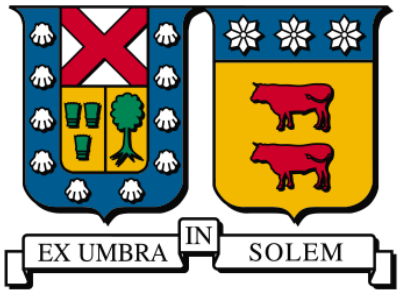
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



"CONTROL AUTOMÁTICO DEL BRAZO ROBOTICO 'MOTOMAN-MH50' UTILIZANDO COMO SENSOR LA CÁMARA 'MOTOSIGHT2D' "

José Felipe Levi Gómez

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELECTRÓNICO

PROFESOR GUÍA: **Francisco Vargas Parra.**

PROFESOR CORREFERENTE: **Héctor Ramírez Estay.**

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☑ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: Control Automático del Brazo Robótico "Motoman-MH50 utilizando como Sensor la Cámara "MotoSight2D"

Nombre del candidato(a): José Felipe Levi Gómez

Carrera / Grado: Ingeniería Civil Electrónica

Campus: Casa Central Valparaíso **Departamento:** Electrónica

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Francisco Javier Vargas Parra, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☑ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por **(marcar una opción)**:

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

_____

_____

_____

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 21/01/2026 **Firma:** 

Estudiante o Candidato(a):

Fecha: 21/01/2026 **Firma:** 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

Quiero agradecer al Centro Científico Tecnológico de Valparaíso (CCTVal) por facilitar el uso de su laboratorio y de los recursos necesarios para el desarrollo de este trabajo de título.

Agradezco al profesor Francisco Vargas, en su rol de profesor guía, por su orientación, disposición y apoyo durante el desarrollo de este trabajo, así como por sus observaciones y sugerencias que contribuyeron a mejorar su calidad final.

Agradezco también a David Aliaga, Daniel Erraz y Sebastián Latorre, con quienes trabajé y fueron parte fundamental de mi desarrollo como ingeniero, así como por las cartas de recomendación que contribuyeron a que este proceso pudiera finalizar de manera exitosa.

Finalmente, agradezco al profesor Héctor Ramírez por su participación en la evaluación de este trabajo y por la valiosa retroalimentación entregada durante dicho proceso.

Dedico este trabajo a mis padres y a mi hermana, quienes me han entregado el cariño, la educación y las herramientas necesarias para alcanzar mis metas, así como el apoyo incondicional que me han brindado desde siempre. Mis logros son también sus logros.

Asimismo, dedico este trabajo a mi familia y amigos, por ser parte del camino recorrido y de la persona que soy hoy.

Resumen

En automatización industrial, una tarea recurrente es el uso de brazos robóticos para operaciones de paletización y manipulación de piezas, las cuales suelen requerir que los objetos se encuentren en posiciones predefinidas dentro del espacio de trabajo. La incorporación de sistemas de visión por computador permite flexibilizar estos procesos, habilitando la detección de objetos en posiciones variables.

En este trabajo se aborda la integración de un sistema de visión por computador a un brazo robótico industrial, con el objetivo de detectar la posición de un objeto y comandar el desplazamiento del robot para su manipulación. El sistema está compuesto por un brazo robótico Motoman MH50 y una cámara MotoSight2D, ambos de la marca Yaskawa. Se implementa una tarea de automatización que consiste en detectar un objeto dentro del área de trabajo y trasladarlo hacia una posición objetivo. Asimismo, se evalúa la factibilidad de implementar estrategias de control automático al movimiento del robot.

Los resultados muestran que el sistema de visión permite detectar la posición del objeto con buena precisión. Sin embargo, se identifica una imprecisión mecánica en una de las articulaciones del robot, lo que afecta la exactitud del posicionamiento final. Además, la arquitectura del sistema no permite la implementación de un control automático en lazo cerrado, debido a la imposibilidad de realizar mediciones continuas mientras el robot se encuentra en movimiento. Como alternativa, se plantea una solución basada en interrupciones que permite reaccionar ante un desplazamiento del objeto detectado. Esta solución permitió detener correctamente el robot y reiniciar el ciclo de detección.

Como conclusión, el sistema permite realizar aplicaciones de paletización y manipulación de piezas de forma precisa y confiable, validando la integración entre el sistema de visión y el brazo robótico. Como trabajo futuro, se propone habilitar la activación automática de interrupciones y estudiar la posibilidad de controlar del robot en tiempo real para implementar estrategias de control automático avanzadas.

Palabras Clave: brazo robótico, automatización industrial, visión por computador, detección de objetos, manipulación de objetos

Abstract

In industrial automation, a recurring task is the use of robotic manipulators for palletizing and object handling operations, which typically require objects to be located at predefined positions within the workspace. The incorporation of computer vision systems increases the flexibility of these processes by enabling the detection of objects at variable positions.

This work addresses the integration of a computer vision system with an industrial robotic arm in order to detect the position of an object and command the robot motion for its manipulation. The system is composed of a Motoman MH50 robotic arm and a MotoSight2D camera, both manufactured by Yaskawa. An automation task is implemented in which an object is detected within the workspace and transported to a target position. Additionally, the feasibility of implementing automatic control strategies for the robot motion is evaluated.

The results show that the vision system is capable of detecting the object position with good accuracy. However, a mechanical inaccuracy was identified in one of the robot joints, which affects the precision of the final positioning. Furthermore, the system architecture does not allow the implementation of closed-loop automatic control, due to the inability to perform continuous measurements while the robot is in motion. As an alternative, an interruption-based solution is proposed to react to object displacement during the robot motion. This solution successfully allowed the robot to stop and restart the detection cycle.

In conclusion, the system enables precise and reliable palletizing and object manipulation applications, validating the integration between the vision system and the robotic arm. As future work, the automatic activation of interruptions is proposed, along with the study of real-time robot control to enable the implementation of advanced automatic control strategies.

Keywords: robotic arm, industrial automation, computer vision, object detection, object picking

Glosario

Siglas

***TCP**:* Tool Centre Point. Punto que representa el extremo útil de la herramienta instalada en el robot.

***UF**:* User Frame. Sistema de coordenadas definido por el usuario para adaptar el movimiento del robot al entorno.

Términos

***Brazo robótico**:* Manipulador industrial compuesto por varias articulaciones que permiten realizar movimientos programados.

***Articulaciones S, L, U, R, B y T**:* Denominación utilizada por Yaskawa para identificar las seis articulaciones del robot.

***Home Position**:* Posición inicial. Posición de referencia definida por el fabricante, correspondiente a la ubicación cero de cada articulación del robot.

***Coordenadas cartesianas (X, Y, Z)**:* Sistema de referencia utilizado para describir posiciones y trayectorias del robot y la cámara.

***Controlador DX200**:* Unidad encargada de ejecutar la programación, procesar imágenes y gestionar las comunicaciones del sistema.

***Encoder**:* Dispositivo incorporado en servomotores que mide la posición angular, permitiendo controlar el movimiento con precisión.

***Cámara MotoSight2D**:* Sensor de visión utilizado para detectar patrones y obtener sus coordenadas.

***Campo de visión**:* Área física que la cámara puede observar y analizar en una captura.

***Grilla de calibración**:* Plantilla con puntos de referencia utilizada para calibrar el sistema de visión.

***Patrón**:* Imagen o figura previamente enseñada a la cámara para ser reconocida y utilizada como referencia de ubicación.

***Trabajo de visión**:* Conjunto de configuraciones y entrenamientos realizados en software 'In-Sight Explorer' ejecutados por la cámara para detectar un patrón y obtener sus coordenadas.

***Macro INSPECT**:* Instrucción utilizada para capturar una imagen y ejecutar un trabajo de visión.

***Interrupción**:* Señal externa que permite detener o modificar el movimiento del robot durante su ejecución.

Unidades y símbolos

***[mm]**:* Milímetros. Unidad de longitud utilizada para medir distancias y desplazamientos.

***[kg]**:* Kilogramos. Unidad de masa utilizada para expresar la carga útil y el peso de componentes.

***[°]**:* Grados. Unidad angular utilizada para medir posiciones y desplazamientos de las articulaciones.

***[N·m]**:* Newton-metro. Unidad de torque utilizada para expresar el momento ejercido por las articulaciones.

***[kVA]**:* Kilovolt–ampere. Unidad de potencia aparente utilizada para especificar el consumo eléctrico del sistema.

***[mm/s]**:* Milímetros por segundo. Unidad de velocidad lineal utilizada para describir el movimiento del robot.

***[°/s]**:* Grados por segundo. Unidad de velocidad angular utilizada para describir la velocidad de las articulaciones.

Índice general

1.. Introducción 1

2.. Sistema Robot-Cámara 5

2.1. Descripción General del Sistema 5

2.2. Robot Motoman MH-50 6

2.3. Cámara de Visión MotoSight2D 14

2.4. Controlador 15

2.5. Mando de control 16

2.6. Computador 17

2.7. Diagrama de funcionamiento 18

2.8. Conclusión del capítulo 19

3.. Funcionamiento práctico del brazo robótico 20

3.1. Pruebas de movimiento del brazo robótico 20

3.2. Calibración del TCP y creación de UF 23

3.3. Análisis del Error 28

3.4. Cámara MotoSight2D e In-Sight Explorer 30

3.5. Aplicación MotoSight2D 33

3.6. Creación de Trabajo de Programación intregrando la Detección de la Cámara 35

3.7. Conclusión del capítulo 36

4.. Aplicación del funcionamiento del robot a un problema de automatización . . 38

4.1. Definición del Problema de Automatización 38

4.2. Análisis y propuesta de solución 39

4.3. Implementación de la solución 42
4.4. Conclusiones del capítulo 49

5.. Control automático en el sistema Robot-Cámara 50
5.1. Análisis del Sistema 50
5.2. Solución Propuesta 52
5.3. Implementación de la Solución 54
5.4. Conclusión del Capítulo 57

6.. Conclusión 59

Índice de figuras

2.1. Diagrama del sistema 6
2.2. (a)Brazo robótico MH50 II-35. (b) Fotografía del robot en el lugar de trabajo. 7
2.3. Esquema del brazo robótico con la rotulación de cada una de sus articulaciones (*Fuente: DX200 INSTRUCTIONS*). 8
2.4. Rango de movimientos y alcance físico del robot (*Fuente: Motoman MH50 Series Brochure*). 9
2.5. Posición de fábrica del brazo robótico (*Fuente: DX200 INSTRUCTIONS*). 10
2.6. Ejemplos de TCP (*Fuente: DX200 INSTRUCTIONS*). 10
2.7. Referencia coordenadas de la herramienta (*Fuente: DX200 INSTRUCTIONS*). 11
2.8. Ejemplo para configurar el TCP de forma automática (*Fuente: DX200 INSTRUCTIONS*). 11
2.9. Esquema de las coordenadas cartesianas (*Fuente: DX200 INSTRUCTIONS*). 12
2.10. Esquema de las coordenadas cartesianas (vista superior y lateral) (*Fuente: DX200 INSTRUCTIONS*). 12
2.11. Esquema de las coordenadas de usuario (UF) (*Fuente: DX200 INSTRUCTIONS*). 12
2.12. Esquema de las coordenadas de usuario (*Fuente: DX200 INSTRUCTIONS*). 13
2.13. Trayectoria de una interpolación lineal (*Fuente: DX200 INSTRUCTIONS*). 13
2.14. Cámara Motosight2D. 14
2.15. Controlador DX200 15
2.16. Mando de Control 16

2.17. Grilla de calibración de 9 puntos (*Fuente: MOTOSIGHT2D INSTRUCTIONS*). 17
2.18. Diagrama de funcionamiento para realizar una tarea de detección y manipulación de un objeto 18

3.1. Izquierda: Mando de control del brazo robótico. Derecha: Encerradas en rojo se encuentran las teclas para realizar los movimientos del robot. . . . 21
3.2. Error asociado al movimiento en el eje X del brazo robótico. 21
3.3. Error asociado al movimiento en el eje Y del brazo robótico. 22
3.4. Error asociado al movimiento en el eje Z del brazo robótico. 22
3.5. Trabajo de programación básico 23
3.6. Fotografía de la herramienta utilizada. 24
3.7. Las 5 posiciones utilizadas para configurar el TCP automáticamente. . . . 25
3.8. Medidas de la herramienta obtenidas automáticamente. 26
3.9. Extremo de la herramienta en el punto ORG para la creación de un nuevo UF. 27
3.10. Extremo de la herramienta en el punto XX para la creación de un nuevo UF. 27
3.11. Extremo de la herramienta en el punto XY para la creación de un nuevo UF. 28
3.12. Gráfico de los datos de la Tabla 3.1. 29
3.13. Pantalla de calibración realizada. Se muestran los ejes configurados encerrado en rojo. 31
3.14. Se indica la zona dentro de la captura en que se encuentra el patrón. . . . 32
3.15. Se indica la zona dentro de la captura en que se encuentra el patrón. . . . 32
3.16. Patrón es verificado mediante sus medidas. 33
3.17. Interfaz aplicación Motosight2D. 34
3.18. Interfaz pestaña “Result”. 34
3.19. Programación básica para utilizar la cámara en una aplicación con el robot. 35
3.20. Resultados del movimiento del robot indicado por la tarea de la cámara. . 36

4.1. Espacio de trabajo delimitado por cinta verde 39
4.2. (1): Esquema de cuadrantes con punto ciego. (2) y (3): Esquema de cuadrantes superpuestos. 41
4.3. Esquema de diez cuadrantes utilizados en el trabajo. 41
4.4. Máquina de estados de la solución al problema de automatización. 42
4.5. (1): Patrón que representa al objeto. (2) Patrón que representa a la ubicación objetivo 43
4.6. Programación trabajo principal (parte 1) 44
4.7. Programación trabajo principal (parte 2) 45
4.8. Programación del trabajo que selecciona el escaneo. 46
4.9. Programación de escaneo 46
4.10. Programación del trabajo de movimiento. 47
4.11. Prueba de funcionamiento 1: Llegada del puntero al patrón (1) objeto y (2) ubicación objetivo. 48
4.12. Prueba de funcionamiento 2: Llegada del puntero al patrón (1) objeto y (2) ubicación objetivo. 48

5.1. Esquema de control de lazo cerrado en cascada que permitiría el control automático del sistema. 51
5.2. Esquema de control de la solución implementada. 53
5.3. Diagrama de conexiones de terminal de entradas y salidas “CN309” del controlador “DX200” 54
5.4. Fotografía del terminal “CN309” del controlador “DX200” utilizado. . . . 55
5.5. Programación del trabajo de movimiento con las instrucciones de la interrupción en rojo. 56
5.6. Cables conectados al terminal “CN309” para activar la interrupción encerrados en rojo. 56
5.7. Secuencia del robot al activar la interrupción 57

Índice de cuadros

2.1. Especificaciones generales (*Fuente: Motoman MH50 Series Brochure*). . 7
2.2. Especificaciones articulaciones (*Fuente: Motoman MH50 Series Brochure*). 9
2.3. Especificaciones cámara (*Fuente: Motosight2D Brochure*). 14

3.1. Mediciones de los movimientos físicamente y los entregados por el mando de control. 29

1. INTRODUCCIÓN

El presente trabajo se enmarca en el área de automatización industrial, específicamente en la utilización de brazos robóticos para realizar el transporte de piezas y paletización en procesos de ensamblado, líneas de producción o montaje de cargas [1, 2]. En este tipo de entornos, los robots son programados para realizar tareas rápidas y repetitivas entre posiciones predefinidas [3], lo que implica que los objetos a manipular deben estar exactamente en esas ubicaciones para poder ser manipulados. Este posicionamiento puede realizarse de forma manual o mediante un sistema externo automatizado, lo que incrementa la complejidad del proceso, introduce posibles errores y limita la adaptabilidad del sistema. Además, cualquier modificación del entorno de trabajo requiere la reprogramación del robot, disminuyendo la eficiencia del proceso [4]. En este contexto, la principal problemática abordada en el proyecto consiste en detectar la posición de un objeto dentro de un espacio de trabajo y manipularlo utilizando un brazo robótico industrial.

El objetivo general de este trabajo consiste en controlar el desplazamiento de un brazo robótico industrial para desarrollar una tarea específica utilizando como mediciones la información entregada por un sistema de visión por computador. La tarea consiste en reconocer la posición de un objeto dentro del espacio de trabajo, ubicar el extremo del brazo robótico sobre él y trasladarlo hacia una posición objetivo, la cual estará definida mediante un patrón visual reconocido por la cámara.

Los objetivos específicos se presentan a continuación:

- Analizar el funcionamiento, las capacidades y las limitaciones del brazo robótico y su controlador, incluyendo aspectos de programación, integración de herramientas y uso de entradas y salidas.

- Realizar la integración de una cámara al controlador del robot y estudiar su interfaz y software asociados, con el fin de generar movimientos del robot a partir de la información visual adquirida.

- Implementar una estrategia de control del movimiento del robot basada en las mediciones de la cámara.

- Extender la estrategia anterior de modo que las tareas del robot se adapten a la posición de los objetos dentro del campo de visión y a posibles desplazamientos de estos durante la operación del sistema.

En los últimos años, se ha realizado un gran avance en la visión por computadora [5]. Los resultados obtenidos por el análisis digital de imágenes o el uso de redes neuronales son lo suficientemente precisos para el desarrollo de aplicaciones complejas, como el control de vehículos autónomos [6, 7] y asistencia en procedimientos quirúrgicos [8,9]. Con respecto a la robótica industrial, la visión por computadora es ampliamente utilizada para tareas de manipulación de objetos en correas transportadoras, en paletización y transporte de piezas o contenedores. En este contexto, la motivación principal de este trabajo es estudiar las capacidades y limitaciones de la integración de la cámara de visión "MotoSight2D", analizando sus posibles aplicaciones, el tipo de detecciones que realiza, su precisión y la comparación del desempeño del sistema con y sin el uso de visión.

El sistema utilizado está compuesto por el brazo robótico "Motoman MH-50 II-35" y la cámara "MotoSight2D", los cuales interactúan con el espacio de trabajo permitiendo detectar y manipular objetos. La programación y la configuración del sistema se realizan en el mando de control, mientras que para trabajar con la cámara se utiliza el programa computacional "In-Sight Explorer". El enfoque utilizado para desarrollar el trabajo será principalmente de manera práctica, estudiando el funcionamiento e integración de cada componente, realizando pruebas individuales y conjuntas, y desarrollando una tarea de automatización que permita establecer un procedimiento aplicable a un entorno industrial real.

La principal contribución de este proyecto es la implementación de una tarea de

automatización basada en el uso conjunto de un brazo robótico y un sistema de visión por computador para la manipulación de piezas no ordenadas previamente dentro del espacio de trabajo [10], las cuales son trasladadas hacia una posición objetivo también detectada mediante visión. Esto permite aumentar la flexibilidad del sistema y reducir los tiempos de programación. Adicionalmente, se estudia la factibilidad de implementar control automático utilizando las herramientas disponibles, con el fin de responder ante el desplazamiento de un objeto mientras el robot se dirige hacia él, considerando dicho desplazamiento como una perturbación del sistema.

Los alcances del proyecto incluyen establecer el procedimiento necesario para poner en funcionamiento el conjunto compuesto por el robot y la cámara, diseñar una solución a un problema de automatización que permita verificar el desempeño del sistema y la creación de estrategias que compensen sus limitaciones. Adicionalmente, se determinan los factores que aportan al desarrollo de un sistema de control automático y aquellos que representan limitaciones. Como proyección del trabajo, se plantea evaluar el desempeño del sistema con la articulación del brazo robótico reparada (desperfecto mecánico en una de sus articulaciones que genera problemas de precisión, mencionado en el desarrollo del trabajo). También se considera estudiar la integración de un sensor externo que permita entregar información de forma continua acerca de las posiciones de los objetos a manipular y la viabilidad de controlar el robot de forma externa para realizar movimientos no programados, pero en base a información proporcionada en tiempo real por distintos sensores, permitiendo estrategias de control más avanzadas.

El documento se estructura en seis capítulos que incluyen la introducción y conclusión, además de cuatro capítulos intermedios que presentan el desarrollo del tema de memoria. A continuación se presenta el contenido de cada capítulo:

1. Introducción. Presenta el contexto, los objetivos, la motivación, los procedimientos, herramientas utilizadas, el alcance y las proyecciones del trabajo de título.

2. Describe el sistema robot-cámara, presentando cada uno de sus elementos e indicando el rol que cumplen, las características relevantes y su principio de funcionamiento. También, presenta el flujo que define cómo trabajar de forma conjunta

para detectar y manipular los objetos.

3. Realiza pruebas de funcionamiento en base al flujo de trabajo presentado en el capítulo 2. Cada prueba verifica el desempeño y las limitaciones de los elementos. Durante su desarollo se encuentra y analiza un error en una de las articulaciones del sistema estableciendo su causa y posibles medidas de compensación. Realiza con éxito el movimiento del brazo robótico hacia un objeto en base a la información entregada por la cámara.

4. Define el problema de automatización planteado en el objetivo general del proyecto. Realiza el análisis del problema y presenta una propuesta de solución que divide el espacio de trabajo en cuadrantes, dado el reducido campo de visión y la resolución de la cámara. Implementa la solución detectando de forma precisa la posición del objeto, sin embargo, el movimiento del robot genera errores de precisión debido al desperfecto en una de sus articulaciones.

5. Realiza un análisis a los elementos del sistema con respecto a la posibilidad de realizar control automático. Establece una relación entre los requerimientos de un sistema de control de lazo cerrado con las capacidades y forma de trabajo de los elementos. Se concluye que la cámara MotoSight2D no es apta para realizar control automático. No obstante, se presenta una solución alternativa en base a interrupciones que resuelve el problema planteado del desplazamiento del objeto, pero no se considera como sistema de lazo cerrado.

6. Conclusiones. Resume los resultados obtenidos, evalúa y compara la integración de la cámara y el trabajo sin ella en factores como la implementación, programación y precisión. Presenta las limitaciones y errores encontrados.

2. SISTEMA ROBOT-CÁMARA

En este capítulo se describe el sistema robot-cámara utilizado en el desarrollo del trabajo de título y cuáles son sus principales componentes. Se da a conocer cuáles son las características y las funcionalidades principales utilizadas en el desarrollo del proyecto, además de cuál es su rol dentro del sistema. El propósito general de este capítulo es entender la estructura y operación del sistema, y la interacción entre los elementos que servirán como base para etapas posteriores del trabajo.

2.1. Descripción General del Sistema

El sistema robot-cámara se compone de un brazo robótico y una cámara, además de un controlador que se encarga de su funcionamiento, un mando de control utilizado para la manipulación y programación del robot, y un computador que ejecuta el programa computacional para la configuración de la cámara. El objetivo principal del sistema es detectar un objeto dentro de un espacio de trabajo y determinar su posición y la de una ubicación objetivo mediante la información entregada por la cámara, y trasladar el objeto desde su posición inicial a la posición objetivo con el brazo robótico. En la Fig. 2.1 se muestra el diagrama general del sistema y la interconexión entre sus componentes. En él se distinguen tres secciones principales: interfaz, procesamiento y sensado y actuación.

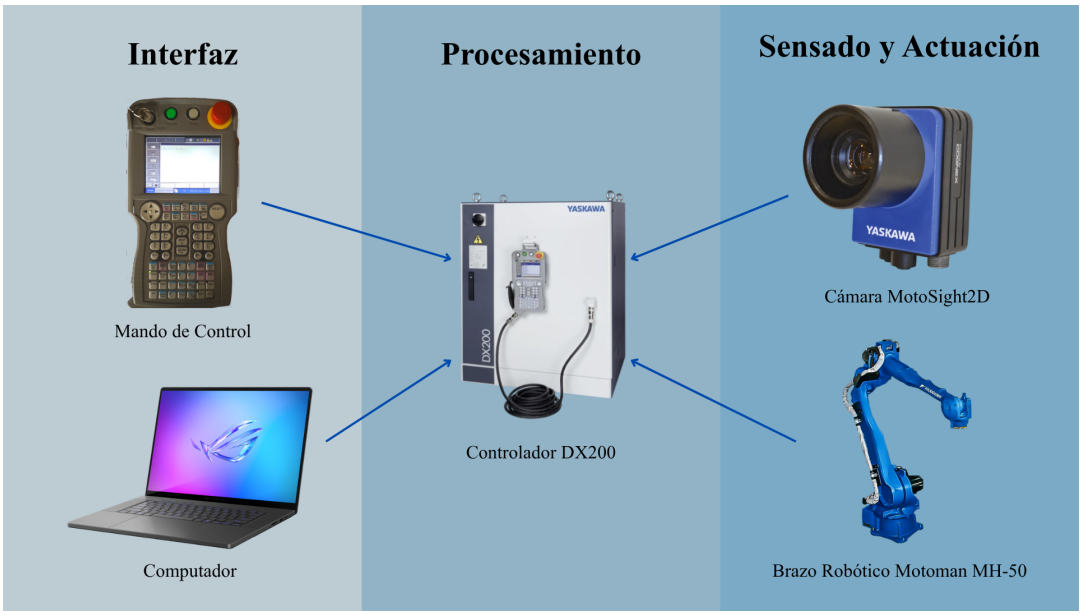


Fig. 2.1: Diagrama del sistema

La interfaz está compuesta por el mando de control y un computador, en los que el usuario interactúa para dar las instrucciones de funcionamiento. Por otra parte, el procesamiento se realiza en el controlador, que ejecuta las tareas programadas y procesa los movimientos del robot e imágenes de la cámara. Finalmente, la actuación y sensado está conformada por el robot y la cámara, que interactúan directamente con el espacio de trabajo, detectando los objetos y ejecutando los movimientos. Esta división permite conocer de manera general el rol de cada componente dentro del sistema. En las secciones posteriores se describen sus características y funciones relevantes para el desarrollo del trabajo de título.

2.2. Robot Motoman MH-50

El brazo robótico utilizado es el “Motoman MH-50 II-35” de “Yaskawa” (Fig. 2.2), un robot industrial de alta velocidad y largo alcance [11]. Está diseñado para otorgar un alto rendimiento para tareas de soldadura, corte de materiales, manipulación de objetos y paletizado, dispensación de sustancias y recubrimientos con pintura [12]. En el presente trabajo, el robot cumple la función de trasladar los objetos detectados por

la cámara desde su posición inicial hasta una posición objetivo dentro del espacio de trabajo.



Fig. 2.2: (a)Brazo robótico MH50 II-35. (b) Fotografía del robot en el lugar de trabajo.

En la Tabla 2.1 se muestran las especificaciones técnicas del robot. Entre las más importantes se encuentran los seis grados de libertad (articulaciones), la carga útil máxima de 35[kg] y alcance máximo de 2538[mm] [11].

Item	**Valor**
Grados de Libertad	6
Carga Útil Máxima	35[kg]
Repetibilidad	$\pm$0.07[mm]
Rango Máximo de Trabajo	2538[mm]
Peso	570[kg]
Fuente de Alimentación	4.0[kVA]

Tab. 2.1: Especificaciones generales (*Fuente: Motoman MH50 Series Brochure*).

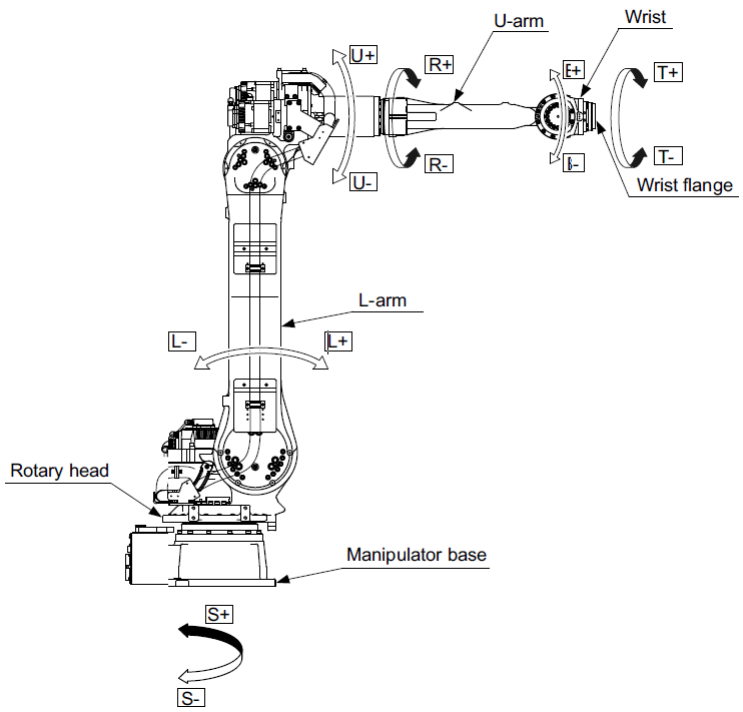


Fig. 2.3: Esquema del brazo robótico con la rotulación de cada una de sus articulaciones (*Fuente: DX200 INSTRUCTIONS*).

Cada articulación se nombra con las letras S, L, U, R, B y T ordenadas desde la base hasta el extremo del brazo robótico (Ver Fig. 2.3) [13] y en la tabla 2.2 se presentan las características de movimiento de cada articulación. Es importante conocer estos parámetros, el rango de movimiento del robot y los límites de cada articulación para la definición de movimientos seguros dentro del espacio de trabajo y evitar golpes contra el entorno. En la Fig. 2.4 se muestra el rango de movimiento y el alcance del brazo robótico desde una vista lateral izquierda y una vista superior [11], lo que sirve para determinar la zona de trabajo para el problema de automatización planteado y su operación segura.

Articulación	Rango Máximo de Movimiento [°]	Máxima Velocidad [°/sec]	Momento Admisible [N·m]	Momento Admisible de Inercia [kg· m^2]
S	± 180	180	-	-
L	$+135/-90$	140	-	-
U	$+251/-160$	178	-	-
R	± 360	250	147	10
B	± 125	250	147	10
T	± 360	360	78	4

Tab. 2.2: Especificaciones articulaciones (*Fuente: Motoman MH50 Series Brochure*).

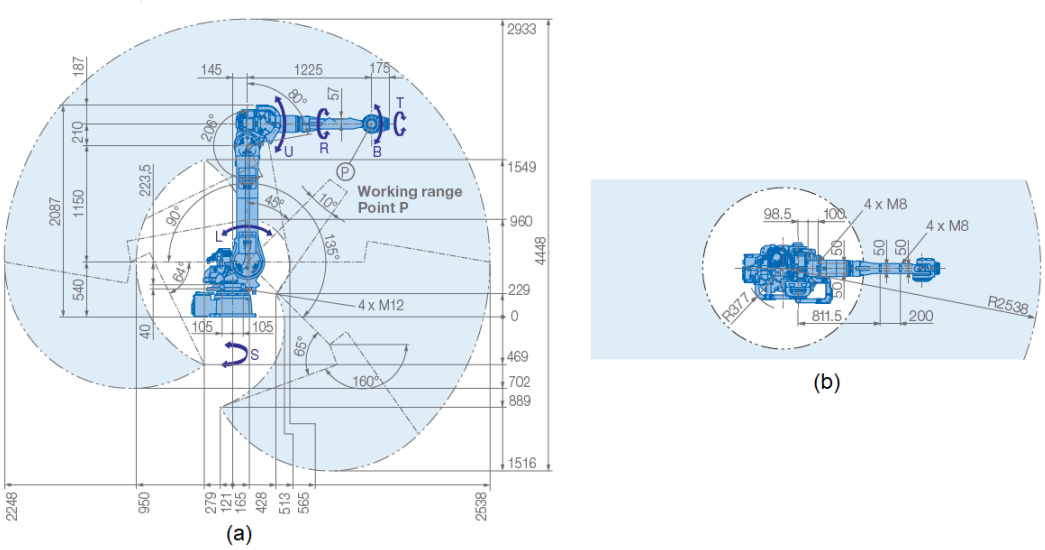


Fig. 2.4: Rango de movimientos y alcance físico del robot (*Fuente: Motoman MH50 Series Brochure*).

Para el correcto funcionamiento del brazo robótico, es necesario realizar la calibración de su posición inicial, denominada “Home Position” en inglés, en donde la posición cero establecida para cada uno de los servomotores coincida con la posición inicial del robot (Fig. 2.5). Esta posición viene calibrada de fábrica, sin embargo, por factores de transporte, instalación o uso, se recomienda verificar periódicamente [13].

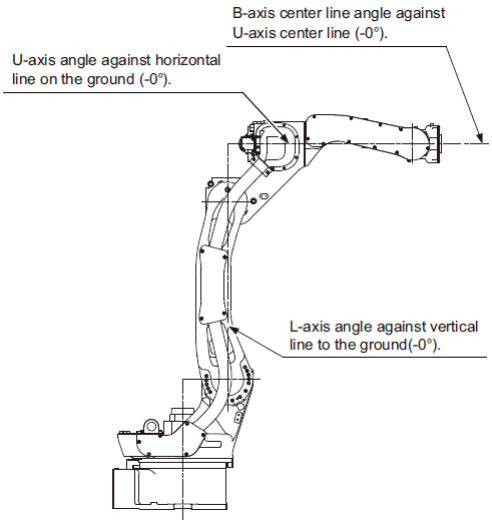


Fig. 2.5: Posición de fábrica del brazo robótico (*Fuente: DX200 INSTRUCTIONS*).

Cuando se ocupa una herramienta se puede establecer un nuevo punto de referencia llamado "Tool Centre Point" en inglés, en adelante TCP. Este punto representa el nuevo extremo del robot y define su propio sistema coordenado. En la Fig. 2.6 se muestran ejemplos de TCP para distintas herramientas y en la Fig. 2.7 un nuevo eje coordenado para la herramienta. El TCP puede ser configurado indicando manualmente las medidas de la herramienta o utilizando una función que permite determinarlo de forma automática guardando 5 posiciones distintas con el extremo de la herramienta en un punto fijo, como se muestra en la Fig. 2.8. El establecimiento de este punto, permitirá al robot calcular correctamente las trayectorias [13].

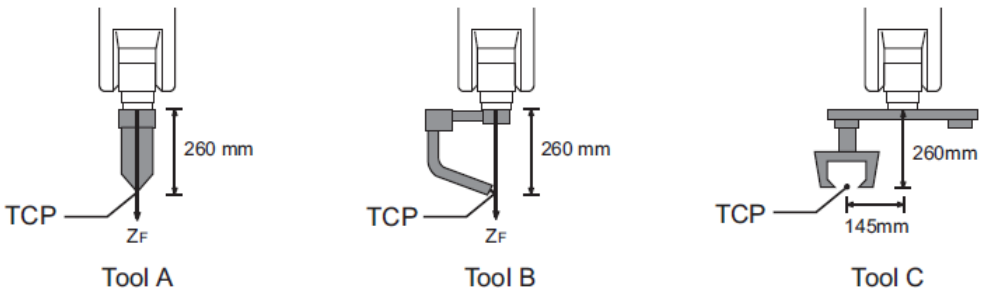


Fig. 2.6: Ejemplos de TCP (*Fuente: DX200 INSTRUCTIONS*).

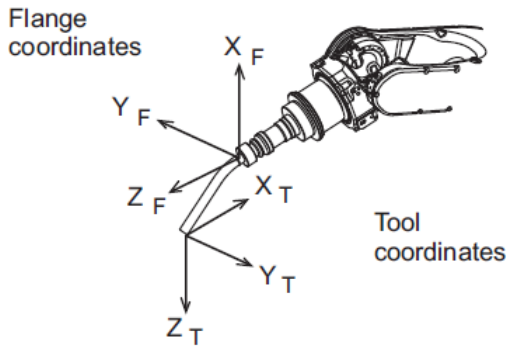


Fig. 2.7: Referencia coordenadas de la herramienta (*Fuente: DX200 INSTRUCTIONS*).

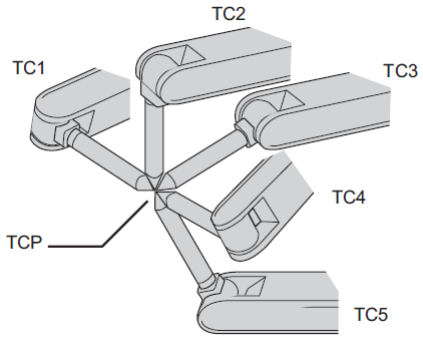


Fig. 2.8: Ejemplo para configurar el TCP de forma automática (*Fuente: DX200 INSTRUCTIONS*).

El brazo robótico trabaja con ejes coordenados XYZ predefinidos orientados con su base (Fig. 2.9 y 2.10). Sin embargo, cuando se opera en estaciones con distintas orientaciones, se pueden definir nuevos sistemas coordenados personalizados, denominados Coordenadas de Usuario, "User Frame" en inglés, en adelante UF. Estos sistemas pueden establecer nuevos orígenes, orientaciones e inclinaciones, permitiendo que el robot se acomode a distintas estaciones de trabajo [13]. En la Fig. 2.11 se muestran dos UF distintos para cada estación de trabajo y en la Fig. 2.12 un UF y los puntos necesarios para su creación. En el desarrollo del proyecto los UF se utilizan para establecer una relación espacial entre el área de trabajo y el campo de visión de la cámara y alinear los resultados de la detección con el movimiento.

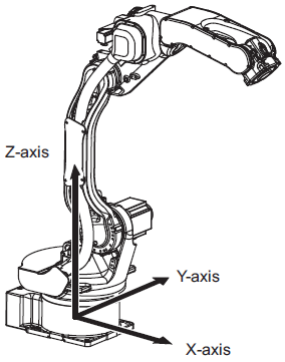


Fig. 2.9: Esquema de las coordenadas cartesianas (*Fuente: DX200 INSTRUCTIONS*).

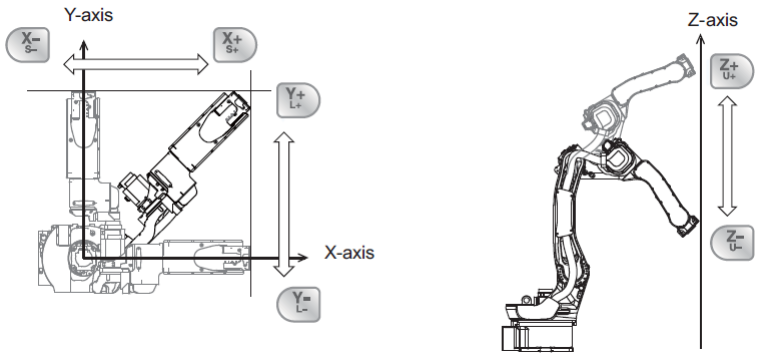


Fig. 2.10: Esquema de las coordenadas cartesianas (vista superior y lateral) (*Fuente: DX200 INSTRUCTIONS*).

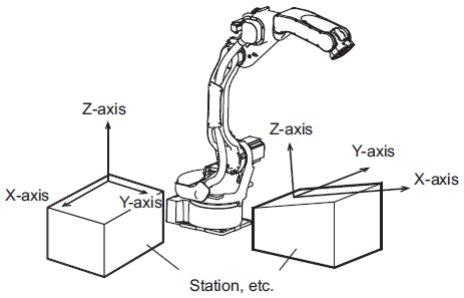


Fig. 2.11: Esquema de las coordenadas de usuario (UF) (*Fuente: DX200 INSTRUCTIONS*).

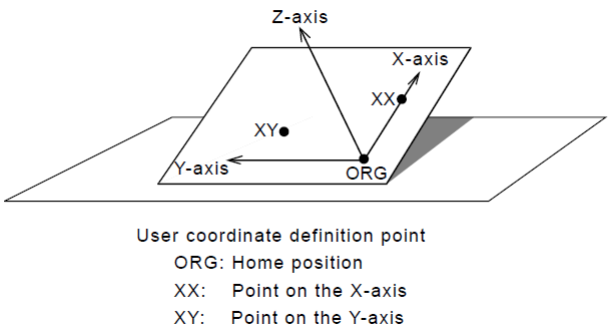


Fig. 2.12: Esquema de las coordenadas de usuario (*Fuente: DX200 INSTRUCTIONS*).

Otra característica importante del movimiento del robot es la interpolación, que define la trayectoria del extremo del robot entre dos puntos consecutivos. Para efectos del desarrollo del proyecto se utilizará la interpolación lineal, en la que la trayectoria entre el punto inicial y final es una línea recta. Su uso se elige ya que otros tipos de interpolaciones no definen una trayectoria conocida o tienen aplicaciones específicas no utilizadas en el trabajo.

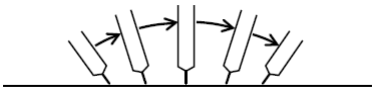


Fig. 2.13: Trayectoria de una interpolación lineal (*Fuente: DX200 INSTRUCTIONS*).

En resumen, el brazo robótico “Motoman MH50 II-35” es el elemento de actuación del sistema, que ejecuta los movimientos necesarios para manipular y trasladar objetos según la información obtenida por la cámara. La correcta calibración de la herramienta y de los sistemas coordenados de usuario (UF) son esenciales para el desarrollo de la aplicación que resuelve el objetivo del trabajo.

2.3. Cámara de Visión MotoSight2D

El sensor de visión utilizado es la cámara “MotoSight2D MS100” de “Yaskawa” (Fig. 2.14), diseñada para integrarse fácilmente a los controladores de la marca [14]. Su función es detectar objetos dentro de su campo visual, determinar sus coordenadas y entregar la información para que el robot ejecute el movimiento hacia el objetivo.



Fig. 2.14: Cámara Motosight2D.

Sus especificaciones técnicas se muestran en la Tabla 2.3 y una de sus características es que opera en escala de grises, lo que junto con su baja resolución, hacen que los objetos a detectar deben tener alto contraste con el fondo y con formas bien definidas, ya que los detalles no serán claros.

Item	**MS2D MS100**
Resolución	800x600
Velocidad del Procesador	1x
fotogramas por seg.	102

Tab. 2.3: Especificaciones cámara (*Fuente: Motosight2D Brochure*).

La cámara y las tareas de visión (las que se crean para que detecte un objeto determinado) pueden ser configuradas a través del programa computacional “In-Sight Explorer” ejecutado en un computador conectado al controlador o directamente desde el mando de control mediante la aplicación “MotoSight2D” [14]. Para el desarrollo del proyecto la cámara se instala en el extremo del brazo robótico, permitiendo definir distintas posiciones de captura y así ampliar el espacio de trabajo al no estar limitado al campo de visión de la cámara.

2.4. Controlador

El controlador (Fig. 2.15) es la unidad de procesamiento del sistema [13]. Se encarga de la comunicación entre el robot, la cámara, el mando de control y el computador. Además, posee terminales de entrada y salidas, de las cuales se hará uso para utilizar interrupciones. Su función principal es ejecutar los trabajos de programación, procesar las imágenes captadas por la cámara y calcular los movimientos del robot. Las configuraciones y los trabajos de programación son realizados a través del mando de control y el computador. Las conexiones fueron realizadas en el momento de instalación del sistema, por lo que, durante el desarrollo del proyecto, la única interacción directa con el controlador fue la conexión a una entrada de interrupción.



Fig. 2.15: Controlador DX200

2.5. Mando de control

El mando de control (Fig. 2.16) es la interfaz entre el usuario y el sistema robot-cámara. En él se ejecuta la manipulación manual del robot, se realizan los procesos de calibración y configuración, y la creación de los trabajos de programación para efectuar tareas automáticas [13]. En el mando también se ejecuta la aplicación MotoSight2D, la cual se encarga de mostrar las imágenes capturadas por la cámara, calibrarla, enseñar los patrones y la asignación de las posiciones detectadas a variables de programación [14]. En este trabajo el mando se utiliza para la manipulación manual, la calibración de la posición inicial, la creación del TCP, la definición de nuevos UF, la programación de secuencias automáticas y la utilización de la aplicación "MotoSight2D" para asignar los resultados de las capturas a variables utilizadas en la programación.



Fig. 2.16: Mando de Control

2.6. Computador

La conexión del computador al controlador “DX200” tiene como propósito la ejecución del programa computacional “In-Sight Explorer” [15], empleado para realizar la configuración y creación de los trabajos de visión. En este programa se ajusta la velocidad de captura, que afecta directamente al contraste de la imagen. También se realiza la calibración mediante el uso de una grilla de nueve puntos (Fig. 2.17), la que permite definir el sistema coordenado del campo de visión de la cámara y el correspondiente UF del robot. En la creación de los trabajos de visión se enseñan los patrones que se quieren reconocer, además de pruebas adicionales de verificación y tolerancia como el tamaño y la orientación. Estos trabajos son guardados en el controlador y ejecutados en la programación cuando se realiza la captura de una imagen.

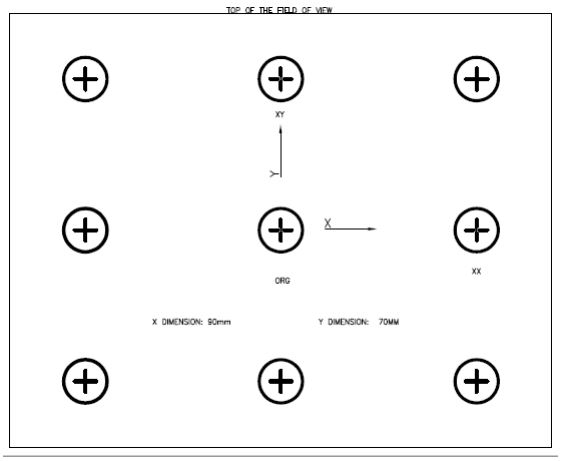


Fig. 2.17: Grilla de calibración de 9 puntos (*Fuente: MOTOSIGHT2D INSTRUCTIONS*).

2.7. Diagrama de funcionamiento

Una vez realizada la descripción general de cada componente del sistema, la Fig. 2.18 presenta un diagrama que indica las etapas necesarias para detectar un objeto dentro del área de trabajo, procesar la información obtenida por la cámara y ejecutar el movimiento del brazo robótico hacia el objeto.

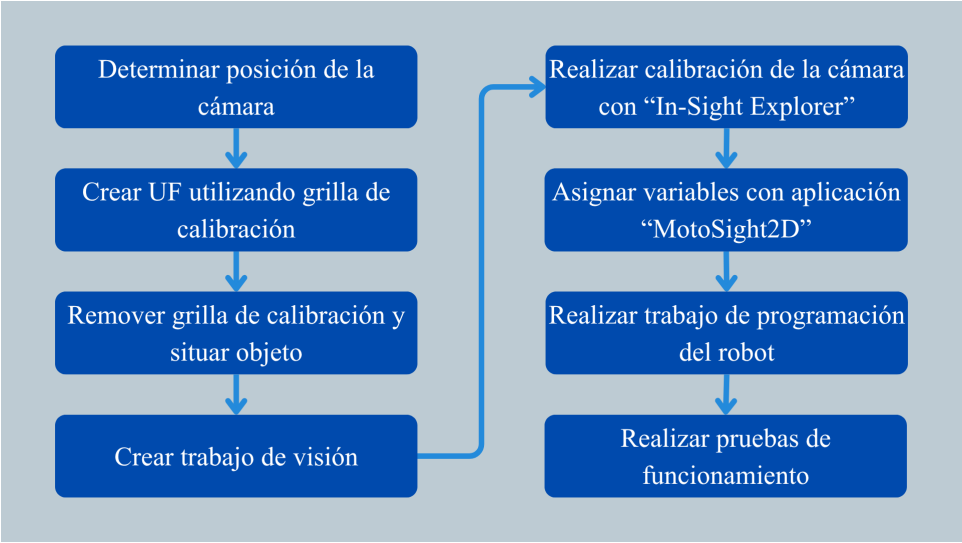


Fig. 2.18: Diagrama de funcionamiento para realizar una tarea de detección y manipulación de un objeto

El proceso comienza estableciendo la posición del robot donde la cámara realizará las capturas abarcando el campo de visión requerido. Posteriormente, se realiza la creación del UF y la calibración de la cámara utilizando la grilla de calibración para ambos en el mismo lugar, estableciendo una relación entre el robot y la cámara.

Una vez calibrado el sistema, se retira la grilla, se coloca el objeto y se crea un trabajo de visión en el programa Ïn-Sight Explorer"para poder detectarlo. Con el trabajo de visión realizado y guardado en el controlador, en la aplicación “MotoSight2D” se asignan las variables que se utilizarán para guardar los resultados de la detección. En el trabajo de programación del robot se sitúa en la posición calibrada para realizar las capturas de imagen y se ejecuta el trabajo de visión para la detección. Finalmente se ejecutan pruebas de funcionamiento para verificar la correcta interacción entre los elementos.

2.8. Conclusión del capítulo

En este capítulo se presentó la descripción general del sistema robot–cámara utilizado en el trabajo, junto con las características y funciones principales de sus componentes: el robot "Motoman MH50 II-35", la cámara "MotoSight2D", el controlador, el mando de control y el computador de configuración.

También se expuso el flujo de funcionamiento que integra todos los elementos, desde la captura de la imagen hasta la ejecución de los movimientos del manipulador. La comprensión de esta estructura resulta fundamental para el desarrollo de los capítulos siguientes, donde se abordarán las pruebas de funcionamiento y la implementación de las aplicaciones prácticas.

3. FUNCIONAMIENTO PRÁCTICO DEL BRAZO ROBÓTICO

Este capítulo tiene como objetivo realizar un análisis del funcionamiento práctico del sistema mediante la realización de pruebas experimentales, las que permitieron entender el comportamiento de cada uno de sus componentes y la relación entre ellos, además de las configuraciones necesarias para que funcionen correctamente. Estas pruebas consistieron en revisar el movimiento y programación del brazo robótico y la configuración del sistema de visión. Durante su desarrollo se detectó un error mecánico en el movimiento del robot, el cual se estudió para encontrar la causa y posibles medidas de compensación.

Como resultado del desarrollo de este capítulo, se establecieron los procedimientos necesarios para detectar un objeto y trasladarse hacia él, lo que constituye una base para el desarrollo de tareas más complejas posteriormente.

3.1. Pruebas de movimiento del brazo robótico

Las primeras pruebas se centraron en el análisis de movimiento del brazo robótico de forma manual con el mando de control (Fig. 3.1), con el objetivo de comprender el uso de los movimientos de las articulaciones y el uso del sistema de coordenadas cartesianas [13].



Fig. 3.1: Izquierda: Mando de control del brazo robótico. Derecha: Encerradas en rojo se encuentran las teclas para realizar los movimientos del robot.

El movimiento independiente de cada una de las articulaciones mostró un correcto funcionamiento y se verificó el rango de movimiento de cada una, limitados por elementos del entorno como paredes y suelo. En el uso de las coordenadas cartesianas se observaron desviaciones en la trayectoria en los distintos ejes y en el ángulo del extremo. Estos errores se presentan en las Fig. 3.2, 3.3 y 3.4. El error no se detecta visualmente en el movimiento de las articulaciones de forma individual.



Fig. 3.2: Error asociado al movimiento en el eje X del brazo robótico.



Fig. 3.3: Error asociado al movimiento en el eje Y del brazo robótico.



Fig. 3.4: Error asociado al movimiento en el eje Z del brazo robótico.

Posteriormente, se realizó una programación básica del robot que consta de una serie de movimientos entre puntos establecidos manualmente. En la Fig. 3.5 se presenta la programación realizada en el mando de control, en donde se indica el tipo de movimiento, la posición objetivo y la velocidad del brazo. Durante esta prueba, el robot llegaba a los puntos establecidos correctamente, pero para los movimientos lineales, se apreciaban leves desviaciones en la trayectoria.

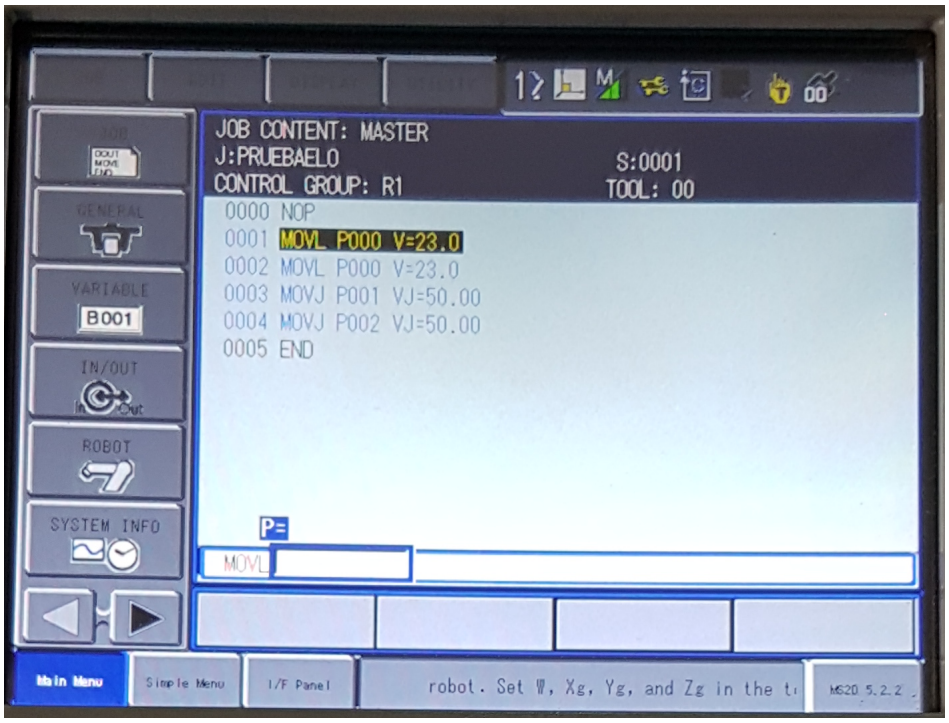


Fig. 3.5: Trabajo de programación básico

El análisis realizado a los resultados obtenidos en esta sección es que mediante el mando de control se pueden definir y guardar posiciones del robot, y estas se pueden utilizar en la programación para realizar tareas repetitivas entre esos puntos. Sin embargo, se encontró el error en las coordenadas cartesianas, que es el principal sistema que se ocupará en el desarrollo del trabajo. En secciones posteriores se encontrarán nuevos errores asociados al movimiento del robot y se hará un análisis para determinar la causa y posibles soluciones.

3.2. Calibración del TCP y creación de UF

Para trabajar correctamente con el sistema fue necesario realizar dos configuraciones principales: indicar el nuevo TCP de la herramienta, con el fin de calcular adecuadamente los movimientos según el nuevo extremo del brazo robótico, y crear coordenadas de usuario personalizadas (UF) para el trabajo conjunto con la cámara [13].

La herramienta utilizada en el desarrollo del proyecto fue un puntero, mostrado en la Fig. 3.6, lo que impidió la manipulación directa de objetos; no obstante, esta acción

se simuló mediante el contacto entre el extremo del puntero y el objeto. La calibración del TCP se realizó mediante el procedimiento automático, estableciendo cinco posiciones, como se muestra en la Fig. 3.7, manteniendo el extremo de la herramienta en un punto fijo. Posteriormente, la verificación del TCP se efectuó mediante movimientos manuales, en los cuales el extremo del puntero no debería variar su posición; sin embargo, se observó que los movimientos no se realizaban en torno al nuevo TCP. Al revisar las medidas calculadas automáticamente, presentadas en la Fig. 3.8, se constató que la dimensión correspondiente al eje Z no coincidía con la longitud real de la herramienta: mientras la medición indicaba 143,267 [mm], el largo real era de 300 [mm], lo que representa una diferencia significativa. Este error se asocia al problema identificado en la sección anterior.



Fig. 3.6: Fotografía de la herramienta utilizada.



Fig. 3.7: Las 5 posiciones utilizadas para configurar el TCP automáticamente.

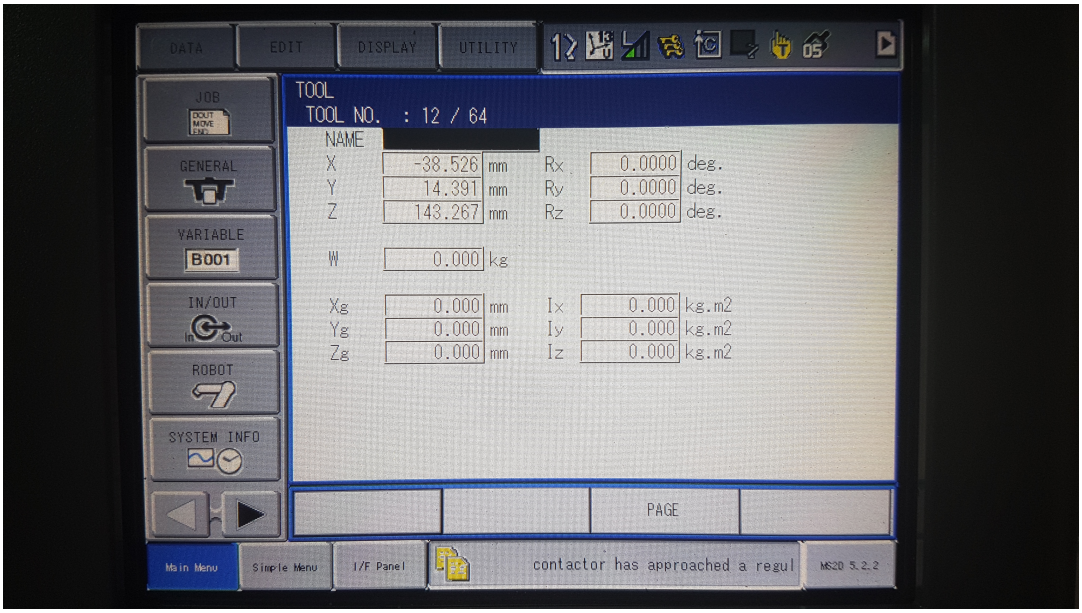


Fig. 3.8: Medidas de la herramienta obtenidas automáticamente.

Posteriormente, se creó un sistema de coordenadas de usuario (UF) utilizando la grilla de calibración de 9 puntos como referencia. En las Fig. 3.9, 3.10 y 3.11 se observan los puntos utilizados para crear el UF. En la verificación del funcionamiento del UF creado se comprobó la posición del nuevo origen y que la orientación de los nuevos ejes coinciden con los de la grilla. Sin embargo, se vuelve a observar el error producido en la sección anterior al realizar movimientos en los ejes predeterminados del robot.

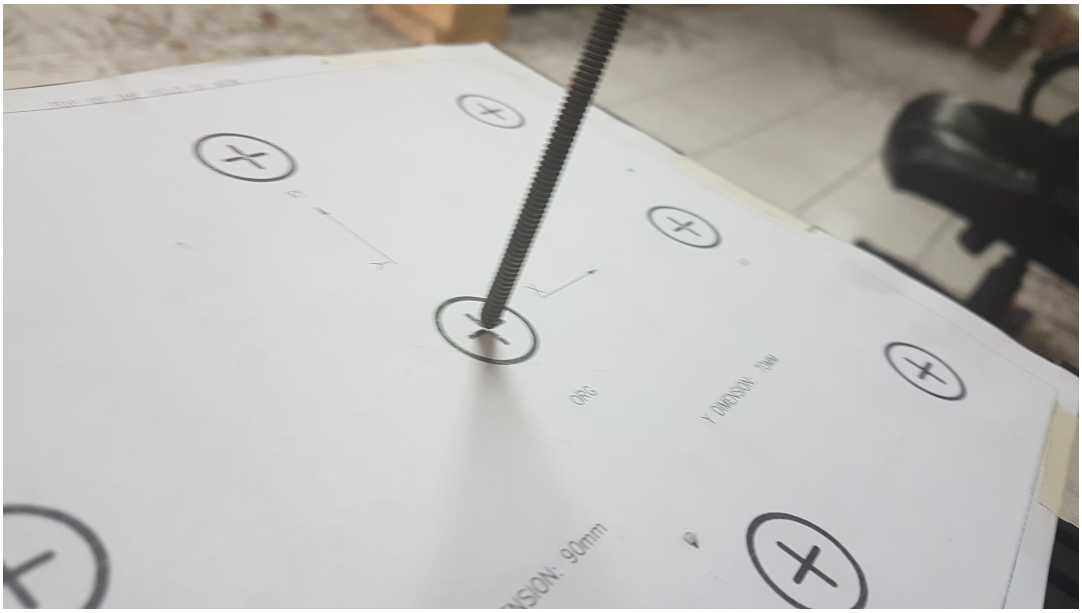


Fig. 3.9: Extremo de la herramienta en el punto ORG para la creación de un nuevo UF.

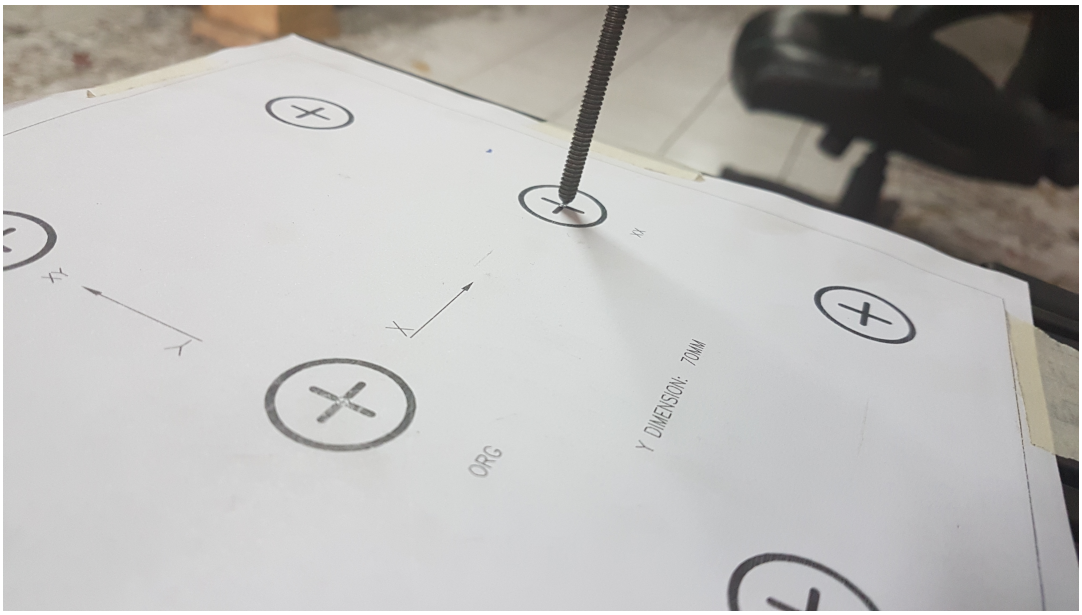


Fig. 3.10: Extremo de la herramienta en el punto XX para la creación de un nuevo UF.

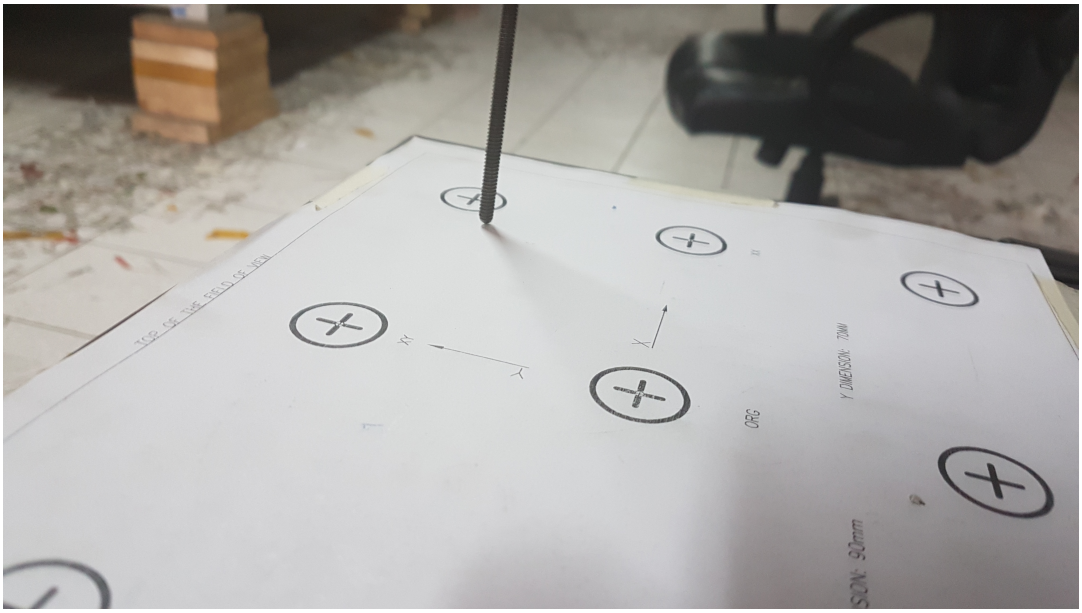


Fig. 3.11: Extremo de la herramienta en el punto XY para la creación de un nuevo UF.

En la siguiente sección se realizará un análisis en profundidad del error, para entender que lo causa, posibles soluciones o maneras de cómo compensarlo, ya que estas configuraciones no permiten realizar de forma correcta el desarrollo del proyecto.

3.3. Análisis del Error

En las secciones anteriores se evidenciaron errores correspondientes al movimiento y a la calibración del TCP. Para dimensionar la influencia del error se realizó un análisis comparativo utilizando un sistema de coordenadas de usuario (UF). El mando de control indica la distancia recorrida desde el origen del UF y se compara con la medición real del movimiento para los ejes X e Y. Los resultados obtenidos se presentan en la Tabla 3.1 y su regresión lineal en la Fig. 3.12. El análisis muestra que la distancia indicada por el mando de control representa el 97 % en el eje Y y solo un 78 % en el eje X, evidenciando una descalibración significativa del sistema.

Distancia Medida [mm]	Distancia Indicada Eje Y [mm]	Distancia Indicada Eje X [mm]
-20	-19.889	-15.286
-40	-39.614	-30.903
-60	-59.172	-46.396
-80	-78.780	-62.178
-100	-97.406	-77.558
20	19.484	15.915
40	38.830	31.367
60	58.505	48.681
80	78.033	62.127
100	97.267	77.523

Tab. 3.1: Mediciones de los movimientos físicamente y los entregados por el mando de control.

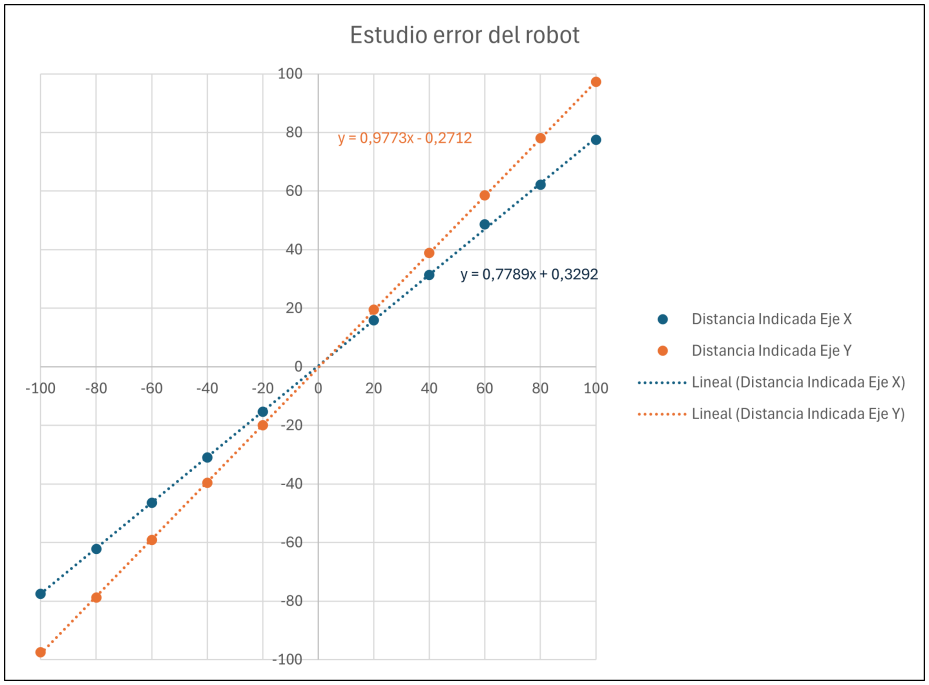


Fig. 3.12: Gráfico de los datos de la Tabla 3.1.

A pesar de realizar nuevamente la calibración de la posición inicial del brazo robótico, el error se siguió manifestando. Se procedió a contactar al soporte técnico de "Yaskawa", al cual se envió el archivo "CMOS.BIN" generado por el controlador, que contiene todas las configuraciones y calibraciones realizadas, para su verificación. El soporte informó que la ejecución del archivo en un robot de prueba no presentaba los errores descritos, lo que descartó un fallo asociado al "software" o "firmware" del controlador.

El análisis cuantitativo y visual sugiere que el error se presenta con mayor notoriedad en el eje X. Sin embargo, para el eje Y el error se va incrementando progresivamente a medida que se aleja de la base del robot, por lo que se procedió a observar el movimiento de las articulaciones.

Finalmente, se establece que la articulación L es la responsable de generar el error, ya que su movimiento influye directamente en el eje X y de forma creciente en el eje Y. Dado que está descartado un problema en el controlador, se concluye que el error corresponde a un desperfecto mecánico, probablemente asociado al encoder del servo-motor de la articulación. La reparación del fallo no está considerada en el desarrollo del proyecto, por lo que se continuará considerando la existencia del error. La detección del error y de su origen permite entender nuevas limitaciones del sistema para el desarrollo de una aplicación de automatización con la cámara, por lo que se establecerán nuevas estrategias para resolver etapas futuras del proyecto.

3.4. Cámara MotoSight2D e In-Sight Explorer

Realizadas las pruebas con el brazo robótico, se procedió a trabajar con la cámara "MotoSight2D" y el programa computacional "In-Sight Explorer" [15]. El objetivo de esta etapa es generar un trabajo de visión realizando la calibración de la cámara y el entrenamiento del patrón a identificar.

Como punto inicial, la cámara se encuentra instalada en el extremo robótico y se establece una posición que detecte el campo de visión deseado. La calibración se realiza mediante la configuración de la velocidad de captura de la cámara, para establecer el contraste de la imagen, y el uso de la grilla de calibración de 9 puntos. En la Fig. 3.13 se ve la detección de los 9 puntos y encerrado en rojo el eje coordenado generado.

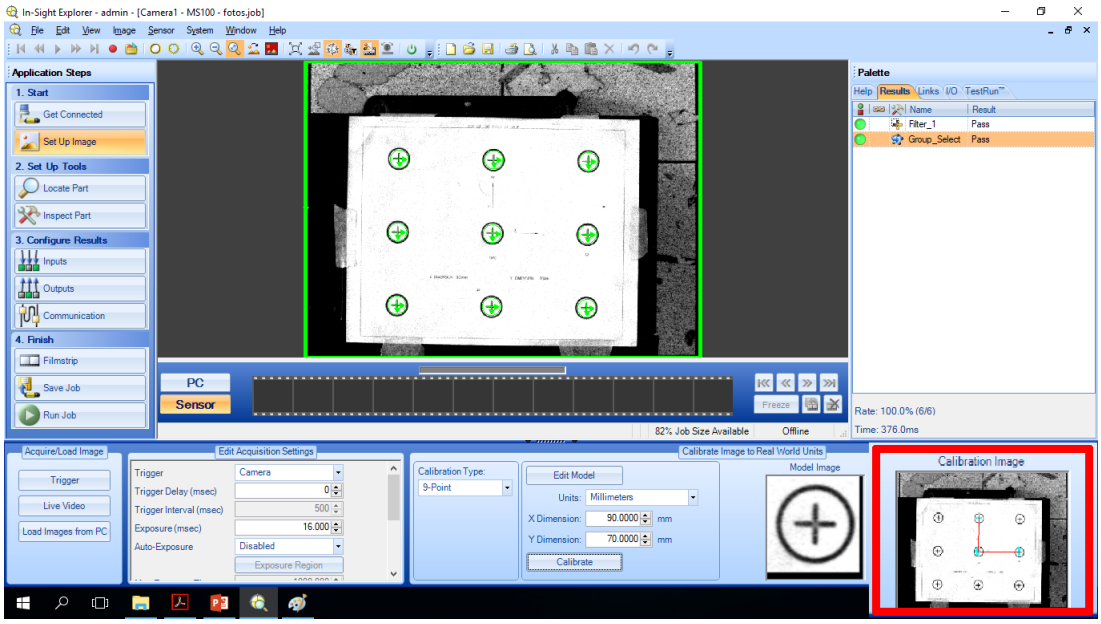


Fig. 3.13: Pantalla de calibración realizada. Se muestran los ejes configurados encerrado en rojo.

Una vez obtenidos los ejes coordenados del campo de visión, se indica el patrón del objeto que se quiere manipular. Para el proyecto se ocupará como objeto una lámina plástica con un patrón similar a un código QR. En la Fig. 3.14 se muestra cómo se debe indicar al programa cuál es el patrón que debe reconocer y en la Fig. 3.15 se ven ejemplos de reconocimiento del patrón además de las coordenadas de su centro y el ángulo con respecto al patrón original.

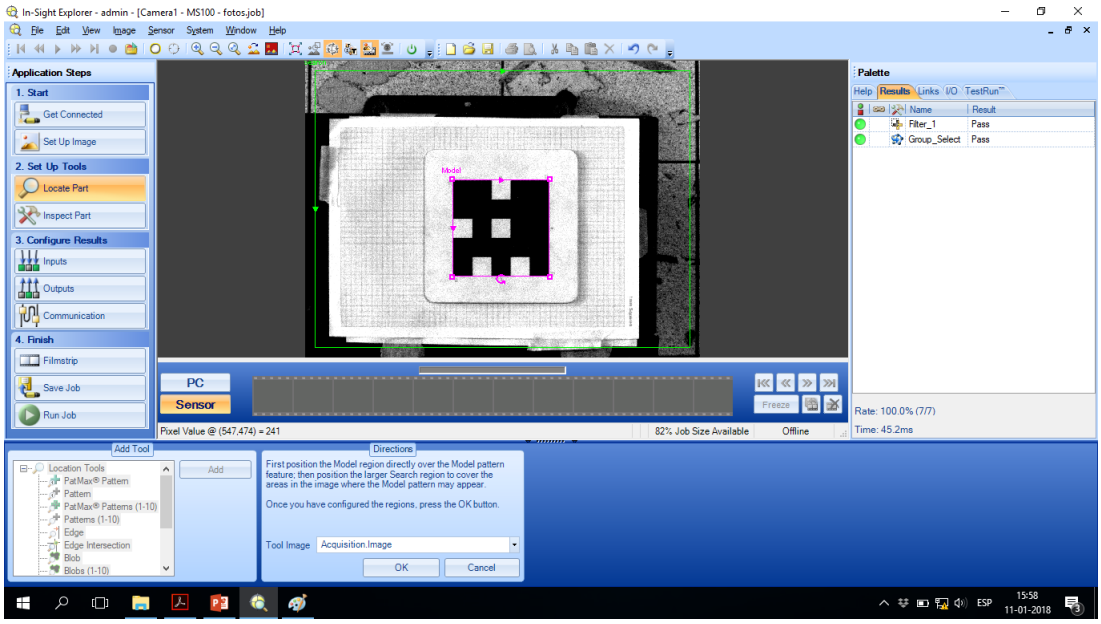


Fig. 3.14: Se indica la zona dentro de la captura en que se encuentra el patrón.

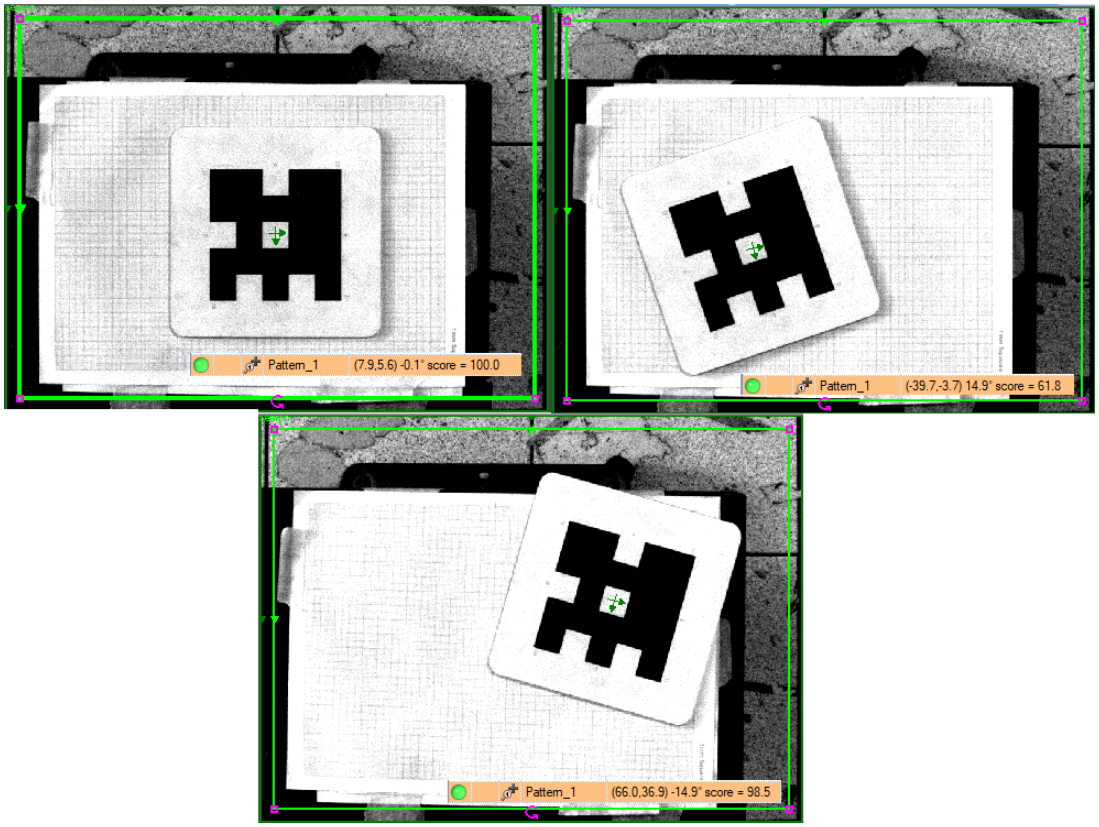


Fig. 3.15: Se indica la zona dentro de la captura en que se encuentra el patrón.

Para verificar con mayor precisión el objeto detectado, se pueden realizar pruebas extras. Es posible determinar el tamaño del objeto y establecer tolerancias para determinar si el objeto debe o no ser detectado. En la Fig. 3.16 se muestran las mediciones realizadas al objeto, de esta manera el trabajo de visión no solo se enfoca en detectar el patrón, sino también de su tamaño.

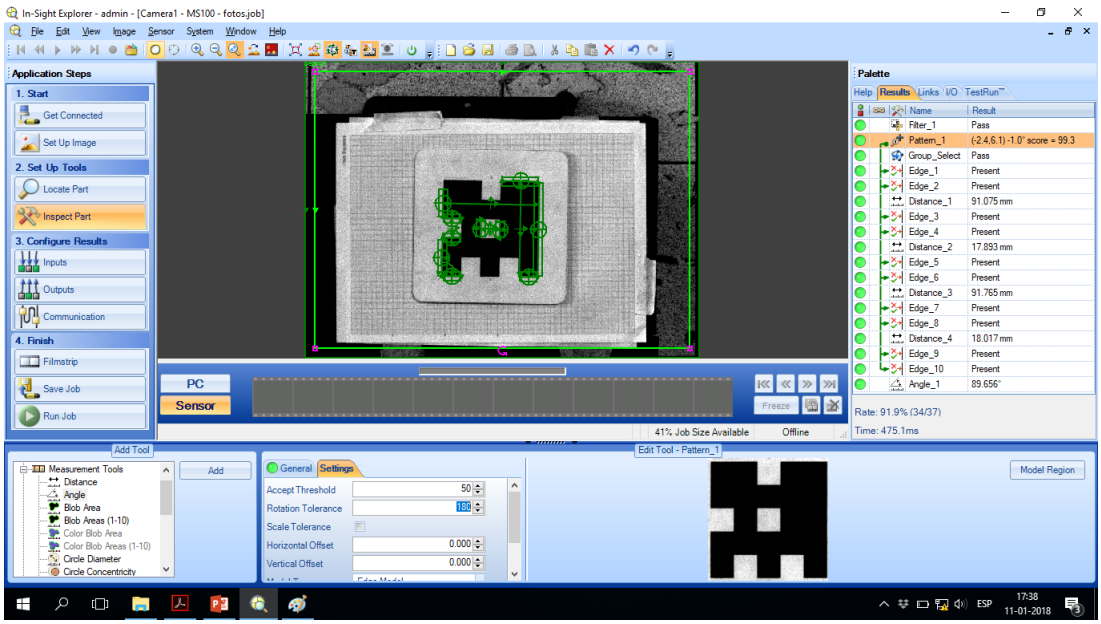


Fig. 3.16: Patrón es verificado mediante sus medidas.

Con las pruebas realizadas, se pudo comprobar el funcionamiento de la detección de la cámara. Se pudo observar que el sistema de visión es capaz de detectar el patrón enseñado con gran precisión y que además se puede configurar para afinar la detección a parámetros medibles. Al completar una tarea de visión, esta se guarda en el controlador "DX200" y puede ser usada para la detección sin la necesidad del "software".

3.5. Aplicación MotoSight2D

La aplicación "MotoSight2D" se utilizó como fase intermedia entre los trabajos de visión y su uso en las tareas de programación. Su principal función es guardar los resultados de la detección en variables que el robot utiliza para moverse hacia el objeto [15]. En la Fig. 3.17 se muestra la interfaz principal de la aplicación con la tarea de

visión realizada en la sección anterior, donde se puede observar la detección del patrón y su centro. La Fig. 3.18 presenta la ventana de configuración de las variables que guardan los datos obtenidos del reconocimiento del patrón.

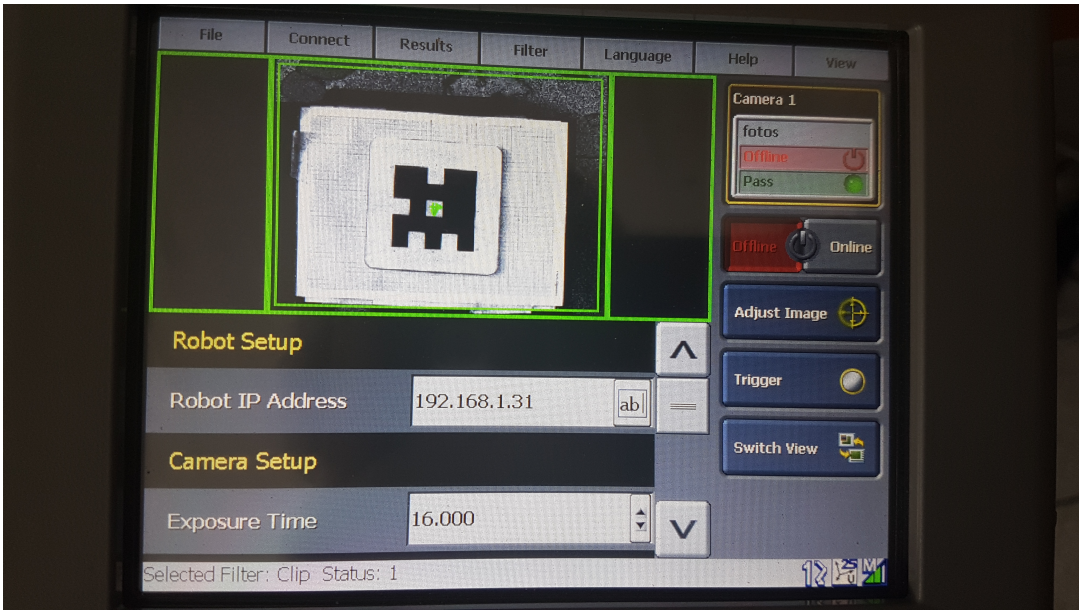


Fig. 3.17: Interfaz aplicación Motosight2D.

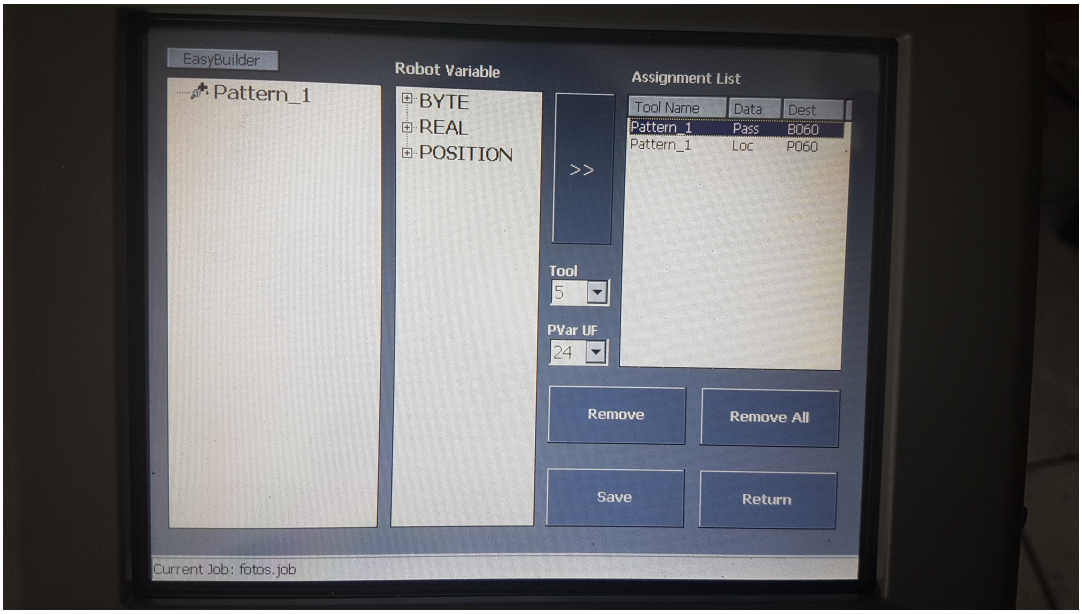


Fig. 3.18: Interfaz pestaña “Result”.

El desarrollo de esta acción permite que el brazo robótico utilice las variables para

moverse a la ubicación del objeto detectado por la cámara. Para esto es necesario realizar correctamente la creación del UF con la grilla de calibración en la misma posición utilizada en la calibración de la cámara.

3.6. Creación de Trabajo de Programación intregrando la Detección de la Cámara

Una vez realizadas las pruebas de funcionamiento de cada uno de los elementos, se procede a realizar su integración. El objetivo es detectar la posición de la lámina con el patrón enseñado y llevar el extremo del robot a su centro.

En la Fig. 3.19 se muestra el trabajo de programación que realiza la tarea planteada [13]. Primero, mueve al robot a la posición en que la cámara fue calibrada. Luego, mediante la macro "INSPECT" se realiza la captura de imagen y se ejecuta la tarea de visión para detectar el patrón [15]. Si reconoce el objeto, las coordenadas son guardadas en las variables configuradas en la aplicación "MotoSight2D". Por último, el robot se mueve en el UF correspondiente a la distancia indicada en las variables, llegando al centro del patrón.

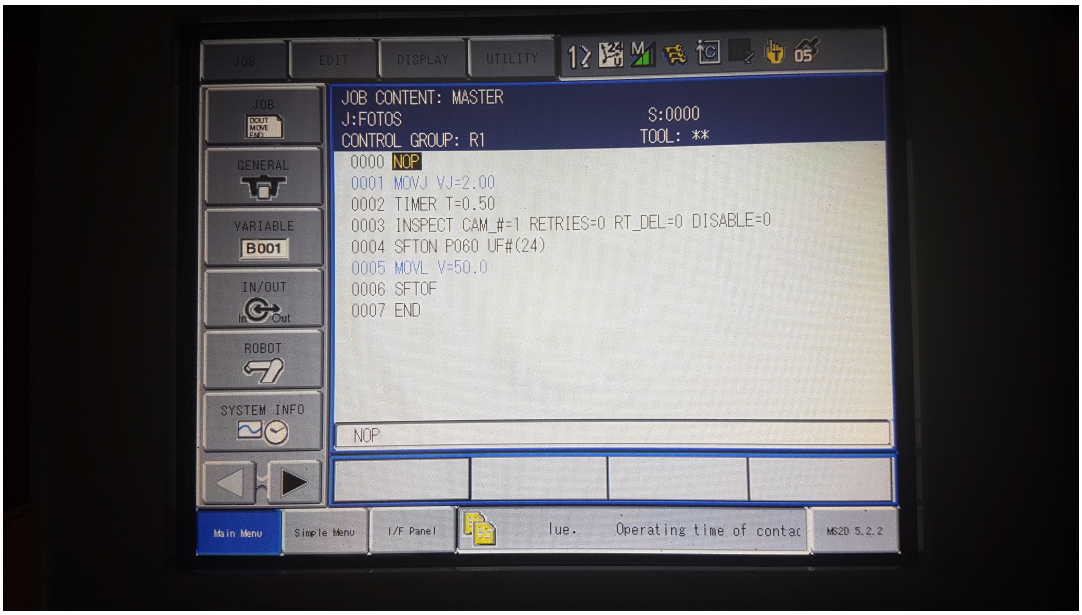


Fig. 3.19: Programación básica para utilizar la cámara en una aplicación con el robot.

Los resultados obtenidos para distintas posiciones del patrón se muestran en la Fig. 3.20. En ella se observa que el puntero del robot acierta con precisión sólo en la primera imagen. Esto es porque coincide con el origen del UF. En cambio, para las demás posiciones, logra detectar el posicionamiento, pero no llega con precisión. La razón de este comportamiento se explica con el error en la articulación L. Se observa que para una posición desplazada en el eje Y, el puntero se mantiene dentro de la zona blanca central del patrón, sin embargo, para desplazamientos en el eje X el error es mayor.

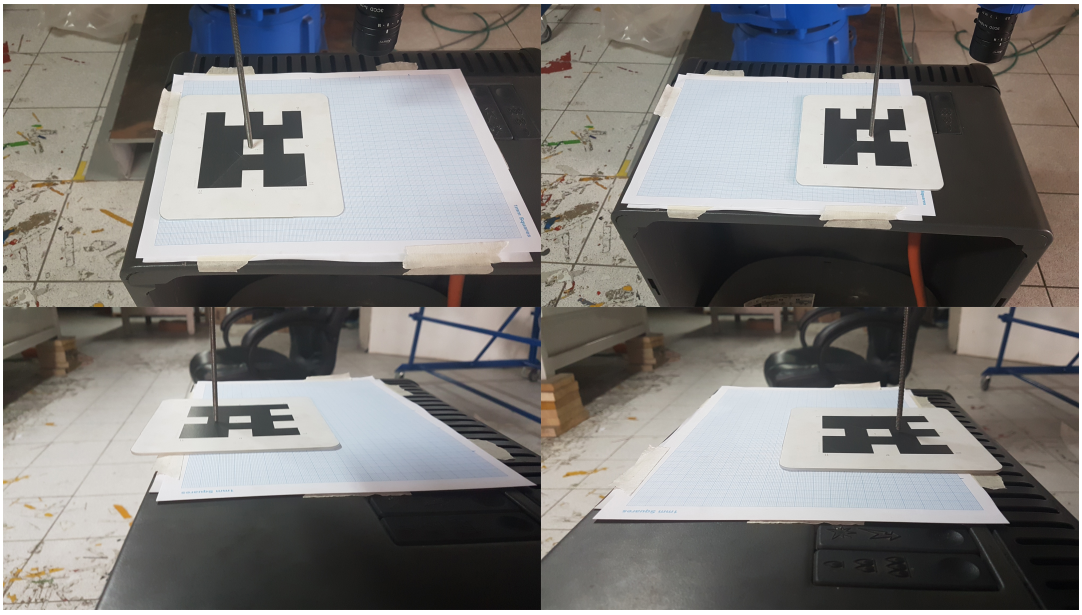


Fig. 3.20: Resultados del movimiento del robot indicado por la tarea de la cámara.

En base a estas pruebas, se puede concluir que la cámara detecta correctamente la posición del objeto, sin embargo, no es capaz de llegar correctamente al centro de la figura. Para el desarrollo del problema de automatización posterior, no se tendrá en cuenta si el robot falla por poco y se dará énfasis a la detección y movimiento hacia el objeto, considerando que si el robot se arregla este error no debería suceder.

3.7. Conclusión del capítulo

Las pruebas realizadas en este capítulo permitieron comprender y evaluar el funcionamiento del sistema y de cada uno de sus componentes. En primer lugar, se analizó

el comportamiento del robot frente a una tarea básica de programación, corroborando la importancia de la correcta definición de coordenadas de usuario (UF) para el trabajo conjunto con la cámara de visión. Por su parte, el desempeño de la cámara demostró su capacidad para detectar y establecer con alta precisión la ubicación del objeto, así como su integración directa y sencilla a la programación del robot mediante el programa computacional, la aplicación del mando de control y el uso del macro que permite capturar y ejecutar la tarea de visión.

El hallazgo de un error mecánico en una de las articulaciones puso de manifiesto la necesidad de revisar de forma sistemática cada uno de los componentes del sistema para verificar su correcto funcionamiento. En este caso, dicho desperfecto afecta directamente tanto al cálculo automático del TCP como a la ejecución de los movimientos del robot, generando errores de precisión. No obstante, esta limitación motiva el planteamiento de una estrategia que permita mitigar dichos errores y continuar con el desarrollo de la solución propuesta.

El conjunto de las pruebas y la creación del trabajo de programación incorporando el sistema de visión establece la base para poder implementar la solución al problema de automatización que se describe en el siguiente capítulo.

4. APLICACIÓN DEL FUNCIONAMIENTO DEL ROBOT A UN PROBLEMA DE AUTOMATIZACIÓN

El propósito de este capítulo es aplicar el conocimiento adquirido en los capítulos anteriores, sobre el sistema robot-cámara, para resolver un problema clásico de automatización industrial, con el fin de evaluar el desempeño general del sistema, la detección de patrones y la precisión de los movimientos. La tarea a realizar consiste en controlar el movimiento del brazo robótico mediante la información entregada por la cámara y manipular objetos en posiciones arbitrarias y trasladarlos a una posición objetivo. Cabe destacar que, como se indicó en el capítulo anterior, la herramienta disponible no permite manipular objetos. No obstante, se considera que el contacto del puntero del robot con el objeto, simula su manipulación. En las secciones siguientes se presenta la definición del problema, el análisis y propuesta de solución, su implementación y resultados obtenidos.

4.1. Definición del Problema de Automatización

Un problema de automatización industrial común es la manipulación de piezas o la paletización de objetos [16]. Estas tareas requieren que los objetos se encuentren en posiciones previamente establecidas, ya que el robot no dispone de la capacidad de identificar por sí mismo la ubicación de las piezas [4]. La integración de una cámara de visión permite superar esta limitación [17], otorgando la capacidad de reconocer la posición de objetos dentro de un espacio de trabajo y adaptar el movimiento del robot de acuerdo a lo detectado. El problema consiste en desarrollar un proceso con el sistema robot-cámara que permita:

1. Detectar un objeto y mover el puntero hacia él

2. Detectar el patrón de la posición objetivo y mover el puntero hacia esa ubicación.

La ejecución de la tarea debe realizarse de manera secuencial, primero debe encontrar el objeto y luego el patrón objetivo. Además, los movimientos se deben determinar únicamente por las coordenadas de usuario (UF) y la información de la detección, a excepción de las posiciones donde la cámara está configurada para realizar las detecciones.

4.2. Análisis y propuesta de solución

En esta sección se analiza la tarea de automatización a desarrollar y se propone una solución que permita trabajar con el sistema robot-cámara considerando las limitaciones mecánicas detectadas en el capítulo anterior. El objetivo principal es definir una estrategia que asegure una detección confiable de los patrones y que, al mismo tiempo, reduzca el efecto del error asociado a la articulación L durante los movimientos del robot.

Para el desarrollo de la tarea se definió un espacio de trabajo rectangular de 2 × 1 [m], delimitado mediante una cinta verde (Fig. 4.1), dentro del cual el sistema debe localizar tanto el objeto a manipular como la posición objetivo. A partir de este espacio, el primer desafío consiste en determinar una posición de la cámara que permita cubrir completamente la zona de trabajo, considerando restricciones como el rango de movimiento del brazo robótico, la resolución de la imagen capturada y la influencia del error mecánico presente en el sistema.



Fig. 4.1: Espacio de trabajo delimitado por cinta verde

Una alternativa para ampliar el campo de visión consiste en alejar la cámara del plano de trabajo o utilizar un lente con mayor ángulo de visión. Sin embargo, alejar la cámara implica posicionar el robot a mayor altura, lo que se encuentra limitado por su rango de operación y, además, reduce la resolución de los objetos detectados, afectando la precisión del reconocimiento. Por otro lado, el uso de lentes con mayor amplitud introduce distorsiones en los bordes de la imagen, lo que no representa una mejora efectiva del campo de visión útil para la aplicación. Debido a estas limitaciones, dichas alternativas no resultan adecuadas para resolver el problema planteado.

Como solución, se propone dividir el espacio de trabajo en múltiples cuadrantes parcialmente superpuestos. Esta estrategia permite mantener una adecuada resolución de imagen en cada captura, mejorar la precisión en la detección de objetos y patrones, y reducir el impacto del error mecánico al trabajar con sistemas de coordenadas de usuario (UF) de menor extensión. La superposición entre cuadrantes se define de manera que sea, al menos, equivalente al tamaño del objeto a detectar, garantizando que este sea identificado independientemente de su posición dentro del espacio de trabajo.

En la Fig. 4.2 se presenta el estudio de la disposición de los cuadrantes que evidencia que una configuración contigua genera zonas ciegas donde el objeto podría no ser detectado (1). En cambio, la utilización de cuadrantes superpuestos (2 y 3) asegura una cobertura completa del espacio de trabajo. A partir de este análisis, se determinó que es necesario definir diez cuadrantes para cubrir la totalidad de la zona delimitada (Fig. 4.3), asignando a cada uno su respectiva calibración de cámara y sistema de coordenadas de usuario.

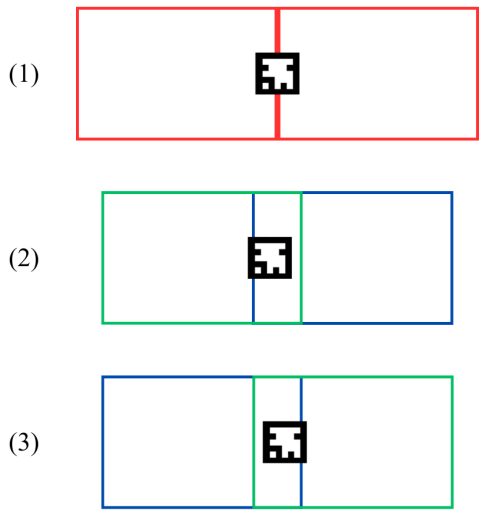


Fig. 4.2: (1): Esquema de cuadrantes con punto ciego. (2) y (3): Esquema de cuadrantes superpuestos.

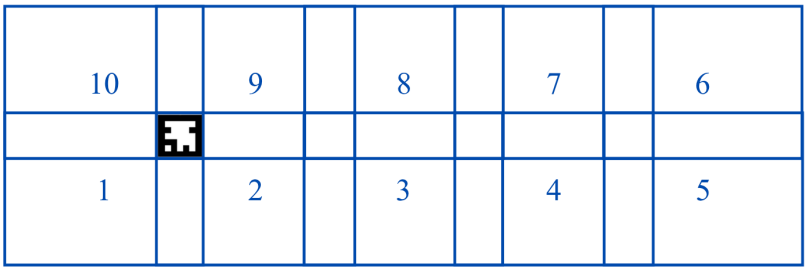


Fig. 4.3: Esquema de diez cuadrantes utilizados en el trabajo.

La adopción de esta estrategia requiere implementar un procedimiento secuencial que permita recorrer cada cuadrante en busca de los patrones definidos. Para ello, se diseñó la máquina de estados de la Fig. 4.4 que organiza el proceso de detección del objeto y de la posición objetivo, así como los movimientos del robot hacia dichas ubicaciones. Esta máquina de estados constituye la base lógica de la solución y define el flujo de ejecución que será implementado en la programación del sistema.

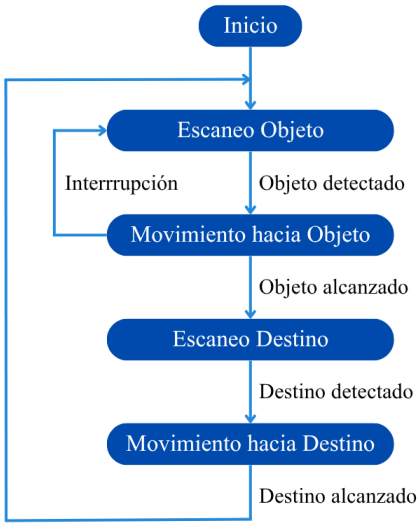


Fig. 4.4: Máquina de estados de la solución al problema de automatización.

En conjunto, la división del espacio de trabajo en cuadrantes, junto con la calibración independiente de la cámara y los sistemas de coordenadas de usuario para cada uno, permite disminuir la influencia del error mecánico del robot y establecer una metodología robusta para la resolución del problema de automatización planteado. Esta estrategia define la base sobre la cual se desarrolla la implementación de la solución descrita en la sección siguiente.

4.3. Implementación de la solución

La implementación de la solución consiste en desarrollar las tareas de visión y los trabajos de programación necesarios para ejecutar la máquina de estados definida en la sección anterior. El objetivo es integrar el sistema robot-cámara de manera que permita detectar el objeto y la posición objetivo dentro del espacio de trabajo, y comandar los movimientos del brazo robótico en función de la información entregada por la cámara.

Debido al error identificado en la articulación L, cada cuadrante del espacio de trabajo debe configurarse de forma independiente. En un sistema ideal, bastaría con definir una única tarea de visión para el objeto, otra para la posición objetivo y un solo sistema de coordenadas de usuario, del cual podrían derivarse los movimientos en todos

los cuadrantes. Sin embargo, el comportamiento anómalo del robot impide replicar con precisión estas configuraciones en distintas zonas del espacio de trabajo. Por esta razón, fue necesario definir diez trabajos de visión independientes —uno por cada cuadrante— tanto para la detección del objeto como para la ubicación objetivo, así como sus correspondientes sistemas de coordenadas de usuario.

Para el desarrollo de la solución se utilizaron dos patrones distintos que se muestran en la Fig. 4.5. El primero (1) corresponde al patrón que representa el objeto a manipular, mientras que el segundo (2) identifica la posición objetivo a la cual debe ser trasladado. Ambos patrones fueron entrenados de manera independiente para cada cuadrante, asegurando una detección consistente dentro del campo de visión definido en cada caso.

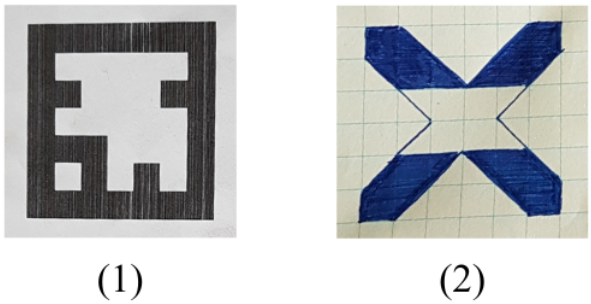


Fig. 4.5: (1): Patrón que representa al objeto. (2) Patrón que representa a la ubicación objetivo

La implementación comienza con la creación de las tareas de visión y de los sistemas de coordenadas de usuario. Para cada cuadrante, se posiciona la grilla de calibración y se ajusta la cámara de modo que su campo de visión abarque completamente dicha grilla. A continuación, se crea el sistema de coordenadas de usuario correspondiente y, mediante el software In-Sight Explorer, se realiza la calibración de la cámara y el entrenamiento del patrón a detectar. Este procedimiento se repite para cada cuadrante y para ambos patrones definidos. Finalmente, utilizando la aplicación MotoSight2D del mando de control, se asignan las variables que almacenan los resultados de la detección, las cuales serán utilizadas posteriormente en la programación del robot.

La programación de la solución se desarrolló íntegramente en el mando de control del robot, siguiendo la lógica de la máquina de estados definida previamente. El sistema

se implementó de manera modular, de modo que cada estado de la máquina esté representado por un bloque de código independiente. En total, se desarrollaron tres trabajos principales: un trabajo principal encargado del control del flujo, un trabajo de escaneo del espacio de trabajo y un conjunto de trabajos de movimiento asociados a cada cuadrante.

El trabajo de programación principal (Fig. 4.6 y 4.7) cumple la función de coordinar la ejecución de la máquina de estados. Este controla la transición entre los estados de escaneo y de movimiento, y determina si la búsqueda y el desplazamiento deben realizarse con respecto al objeto o a la posición objetivo. Para ello, el programa llama al trabajo de escaneo y, una vez detectado el patrón en un cuadrante específico, llama al trabajo de movimiento correspondiente a dicho cuadrante.

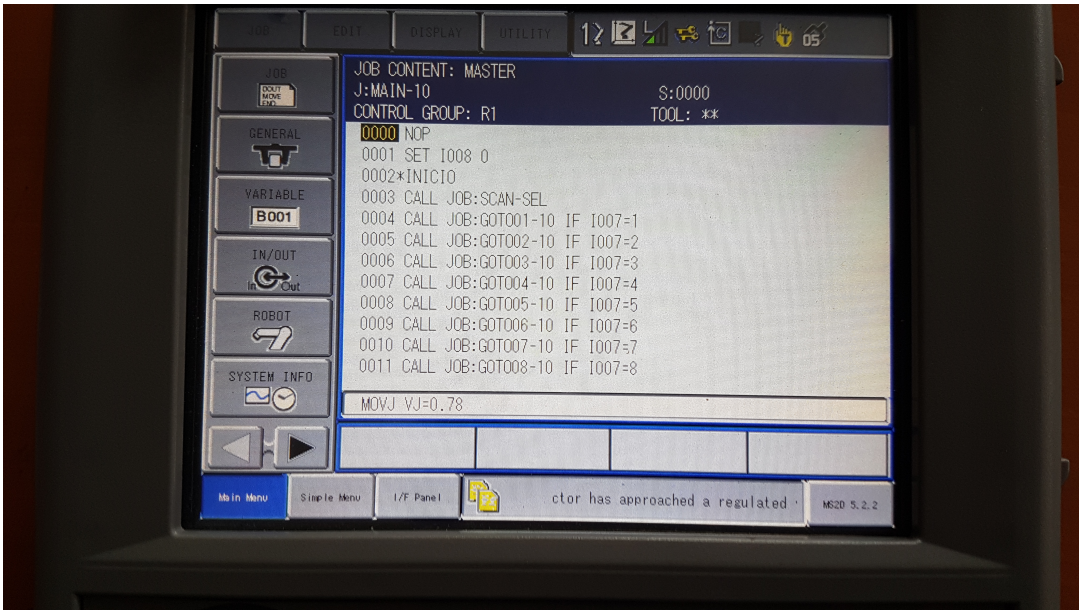


Fig. 4.6: Programación trabajo principal (parte 1)



Fig. 4.7: Programación trabajo principal (parte 2)

El trabajo de escaneo (Fig. 4.8 y 4.9) se encarga de recorrer secuencialmente los cuadrantes definidos, seleccionando el trabajo de visión asociado a cada uno. En cada iteración, la cámara MotoSight2D realiza la captura y el análisis de la imagen, verificando la presencia del patrón correspondiente. Si el patrón es detectado en alguno de los cuadrantes, las coordenadas obtenidas se almacenan en las variables definidas y el flujo del programa retorna al trabajo principal. En caso contrario, el sistema continúa escaneando los cuadrantes restantes hasta completar el recorrido.

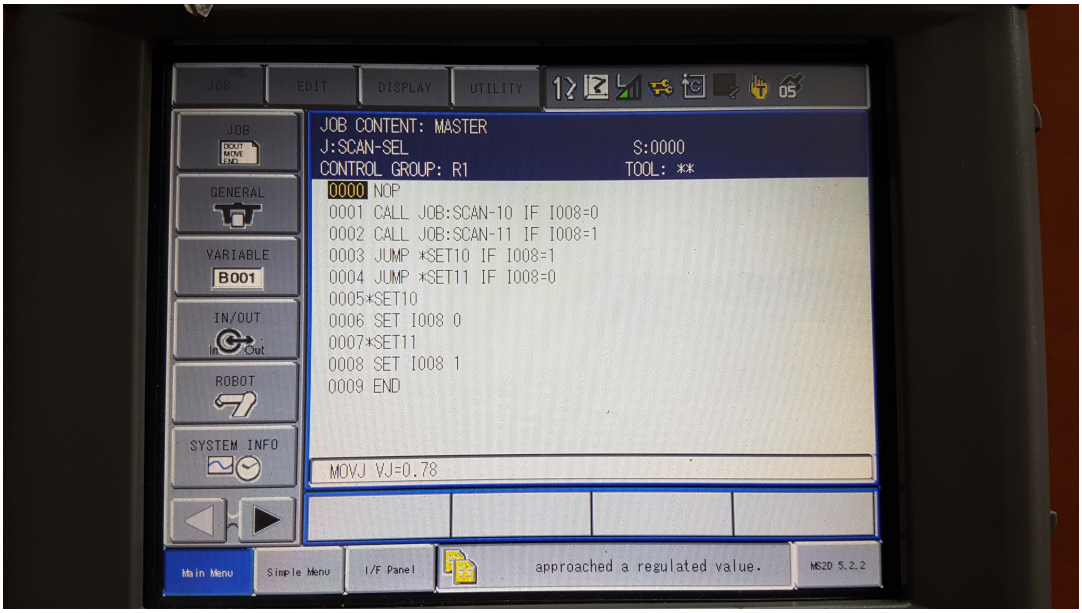


Fig. 4.8: Programación del trabajo que selecciona el escaneo.

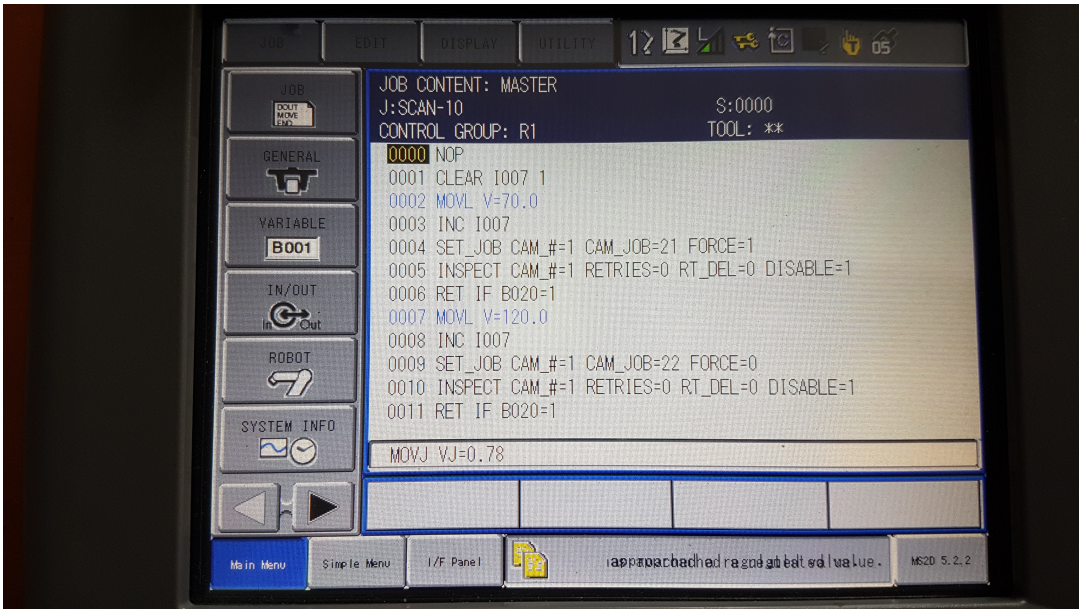


Fig. 4.9: Programación de escaneo

Una vez identificada la posición del patrón y el cuadrante en el que se encuentra, se ejecuta el trabajo de movimiento (Fig. 4.10) asociado a dicho cuadrante. Cada uno de estos trabajos comparte la misma estructura de programación, diferenciándose únicamente en el sistema de coordenadas de usuario utilizado. El movimiento del robot se

realiza tomando como referencia el origen del UF y desplazándose según las coordenadas obtenidas en la detección. Al finalizar el movimiento, el flujo del programa retorna al trabajo principal, permitiendo continuar con el ciclo de operación.

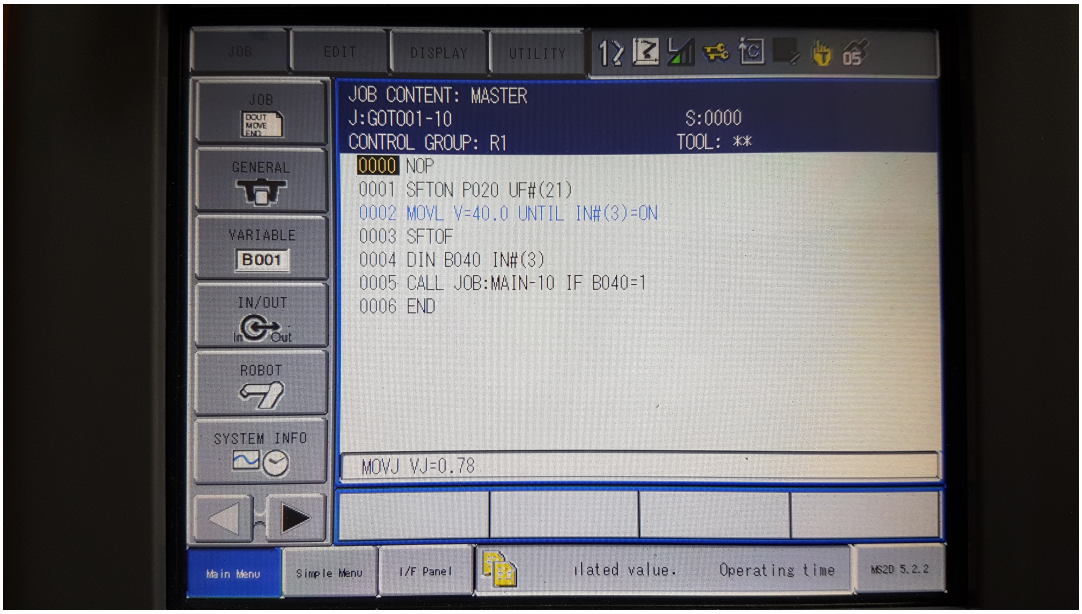
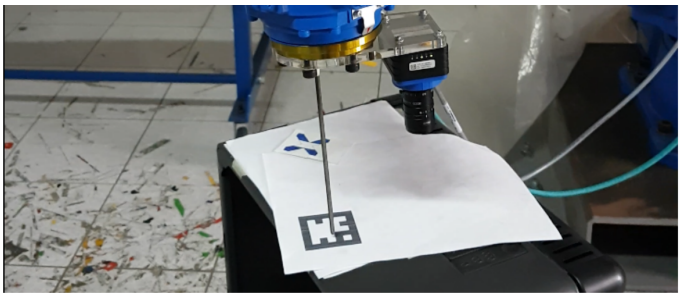


Fig. 4.10: Programación del trabajo de movimiento.

Esta estructura de programación permite ejecutar de forma automática y secuencial el proceso de escaneo y movimiento para ambos patrones, de manera intercalada. De este modo, el sistema es capaz de localizar el objeto dentro del espacio de trabajo y desplazar el extremo del robot hacia la posición objetivo, independientemente de la ubicación inicial de los patrones dentro de la zona delimitada.

(1)



(2)

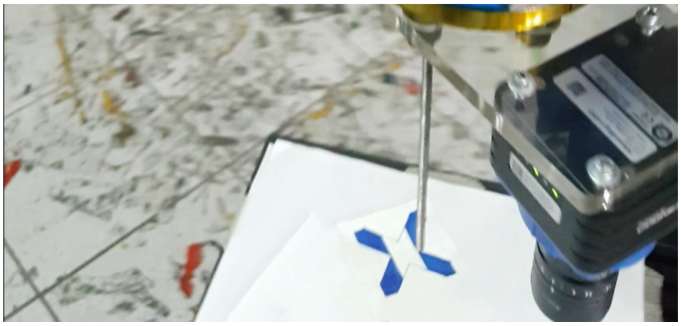
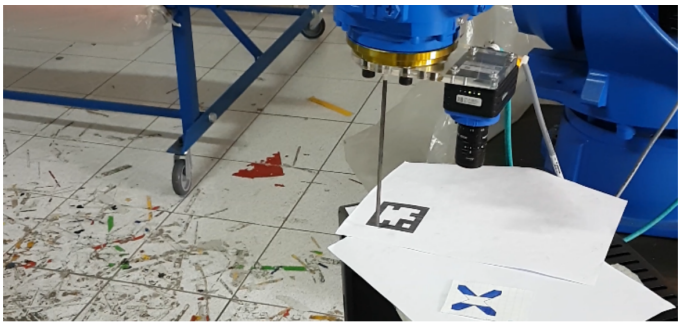


Fig. 4.11: Prueba de funcionamiento 1: Llegada del puntero al patrón (1) objeto y (2) ubicación objetivo.

(1)



(2)

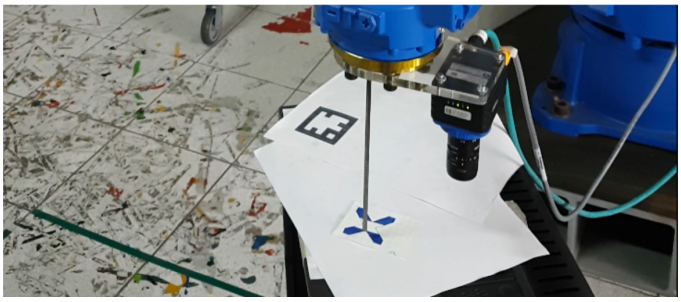


Fig. 4.12: Prueba de funcionamiento 2: Llegada del puntero al patrón (1) objeto y (2) ubicación objetivo.

En las Fig. 4.11 y 4.12 se muestran los resultados, a partir de los cuales se observa que el sistema realiza correctamente el escaneo del espacio de trabajo y detecta los

patrones de forma consistente. Al encontrar el objeto, el robot ejecuta adecuadamente la tarea de movimiento correspondiente y se desplaza hacia la posición detectada. No obstante, debido al error mecánico en la articulación L, el puntero no alcanza con total precisión el centro de los patrones, especialmente en desplazamientos asociados al eje X. A pesar de esta limitación, los resultados confirman el correcto funcionamiento de la solución implementada y un desempeño adecuado del sistema robot-cámara, considerando las restricciones mecánicas existentes.

4.4. Conclusiones del capítulo

En este capítulo se implementó y evaluó una solución de automatización basada en la integración de un sistema robot-cámara, cumpliendo con los objetivos planteados para la detección y manipulación de objetos dentro de un espacio de trabajo. La solución permitió verificar el correcto funcionamiento de la detección de patrones mediante visión por computador y la ejecución de los movimientos del brazo robótico siguiendo la lógica de una máquina de estados. En este contexto, el sistema demostró un desempeño consistente en la identificación del objeto y de la posición objetivo, así como en la secuencia de escaneo y desplazamiento.

No obstante, el desempeño del movimiento del robot se encuentra condicionado por el error mecánico presente en la articulación L, lo que limita la precisión en el posicionamiento final del extremo del robot. La estrategia de división del espacio de trabajo en cuadrantes permitió reducir el impacto de este error y evaluar el funcionamiento general del sistema, pero no constituye una solución de compensación válida para aplicaciones reales de manipulación.

5. CONTROL AUTOMÁTICO EN EL SISTEMA ROBOT-CÁMARA

En este capítulo se aborda uno de los objetivos específicos del trabajo de título. Este objetivo plantea extender la estrategia implementada anteriormente para que el movimiento del robot se adapte al caso en que el objeto se mueva de la posición detectada mientras el brazo robótico se encuentre en movimiento hacia él. La solución desarrollada en el capítulo anterior se puede considerar como un sistema de lazo abierto, ya que los movimientos del robot solo dependen de las coordenadas entregadas por la cámara. No existe retroalimentación que confirme si el robot llega o no correctamente al objeto.

A continuación, se presenta un estudio del sistema y sus elementos en relación a la posibilidad de implementar un sistema de control automático y la búsqueda de una solución ante el desplazamiento del objeto.

5.1. Análisis del Sistema

El propósito de este análisis es identificar las posibilidades y limitaciones del sistema robot–cámara para la implementación de estrategias de control automático, considerando tanto un enfoque conceptual como las restricciones prácticas de la arquitectura desarrollada.

Desde un punto de vista teórico, un sistema de control automático en lazo cerrado busca que una variable de interés alcance y mantenga un valor deseado mediante la medición continua de la salida del sistema y su comparación con una referencia. A partir de esta comparación se genera una señal de error que permite ajustar la acción de control de manera dinámica [18]. En el caso del sistema robot–cámara, el objetivo de control puede describirse como la reducción de la distancia entre el extremo del brazo robótico y el objeto detectado por el sistema de visión, idealmente hasta un valor nulo, constituyendo

esta distancia la variable de interés del sistema.

El sistema desarrollado hasta este punto opera de forma secuencial: la cámara MotoSight2D se posiciona en una ubicación previamente calibrada en conjunto con el brazo robótico, realiza la captura de una imagen y determina las coordenadas del objeto detectado. A partir de esta información, el robot ejecuta un movimiento hacia la posición estimada del objeto, sin considerar información adicional durante la ejecución del desplazamiento.

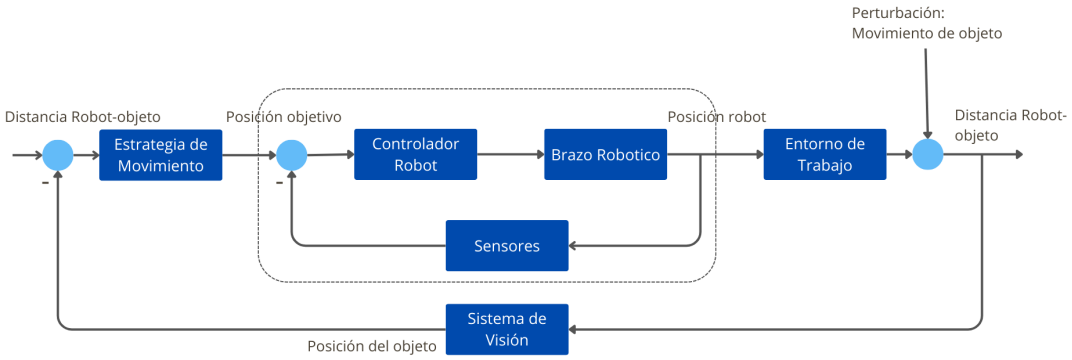


Fig. 5.1: Esquema de control de lazo cerrado en cascada que permitiría el control automático del sistema.

La Figura 5.1 presenta un esquema conceptual del sistema [19], representado como un sistema de control de lazo cerrado en cascada [20]. En este modelo, el lazo interno corresponde al control de posición del manipulador, el cual es implementado por el controlador del robot junto con sus sensores internos. El lazo externo considera la interacción entre el robot y el entorno, incorporando la información entregada por el sistema de visión para corregir la posición del robot en función de la posición del objeto.

Bajo un enfoque ideal de control automático, el sistema podría modelarse como un lazo cerrado en el cual la posición del robot se ajusta en función del error entre la posición deseada y la posición medida del objeto. Sin embargo, la implementación práctica de este tipo de control automático no es posible con la arquitectura actual del sistema. Estas limitaciones se deben principalmente a las características del sistema de visión y a su forma de integración con el controlador del robot. Para la utilización de la cámara MotoSight2D en la programación del robot se emplea la macro INSPECT, la cual forma

parte de la secuencia de ejecución del programa. Esto implica que no es posible realizar capturas de imagen mientras el robot se encuentra en movimiento. Las coordenadas detectadas por la cámara se relacionan con el movimiento del robot mediante un proceso de calibración y la definición de sistemas de coordenadas de usuario (UF). Si la captura se realiza desde una posición distinta a la calibrada, dicha relación se pierde. Adicionalmente, al acercarse al objeto, el campo de visión de la cámara se ve reducido, lo que afecta la capacidad de detección. Por otra parte, el controlador del robot no permite incorporar la información del sistema de visión mediante una retroalimentación continua en tiempo real.

Debido a estas restricciones, el sistema de visión no es capaz de entregar mediciones continuas de la posición del objeto durante el movimiento del robot, lo que impide la implementación de un control automático en lazo cerrado. La necesidad de detener el robot para realizar nuevas capturas, junto con la pérdida de la referencia calibrada, reduce además la eficiencia del sistema para tareas de posicionamiento rápido.

Ante estas limitaciones, se plantea una solución alternativa que permita gestionar cambios en la posición del objeto sin recurrir a un control automático en lazo cerrado. Dicha solución se basa en el uso de una interrupción generada por un sensor externo, que informa al controlador del robot sobre la ocurrencia de un evento, permitiendo detener el movimiento y reiniciar el proceso de detección. Para efectos del presente proyecto, la activación de la interrupción se realiza de forma manual, quedando como trabajo futuro la incorporación de un sensor capaz de detectar automáticamente el fenómeno descrito.

5.2. Solución Propuesta

Como se analizó en el subcapítulo anterior, la arquitectura actual del sistema robot–cámara no permite la implementación de un control automático en lazo cerrado, debido principalmente a la imposibilidad de realizar mediciones continuas de la posición del objeto durante el movimiento del robot. Frente a estas limitaciones, se propone una solución alternativa que permita mejorar la capacidad de reacción del sistema ante cambios en la posición del objeto, manteniendo la arquitectura existente.

La solución propuesta se basa en un esquema de operación en lazo abierto con supervisión [21] mediante interrupciones externas, cuyo funcionamiento general se ilustra en la Fig. 5.2. En este enfoque, el sistema de visión se utiliza únicamente para la detección discreta de la posición del objeto, a partir de la cual se define una posición objetivo para el brazo robótico. El controlador del robot ejecuta el movimiento hacia dicha posición, sin realimentación continua proveniente del sistema de visión.

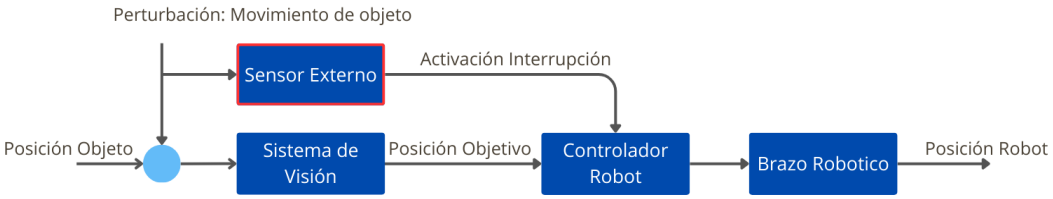


Fig. 5.2: Esquema de control de la solución implementada.

En el esquema propuesto (Fig. 5.2), la perturbación asociada al desplazamiento del objeto durante la ejecución del movimiento se modela como una entrada externa al sistema. Dicha perturbación es detectada por un sensor externo, el cual no forma parte del lazo de control principal. Este sensor genera una señal de interrupción que ingresa directamente al controlador del robot, permitiendo modificar la secuencia de ejecución del programa. La activación de la interrupción provoca la detención inmediata del movimiento en curso y fuerza el reinicio del ciclo de trabajo, el cual contempla una nueva captura de imagen por parte del sistema de visión y el cálculo de una nueva posición objetivo. De este modo, el sistema no corrige el movimiento en tiempo real, sino que reacciona ante eventos relevantes del entorno, asegurando que cada desplazamiento del robot se base en información actualizada y coherente con la posición real del objeto.

Desde el punto de vista del control, la solución propuesta no constituye un control automático en lazo cerrado, sino una estrategia de control en lazo abierto con supervisión. Este enfoque permite mejorar la robustez del sistema robot–cámara frente a perturbaciones externas, manteniendo una arquitectura simple y compatible con las limitaciones del hardware y software disponibles. Si bien no se incorpora realimentación continua de la posición del objeto, la inclusión de un mecanismo de supervisión mediante interrupciones proporciona una solución práctica y coherente desde el punto de vista de

la automatización industrial.

5.3. Implementación de la Solución

Para poder utilizar una interrupción se utiliza el terminal de entradas y salidas "CN309" del controlador "DX200". En la Fig. 5.3 se ve la disposición de las entradas y en la Fig. 5.4 se muestra una imagen del terminal. Para el trabajo de programación se utiliza la entrada IN03 que corresponde al terminal B4. Para activar la interrupción se debe realizar la conexión entre los terminales B4 y B7.

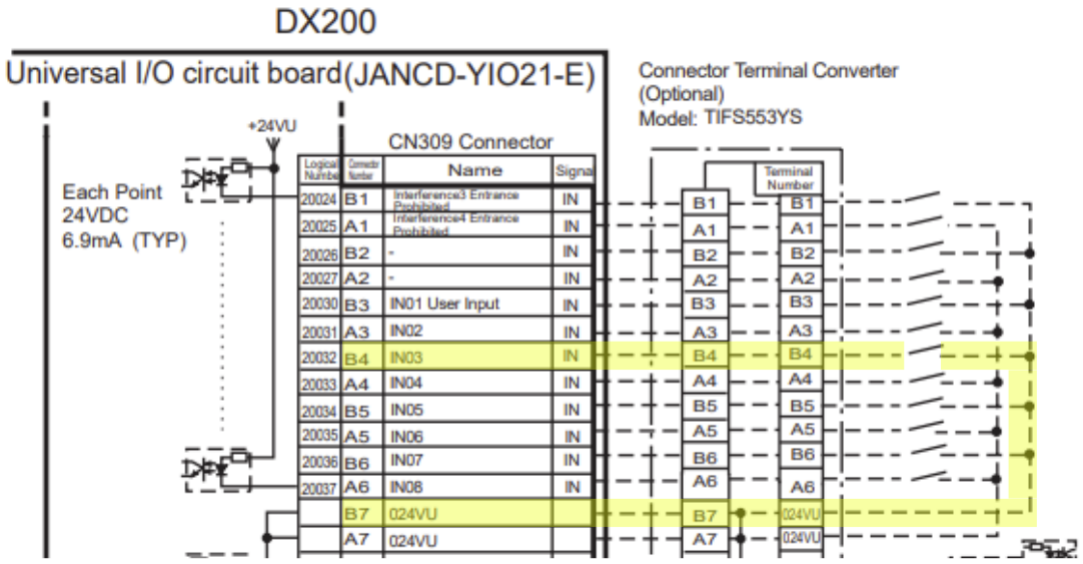


Fig. 5.3: Diagrama de conexiones de terminal de entradas y salidas "CN309" del controlador "DX200"

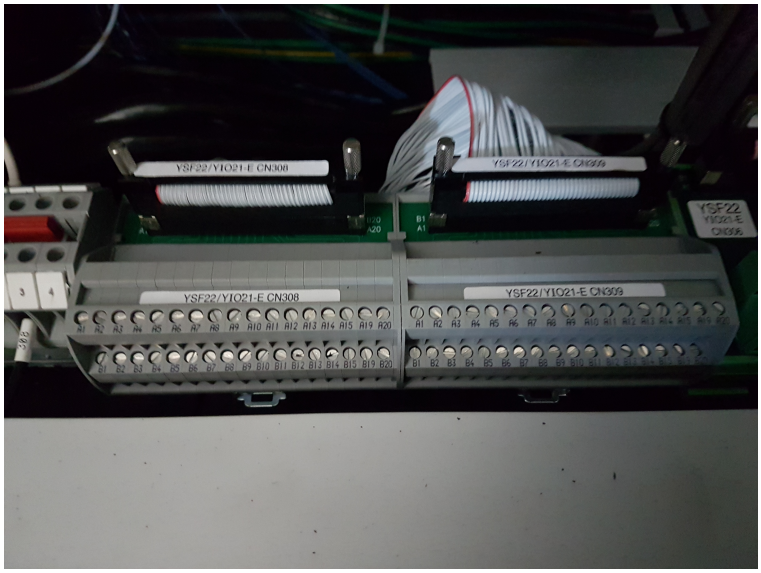


Fig. 5.4: Fotografía del terminal “CN309” del controlador “DX200” utilizado.

La interrupción se incorpora al trabajo de movimiento del capítulo anterior. El código se muestra en la Fig. 5.5 donde se destaca lo correspondiente a la interrupción. El comportamiento programado consiste en que el robot se dirige hacia el objeto hasta que la entrada “IN03” tenga un valor lógico alto. Cuando esto ocurre, el robot detiene su desplazamiento y el flujo del programa vuelve al trabajo principal iniciando nuevamente el trabajo de escaneo del objeto.

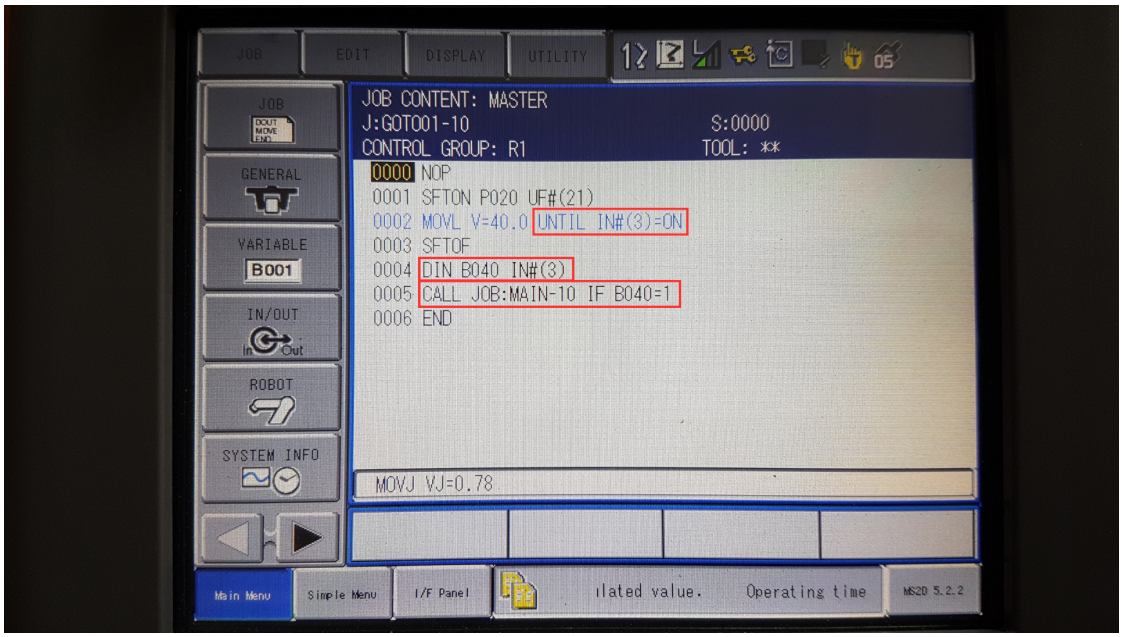


Fig. 5.5: Programación del trabajo de movimiento con las instrucciones de la interrupción en rojo.

En la Fig. 5.6 se muestran los cables conectados a la terminal, utilizados para verificar si la interrupción funciona correctamente.

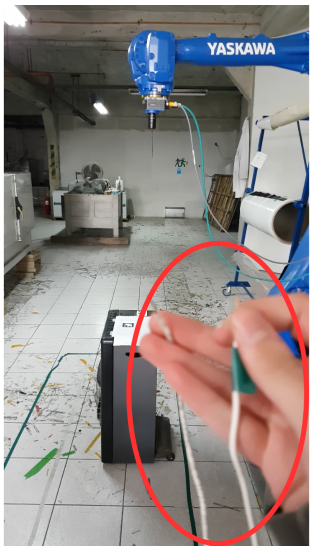


Fig. 5.6: Cables conectados al terminal “CN309” para activar la interrupción encerrados en rojo.

En la Fig. 5.7 se muestra una secuencia que demuestra el funcionamiento de la interrupción.

- En (1) se muestra una imagen mientras el robot se desplaza hacia el objeto. Al hacer contacto entre los cables, se activa la interrupción deteniendo su movimiento.
- En (2) el robot se encuentra en una posición más alejada que en el punto anterior, ya que el robot se detuvo y se dirige nuevamente hacia la posición para iniciar el escaneo.
- En (3) el sistema se encuentra realizando nuevamenet el escaneo.

En base a las pruebas realizadas, queda demostrado que el sistema es capaz de detener el movimiento del robot ante interrupciones y volver a un estado que permita encontrar el objeto en su posición correcta.



(1) (2) (3)

Fig. 5.7: Secuencia del robot al activar la interrupción

5.4. Conclusión del Capítulo

En este capítulo se analizó la factibilidad de implementar un control automático en lazo cerrado para el sistema robot–cámara. El estudio evidenció que, debido a las limitaciones propias del sistema de visión y a su integración con el controlador del robot, no es posible realizar mediciones continuas durante el movimiento ni ajustar la trayectoria en tiempo real. Frente a este escenario, se propuso una solución alternativa basada en una

estrategia de lazo abierto con supervisión mediante interrupciones, que permite detener el movimiento del robot y reiniciar el proceso de detección cuando el objeto cambia de posición durante la ejecución.

Las pruebas realizadas permitieron comprobar que el uso de interrupciones cumple con el objetivo planteado, mejorando la capacidad de reacción del sistema frente a perturbaciones externas. No obstante, la activación de la interrupción se realiza de forma manual, por lo que se plantea como trabajo futuro la incorporación de un mecanismo automático de detección que permita activar la interrupción sin intervención del operador.

6. CONCLUSIÓN

El presente trabajo tuvo como objetivo general controlar los movimientos de un brazo robótico mediante la información entregada por una cámara de visión. La tarea principal fue realizar la manipulación de un objeto que fuese detectado dentro de un espacio de trabajo. Para ello se utilizó el sistema robot-cámara, compuesto principalmente por el brazo robótico "Motoman MH-50" y la cámara de visión "MotoSight2D", gestionados por el uso del mando de control y el programa computacional "InSight Explorer". Durante el trabajo se realizaron procedimientos de calibración en ambos dispositivos y se realizaron pruebas de funcionamiento para evaluar el desempeño del sistema en conjunto. Además, se implementó la tarea de automatización para encontrar un objeto sin una posición predefinida, dentro de un espacio de trabajo, y llevar al robot a dicha posición para trasladarlo, de forma simulada, a un lugar objetivo marcado por otro patrón visual.

Los resultados demostraron que la cámara "MotoSight2D" detecta objetos y patrones de manera precisa y las coordenadas que entrega coinciden con la posición espacial real del objeto detectado. Gracias a la calibración y la creación de coordenadas de usuario (UF) utilizando la grilla de nueve puntos, se logró establecer la relación entre la información de la cámara y el posicionamiento del robot. De esta forma se ejecutaron con éxito movimientos automáticos hacia los objetos detectados, cumpliendo con el objetivo general del trabajo.

Adicionalmente, se analizó la posibilidad de implementar control automático de lazo cerrado en el sistema robot–cámara, concluyendo que la cámara de visión no es apta para este fin debido a la imposibilidad de realizar mediciones continuas durante el movimiento del manipulador. Como alternativa, se implementó una solución que permite reaccionar ante el desplazamiento del objeto mientras el robot se encuentra en movimiento, mediante la detención del brazo robótico y la ejecución de un nuevo ciclo de escaneo

para recalcular la posición objetivo.

El desarrollo del trabajo también dejó ver las limitaciones del sistema. El brazo robótico presentó un desperfecto mecánico en una de sus articulaciones asociado al "encoder" del servomotor, el cual provocó errores en las trayectorias del robot y en la precisión de las posiciones determinadas por la cámara.

El trabajo permitió establecer un procedimiento completo para la integración del sistema robot–cámara, desde las configuraciones básicas de cada elemento hasta la ejecución de tareas de manipulación con el conjunto de la cámara y el robot. Se generó el conocimiento práctico necesario para poder operar el robot mediante el mando de control, la generación de tareas de visión con "In-Sight Explorer" y el uso de interrupciones realizando conexiones al terminal de entradas y salidas del controlador "DX200". De esta forma se sientan las bases para utilizar el robot en tareas de automatización industrial utilizando la cámara como detector de objetos. Además, se presentan las oportunidades y limitaciones del sistema, como información útil para el desarrollo de soluciones más complejas.

Para trabajos futuros, se recomienda realizar pruebas con la articulación del brazo robótico reparado, y adaptar la programación de los cuadrantes, de una manera más sencilla y que actualmente no es posible realizar. También se propone realizar la búsqueda e implementación de un sensor que permita detectar el movimiento del objeto mientras el robot se encuentra en movimiento y que active la interrupción de forma automática. Finalmente, se propone explorar maneras alternativas de abordar la implementación de un sistema de control de lazo cerrado. Investigando la posibilidad de controlar los movimientos del robot en tiempo real de forma externa, lo que ayudaría a que el robot reciba instrucciones de sensores que determinen su movimiento de forma constante.

Bibliografía

[1] Haider AF Almurib, Haidar Fadhil Al-Qrimli, and Nandha Kumar. A review of application industrial robotic design. In *2011 Ninth International Conference on ICT and Knowledge Engineering*, pages 105–112. IEEE, 2012.

[2] Robotnik. Guía completa de robots manipuladores: ventajas y aplicaciones, 2024. [consulta: 04 enero 2026].

[3] Saunak Saha, Amitangshu Mukherjee, Sanchoy Das, and Babak Hoseini. Intelligent task repeatability of an industrial robotic arm. In *2016 IEEE 7th Power India International Conference (PIICON)*, pages 1–5. IEEE, 2016.

[4] Jordi Termes Sabater. Manipulación de objetos mediante un brazo robótico usando técnicas de aprendizaje por refuerzo. 2019.

[5] Eddie Angel Sobrado Malpartida. *Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot*. Pontificia Universidad Catolica del Peru (Peru), 2011.

[6] Henil Gajjar, Stavan Sanyal, and Manan Shah. A comprehensive study on lane detecting autonomous car using computer vision. *Expert Systems with Applications*, 233:120929, 2023.

[7] Xingshuai Dong and Massimiliano L Cappuccio. Applications of computer vision in autonomous vehicles: Methods, challenges and future directions. *arXiv preprint arXiv:2311.09093*, 1, 2023.

[8] Daichi Kitaguchi, Nobuyoshi Takeshita, Hiro Hasegawa, and Masaaki Ito. Arti-

ficial intelligence-based computer vision in surgery: Recent advances and future perspectives. *Annals of gastroenterological surgery*, 6(1):29–36, 2022.

[9] Pietro Mascagni, Deepak Alapatt, Luca Sestini, Maria S Altieri, Amin Madani, Yusuke Watanabe, Adnan Alseidi, Jay A Redan, Sergio Alfieri, Guido Costamagna, et al. Computer vision in surgery: from potential to clinical value. *npj Digital Medicine*, 5(1):163, 2022.

[10] Kazuyuki Nagata and Takao Nishi. Modeling object arrangement patterns and picking arranged objects. *Advanced Robotics*, 35(16):981–994, 2021.

[11] Yaskawa. *Motoman MH50 Series Brochure*. Yaskawa America, Inc. Motoman Robotics Division, n.d.

[12] Yaskawa America, Inc. Mh series industrial robots for assembly & handling, 2026. [consulta: 04 enero 2026].

[13] Yaskawa. *DX200 INSTRUCTIONS*. Yaskawa Electric Corporation, manual no. recto-a220 edition, 2025.

[14] Yaskawa. *Vision System: Camera & Software Motosight2D*. YASKAWA Europe GmbH. Robotics, n.d.

[15] Yaskawa. *MOTOSIGHT™2D INSTRUCTIONS*. Yaskawa Electric Corporation, manual no. 169004-1cd edition, n.d.

[16] Vivek Vijayakumar and Fabio Sgarbossa. A literature review on the level of automation in picker-to-parts order picking system: research opportunities. *IFAC-PapersOnLine*, 54(1):438–443, 2021.

[17] Kevin Joel Martínez Andino et al. Diseño y evaluación de un brazo robótico autónomo controlado por visión artificial para entornos industriales. *Innovarium International Journal*, 1(1):1–12, 2023.

[18] Evgeniĭ Pavlovich Popov. *The dynamics of automatic control systems*. Elsevier, 2014.

[19] Loukas Prezas, Zoi Arkouli, George Michalos, Panagiotis Angelakis, Christos Gkournelos, and Sotiris Makris. A multi-purpose robot perception system enabling closed-loop control for zero defect manufacturing in gluing processes of large parts. *Robotics and Autonomous Systems*, 181:104778, 2024.

[20] Yongho Lee, Sunggook Oh, and Sunwon Park. Enhanced control with a general cascade control structure. *Industrial & engineering chemistry research*, 41(11):2679–2688, 2002.

[21] Ana González-Marcos, J Ordieres-Meré, V Muñoz-Munilla, and Fernando Alba-Elías. An intelligent supervision system for open loop controlled processes. *Journal of Intelligent Manufacturing*, 24(1):15–24, 2013.