaz de simulación en panel HMI



3.2 Análisis de resultados

En favor de verificar la operación simulada del sistema de *Servo-Visión*, se procedió con el desarrollo de un experimento basado en la localización y el seguimiento para 60 posiciones

distintas del objeto (i.e. la esfera) en el plano de movimiento, almacenados en una base de datos mostrada en la Fig. 19.

A partir de ello se presentan los resultados contenidos en la Tabla 2, donde el *error de localización* para las coordenadas del objeto en la i -ésima posición de la base de datos se determina mediante un criterio de *distancia euclidiana*:

$$e_i = \frac{d_D}{d_C} \times 100\%,$$

donde para un punto de *coordenadas conocidas* (x_C, y_C) la distancia euclidiana conocida d_C se define como:

$$d_C = \sqrt{x_C^2 + y_C^2}$$

mientras que para un punto de *coordenadas automáticas* (x_A, y_A) calculadas por el sistema de visión artificial, la distancia euclidiana d_D del vector diferencia se define como:

$$d_D = \sqrt{(x_C - x_A)^2 + (y_C - y_A)^2}$$

Note que si $x_C = x_A$ e $y_C = y_A$ el error e_i se hace cero, mientras que si $x_A = y_A = 0$ el error será del 100 %.

A partir de ello, se verifica de los resultados una aproximación bastante satisfactoria para las coordenadas del objeto por parte del sistema de visión artificial, con porcentajes de error que superan el 4 % solo en dos de los puntos analizados.

De otro lado, en la misma tabla se incluye la conversión de coordenadas cartesianas a coordenadas polares mediante el cálculo cinemático para los ángulos α_i y β_i de las articulaciones del manipulador para cada punto de prueba. En este caso, el *error de posicionamiento* para la rutina desarrollada en TIA Portal es fijo y se define por la resolución del movimiento angular, correspondiendo con ± 0.1 [°] debido a la precisión decimal con la cual se presentan los datos en el panel HMI.

Figura 19.

Base de datos del registro de imágenes

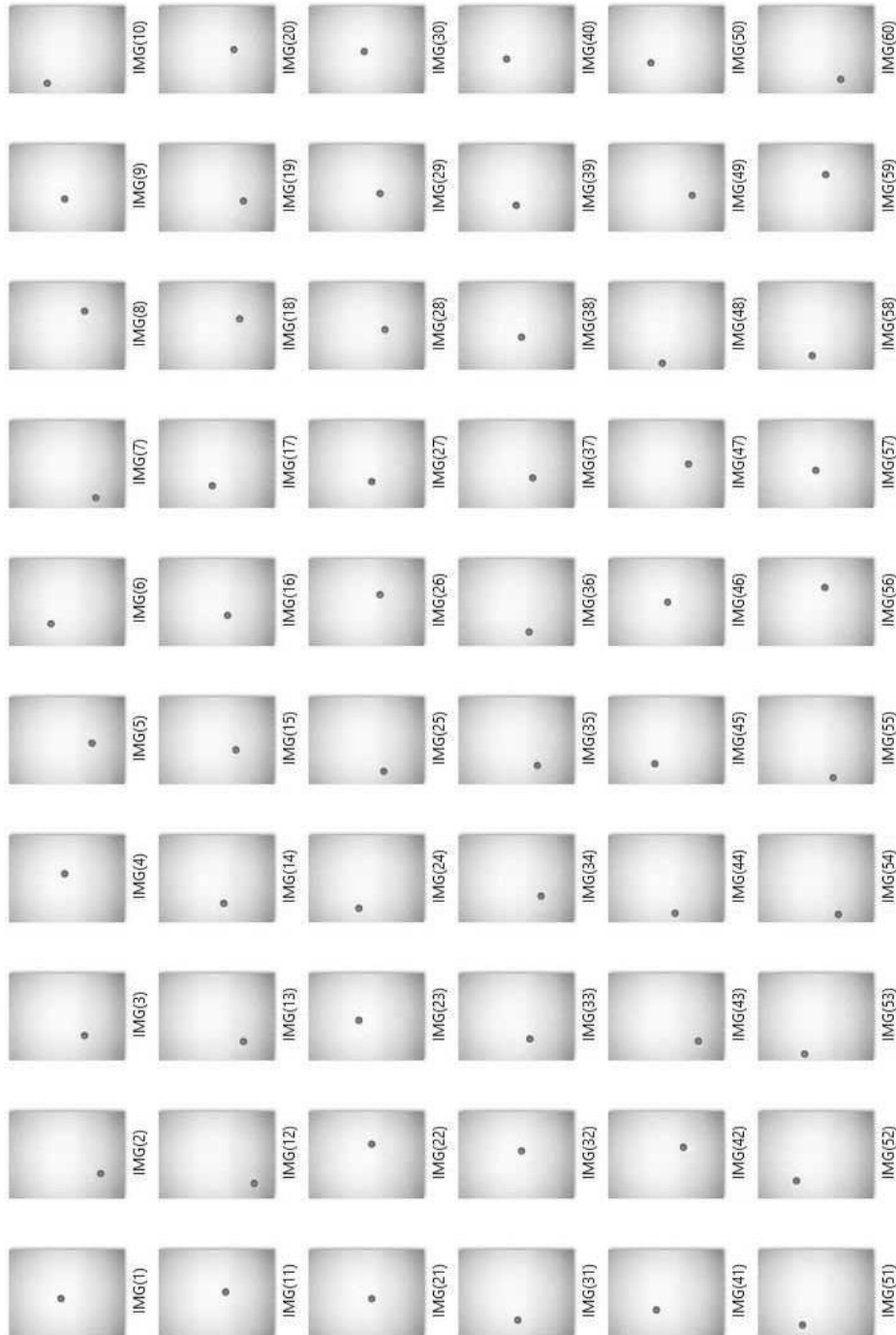


Tabla 2.

Pruebas para localización y seguimiento automático de coordenadas

Punto de prueba	$(x_C, y_C)[cm]$	$(x_A, y_A)[cm]$	ei [%]	ai [°]	βi [°]
1	(26.6, 44.3)	(26.2, 43.6)	1.56	75.1	35.3
2	(16.8, 16.3)	(17.1, 16.2)	1.35	0.1	132.6
3	(16.9, 27.3)	(17.1, 27.3)	0.62	17.9	127.3
4	(33.2, 38.9)	(33.3, 41.1)	4.31	60.7	36.8
5	(27.9, 22.2)	(28.1, 22.1)	0.63	0.9	100.2
6	(15.1, 49.5)	(15.3, 50.5)	1.95	83.4	58.2
7	(6.5, 19.5)	(6.9, 19.6)	2.01	27.6	167.6
8	(40.3, 26.8)	(40.2, 27.3)	1.05	13.1	65.8
9	(22.0, 41.0)	(22.3, 41.1)	0.68	85.9	24.6
10	(6.5, 53.1)	(6.9, 53.0)	0.77	89.0	73.3
11	(28.9, 33.2)	(30.7, 33.3)	4.10	20.8	88.0
12	(10.2, 13.6)	(10.4, 13.7)	1.32	12.2	161.7
13	(13.0, 20.6)	(13.0, 21.1)	2.05	15.1	144.5
14	(12.8, 34.2)	(13.0, 34.6)	1.22	33.2	128.8
15	(23.4, 26.3)	(23.5, 26.3)	0.28	10.4	111.2
16	(20.8, 31.8)	(21.0, 32.1)	0.95	21.8	113.5
17	(15.1, 42.4)	(15.2, 42.6)	0.50	97.1	29.6
18	(34.6, 23.4)	(34.8, 23.7)	0.86	3.4	82.5
19	(20.6, 21.2)	(21.0, 21.1)	1.39	3.7	120.0
20	(29.9, 27.7)	(30.1, 27.7)	0.49	10.3	93.7
21	(26.1, 35.6)	(26.1, 36.0)	0.91	26.2	96.8
22	(37.1, 35.4)	(37.4, 35.9)	1.14	57.3	24.2
23	(27.3, 45.0)	(27.5, 44.8)	0.54	69.4	42.5
24	(9.4, 44.8)	(9.6, 44.8)	0.44	52.1	117.3
25	(8.5, 27.5)	(8.8, 27.7)	1.25	30.4	148.6
26	(35.0, 30.3)	(35.2, 30.2)	0.48	65.5	2.9

Punto de prueba	$(x_C, y_C)[cm]$	$(x_A, y_A)[cm]$	ei [%]	ai [°]	β_i [°]
27	(17.7, 36.2)	(18.0, 35.9)	1.05	96.5	10.9
28	(27.3, 26.9)	(27.5, 26.8)	0.58	9.2	100.7
29	(25.8, 30.5)	(26.1, 30.3)	0.90	15.9	102.3
30	(28.5, 41.1)	(28.7, 41.1)	0.40	73.0	28.5
31	(11.6, 38.3)	(11.4, 38.5)	0.71	106.6	21.0
32	(32.2, 36.2)	(32.7, 36.0)	1.11	68.9	16.1
33	(14.6, 30.1)	(14.8, 30.3)	0.85	24.9	130.4
34	(17.9, 22.4)	(18.0, 22.5)	0.49	9.3	128.2
35	(12.8, 25.2)	(12.8, 25.1)	0.35	20.6	141.2
36	(9.4, 30.7)	(9.6, 30.8)	0.70	32.7	142.1
37	(20.4, 28.3)	(20.6, 28.2)	0.64	15.8	117.7
38	(22.2, 36.0)	(22.3, 36.0)	0.24	88.8	10.4
39	(17.7, 39.5)	(18.0, 39.6)	0.73	94.7	20.6
40	(22.4, 46.2)	(23.6, 46.3)	2.35	76.2	43.7
41	(18.3, 45.8)	(18.4, 46.3)	1.03	49.7	96.1
42	(35.2, 26.9)	(35.2, 27.7)	1.81	10.8	80.0
43	(13.2, 17.1)	(13.2, 17.3)	0.93	9.4	146.7
44	(6.2, 33.4)	(5.9, 32.8)	1.97	118.3	13.7
45	(14.0, 47.2)	(14.0, 47.4)	0.41	54.1	102.6
46	(29.9, 38.5)	(30.1, 38.5)	0.41	72.6	21.3
47	(29.9, 24.2)	(30.1, 24.2)	0.52	4.0	94.8
48	(4.3, 42.2)	(4.4, 42.2)	0.26	114.5	36.5
49	(24.8, 21.6)	(24.9, 21.6)	0.30	1.4	108.9
50	(20.6, 50.1)	(21.0, 50.0)	0.76	67.3	67.2
51	(8.0, 48.3)	(8.0, 48.9)	1.23	99.9	52.0
52	(12.2, 52.9)	(12.2, 53.1)	0.37	77.0	77.2
53	(4.5, 47.5)	(4.4, 47.4)	0.30	107.6	50.2
54	(5.5, 24.6)	(5.4, 24.2)	1.64	34.0	163.4
55	(3.9, 27.5)	(4.3, 27.7)	1.61	38.8	159.8

Punto de prueba	$(x_C, y_C)[cm]$	$(x_A, y_A)[cm]$	ei [%]	ai [°]	βi [°]
56	(40.1, 33.6)	(40.2, 33.4)	0.43	27.6	57.5
57	(25.5, 39.5)	(25.7, 39.7)	0.60	33.5	92.5
58	(9.2, 42.2)	(9.6, 42.2)	0.93	106.6	31.7
59	(38.9, 33.0)	(39.0, 32.8)	0.44	55.9	16.7
60	(9.4, 22.2)	(9.6, 22.5)	1.50	23.6	153.9

4. Conclusiones

A partir de los desarrollos presentados y los resultados obtenidos en el presente trabajo de grado, es posible enunciar la siguiente conclusión general:

Se realizó el cálculo y la programación para las trayectorias de movimiento de un *brazo robótico PUMA* en el *laboratorio de control e instrumentación de la E3T-UIS*, a partir del diseño e implementación para un sistema de *Servo-Visión* que permitió realizar la detección automática de las coordenadas de un objeto en un plano de movimiento, posteriormente convertidas (utilizando *cálculo cinemático*) en ángulos para el posicionamiento de las articulaciones de brazo y antebrazo del manipulador robótico, siendo además ejecutadas mediante simulación en un PLC *SIEMENS S7-1500* empleando rutinas en lenguaje escalera programadas en TIA Portal.

De manera más puntual:

Se desarrolló una rutina computacional para calcular las coordenadas de un objeto en el espacio a partir de técnicas de *visión artificial*. Tal y como fue abordado en la *Sección 2.2*, se realizó la implementación computacional en MATLAB para un sistema de *visión artificial* basado

en el algoritmo del cálculo del *centroide*, para detectar las coordenadas de posición de una esfera en un plano de movimiento. La Fig. 6 ilustra las etapas del sistema de *visión artificial* implementado. Asimismo, las *Subsecciones* 2.2.1, 2.2.2 y 2.2.3 abordan respectivamente las etapas de captura, preproceso y proceso de imagen, con sus respectivos resultados asociados.

Se determinaron las trayectorias de movimiento para un *brazo robótico PUMA* aplicando análisis cinemático sobre la información suministrada por el algoritmo de *visión artificial*. En particular, la *Sección* 2.1 presento las bases teóricas para definir la *cinemática de un robot* y particularmente para justificar el *cálculo cinemático* en un plano de movimiento para el manipulador *PUMA*. Posteriormente, habiendo definido el *problema cinemático inverso* en términos de los ángulos α y β para las articulaciones del brazo y antebrazo del manipulador robótico, se realizó un vínculo entre el algoritmo de *cálculo cinemático* y el sistema de *visión artificial* a partir de la configuración del esquema de *Servo-Visión* ilustrado en la Fig. 15, tomando como base el prototipo experimental acondicionado por *J. A. Monsalve* en (Monsalve, 2019).

Se ejecutaron las trayectorias en una interfaz que simulo el movimiento del *brazo robótico PUMA MA2000* a través de instrucciones programadas en un PLC. Tomando en cuenta las restricciones de acceso al campus universitario debidas a la emergencia sanitaria causada por el virus SARS-CoV-2 (COVID 19), la verificación del esquema de *Servo-Visión* fue realizada a partir de experimentos de simulación numérica en el TIA Portal de *SIEMENS* según se describe en la *Sección* 3.1. A partir de ello, empleando la base de datos ilustrada en la Fig. 19 para 60 posiciones de prueba diferentes del objeto de análisis, fue posible constituir los resultados incluidos en la Tabla 2 para verificar una apropiada aproximación en la *localización de coordenadas* por parte del algoritmo de *visión artificial* y su correspondiente mapeo a posiciones angulares a través de *cálculo cinemático*. Por su parte, el *error de posicionamiento* para la rutina ejecutada en el PLC

no se estimó de manera explícita ya que depende enteramente de la precisión del movimiento de rotación de los dispositivos actuadores (i.e. motores y servomotores) en el manipulador robótico.

5. Recomendaciones

La configuración de los pulsos en el bloque PULSEGEN programado para la emulación en TIA Portal del movimiento de los motores M3 y M4, se realizó asumiendo valores de tiempo arbitrarios que facilitarían visualizar los cambios de ángulo en la interfaz del panel HMI para efectos de simulación.

En el momento de realizarse una verificación experimental sobre el prototipo real en laboratorio, se recomienda modificar la asignación de tiempos en las tramas de pulsos PWM utilizadas para manipular el movimiento de motores y actuadores, de manera acorde a la caracterización realizada por *J. A. Monsalve* en (Monsalve, 2019).

5.1 Trabajo futuro

Evidentemente, la acción inmediata que debe ser realizada al momento de superarse la emergencia sanitaria causada por el virus SARS-CoV-2 (COVID 19), corresponde con la implementación real en laboratorio para el esquema de *Servo-Visión* propuesto y la verificación experimental para la programación automática de trayectorias en el manipulador robótico *PUMA*.

Referencias Bibliográficas

- Allauca, S. A., & Estévez, M. A. (2019). *Diseño y construcción de un robot PUMA con un sistema de simulación en tiempo real*. Ecuador: Trabajo de grado (Ingeniero Mecatrónico), Universidad de las Fuerzas Armadas. Departamento de Ciencias de la Energía y Mecánica.
- Álvarez, Y. G. (2014). *Diseño y construcción de un vehículo aéreo no tripulado UAV (Unmanned Aerial Vehicle)*. Bucaramanga: Trabajo de grado (Ingeniero Mecánico), Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Mecánica.
- Amaya, Y. D., & Ruiz, J. A. (2005). *Localización dinámica de móviles y obstáculos en una escena controlada para aplicaciones en robótica*. Bucaramanga: Trabajo de grado (Ingeniero Electrónico), Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de ingenierías Eléctrica, Electrónica y Telecomunicaciones.
- Arango, J. E. (2014). *Payload análisis and control of manipulators for human interactive environments*. Manizales: Tesis doctoral (doctor en ingeniería), Universidad Nacional de Colombia.
- Ardila, P. M., Dulcey, D. C., Mendoza, Y. R., & Mendoza, C. U. (2012). *Seminario de investigación en robótica industrial*. Bucaramanga: Trabajo de grado (Ingeniero Mecánico), Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Mecánica.
- Beltrán, N. F. (2011). *Diseño e implementación de la tarjeta para control de robot PUMA MA2000*. Bucaramanga: Trabajo de grado (Ingeniero Electrónico), Universidad Industrial

de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de ingenierías Eléctrica, Electrónica y Telecomunicaciones.

Buitrago, J. E., & Cediell, N. G. (2010). *Robot cartesiano neumático para el laboratorio de sistemas mecatrónicos de la escuela de ingeniería mecánica, diseño y construcción*. Bucaramanga: Trabajo de grado (Ingeniero Mecánico), Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Mecánica.

Carreño, J. J. (2015). *Control robusto de la dinámica de un robot manipulador, considerando incertidumbre paramétrica*. Bucaramanga: Trabajo de grado (Maestría en Ingeniería Electrónica), Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de ingenierías Eléctrica, Electrónica y Telecomunicaciones.

Carrera, I. G. (2011). *Diseño, construcción y control de un robot domestico para asistencia y entrenamiento de la movilidad*. Madrid: Tesis doctoral (doctor en ingeniería industrial), Universidad Politécnica de Madrid.

Cuevas, E., Zaldivar, D., & Pérez, M. (2010). *Procesamiento digital de imágenes con MATLAB y Simulink*. Ciudad de México: Grupo Editorial Alfaomega.

Gutierrez, A. L., & Nogales, C. A. (2011). *Sistema para el procesamiento automático de contornos poligonales basado en Servo-Visión*. Bogotá: Trabajo de grado (Ingeniero en Automatización), Universidad de La Salle. Facultad de Ingeniería .

Gutiérrez, M. C. (2010). *Rediseño e implementación de control articular de robot PUMA de 6 grados de libertad y su respectivo sistema de alimentación eléctrica*. Bucaramanga: Trabajo de grado (Ingeniero Electrónico), Universidad Industrial de Santander, Facultad de Ingenierías Físico-Mecánicas. Escuela de ingenierías Eléctrica, Electrónica y Telecomunicaciones.

- Herrera, J. A., & Rincón, R. A. (2012). *Diseño y construcción del prototipo experimental de la estructura para un vehículo aéreo no tripulado (UAV) tipo quadrotor*. Bucaramanga: Trabajo de grado (Ingeniero Mecánico), Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingeniería Mecánica.
- Jauregui, A. L., & Vianchada, C. A. (2010). *Diseño de un sistema de control de cabeceo y alabeo de un vehículo aéreo no tripulado (UAV)*. Bucaramanga: Trabajo de grado (Ingeniero Electrónico), Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de ingenierías eléctrica, electrónica y telecomunicaciones.
- Leguizamón, D. A. (2016). *Simulación de movimientos de un brazo robótico con 5 grados de libertad en el procedimiento de vacunación de ganado, utilizando MATLAB*. Bogotá: Trabajo de grado (Ingeniero Mecánico), Universidad Libre. Facultad de Ingeniería.
- Monsalve, J. A. (2019). *Acondicionamiento y puesta a punto del brazo robótico TQ MA2000*. Bucaramanga: Trabajo de grado (Ingeniero Electrónico), Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de ingenierías Eléctrica, Electrónica y Telecomunicaciones.
- Olivares, J. A. (2016). *Interfaz de usuario para el robot PUMA 560 de UNIMATION*. México: Trabajo de grado (Maestría en Sistemas de Manufactura), Tecnológico de Monterrey. División de Ingeniería y Arquitectura.
- Ortiz, I. R. (2012). *Sintonización de controladores PID para un robot manipulador PUMA MA2000*. Bucaramanga: Trabajo de grado (Ingeniero Electrónico), Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de ingenierías Eléctrica, Electrónica y Telecomunicaciones.
- Peña, A. G., Pérez, J. G., & Suarez, M. F. (2005). *Prototipo de un robot tipo PUMA de 6 grados*

- de libertad*. Bucaramanga: Trabajo de grado (Ingeniero Electrónico), Universidad Industrial de Santander. Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica y Telecomunicaciones.
- Poveda, D. K. (2014). *Identificación y control de un robot manipulador PUMA MA2000 de seis grados de libertad*. Bucaramanga: Trabajo de grado (Ingeniero Electrónico), Universidad Industrial de Santander. de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y Telecomunicaciones.
- Reyes, F. (2011). *Control de robots manipuladores*. Ciudad de México: Grupo Editorial Patria.
- Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2010). *Robotics*. New York: Springer Science & Business Media.
- Vaca, S. D. (2012). *Diseño y simulación de un robot manipulador industrial tipo PUMA capaz de levantar pesos de hasta 20 kg*. Ecuador: Trabajo de grado (Ingeniero Mecánico), Escuela Politécnica Nacional. Facultad de Ingeniería Mecánica.