



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS
INDUSTRIALES

TRABAJO FIN DE GRADO

**SISTEMA DE PICKING DE PALÉS PARA UN
ROBOT INDUSTRIAL EQUIPADO CON UNA
CÁMARA RGB-D**

Autor: Lionel Güitta López

Director: Jaime Boal Martín-Larrauri

Codirector: José Antonio Rodríguez Mondéjar

Madrid

2020-2021

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
“Sistema de picking de palés para un robot industrial equipado con una cámara RGB-D”
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el
curso académico 2020/21 es de mi autoría, original e inédito y
no ha sido presentado con anterioridad a otros efectos.
El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido
tomada de otros documentos está debidamente referenciada.



Fdo.: Lionel Güitta López

Fecha: 28/ 06/ 2021

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Firmado digitalmente por
BOAL MARTIN LARRAURI
JAIME - 05304600H
Fecha: 2021.06.29 10:24:56
+02'00'

Fdo.: Jaime Boal Martín-Larrauri

Fecha://

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Firmado por RODRIGUEZ MONDEJAR JOSE ANTONIO -
29057313X el día 29/06/2021 con un certificado
emitido por AC FNMT Usuarios

Fdo.: José Antonio Rodríguez Mondéjar

Fecha://



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO **SISTEMA DE PICKING DE PALÉS PARA UN ROBOT INDUSTRIAL EQUIPADO CON UNA CÁMARA RGB-D**

Autor: Lionel Güitta López

Director: Jaime Boal Martín-Larrauri

Codirector: José Antonio Rodríguez Mondéjar

Madrid

2020-2021

Agradecimientos

Este trabajo no habría sido posible sin el apoyo de toda mi familia y amigos, así como de la indispensable guía de Jaime a lo largo de todo el proyecto.

Muchas gracias a todos por estar ahí apoyándome siempre en todo lo que necesitare.

Índice de la memoria

RESUMEN DEL PROYECTO	<i>ix</i>
ABSTRACT	<i>xiii</i>
Capítulo 1. Introducción	<i>1</i>
1.1 Motivación del proyecto	<i>3</i>
1.2 Objetivos	<i>4</i>
Capítulo 2. Estado de la cuestión	<i>7</i>
2.1 REDES NEURONALES	<i>7</i>
2.2 REDES NEURONALES EN SISTEMAS DE PICK AND PLACE	<i>10</i>
2.2.1 USO DE MODELOS 3D	<i>11</i>
2.2.2 MÉTODO AGNÓSTICO	<i>12</i>
2.3 HERRAMIENTAS DE AGARRE	<i>14</i>
2.3.1 HERRAMIENTAS ANTROPOMÓRFICAS	<i>14</i>
2.3.2 HERRAMIENTAS PARA OBJETOS DE FORMA Y TAMAÑO ESPECÍFICO	<i>15</i>
2.3.3 HERRAMIENTAS MÚLTIPLES	<i>15</i>
Capítulo 3. Arquitectura del sistema	<i>19</i>
Capítulo 4. Captura de imágenes y detección de piezas	<i>23</i>
4.1 SISTEMA DE PARTIDA	<i>24</i>
4.2 TRASLACIÓN DEL SISTEMA DE COORDENADAS	<i>28</i>
4.3 EXTRACCIÓN DE COORDENADAS POR CALIBRACIÓN A PARTIR DE PUNTOS FIJOS . 28	
4.3.1 CALIBRACIÓN A PARTIR DE PUNTOS EXTREMOS	<i>29</i>
4.4 AJUSTE EXACTO POR APROXIMACIÓN	<i>35</i>
4.5 EXTRACCIÓN DE LA ALTURA Y DEL ÁNGULO DE LA PIEZA	<i>40</i>
Capítulo 5. Acciones del brazo robótico	<i>41</i>
5.1 ZONA DE TRABAJO	<i>41</i>
5.2 MOVIMIENTOS DEL ROBOT	<i>42</i>
5.3 GESTIÓN DE LAS CINTAS TRANSPORTADORAS	<i>48</i>
5.3.1 RETENEDORES Y SENSORES	<i>49</i>

5.3.2 DESVIADOR.....	52
5.4 PROGRAMACIÓN.....	52
Capítulo 6. Resultados.....	57
6.1 TIEMPOS MATLAB.....	57
6.2 TIEMPOS ROBOT.....	59
Capítulo 7. Conclusiones.....	63
Capítulo 8. Futuros desarrollos.....	65
Capítulo 9. Referencias.....	67
Anexo I Desarrollo de la aplicación de MATLAB.....	71
Anexo II Carcasa de la cámara.....	75
Anexo III Alineación con los ODS.....	77
Anexo IV Tablas de tiempos de los procesos del robot.....	79

Índice de figuras

Figura 1: Brazo robótico IRB120 de ABB	2
Figura 2: Área de trabajo para el montaje de las figuras con figura de ejemplo	2
Figura 3: Cámara Intel RealSense D435	2
Figura 4: Planta completa con robot, cintas y área de montaje	3
Figura 5: Esquema del proceso.....	4
Figura 6: Arquitectura de la red neuronal convolucional AlexNet (Fuente: [10])	8
Figura 7: Arquitectura de la red neuronal convolucional de VGG16 (Fuente: [11])	8
Figura 8: Proceso R-CNN (Fuente: [12])	9
Figura 9: Proceso Faster R-CNN (Fuente: [12])	9
Figura 10: Proceso YOLO (Fuente: [12]).....	9
Figura 11: Análisis de métodos de agarre a partir de modelos 3D de objetos (Fuente: [15])	12
Figura 12: Esquema de análisis agnóstico para planificación de agarres (Fuente: [17])	13
Figura 13: Modelo 3D de la herramienta propuesta en [15] (Fuente: [15])	14
Figura 14: Robot con herramienta de agarre antropomórfica propuesto en [16] (Fuente: www.nasa.gov)	14
Figura 15: Herramienta propuesta para trabajos microscópicos (Fuente: [20])	15
Figura 16: Modelo de la herramienta con (A) Pinza de succión (B) Motor para el cambio de herramienta y (C) Pinza mecánica (Fuente [21]).....	16
Figura 17: Vista inferior pinza de agarre abierta	17
Figura 18: Vista inferior Pinza de Agarre Cerrada.....	17
Figura 19: Vista frontal Pinza de Agarre abierta	17
Figura 20: Vista frontal Pinza de Agarre cerrada	17
Figura 21: Esquema del proceso de construcción de una figura	19
Figura 22: Interfaz inicial de la aplicación de MATLAB	20
Figura 23: Interfaz tras análisis de palé con pieza.....	21
Figura 24: Interfaz tras análisis de palé sin pieza.....	22
Figura 25: Vista inferior de la pinza y la cámara en la muñeca del robot.....	23
Figura 26: Interfaz inicial de la aplicación de partida	25
Figura 27: Resultado del procesado de imágenes del sistema partida.....	25

Figura 28: Sistema de coordenadas de la cámara	27
Figura 29: Vista lateral de los orígenes de los sistemas coordenadas del robot y de la cámara	27
Figura 30: Vista de planta de la disposición de los sistemas de coordenadas	27
Figura 31: Puntos límite del palé para calibración	30
Figura 32: Interfaz del flexpendant para la obtención de coordenadas del robot.....	31
Figura 33: Extracción de coordenadas de la pieza en el sistema del robot empleando el flexpendant	31
Figura 34: Punto de partida de la calibración	33
Figura 35: Desplazamiento en el eje X de la cámara	33
Figura 36: Desplazamiento en el eje Y de la cámara	33
Figura 37: Ajuste exacto primera fase	36
Figura 38: Ajuste exacto segunda fase	36
Figura 39: Ajuste exacto tercera fase	36
Figura 40: Ajuste exacto cuarta fase.....	36
Figura 41: Resultado de la anotación de todos los puntos.....	37
Figura 42: Prueba del ajuste exacto	39
Figura 43: Distancia en el eje Z de la pinza hasta la pieza a recoger	40
Figura 44: Alcance del brazo robótico IRB 120.....	41
Figura 45: Zonas alcanzables por el robot en el área de trabajo.....	42
Figura 46: Posición de referencia/reposo frente al palé bloqueado.....	43
Figura 47: Recogida de la pieza tras análisis de la imagen	44
Figura 48: Figura 1 anotada.....	45
Figura 49: Figura 2 anotada.....	45
Figura 50: Pieza insertada sin rotación de la pinza	47
Figura 51: Pieza insertada con la pinza rotada 90°	47
Figura 52: Pieza insertada con la pinza rotada 90° (vista lateral).....	47
Figura 53: Esquema de los sensores de la planta	48
Figura 54: Retenedor cinta principal expuesto	49
Figura 55: Retenedor cinta principal recogido	49
Figura 56: Retenedor de ajuste expuesto	50
Figura 57: Retenedor de ajuste recogido	50
Figura 58: Retenedor de ajuste parando palé.....	50

Figura 59: Bloqueadores recogidos	51
Figura 60: Bloqueadores expuestos	51
Figura 61: Vista inferior de palé con agujeros para bloqueo resaltados.....	51
Figura 62: Desviador en activo.....	52
Figura 63: Desviador desactivado	52
Figura 64: Aproximación de palé al retenedor principal.....	53
Figura 65: Desvío de palé.....	53
Figura 66: Incorporación de palé bloqueado a la cinta principal	54
Figura 67: Llegada del palé desviado a la posición de bloqueo	54
Figura 68: Diagrama de cajas de los tiempos de app_picture	58
Figura 69: Gráfico de cajas de la función app_picture despreciando outliers superiores ...	58
Figura 70: Gráfico de cajas de los tiempos de envío de datos al robot	58
Figura 71: Gráfico de cajas de tiempo total de procesado de una pieza.....	58
Figura 72: Esquema de los puntos de medida de tiempos del robot.....	60

Índice de tablas

Tabla 1: Coordenadas de los puntos límite del palé.....	32
Tabla 2: Coordenadas puntos próximos.....	34
Tabla 3: Puntos obtenidos del ajuste exacto.....	38
Tabla 4: Estadísticos de los tiempos obtenidos.....	59
Tabla 5: Estadísticos de la construcción de la primera figura con la primera forma.....	61
Tabla 6: Estadísticos de la construcción de la segunda figura con la primera forma.....	61
Tabla 7: Estadísticos de la construcción de la primera figura con la segunda forma.....	61
Tabla 8: Estadísticos de la construcción de la segunda figura con la segunda forma.....	62
Tabla A.4. 1: Tiempos de ensayos de la figura 1 con la forma 1.....	79
Tabla A.4. 2: Estadísticos de los tiempos de la figura 1 con la forma 1 desglosados por pieza.....	80
Tabla A.4. 3: Tiempos de ensayos de la figura 2 con la forma 1.....	81
Tabla A.4. 4: Estadísticos de los tiempos de la figura 2 con la forma 1 desglosados por pieza.....	82
Tabla A.4. 5: Tiempos de ensayos de la figura 1 con la forma 2.....	83
Tabla A.4. 6: Estadísticos de los tiempos de la figura 1 con la forma 2 desglosados por pieza.....	84
Tabla A.4. 7: Tiempos de ensayos de la figura 2 con la forma 2.....	85
Tabla A.4. 8: Estadísticos de los tiempos de la figura 2 con la forma 2 desglosados por pieza.....	86

RESUMEN DEL PROYECTO

SISTEMA DE PICKING DE PALÉS PARA UN ROBOT INDUSTRIAL EQUIPADO CON UNA CÁMARA RGB-D

Autor: Güitta López, Lionel

Director: Boal Martín-Larrauri, Jaime

Codirector: Rodríguez Mondéjar, José Antonio

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas)

El objetivo de este proyecto es conseguir que un robot industrial ABB IRB120 de la minifábrica de ICAI, equipado con una cámara Intel RealSense D435 que proporciona imágenes de color y profundidad, sea capaz de coger de forma autónoma piezas de LEGO, que circulan en palés por la cinta transportadora, y construir distintas figuras en el área de trabajo designada.

Palabras clave: Pick and Place, Visión Artificial, YOLO, R-CNN, Automatización.

1. Introducción

Hoy en día, existen gran cantidad de robots industriales en fábricas de todo tipo, con entornos dinámicos y repitiendo la misma tarea de coger un objeto o pieza y colocarla en un sitio diferente, lo que comúnmente se denomina *Pick and Place*. Dichas piezas pueden estar en diversas posiciones dispuestas de múltiples maneras, el robot que opere con ellas ha de ser capaz de detectar esto y actuar en consecuencia. Estas tareas se desarrollaban haciendo que las piezas llegasen siempre en la misma posición o empleando múltiples sensores y complejos cálculos que implicaban una gran dificultad para desarrollar la tarea [1]. Con la llegada de la Industria 4.0, este tipo de tareas se está desarrollando con el objetivo de flexibilizar y dotar de mayor precisión a los sistemas gracias a los sistemas de visión artificial basados en redes neuronales [2] y al empleo de diversas herramientas para agarrar dichas piezas.

2. Definición del proyecto

Con estas condiciones, se van a implantar estas nuevas características en un robot industrial ABB IRB120 [3] instalado en la minifábrica de ICAI, como el de la Figura 1, con el fin de capacitarlo con la habilidad para montar varias figuras de LEGO en el área de trabajo de la Figura 2. La figura que se construirá será escogida por el operario a partir de dos propuestas.

Las figuras serán montadas por el brazo robótico a partir de piezas que reconocerá y recolectará de palés que se moverán por cintas transportadoras dentro de la región de trabajo del robot, y que gestionará el mismo.

Este proyecto parte del sistema de visión artificial aplicado a un robot industrial de Ignacio Ortiz de Zúñiga Mingot [4]. Dicho sistema era capaz de reconocer la posición de las piezas, la orientación y su color a partir de las imágenes de color y profundidad obtenidas con la cámara Intel RealSense D435 [6].

3. Descripción del sistema

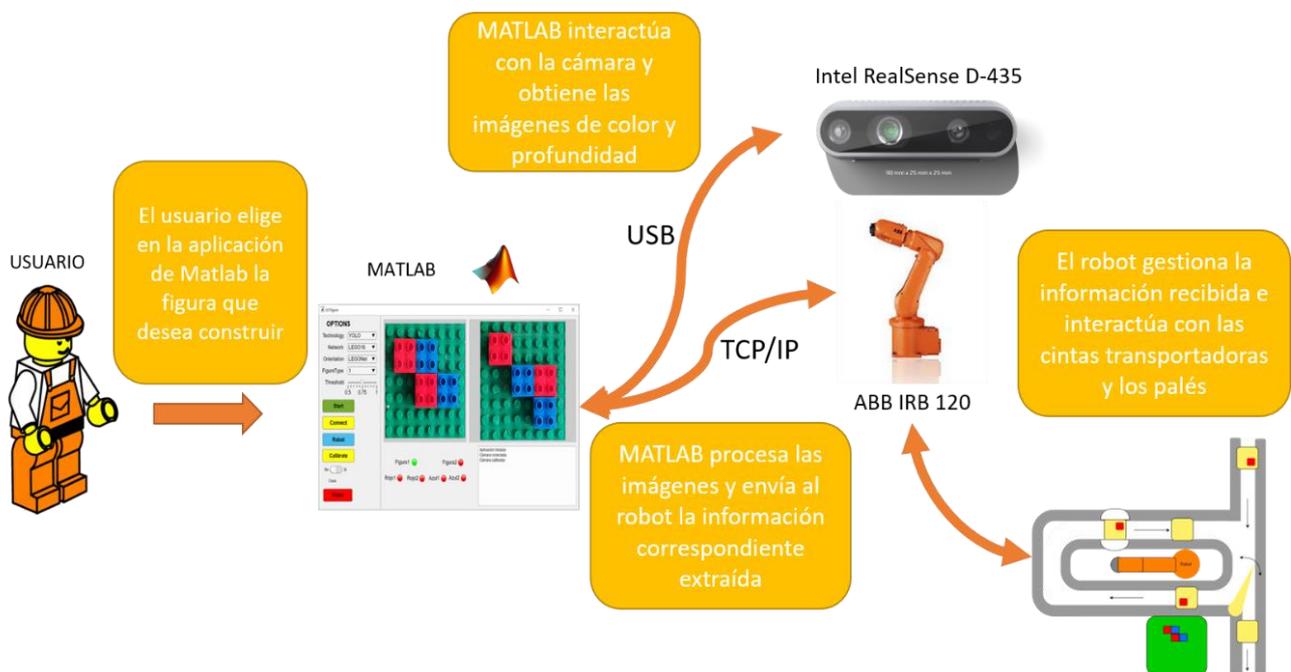


Figura Res. a: Esquema de funcionamiento del sistema

El proceso comienza con la elección por parte de un usuario de la figura que desea construir a través de la aplicación desarrollada en MATLAB. Una vez escogida, MATLAB se comunica con el robot para indicarle el inicio de construcción de una de las figuras y este se dispone a desviar un palé hacia su zona para la inspección de este. Una vez que el palé llega a la posición indicada, el robot se lo comunica al ordenador y este interactúa con la cámara para obtener imágenes para su estudio. Una vez analizadas, se determina la presencia de una pieza de LEGO o no, su localización y si corresponde a la figura en construcción. En caso afirmativo el robot procederá a coger la pieza y trasladarla a su correspondiente posición. En caso negativo el palé se liberará de su posición de bloqueo para la incorporación a la cinta

principal. Una vez colocada la pieza, el robot volverá a su posición inicial y procederá a desviar un nuevo palé si la figura no se ha terminado de construir. En el caso de que se haya completado la figura, el robot se mantendrá a la espera de nuevas órdenes por parte del ordenador.

4. Resultados

Este proyecto no tiene unos resultados numéricos como tal. El objetivo principal era conseguir construir las figuras de LEGO empleando el robot, la cámara y las cintas transportadoras conjuntamente a partir del del sistema de visión artificial desarrollado por Ignacio Ortiz de Zúñiga Mingot.

Se realizaron numerosas pruebas a lo largo del proyecto y actualmente la tarea se completa con un 100% de éxito. Por lo tanto, se puede concluir que el objetivo se cumplió. El operario solo tiene que escoger la figura que desea construir. Tras esto el robot, MATLAB y la cámara trabajaran conjuntamente para construirla en el área de destino y el operario solo tiene que supervisar el montaje de esta.

5. Conclusiones

Sobre el procesado de las imágenes, se analizaron distintos métodos para la integración del sistema de visión artificial partiendo desde un enfoque más flexible. Después, tras descartar este debido a la falta de precisión que planteaba, se consiguió un sistema personalizado para el robot, con gran precisión y robustez para el proyecto. Sin embargo, este conlleva la calibración inicial de todos los puntos del palé, para conseguir que el robot sea capaz de coger la pieza con exactitud y sin ocasionar daños en la misma ni en el brazo robótico.

Respecto a la gestión de las cintas trasportadoras y los movimientos del robot se puede concluir que la estructura del sistema se puede adaptar y emplear en otros robots de la fábrica de ICAI. Además, el empleo de diversas funciones dota de flexibilidad al sistema para su fácil adaptación a un nuevo entorno o el montaje de nuevas figuras.

6. Referencias

- [1] C.-Y. Tsai, C.-C. Wong, C.-J. Yu, C.-C. Liu y T.-Y. Liu, «A Hybrid Switched Reactive-Based Visual Servo» Marzo 2015. Disponible: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6920015>. [Último acceso: 21 Enero 2021].
- [2] S. Soltan, A. Oleinikov, M. F. Demirci y A. Shintemirov, «Deep Learning-Based Object Classification and Position Estimation Pipeline for Potential Use in Robotized Pick-and-Place Operations» 18 Agosto 2020. Disponible: https://www.mdpi.com/2218-6581/9/3/63?type=check_update&version=1. [Último acceso: 27 Diciembre 2020].
- [3] ABB, «IRB 120» Disponible: <https://new.abb.com/products/robotics/es/robots-industriales/irb-120>. [Último acceso: 10 Enero 2021].
- [4] I. Ortiz de Zúñiga Mingot, «Optimización de un sistema de visión artificial de un robot industrial» Trabajo de Fin de Grado, ICAI.
- [5] «Realsense D435». Disponible: <https://www.intereasense.com/depth-camerad435/>. [Último acceso: 28 Diciembre 2020].

ABSTRACT

PALLET PICKING SYSTEM FOR AN INDUSTRIAL ROBOT EQUIPPED WITH AN RGB-D CAMERA

Author: Güitta López, Lionel

Supervisor: Boal Martín-Larrauri, Jaime

Co-supervisor: Rodríguez Mondéjar, José Antonio

Collaborating Entity: ICAI – Universidad Pontificia Comillas)

The objective of this project is to make an ABB IRB120 industrial robot at ICAI's mini-factory, equipped with an Intel RealSense D435 camera that provides color and depth images, capable of autonomously picking up LEGO pieces, which circulate on pallets on the moving belt, and building different figures in the designated work area.

Keywords: Pick and Place, Artificial Vision, YOLO, R-CNN, Automation

1. Introduction

Nowadays, there are a large number of industrial robots in factories of all kinds, with dynamic environments and repeating the same task of picking up an object or part and placing it in a different location, commonly referred to as Pick and Place. Such parts can be in various positions and arranged in multiple ways, the robot operating with them has to be able to detect this and act accordingly. These tasks were developed by having the parts always arrive in the same position or by using multiple sensors and complex calculations that implied a great difficulty to develop the task [1]. With the advent of Industry 4.0, these types of tasks are being developed with the aim of making the systems more flexible and more precise thanks to artificial vision systems based on neural networks [2] and the use of various tools to grasp these parts.

2. Project definition

With these conditions, these new features are going to be implemented in an ABB IRB120 [3] industrial robot installed in the ICAI mini-factory, like the one in Figure 1, in order to train it with the ability to assemble several LEGO figures in the work area in Figure 2. The figure to be built will be chosen by the operator from two proposals. The figures will be assembled by the robotic arm from parts that it will recognize and collect from pallets that

will move on conveyor belts within the robot's working region, and that will be managed by the robot.

This project is based on the artificial vision system applied to an industrial robot by Ignacio Ortiz de Zúñiga Mingot [4]. This system was able to recognize the position of the parts, the orientation and their color from the color and depth images obtained with the Intel RealSense D435 camera [6].

3. System description

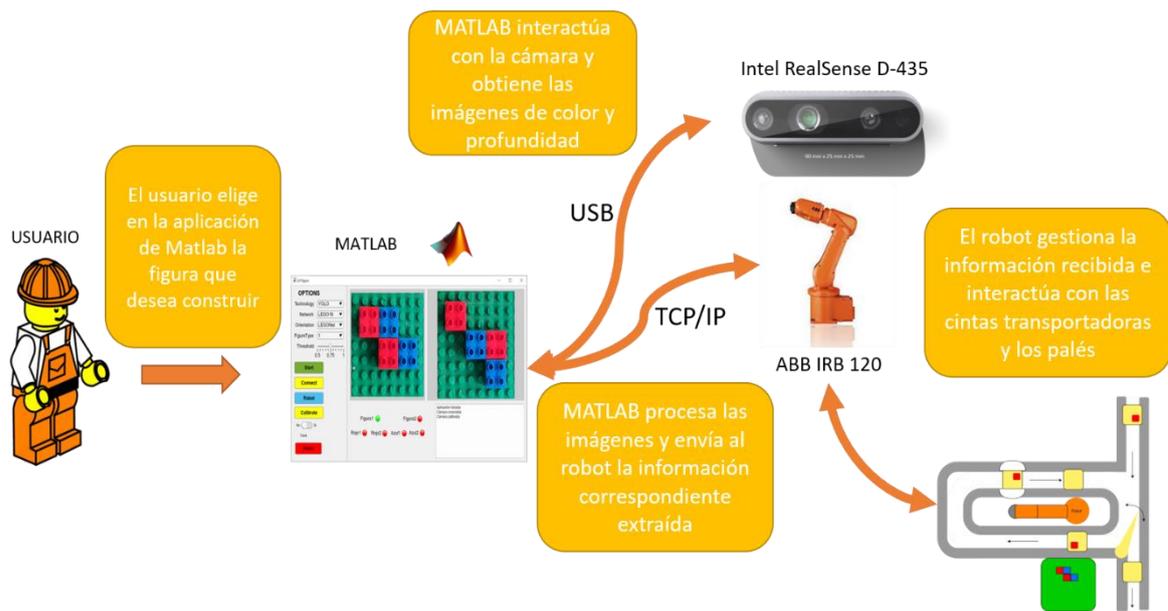


Figure 1: Schematic diagram of system operation

The process begins with a user choosing the figure to build through the MATLAB application. Once chosen, MATLAB communicates with the robot to indicate the start of construction of one of the figures and the robot is ready to divert a pallet to its area for inspection. Once the pallet reaches the indicated position, the robot communicates it to the computer and the computer interacts with the camera to obtain images for study. Once analyzed, the presence of a LEGO piece is determined, its location and if it corresponds to the figure under construction. If so, the robot will proceed to pick up the piece and move it to its corresponding position. If not, the pallet will be released from its blocking position for the incorporation to the main conveyor. Once the piece has been placed, the robot will return to its initial position and will proceed to divert a new pallet if the figure has not been

completed. If the figure has been completed, the robot will wait for further orders from the computer.

4. Results

This project does not have numerical results as such. The main objective was to build the LEGO figures using the robot, the camera and the conveyor belts together from the artificial vision system developed by Ignacio Ortiz de Zúñiga Mingot.

Numerous tests were performed throughout the project and currently the task is completed with a 100% success rate. Therefore, it can be concluded that the objective was met. The operator only has to choose the figure he wants to build. After that the robot, MATLAB and the camera will work together to build it in the target area and the operator only has to supervise the assembly of the figure.

5. Conclusions

Regarding image processing, different methods for the integration of the artificial vision system were analyzed, starting from a more flexible approach. After discarding this approach due to its lack of precision, a customized system for the robot was obtained, with high precision and robustness for the project. However, this involves the initial calibration of all points of the pallet, to ensure that the robot is able to pick up the part accurately and without damaging it or the robotic arm.

With regard to the management of the conveyor belts and the robot's movements, it can be concluded that the system structure can be adapted and used in other robots in the ICAI factory. In addition, the use of various functions provides flexibility to the system for easy adaptation to a new environment or the assembly of new figures.

6. Sources

- [1] C.-Y. Tsai, C.-C. Wong, C.-J. Yu, C.-C. Liu y T.-Y. Liu, «A Hybrid Switched Reactive-Based Visual Servo» Marzo 2015. Disponible: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6920015>. [Último acceso: 21 Enero 2021].

- [2] S. Soltan, A. Oleinikov, M. F. Demirci y A. Shintemirov, «Deep Learning-Based Object Classification and Position Estimation Pipeline for Potential Use in Robotized Pick-and-Place Operations» 18 Agosto 2020. Disponible: https://www.mdpi.com/2218-6581/9/3/63?type=check_update&version=1. [Último acceso: 27 Diciembre 2020].
- [3] ABB, «IRB 120» Disponible: <https://new.abb.com/products/robotics/es/robots-industriales/irb-120>. [Último acceso: 10 Enero 2021].
- [4] I. Ortiz de Zúñiga Mingot, «Optimización de un sistema de visión artificial de un robot industrial» Trabajo de Fin de Grado, ICAI.
- [5] «Realsense D435». Disponible: <https://www.intereasense.com/depth-camerad435/>. [Último acceso: 28 Diciembre 2020].

Capítulo 1. Introducción

Actualmente, la industria se ve inmersa en una nueva revolución hacia la Industria 4.0, o también conocida como industria inteligente. La Industria 4.0 combina técnicas avanzadas de producción y operaciones con tecnologías inteligentes como la robótica, la analítica, la inteligencia artificial o el Internet of Things (IoT) [1].

En este contexto, la producción se mueve hacia la automatización y digitalización completa de los procesos. Gracias a la introducción de los elementos de la Industria 4.0, estos procesos de producción son cada vez más eficientes. La comunicación entre distintos componentes de la cadena de producción hace que colaboren sin intervención humana y de un modo más fluido y dinámico.

Hoy en día, existen una gran cantidad de robots industriales en fábricas de todo tipo, con entornos dinámicos y repitiendo la misma tarea de coger un objeto o pieza y colocarla en un sitio diferente, lo que comúnmente se denomina *Pick and Place*. Dichas piezas pueden estar en diversas posiciones dispuestas de múltiples maneras, el robot que opere con ellas ha de ser capaz de detectar esto y actuar en consecuencia. Estas tareas se desarrollaban gracias al uso de múltiples sensores o de complejos cálculos que implicaban una gran dificultad para desarrollar la tarea [2]. Con la llegada de la Industria 4.0, este tipo de tareas se está desarrollando con el objetivo de flexibilizar y dotar de mayor precisión a los sistemas gracias a los sistemas de visión artificial basados en redes neuronales [3].

Con estas condiciones, se van a implantar estas nuevas características en un robot industrial ABB IRB120 [4] instalado en la minifábrica de ICAI, como el de la Figura 1, con el fin de capacitarlo con la habilidad para montar una figura de LEGO en el área de trabajo de la Figura 2. La figura que se construirá será escogida por el operario a partir de dos propuestas. Las figuras serán montadas por el brazo robótico a partir de piezas que reconocerá y recolectará de palés que se moverán por cintas transportadoras dentro de la región de trabajo del robot.



Figura 1: Brazo robótico IRB120 de ABB

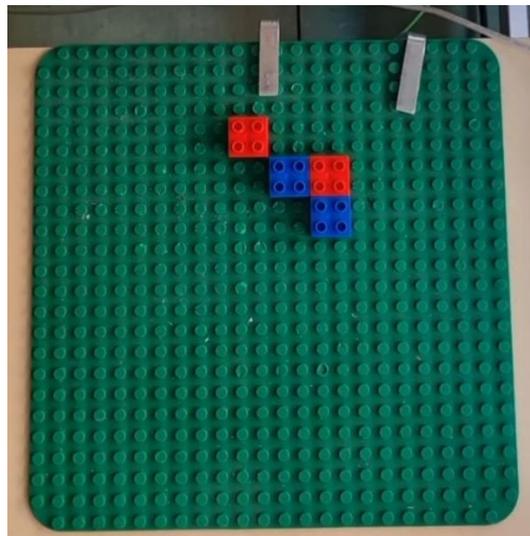


Figura 2: Área de trabajo para el montaje de las figuras con figura de ejemplo



Figura 3: Cámara Intel RealSense D435

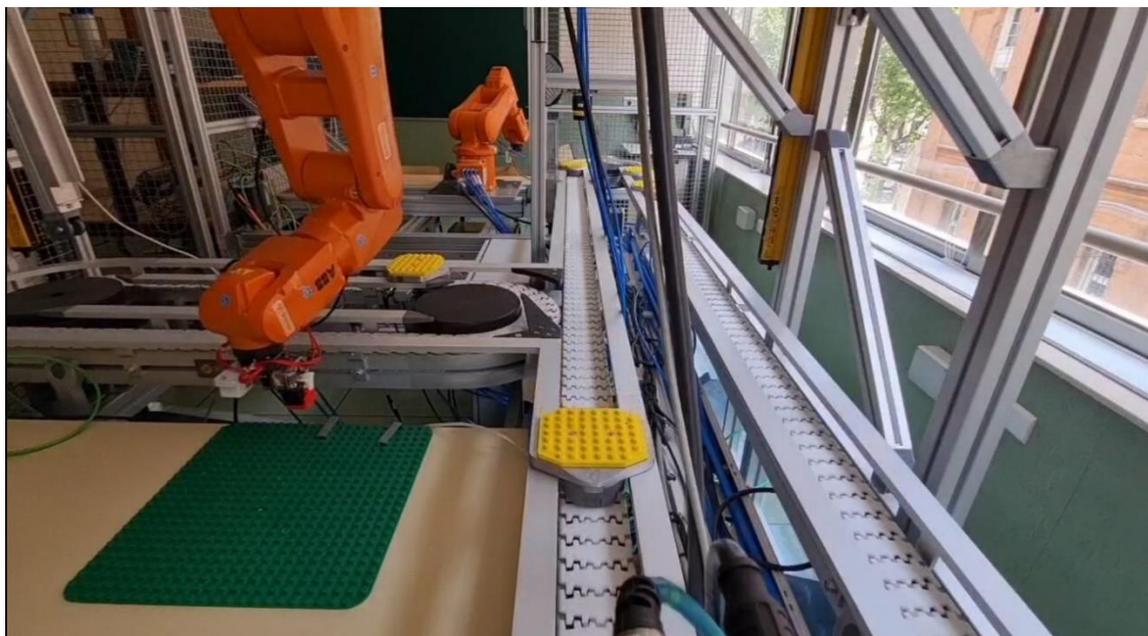


Figura 4: Planta completa con robot, cintas y área de montaje

Este proyecto parte del sistema de visión artificial aplicado a un robot industrial de Ignacio Ortiz de Zúñiga Mingot [5]. Dicho sistema es capaz de reconocer la posición de las piezas, la orientación y su color a partir de las imágenes de color y profundidad obtenidas con la cámara Intel RealSense D435 [6] de la Figura 3. Este se implantará y se integrará en la planta constituida por el robot de la Figura 1, la zona de montaje de la Figura 2 y las cintas transportadoras que se observan en la Figura 4.

1.1 MOTIVACIÓN DEL PROYECTO

A medida que la industria avanza, la automatización de procesos está cada día más presente en los sistemas de producción reales. Así pues, los sistemas crecen en autonomía, flexibilidad y eficacia. Entre otras, dotarlos de la capacidad de comunicarse entre ellos y otorgarles un sistema de visión artificial, son técnicas que hacen que dichas características crezcan.

A pesar de que, en una fábrica, la situación natural es que estos sistemas vayan a trabajar con la intervención mínima de operario gracias a la capacidad autónoma que tienen, se tiene

que aprender a usarlos y a dotarlos de dichas capacidades para saber cómo mejorarlos y cómo trabajan.

Es por ello por lo que, con la intención de integrar nuevos componentes de la Industria 4.0 en ICAI y dar a los alumnos una visión de cómo son los procesos de fabricación, hacia donde avanzarán, como se manipulan, y facilitar su aprendizaje; se propone el proyecto de la implantación de un sistema de *pick-and-place* en un robot industrial con cámara RGB-D.

1.2 OBJETIVOS

Este trabajo parte del que realizó Ignacio Ortiz de Zúñiga Mingot [5]. En su trabajo optimizó el sistema de visión artificial de un robot industrial. El sistema era capaz de identificar y extraer características de piezas de LEGO con una cámara RGB-D. El objetivo de este proyecto se basa en implantar este sistema en uno de los robots industriales de la universidad y que trabaje de forma coordinada con el resto de los elementos de la instalación para construir una figura preprogramada con bloques de LEGO en un área de trabajo fija.

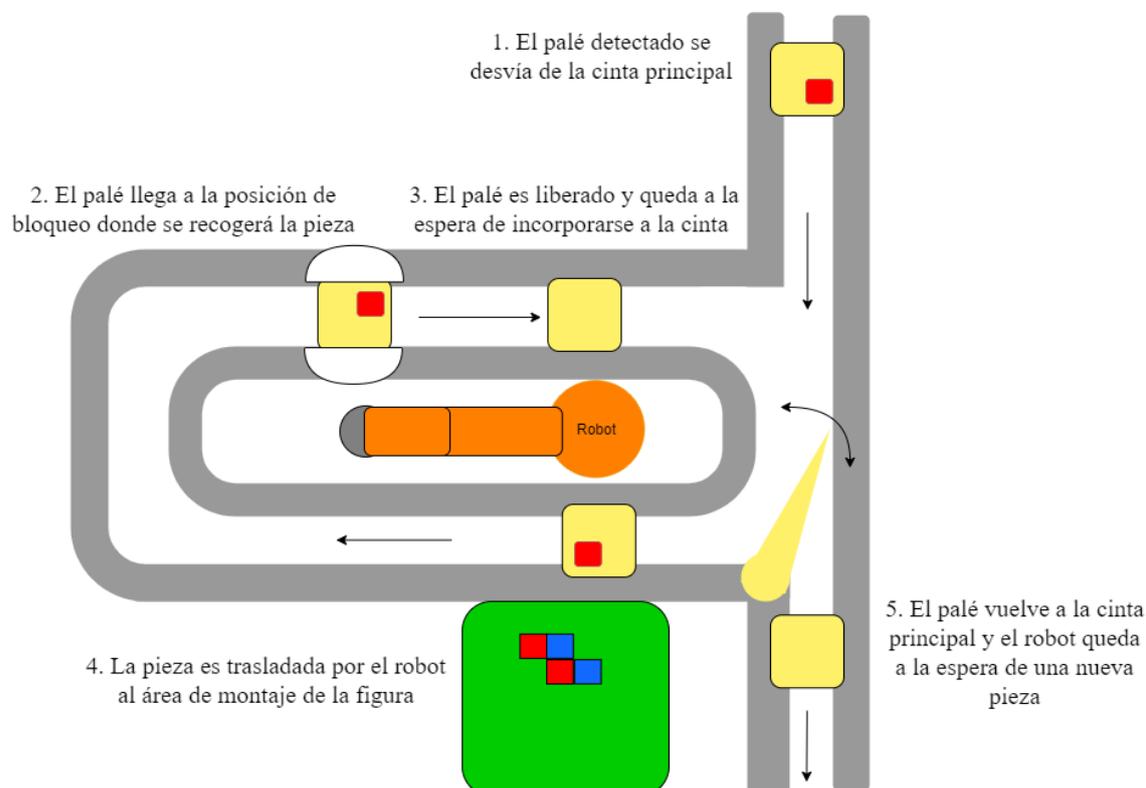


Figura 5: Esquema del proceso

En la Figura 5 se expone visualmente el proceso de montaje de la figura y el recorrido del palé. A continuación, se explica detalladamente el mismo. El proceso comenzaría con la elección, por parte de un operario, de la figura que se desee montar desde el ordenador a través de una interfaz. Tras esto, el robot tendrá que detectar la llegada de un palé por la cinta transportadora a las proximidades de su zona y desviarlo a su zona de trabajo (punto 1 de la Figura 5). El palé llevará una única pieza de LEGO de color y posición en el palé desconocidas.

Una vez desviado el palé y retenido en una posición próxima al robot (punto 2 de la Figura 5), este tiene que posicionarse sobre él y, gracias al sistema de visión artificial implantado, detectar el color de la pieza y su posición.

Si la pieza es necesaria para la construcción de la figura que se solicita, el robot debe cogerla y ponerla en su sitio correspondiente (punto 4 de la Figura 5). Durante el traslado de la pieza, se liberará el palé bloqueado (punto 3 de la Figura 5), pero no se incorporará a la cinta principal hasta que se detecte una ocasión segura para ello (punto 5 de la Figura 5). Cuando el robot haya dejado la pieza y vuelto a la posición de origen, se reinicia el proceso quedando a la espera de un nuevo palé.

Por otro lado, si la pieza detectada no era necesaria o no se detecta ninguna, el robot liberará el palé de la posición de bloqueo, pero lo mantendrá en la cinta transportadora de su área sin incorporarlo a la cinta principal hasta que se asegure que no colisionará con otro palé que circule por esta.



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

CAPÍTULO 1: INTRODUCCIÓN

Capítulo 2. Estado de la cuestión

Con el avance de la tecnología, día a día se han abierto las puertas a nuevos procesos y a la mejora de otros, gracias a distintas técnicas que antes eran imposibles de implantar debido a la limitación de su contexto temporal. Hasta hace no mucho, las tareas que realizaban distintos robots en una cadena de montaje eran muy poco flexibles y requerían de muchos sensores y cálculos [7].

Actualmente, estos sistemas se han flexibilizado y automatizado gracias a la introducción de la visión artificial, que suponía una clara mejora. Estos sistemas han pasado a ser una parte importante dentro de la Industria 4.0 gracias al gran potencial que presenta y que está demostrando gracias a las redes neuronales profundas y su aplicación en la industria.

2.1 REDES NEURONALES

Dentro de los sistemas de visión artificial el uso de redes convolucionales para el procesamiento de las imágenes y la detección de objetos es una técnica muy extendida en evolución.

Los problemas principales que presentan y que se busca optimizar es la gran cantidad de imágenes y capacidad computacional que requiere su entrenamiento y evaluación. Por otro lado, según el contexto y el *hardware*, la velocidad real de su aplicación debe cumplir los requisitos necesarios.

Estas redes pueden estar compuestas por diversas arquitecturas de capas cuyo desempeño final puede variar [3] dando buenos resultados y soluciones a uno de los problemas mencionados, pero fallando en otro. Con el paso del tiempo y el desarrollo de estas, ciertas configuraciones de arquitecturas predominaron sobre el resto debido a su mayor eficacia y eficiencia en cuanto a la necesidad de recursos computacionales. Ejemplos de estas son AlexNet [8] expuesta en la Figura 6, diseñada por Alex Krizhevsky y VGG16 [9] expuesta en la Figura 7, diseñada por K. Simonyan y A. Zisserman. Estas arquitecturas tienen la capacidad de clasificar hasta 1000 clases de objetos distintos presentes en imágenes con una precisión del 84,7% y 92,7% respectivamente en sus primeros desarrollos.

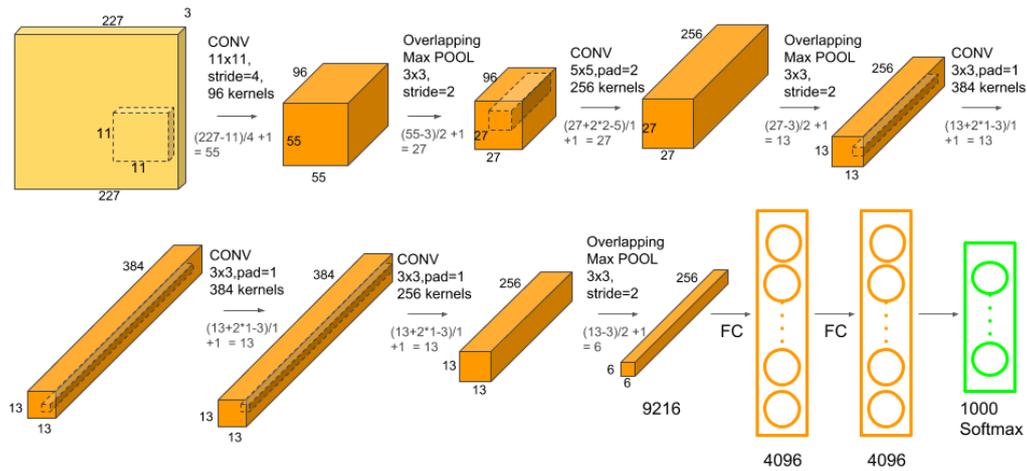


Figura 6: Arquitectura de la red neuronal convolucional AlexNet (Fuente: [10])

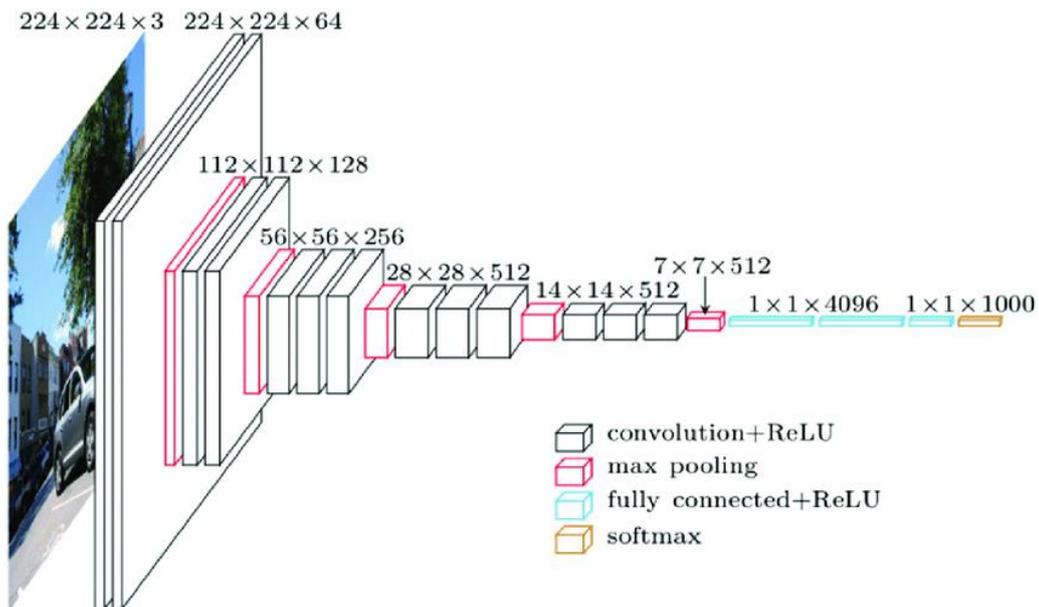


Figura 7: Arquitectura de la red neuronal convolucional de VGG16 (Fuente: [11])

Para la detección de los objetos en imágenes y la extracción de su ubicación en esta, se crearon arquitecturas de redes como las *Region-proposal Convolutional Neural Network* (R-CNN), las *Faster Region-proposal Convolutional Neural Network* o las *You Only Look Once* (YOLO). Estas se basan en clasificadores de objetos como los mencionados anteriormente y como se ha comentado, añaden la capacidad de localizarlos en la imagen.

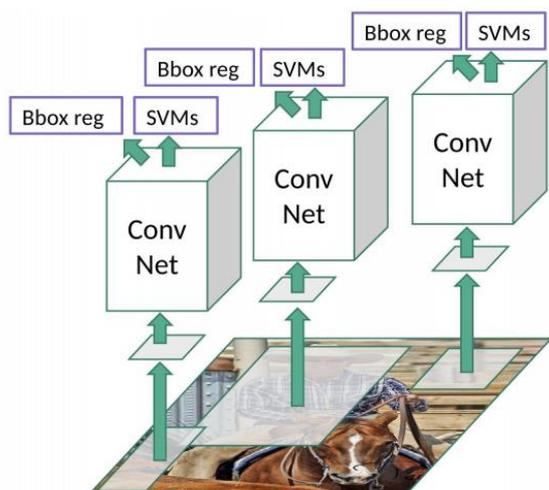


Figura 8: Proceso R-CNN (Fuente: [12])

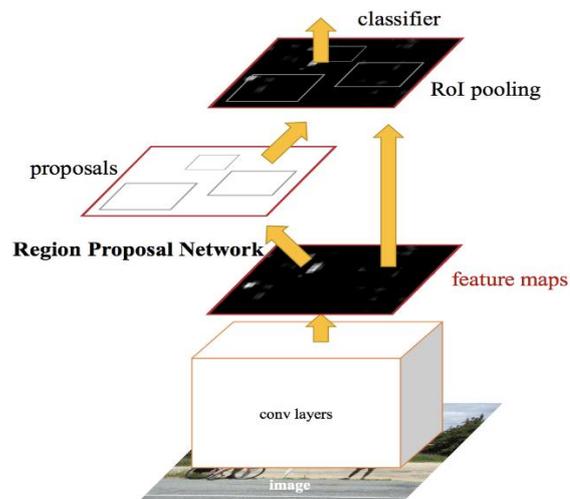


Figura 9: Proceso Faster R-CNN (Fuente: [12])

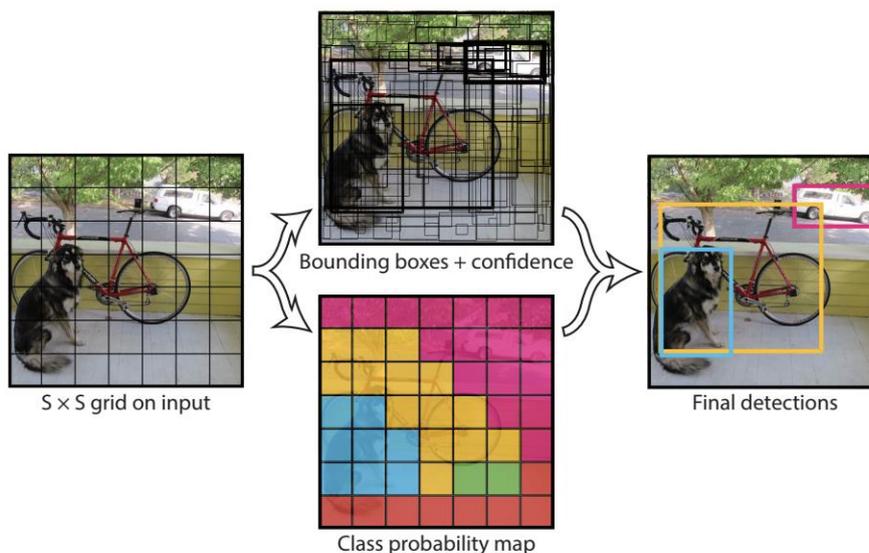


Figura 10: Proceso YOLO (Fuente: [12])

Como introducción a estas arquitecturas se comenta a continuación el proceso que siguen estas descrito con mayor detalle en [12].

La detección de objetos empleando el método de R-CNN, expuesto en la Figura 8, consiste en la propuesta de 2000 regiones de la imagen para ser analizadas y su posterior procesado

para la detección de objetos en cada una de estas regiones. Este proceso conlleva una elevada carga computacional que se reduce con el método de Faster R-CNN.

Como se ha comentado el proceso de Faster R-CNN reduce la carga computacional que conlleva R-CNN y aumenta enormemente la rapidez. Esto es debido a que la imagen de partida sirve como entrada a una primera red convolucional con la que se obtendrán las regiones de interés que se analizarán más tarde para la detección de objetos en estas. En vez de tener 2000 regiones que analizar, se reduce a las que extraiga la primera red convolucional. Visualmente, el proceso se expone en la Figura 9.

Por último, el método de YOLO divide la imagen inicial en una malla de dimensiones $S \times S$, para cada celda detecta N *bounding boxes* que contienen un objeto dentro de ellas y le asigna a cada una un nivel de confianza y , por último, de las N *bounding boxes* detectadas previamente, se excluyen las que tienen asociadas un nivel de confianza más bajo. Visualmente, el proceso se expone en la Figura 10.

Estas técnicas son empleadas con diversos objetivos y en distintas tareas en la robótica industrial. una de estas es el conocido comúnmente como *pick and place*, donde el robot debe detectar y coger el objeto que desea (*pick*) y colocarlo en la ubicación que requiera (*place*), una tarea muy usual en sistemas de producción o clasificación.

Cabe destacar que la tarea del *pick and place* se ha desarrollado enormemente en competiciones como el *Amazon Robotics Challenge* (ARC) de 2016 [13] y 2017, donde múltiples universidades y equipos de investigación desarrollaron sus propios sistemas de *pick and place* clasificando múltiples objetos en distintos contenedores ubicados en diferentes posiciones.

2.2 REDES NEURONALES EN SISTEMAS DE *PICK AND PLACE*

Para la tarea de *pick and place* se han de tener en cuenta diversos factores como que el robot sea capaz identificar objetos, localizarlos. Tras esto, una de las cuestiones más importantes es cómo debe coger el objeto y, por último, dónde debe situarlo y cómo.

Dividiendo la tarea en sus dos partes principales, coger la pieza deseada y dejarla, el sistema de visión artificial es clave en la primera etapa del proceso, donde se han de localizar los objetos y saber cómo cogerlos. Para esta primera fase se pueden emplear sistemas de menor

o mayor complejidad dependiendo de las distintas clases de objetos que se vayan a clasificar, si se quiere coger de un modo especial y de la cantidad de herramientas de las que dispone el robot para coger el objeto. Existen sistemas basados en modelos 3D de los objetos, otros más sencillos que emplean solamente su ubicación, y otros agnósticos a los objetos que cogen.

2.2.1 USO DE MODELOS 3D

Existen sistemas que basan la tarea de coger los objetos en modelos 3D de estos para buscar en ellos la zona óptima de agarre y qué herramienta emplear.

En [14] se emplea un sistema de redes neuronales que permite identificar objetos y localizarlos. Además, construye una nube de puntos para cada objeto identificado que servirá de base para analizar cuál será el mejor agarre. En la propuesta que realizan, emplean dos algoritmos que proporcionarán los datos necesarios para saber qué objeto de los identificados se debe coger.

- El primer algoritmo emplea la técnica de HAF (*Height Accumulated Features*), para destacar las zonas de cada nube de puntos con mayor altura, es decir, para cada objeto, extrae la zona de mayor altura. Las características extraídas se pasan después a una SVM (*Support Vector Machine*) que analizará y obtendrá cuál de los objetos identificados es el óptimo para coger.
- El segundo simplifica la nube de puntos a un cilindro o un plano según a lo que más se ajuste esta y con ello se discierne sobre la viabilidad del uso de una herramienta u otra. Con los dos métodos y experimentando con distintos objetos llegaron a establecer qué método de agarre emplear para cada objeto en concreto.

En [15], se parte de modelos 3D de objetos y se propone un método para la obtención del mejor agarre que puede realizar una herramienta con forma de mano humana. Para cada modelo de los objetos, analizaban múltiples posiciones de la mano donde podría efectuar un agarre posible, es decir, donde pudiese cerrarse la mano para coger el objeto. Esto es lo que se visualiza en la Figura 11. También se puede analizar cómo se adapta la mano al objeto o viceversa, es decir, comprobando las colisiones entre mano y objeto. De este análisis se

obtienen múltiples agarres posibles a partir de los cuales se establece el mejor basándose en la estabilidad del agarre y la preferencia en cuanto al contacto con superficies planas antes que curvas. El problema que presenta es que requiere de una gran capacidad computacional y su aplicación en tiempo real requiere de entrenamientos previos. En [16] solucionan este problema analizando la intersección de los volúmenes de la mano y del objeto en distintas posiciones, lo que requiere menor carga computacional y permite su aplicación en tiempo real.

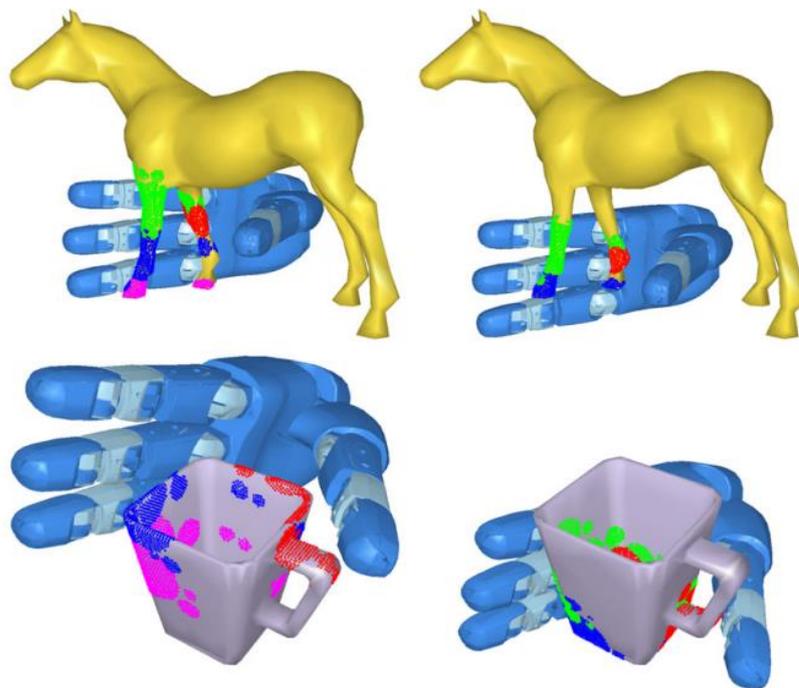


Figura 11: Análisis de métodos de agarre a partir de modelos 3D de objetos (Fuente: [15])

2.2.2 MÉTODO AGNÓSTICO

Existen sistemas agnósticos a los objetos, esto quiere decir, que no dependen del objeto identificado, analizan la imagen buscando el objeto con mayor probabilidad de ser agarrado con éxito y tras cogerlo, lo identifican.

Esto es lo que se plantea en [17] y se visualiza en la Figura 12, partiendo de unas técnicas de agarre básicas predefinidas, una red neuronal analiza la imagen en busca de la zona que

mejor se adapta a uno de los agarres predefinidos y da sus probabilidades de éxito. Al tener dos herramientas de agarre, la imagen es procesada en dos redes neuronales en paralelo, una que analizará las mejores zonas de actuación de una de las herramientas y la segunda red para la segunda herramienta. De los datos obtenidos, la zona con mayor probabilidad de éxito de ambas herramientas será donde se efectuará el agarre con la herramienta asociada a dicha zona.

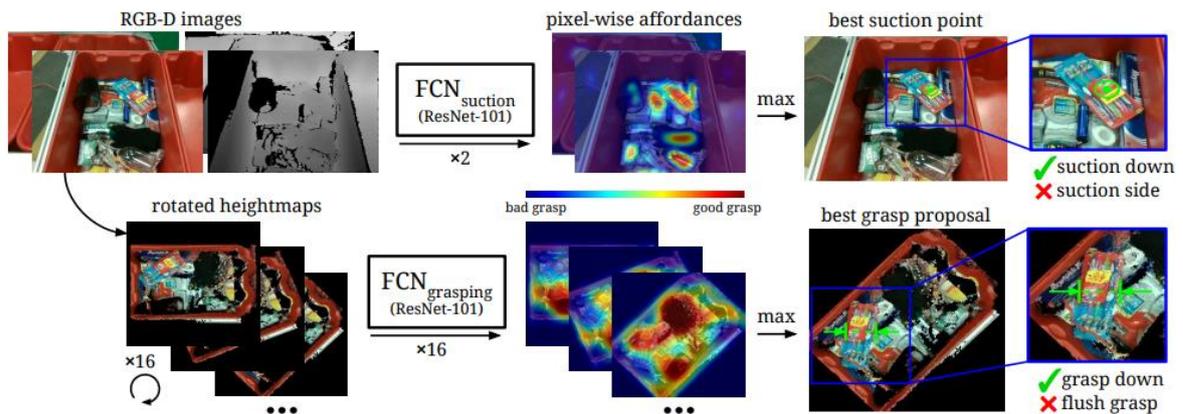


Figura 12: Esquema de análisis agnóstico para planificación de agarres (Fuente: [17])

Muchos de los sistemas mencionados anteriormente están limitados a los objetos con los que han sido entrenados, pero una característica del último mencionado es que, al hacer el sistema agnóstico al objeto, se pueden introducir objetos ligeramente modificados y compararlos con los que ya se conocen, como se propone en [18]. Estos que serán “nuevos” en el sistema se compararán con imágenes de los que se tienen datos. Las imágenes del objeto nuevo y del conocido se procesarán por separado en una red neuronal que extraerá las características principales de ambas. Estas características se compararán y se extraerá la similitud con una función que relacionará ambas imágenes. Según el resultado se podrá asumir que el objeto que en principio era nuevo ahora pertenece al grupo con el que se ha comparado o no, dando una mayor robustez al sistema frente a posibles cambios en las imágenes analizadas.

2.3 HERRAMIENTAS DE AGARRE

Una parte importante de la tarea de coger los objetos es la herramienta instalada en el extremo del robot que interactuará físicamente con ellos, de esta depende como se entrenará el sistema, que técnicas se aplicarán para su desarrollo y la flexibilidad que tenga el robot para agarrar diversos objetos en diversas posiciones. Es por ello por lo que existe especial interés en el diseño de un sistema de agarre flexible, eficiente y robusto para múltiples objetos.

2.3.1 HERRAMIENTAS ANTROPOMÓRFICAS

Existen diseños de sistemas antropomórficos como los comentados en [15] [16] y expuestos en la Figura 13 y la Figura 14 respectivamente. Estos sistemas están orientados a su uso en un trabajo colaborativo entre personas y robots y puede ser clave para el diseño de robots antropomórficos. Su aplicación está limitada debido a que requieren de gran capacidad computacional, pero su desarrollo está abriendo las puertas a su uso en tiempo real gracias a las técnicas comentadas anteriormente. Además, le darían un componente más humano al robot en tareas colaborativas con operarios.

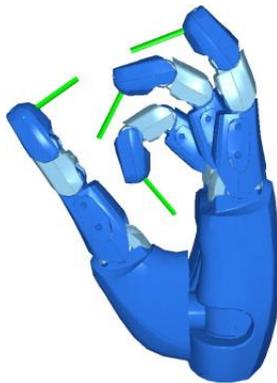


Figura 13: Modelo 3D de la herramienta propuesta en [15] (Fuente: [15])



Figura 14: Robot con herramienta de agarre antropomórfica propuesto en [16] (Fuente: www.nasa.gov)

2.3.2 HERRAMIENTAS PARA OBJETOS DE FORMA Y TAMAÑO ESPECÍFICO

Con el objetivo de realizar tareas de gran precisión u orientadas a objetos específicos, se han diseñado distintas herramientas.

En el caso de [19], se diseña una herramienta específica para el agarre preciso de objetos con forma cuadrada. Para ello el sistema no emplea visión artificial, si no 8 sensores laser que miden distintos parámetros que permitirán, a través de cálculos estimar la posición del objeto y su orientación para proceder a cogerlo. Este diseño dotó al robot de una precisión con un error medio de 0.73mm en la posición y 0.104° en la orientación.

Por otra parte, en [20] se propone el diseño de una herramienta para manipulación de objetos microscópicos que se expone en la Figura 15. En este, el problema recae en la colocación de los objetos, ya que, tras el agarre, estos se adhieren a la herramienta y resulta complicado dejarlos en su posición objetivo con precisión. El diseño que plantean consigue solventar este problema con una precisión de $0.45 \pm 0.24 \mu\text{m}$. Para el reconocimiento de los objetos emplean un sistema de visión artificial basado en el detector de bordes de Canny y la transformada de Hough. Este diseño y los resultados de precisión que plantea tras los experimentos dan paso a la automatización de los procesos de *pick and place* a escala microscópica.

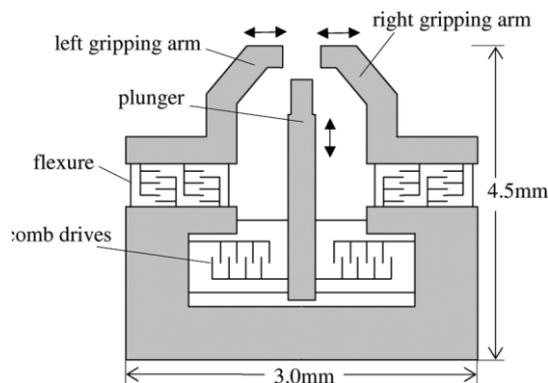


Figura 15: Herramienta propuesta para trabajos microscópicos (Fuente: [20])

2.3.3 HERRAMIENTAS MÚLTIPLES

Atendiendo al problema de la diversidad de objetos y formas que pueda encontrar un robot, se plantea la posibilidad de equiparlo con varias herramientas para que según el objeto y su posición el robot emplee la más adecuada. En [21] se analiza el diseño del sistema de agarre ganador del ARC de 2017 con dos herramientas implantadas, una pinza de succión y otra mecánica. Con ello, las posibilidades de éxito en el agarre de un objeto no están limitadas a una sola herramienta y demuestra su utilidad con elevadas tasas de éxito en el agarre para ambas herramientas y múltiples objetos.

Por otra parte, su uso requiere un análisis adicional debido a que se ha de elegir qué método emplear para cada objeto. A pesar de esto último, este sistema plantea más ventajas que inconvenientes y versiones de este diseño han sido empleadas en múltiples propuestas para las competiciones del ARC como la ganadora de 2016 y la de la Figura 16, ganadora de 2017.

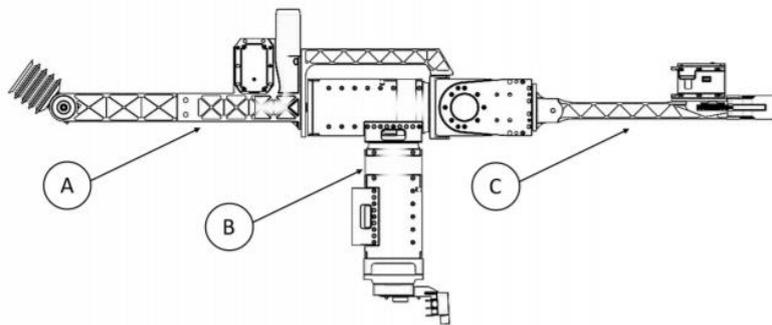


Figura 16: Modelo de la herramienta con (A) Pinza de succión (B) Motor para el cambio de herramienta y (C) Pinza mecánica (Fuente [21])

En cuanto a la segunda fase del proceso, la tarea de dejar el objeto obtenida en su sitio correspondiente puede ser más o menos sencilla según la precisión que se desee. La mayoría de los sistemas de clasificación no requieren de complejos métodos para dejar el objeto en un contenedor, se lleva el robot a una posición próxima y se suelta la pieza.

Para este proyecto el objetivo no consistirá en clasificar múltiples objetos de diversas formas. Razón por la cual el sistema de agarre que se empleará será una pinza capaz de coger las piezas cuadradas de LEGO. Esta es como la que se muestra en la Figura 17, la Figura 18, la Figura 19 y la Figura 20. Por otro lado, el sistema no requiere identificar múltiples

superficies de agarre. En este caso se obtendrá la posición de la pieza y con ello se procederá al agarre de esta y a su traslado a su ubicación correspondiente.

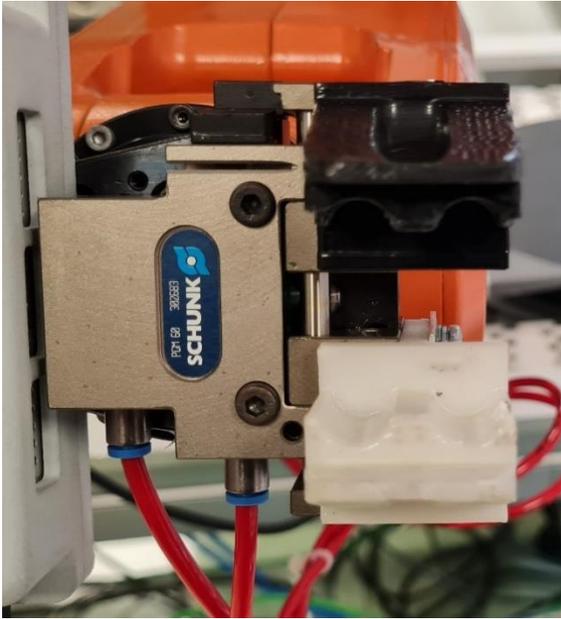


Figura 17: Vista inferior pinza de agarre
abierta



Figura 18: Vista inferior Pinza de Agarre
Cerrada

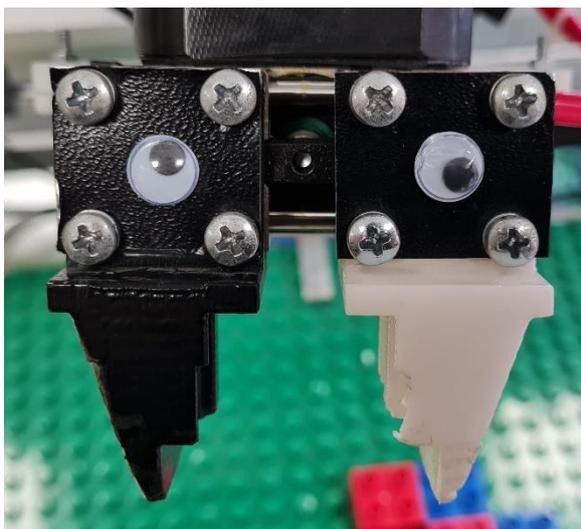


Figura 19: Vista frontal Pinza de Agarre
abierta

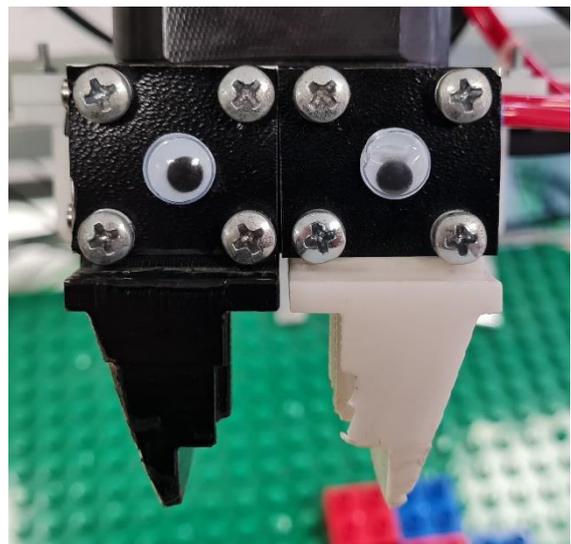


Figura 20: Vista frontal Pinza de Agarre
cerrada

Ignacio Ortiz de Zúñiga Mingot, en su TFG [5], aplicó los métodos y arquitecturas de redes neuronales mencionados para optimizar un sistema de visión artificial cuyo propósito era la detección de piezas de LEGO y la extracción de sus características. Las conclusiones de sus experimentos muestran el gran potencial del uso del método YOLO basado en la arquitectura VGG16, ligeramente modificada y reentrenada, para la ubicación y extracción de características de las piezas; y el uso de una arquitectura basada en AlexNet para identificar el ángulo de rotación de las piezas en la imagen. Estos sistemas se aplicarán en este proyecto para cumplir con los objetivos especificados en el apartado 0.

Capítulo 3. Arquitectura del sistema

Como se ha comentado anteriormente, el objetivo final de este proyecto es que el brazo robótico construya varias figuras de LEGO determinadas de manera autónoma, a partir de las piezas que le lleguen en palés que circulan por cintas transportadoras.

Para el desarrollo de este proyecto existen tres elementos clave, el brazo robótico IRB 120 de la marca ABB, la cámara Intel RealSense D435 y el software de MATLAB. En este capítulo se va a introducir el papel de cada uno de ellos en el sistema global y las comunicaciones entre ellos describiendo el proceso completo. En la Figura 21 se muestra un esquema a modo de resumen.

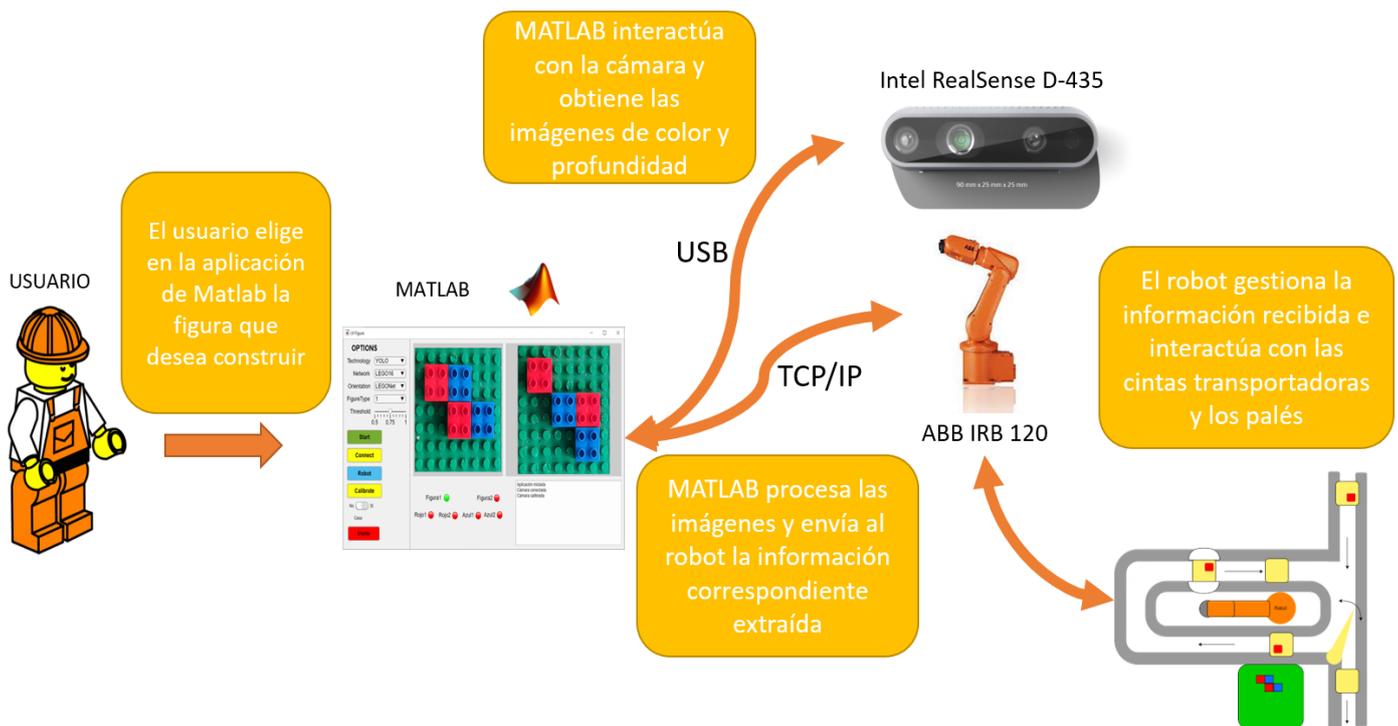


Figura 21: Esquema del proceso de construcción de una figura

Al arrancar la aplicación desarrollada en MATLAB, lo primero que se hace es comprobar que la cámara está conectada a través de un cable USB y calibrarla para obtener correctamente las imágenes de profundidad y sus medidas.

En el caso de no estuviese conectada, salta un error y no se puede continuar con el desarrollo. Si todo está conectado correctamente se prosigue con la aplicación.

Por otro lado, una vez arrancado el robot, e iniciado el programa en él, este queda a la espera de la conexión con MATLAB a través del protocolo TCP/IP.

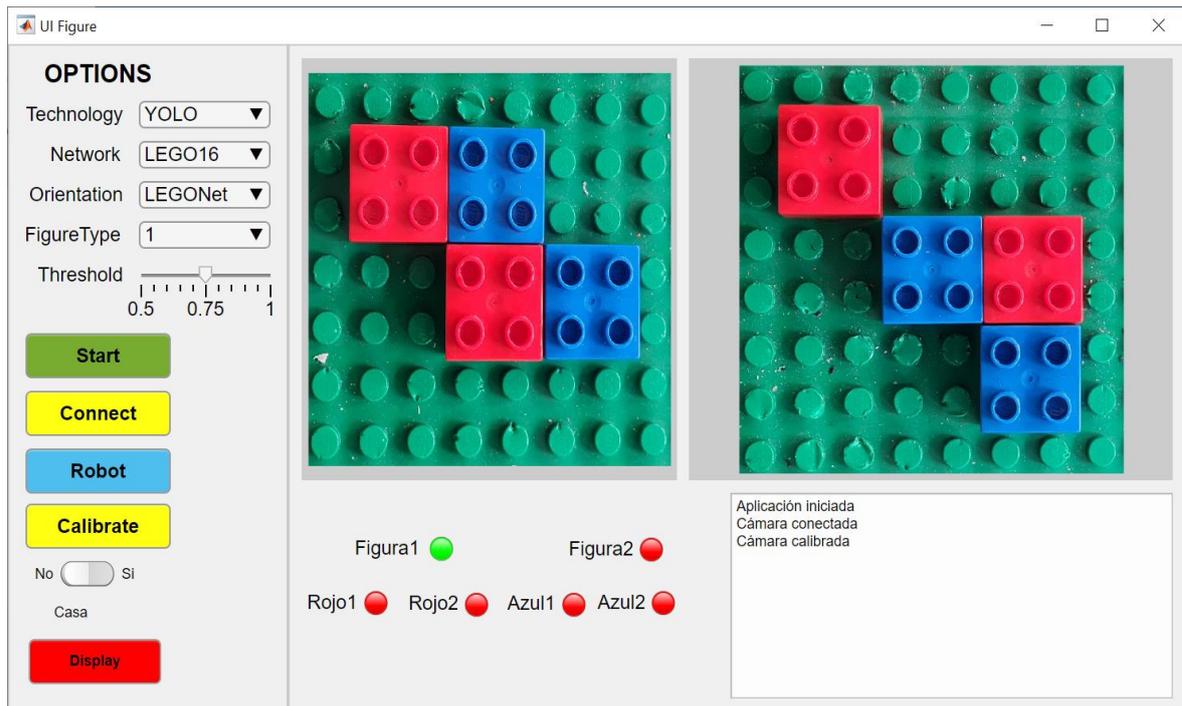


Figura 22: Interfaz inicial de la aplicación de MATLAB

El usuario visualizará una interfaz como la expuesta en la Figura 22 y que se explica detalladamente en el Anexo I. Con la interfaz expuesta, el usuario deberá pulsar en el botón “Connect” para completar la conexión entre el robot y MATLAB. Con ello, el robot queda a la espera de instrucciones del ordenador que le indiquen que ha comenzado la construcción de una figura. El usuario deberá escoger que tecnología desea emplear tanto para la detección de las figuras, como para la extracción de la orientación de las piezas. Además, tendrá que definir cuál de las dos figuras expuestas en la interfaz desea construir. Tras ello pulsará el botón “Start”

Tras pulsar el botón MATLAB le comunicará al robot que ha comenzado la construcción de una figura y este activará retenedores en la cinta principal para detectar correctamente la

aproximación de un palé a la zona de desvío. Cuando esto ocurra, el palé será desviado a la cinta próxima al área del robot y bloqueado en una zona cercana a este.

Tras bloquear el panel, el robot le comunicará a MATLAB que está en la posición indicada para capturar imágenes del palé bloqueado y MATLAB obtendrá las imágenes de la cámara. Estas imágenes se procesarán, se determinará si hay una pieza en el palé o no y si se debe de coger o no. Además, se mostrarán las imágenes con su respectivo análisis al usuario a través de la interfaz, como se observa en la Figura 23 y la Figura 24.

Si no es necesario coger la pieza, debido a que no pertenece a la figura que se está construyendo o bien a que ya se han cogido las necesarias de ese tipo, el robot liberará el palé y volverá a bloquear la cinta principal para desviar uno nuevo.

Por otro lado, si hay que coger la pieza, MATLAB se lo transmitirá al robot junto con las coordenadas de esta y características de la misma, como su color o la figura a la que pertenece.

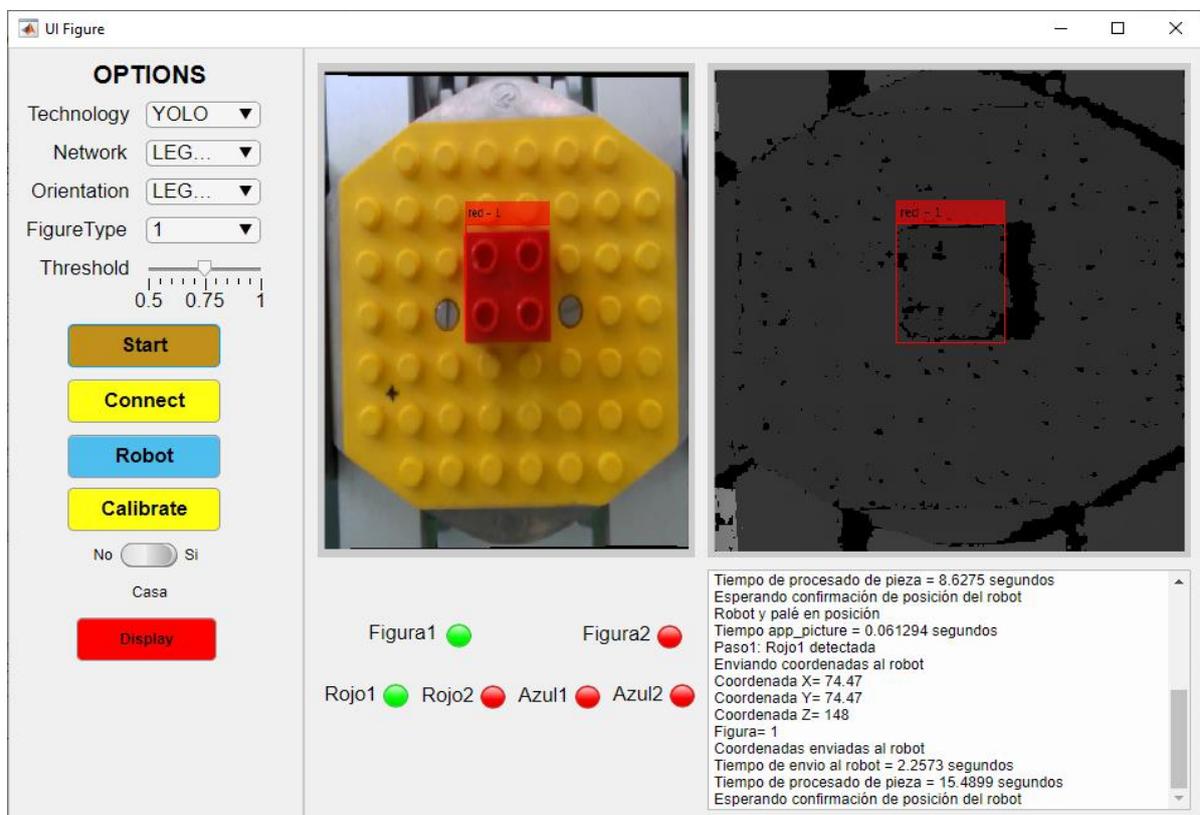


Figura 23: Interfaz tras análisis de palé con pieza

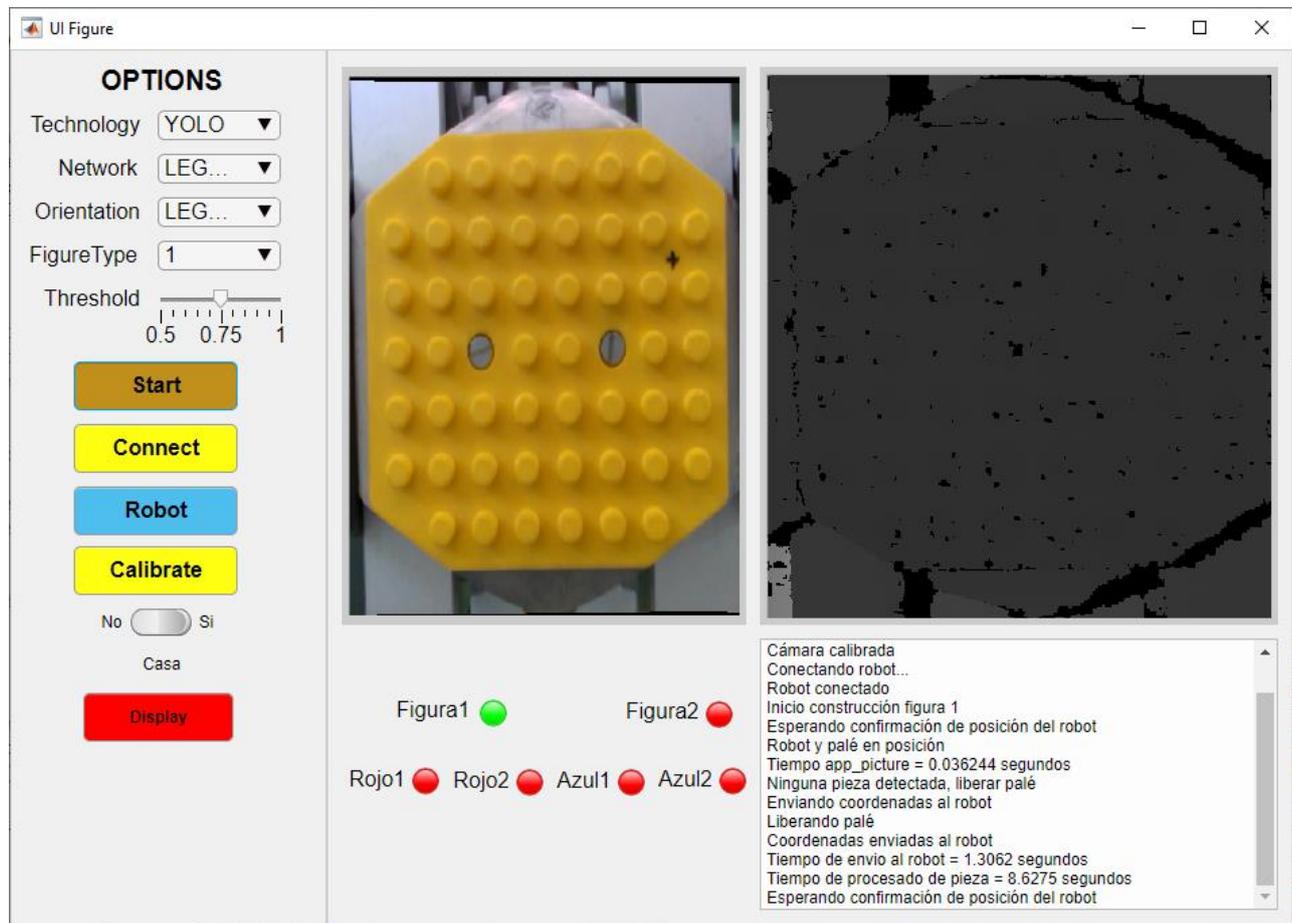


Figura 24: Interfaz tras análisis de palé sin pieza

El robot recibe la información y con ello se traslada a la ubicación de la pieza para cogerla y llevarla a su correspondiente ubicación. En el proceso de traslado de la pieza, además, se liberará el palé de la posición de bloqueo.

Una vez colocada la pieza, el robot volverá a su posición de origen y activará los retenedores de la cinta principal si aún no se ha terminado de construir la figura. En cambio, si la figura ha sido finalizada, el robot aguardará en la posición de origen a la espera de nuevas órdenes de MATLAB.

Cabe destacar que, si durante el proceso las comunicaciones entre la cámara y MATLAB o el robot y MATLAB se interrumpen, el sistema da un error y se detiene la construcción de la figura.

Capítulo 4. Captura de imágenes y detección de piezas

A lo largo de este capítulo se va a desarrollar el proceso de detección de piezas y su localización en el sistema de coordenadas del robot a partir del sistema de visión artificial de partida. Se detallarán las aproximaciones iniciales al problema y la evolución para su óptima función.

Para la captura de las imágenes se empleará la cámara Intel RealSense D435. Esta está ubicada en la muñeca del robot IRB 120, próxima a la pinza de agarre como se puede ver en la Figura 25. Se instaló en la muñeca del robot gracias a una carcasa que actúa como protección ante posibles colisiones.

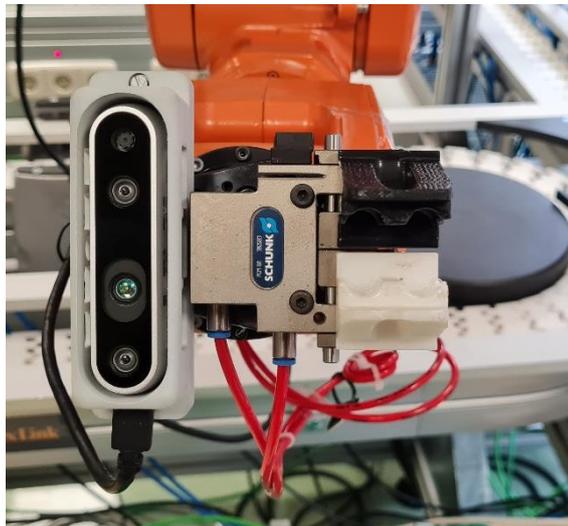


Figura 25: Vista inferior de la pinza y la cámara en la muñeca del robot

4.1 SISTEMA DE PARTIDA

El sistema de visión artificial del que parte este proyecto es el de Ignacio Ortiz de Zúñiga Mingot, que se expone en [5]. La interfaz de partida incluía una tabla con los resultados del análisis de las imágenes. Esto es, para cada pieza detectada obtenía, el color de esta, el nivel de confianza con el que ha identificado la pieza, las coordenadas del centro de la *bounding box* en la que está la pieza, las dimensiones de la *bounding box*, el ángulo de orientación de la pieza en la imagen y la altura de esta. La interfaz de inicio se expone en la Figura 26.

Como en la versión final de la aplicación, al principio se comprueba la conexión con la cámara, que está conectada, y con ello se calibra la cámara para que la imagen de profundidad se obtenga con las medidas correctas. Para iniciar el sistema y comenzar con el análisis de las piezas se debe elegir el tipo de tecnología en la que se basará para clasificar las piezas (AlexNet o VGG16), la tecnología para localizarlas en la imagen (R-CNN, Faster R-CNN o YOLO), la forma en la que se obtiene la rotación (AlexNet, VGG16 o la Transformada de Hough) y pulsar el botón “*Start*”. Entonces, la cámara empieza a captar imágenes continuamente y a procesarlas con los parámetros indicados, el proceso es casi inmediato y se muestran los resultados en pantalla continuamente como se puede observar en a Figura 27.

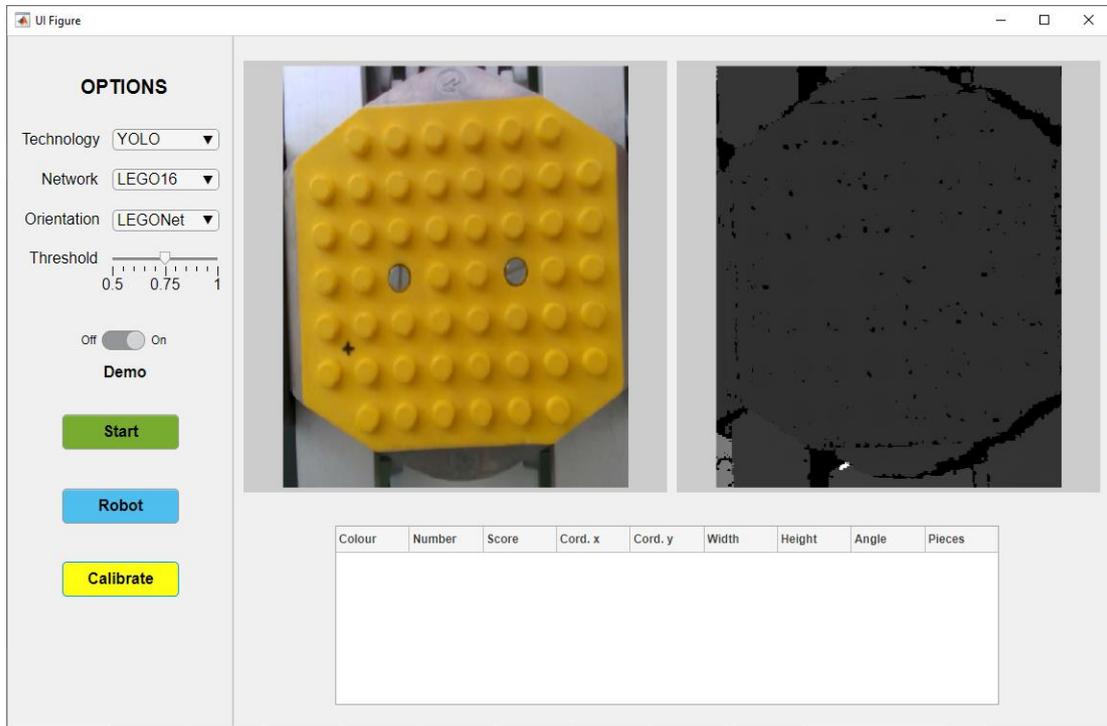


Figura 26: Interfaz inicial de la aplicación de partida

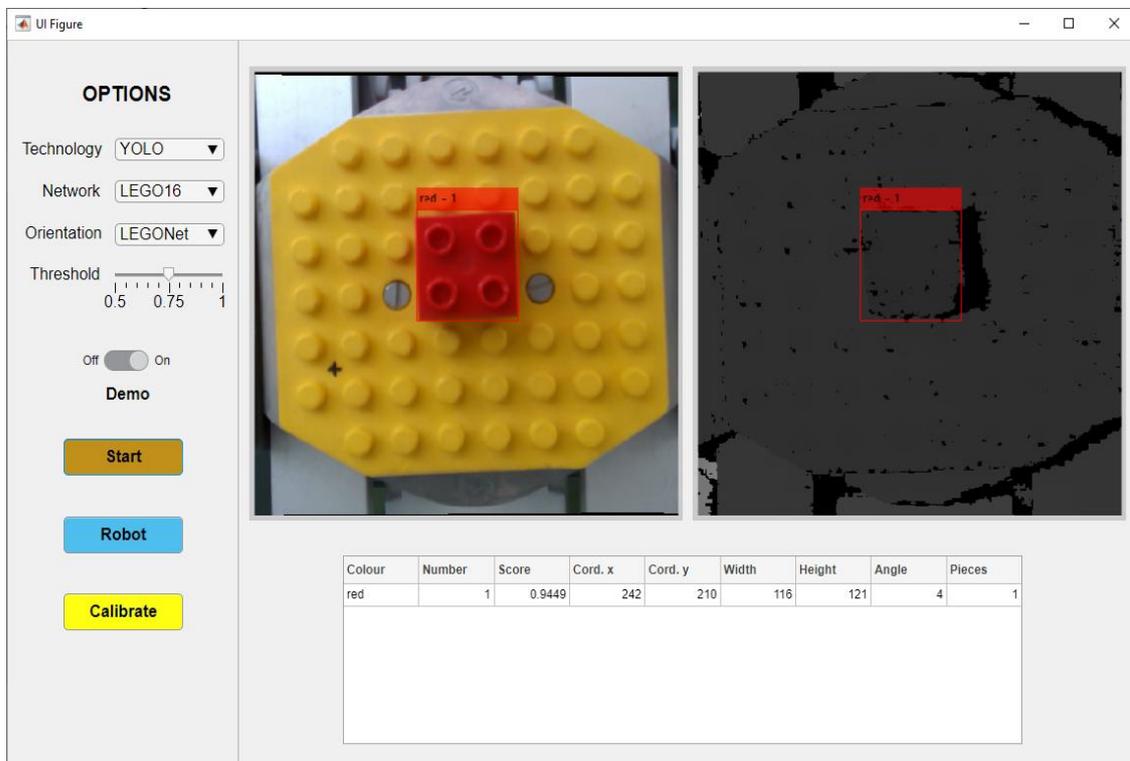


Figura 27: Resultado del procesado de imágenes del sistema partida

Para poder comunicar MATLAB con la cámara, Ignacio creo una clase en MATLAB a través de la cual el usuario que modifique los códigos pueda acceder a las funciones de la cámara de una manera intuitiva. Esta clase se creó como herencia de la clase Pipeline del wrapper de MATLAB desarrollado por Intel [23], cuyas funciones eran menos intuitivas para su uso.

Cabe destacar que, aunque la nueva aplicación de MATLAB empleada, que se desarrolla en el Anexo I, ha evolucionado hacia un punto en el que se centra en el proceso de construcción de las figuras, la aplicación inicial que se ha expuesto se continuó empleando para la extracción de las coordenadas de las piezas en la imagen directamente desde la interfaz.

De las imágenes extraídas por la cámara se obtienen las coordenadas de las piezas detectadas en el sistema de coordenadas de la cámara, con el origen en la esquina superior derecha de la imagen, y con las medidas de la cámara, es decir, en píxeles. Por lo tanto, se deben realizar dos operaciones: cambiar el sistema de medida de píxeles a milímetros, que es la magnitud con la que opera el robot; y cambiar el sistema de coordenadas del que se emplea con la cámara al robot.

A continuación, en la Figura 28 se muestra un ejemplo de imagen analizada con el origen del sistema de coordenadas de la cámara resaltado. Por otro lado, en la Figura 29 y la Figura 30, se expone la situación del sistema de coordenadas del robot frente al de la cámara desde la vista de la planta y un lateral.

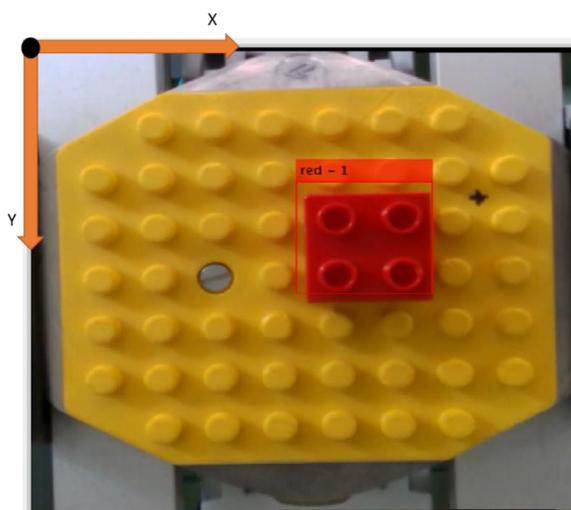


Figura 28: Sistema de coordenadas de la cámara

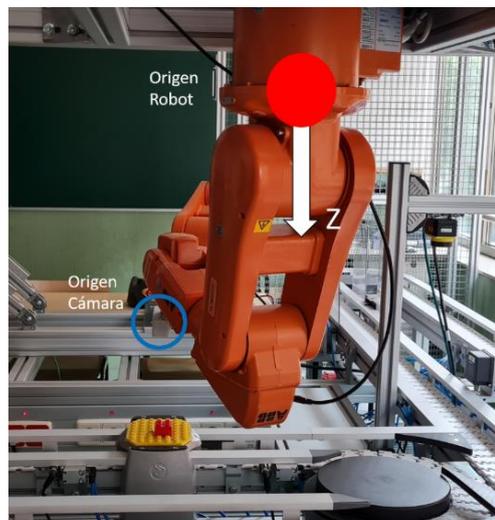


Figura 29: Vista lateral de los orígenes de los sistemas coordenadas del robot y de la cámara

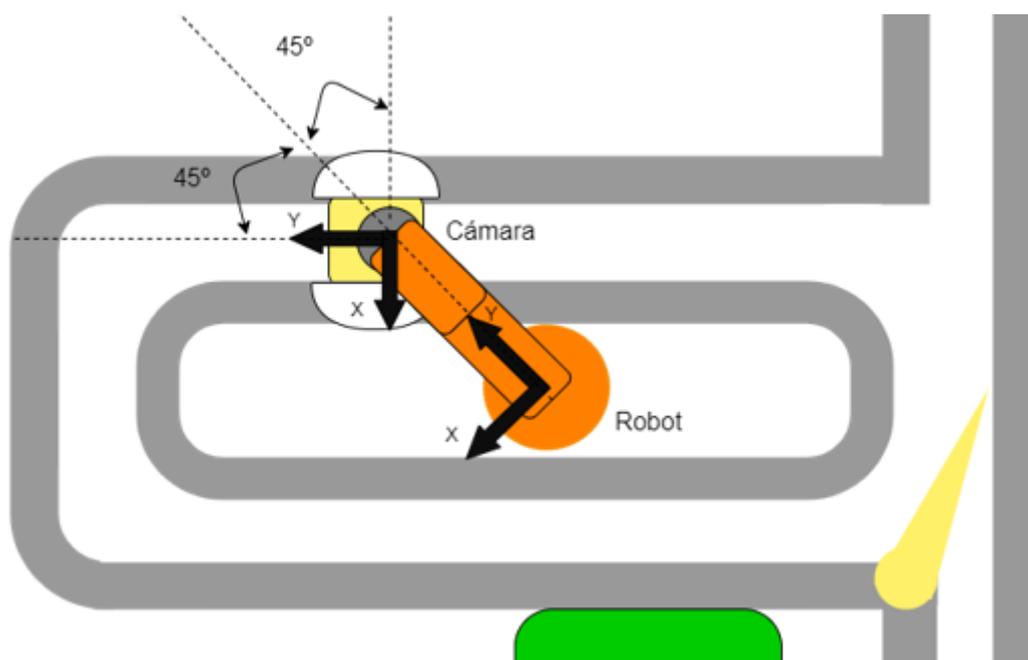


Figura 30: Vista de planta de la disposición de los sistemas de coordenadas

Al robot se le deben enviar las variaciones de distancia en milímetros respecto del punto en el que se encuentra para llegar a la pieza detectada y poder cogerla. Estas variaciones deben ser en los ejes X, Y y Z del robot. Por lo que además de dar las variaciones en las unidades correctas, se han de establecer en las direcciones indicadas. Por esta razón se deben realizar

dos tareas principales para la extracción de las coordenadas en el sistema del robot. La primera tarea es orientar ambos sistemas del mismo modo para que las variaciones estén en la misma dirección y la segunda es la transformación de las unidades de la cámara (píxeles) a las del robot (milímetros).

4.2 TRASLACIÓN DEL SISTEMA DE COORDENADAS

Como se ha comentado anteriormente y como se puede observar en la Figura 30, el sistema de coordenadas de la cámara, en el que se obtienen las coordenadas de la pieza detectada, se encuentra rotado respecto del robot. La rotación es de aproximadamente 45 grados. Razón por la que las coordenadas obtenidas en el sistema de la cámara han de ser transformadas a coordenadas del robot. Observando la Figura 30 se pueden extraer las ecuaciones (1) y (2) para la transformación de coordenadas.

$$x_{rob} = x_{cam} * \cos(45) + y_{cam} * \cos(45) \quad (1)$$

$$y_{rob} = -x_{cam} * \sen(45) + y_{cam} * \sen(45) \quad (2)$$

En este proyecto el eje Z o normal a la cara superior de las piezas se va a tratar de manera especial. Como solo se espera una sola pieza, sólo habrá una altura y la variación en el eje Z en el sistema del robot será siempre la misma. Se desarrolla en mayor detalle en el apartado 4.5

4.3 EXTRACCIÓN DE COORDENADAS POR CALIBRACIÓN A PARTIR DE PUNTOS FIJOS

En el apartado anterior se ha desarrollado la primera tarea para la obtención de las coordenadas en el sistema del robot, la transformación de un sistema de coordenadas a otro. En este apartado se va a desarrollar la segunda tarea, el paso de las unidades del sistema de la cámara a las del sistema del robot.

Este problema se puede afrontar de diferentes maneras, en este caso se van a analizar dos de ellas. La primera es la calibración de los puntos a partir de los puntos más extremos y por lo tanto más difíciles de detectar por la cámara, y la segunda es la calibración de la medida de una unidad de distancia entre píxeles y milímetros.

Cabe destacar de igual manera que, si se emplea la arquitectura de YOLO o Faster R-CNN con el clasificador de objetos basado en AlexNet, la imagen que se procesará tendrá dimensiones de 227x227 píxeles; si el clasificador es el basado en VGG16 será de 224x224 píxeles, aunque durante el procesado se reajustarán para trabajar en dimensiones de 480x480. Las coordenadas obtenidas tras el procesado de las imágenes tendrán esos máximos. Por otro lado, si se emplea R-CNN las dimensiones de la imagen de entrada serán de 480x480 píxeles directamente.

4.3.1 CALIBRACIÓN A PARTIR DE PUNTOS EXTREMOS

Los puntos extremos son los puntos en los que se localizarán las piezas que van a ser detectadas por la cámara con mayor dificultad, en este caso, debido a su situación en la imagen. En la Figura 31 se visualiza cuáles van a ser estos puntos en el palé. Son concretamente esos puntos debido a que son los que se van a encontrar más próximos a los bordes de la imagen y por ello pueden ser más difíciles de identificar.

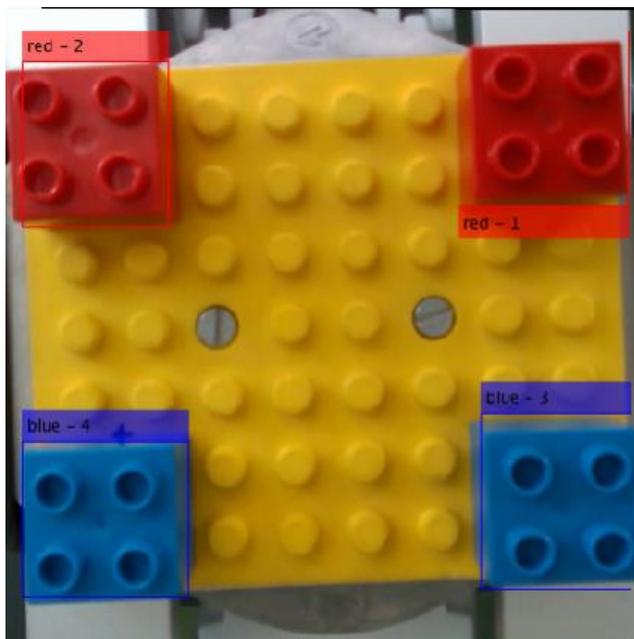


Figura 31: Puntos límite del palé para calibración

A continuación, se procedió a la extracción de las coordenadas de los puntos de calibración con el robot y la aplicación de MATLAB desarrollada por Ignacio. Dicha aplicación devolvía directamente el centro de la *bounding box*, lo que consideraremos el punto al que deberá llegar el robot para coger la pieza, además se hará uso de las ecuaciones (1) y (2) para obtener las coordenadas en unidades de la cámara, pero en ejes del robot.

Por otro lado, el robot dispone del Flexpendant, expuesto en la Figura 32, una herramienta con la cual se pueden desplazar las diferentes articulaciones del robot con un *joystick* y tratar con las señales que el robot puede manejar. También consta de un sistema de seguridad que habilita y deshabilita el robot en función de si el operario esta accionando un pulsador o no. Además, esta herramienta consta de una pantalla a través de la cual el operario puede conocer distintos detalles del robot, como el valor de ciertas señales o las propias coordenadas de la posición del robot.



Figura 32: Interfaz del flexpendant para la obtención de coordenadas del robot



Figura 33: Extracción de coordenadas de la pieza en el sistema del robot empleando el flexpendant

A continuación, se muestra en la Tabla 1: Coordenadas de los puntos límite del palé

los datos extraídos de la calibración.

Punto 1					Punto 2						
Robot		Cámara			Robot		Cámara				
x	-118.10	x'	74.00	Δx	130.11	x	-45.50	x'	402.50	Δx	361.69
y	409.40	y'	110.00	Δy	25.46	y	341.70	y'	109.00	Δy	-207.54
Punto 4					Punto 3						
Robot		Cámara			Robot		Cámara				
x	-56.00	x'	72.00	Δx	325.55	x	16.00	x'	414.00	Δx	569.36
y	468.30	y'	388.40	Δy	223.73	y	398.00	y'	391.20	Δy	-16.12

Tabla 1: Coordenadas de los puntos límite del palé

En la Tabla 1 se puede observar que, para cada punto, bajo la cámara se tienen dos valores. El valor asociado a X' e Y' es el obtenido directamente de la aplicación de MATLAB, en cambio, el valor asociado a ΔX y ΔY son los valores de X' e Y' transformados al sistema de coordenadas del robot. Es decir, empleando las ecuaciones (1) y (2) se ha obtenido el valor de ΔX y ΔY a partir de X' e Y'

Conocida la relación entre las coordenadas de la cámara y las variaciones del punto de origen a las piezas en el sistema del robot, se puede establecer un cálculo de coordenadas de puntos intermedios por interpolación lineal. De esta manera, nos aseguramos de que para una pieza localizada en un punto que se encuentre dentro de la región del palé se extraigan sus coordenadas a través de la interpolación.

4.3.2 CALIBRACIÓN A PARTIR DE DOS PUNTOS PRÓXIMOS

Este método es similar al anterior, pero en este caso se busca obtener la distancia entre una posición de una pieza y la siguiente más próxima. Como vamos a analizar piezas cuadradas y no hay diferencia entre un desplazamiento en la dirección horizontal del palé y la vertical, se puede presuponer que no existirá diferencia entre obtener la distancia vertical o la horizontal. En la Figura 34, la Figura 35 y la Figura 36 se visualiza el concepto de desplazamientos verticales y horizontales



Figura 34: Punto de partida de la calibración



Figura 35: Desplazamiento en el eje X de la cámara



Figura 36: Desplazamiento en el eje Y de la cámara

Conociendo la distancia en píxeles entre los dos puntos en coordenadas de la cámara, se puede transformar al sistema del robot y medir el valor de dicha distancia entre dos posiciones de igual manera a como se realizó en el apartado anterior. Con ello se tiene la equivalencia entre una posición en unidades reales y píxeles. De manera que, a partir de un

nuevo punto se puede establecer a cuantas posiciones del origen se encuentra y transformarlo a unidades reales.

En la Tabla 2 se muestran los resultados de las medidas de los puntos en coordenadas de la cámara y el robot.

Punto 1						Punto 2					
Robot		Cámara				Robot		Cámara			
x	-114.60	x'	72.00	Δx	125.87	x	-103.15	x'	125.00	Δx	161.22
y	410.70	y'	106.00	Δy	24.04	y	398.90	y'	103.00	Δy	-15.56
Punto 3											
Robot		Cámara									
x	-102.65	x'	70.00	Δx	157.68						
y	422.30	y'	153.00	Δy	58.69						

Tabla 2: Coordenadas puntos próximos

La estructura de la Tabla 2 es la misma que la de la Tabla 1: Coordenadas de los puntos límite del palé

. En este caso los puntos son los extraídos de la Figura 34 ,la Figura 35 y la Figura 36. El problema que presenta es que se asume que no va a existir una gran distorsión entre los

4.3.3 CONCLUSIONES DE LOS MÉTODOS DE CALIBRACIÓN

De los ensayos y pruebas realizados con ambos métodos se puede extraer las siguientes conclusiones.

- El método de YOLO es el que proporciona una *bounding box* más ajustada a la pieza y más precisa. El uso de Faster R-CNN conlleva a un mayor nivel de confianza, pero con una *bounding box* más grande que aumenta la posibilidad de obtener un mayor error entre el centro de la misma y el centro de la pieza contenida en ella.
- Relativo al método de puntos extremos, al analizar los casos límite de una imagen, obtener las coordenadas del resto de los puntos debería ejecutarse con mayor facilidad.
- Para ambos métodos, sólo se requiere extraer la información de un par de puntos para calibrarlo completamente. En el caso de los puntos extremos, son cuatro con

posibilidad de poder reducirlos a tres que no estén alineados en una sola dirección. Para el caso de los puntos próximos, sirve con evaluar también tres puntos, de igual manera esto se podría reducir a dos ya que se busca identificar a cuanto equivale la distancia entre una posición de la pieza y la siguiente más cercana. Al ser cuadradas las piezas, podemos suponer que será igual en el eje horizontal y vertical.

- Relativo al método de puntos extremos. Tras varias pruebas y ensayos con piezas ubicadas en diversas posiciones se obtuvo un error de aproximadamente 2mm. En este caso, por la condición del robot y su estructura de funcionamiento, este error puede hacer que, aunque se aproxime a la posición de la pieza, al cerrar la pinza no se cierre justo sobre ella y por la presión entre la pieza el palé, la pinza no pueda mover la pieza del palé y el robot quede bloqueado.
- A pesar de añadir nuevos puntos de calibración para el primer método y así obtener una mayor precisión el resultado no mejoraba, llegaba a empeorar, dando lugar a más situaciones de bloqueo del robot a la hora de coger las piezas.
- El error puede verse incrementado por varias razones. La principal, que la rotación de un sistema de coordenadas respecto del otro no sea exactamente de 45° , seguida de que el centro de la *bounding box* puede no coincidir con el de la pieza del palé, por lo que el robot irá al de la *bounding box* y no logrará alcanzar la pieza. Este se podría solventar con un reentrenamiento de las redes neuronales, para dotarlas de mayor precisión.
- Ante alguna ligera variación en la posición de captura de fotos o de la situación del palé en la imagen el sistema se descalibra y da lugar a error. No presenta gran robustez ninguno de los dos.

4.4 AJUSTE EXACTO POR APROXIMACIÓN

Debido a la falta de precisión y robustez que ofrecían los métodos de calibración, se cambió el enfoque por una aproximación más personalizada para este caso de estudio. Al no tener un área de estudio con una gran cantidad de opciones para la recogida de las piezas, se plantea la posibilidad de extraer las coordenadas de todos los puntos posibles en los que se puede encontrar una pieza tanto en el sistema del robot como el de la cámara.

Para este método, no se ha empleado la aplicación de [5] para obtener las coordenadas en el sistema de la cámara. Se desarrolló un código que permitía tomar una imagen, anotar en ella todos los puntos de interés y obtener sus coordenadas. Hicieron falta cuatro imágenes para extraer todos los puntos del caso de estudio, estas son las expuestas en la Figura 37, la Figura 38, la Figura 39 y la Figura 40

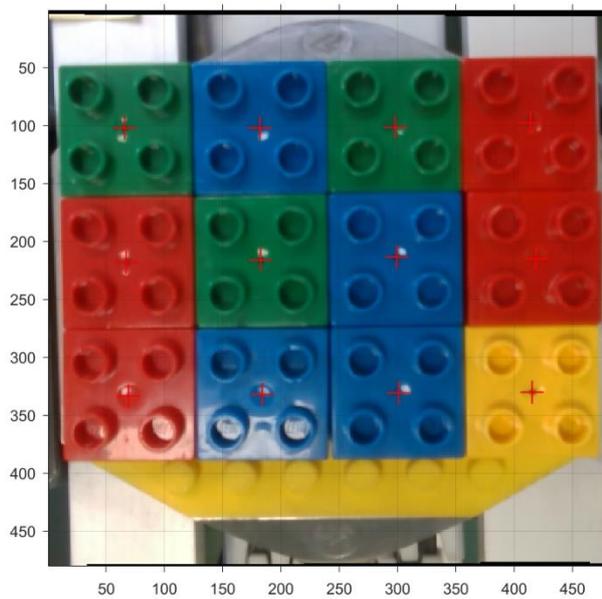


Figura 37: Ajuste exacto primera fase

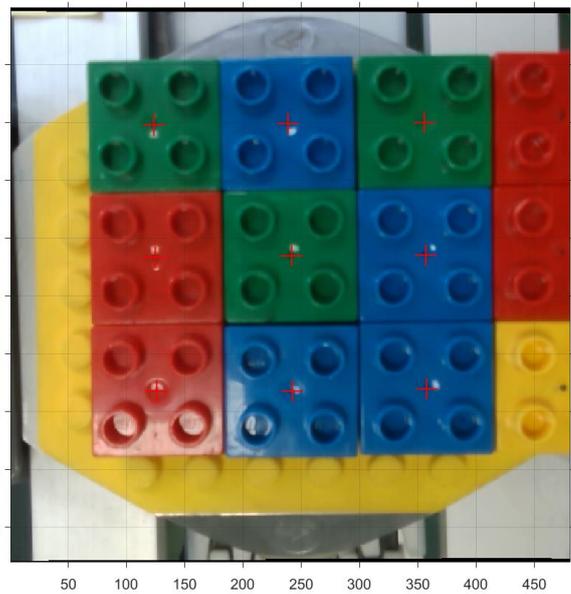


Figura 38: Ajuste exacto segunda fase

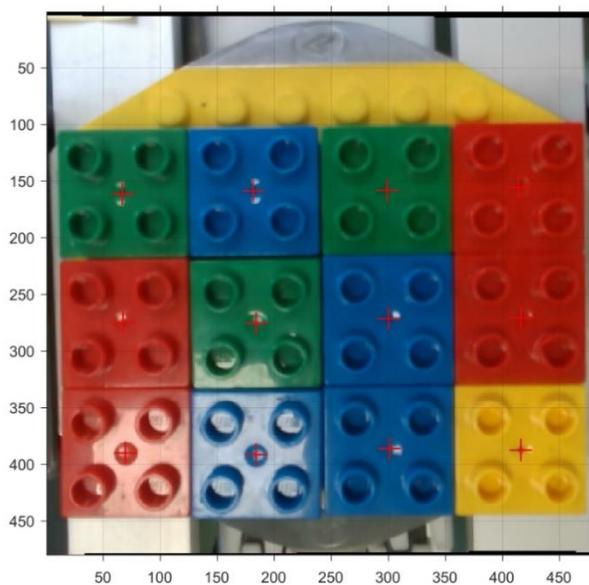


Figura 39: Ajuste exacto tercera fase

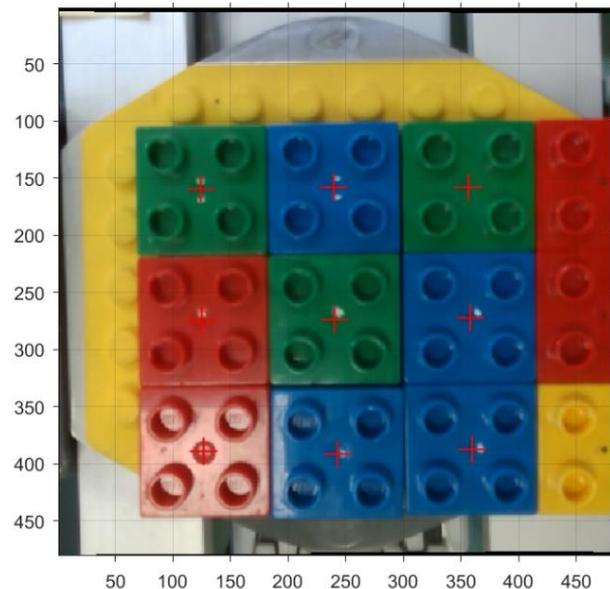


Figura 40: Ajuste exacto cuarta fase

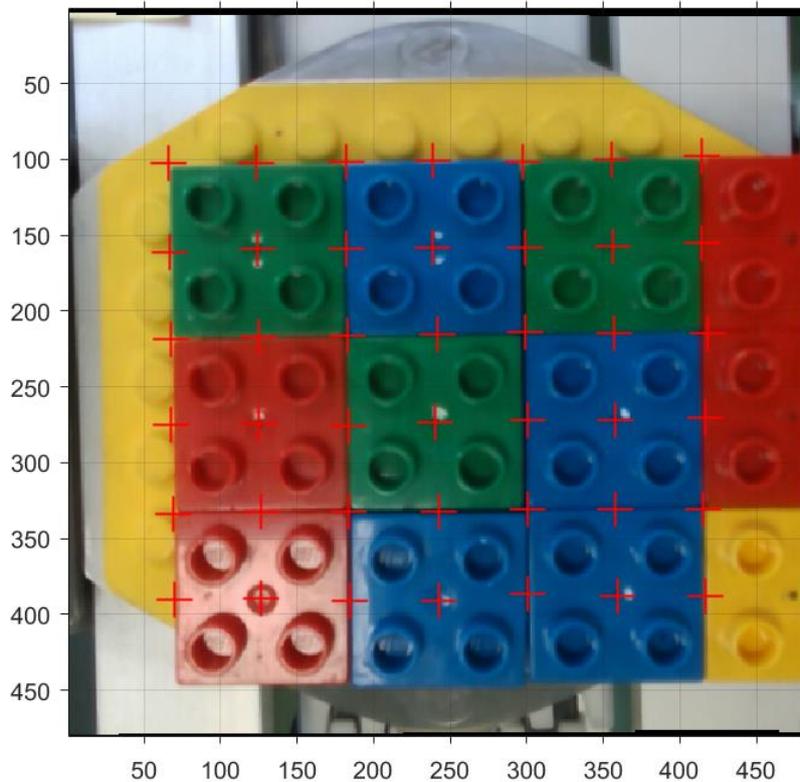


Figura 41: Resultado de la anotación de todos los puntos

En la Figura 41 se muestran todos los puntos anotados en una sola imagen. De todos ellos se tienen las coordenadas en el sistema de la cámara. Para su obtención en el sistema del robot se tomaron los puntos moviendo el brazo a cada uno de ellos y anotando las coordenadas de cada uno en el eje X e Y.

De esta manera tendremos asociado cada punto en coordenadas de la cámara directamente con su homólogo en coordenadas del robot. En la Tabla 3 se muestra el resultado de toda la calibración.



Xcam	Ycam	Xrob	Yrob
65.86	102.37	-114.60	410.70
66.92	161.65	-102.65	422.30
67.45	218.27	-90.70	433.90
67.45	274.90	-79.40	446.40
69.03	333.64	-68.10	458.90
69.56	390.27	-56.80	471.40
123.54	102.37	-103.15	398.90
124.07	159.00	-90.95	410.53
124.60	216.69	-78.75	422.15
125.13	274.37	-67.30	434.23
126.19	332.05	-55.85	446.30
126.72	389.21	-44.40	458.38
181.76	101.85	-91.70	387.10
181.76	159.00	-79.25	398.75
182.82	216.16	-66.80	410.40
183.87	275.43	-55.20	422.05
183.87	332.05	-43.60	433.70
184.40	391.33	-32.00	445.35
238.38	100.79	-79.85	376.25
238.91	157.94	-67.38	387.45
239.97	273.31	-43.30	410.40
241.56	215.10	-54.90	398.65
242.09	332.58	-31.70	422.15
242.62	391.33	-23.00	430.94
297.66	101.32	-68.00	365.40
298.71	157.94	-55.50	376.15
299.24	213.51	-43.00	386.90
300.30	271.72	-31.40	398.75
300.83	331.00	-19.80	410.60
300.83	386.03	-14.00	416.53
355.34	100.26	-55.70	353.35
356.40	157.41	-43.55	364.55
356.93	214.57	-31.40	375.75
357.46	271.72	-20.08	387.88
357.46	330.47	-8.75	400.00
359.57	387.62	-0.33	409.16
414.08	97.61	-43.40	341.30
414.08	155.30	-31.60	352.95
415.14	330.47	2.30	389.40
416.20	270.67	-8.75	377.00
416.20	387.62	13.35	401.80
418.32	214.57	-19.80	364.60

Tabla 3: Puntos obtenidos del ajuste exacto

El problema ahora se reduce a ajustar el punto que obtenemos de las coordenadas de la cámara a uno de estos, el que se encuentre más próximo a este. Esto se hace calculando la distancia euclídea entre el punto obtenido por el procesado de la imagen y los puntos de referencia. El que menor distancia tenga es al que corresponde, y con ello obtenemos el destino del robot. Haciendo una simple resta con el punto de origen de captación de las imágenes obtenemos las variaciones en los ejes que necesitamos para enviar al robot.

Poniendo un ejemplo, si la cámara detectase que el centro de la *bounding box* se encuentra en las coordenadas (250,150) a partir de la distancia de este punto a los puntos de la cámara de la Tabla 3 podemos obtener que el punto más cercano en coordenadas de la cámara es el (238.9,157.9) y con ello, que el robot se debe desplazar hacia el punto (-67.38, 387.45). Visualmente, en la Figura 42 se puede ver en azul el que podría ser el punto obtenido tras el procesamiento de la imagen y la detección de la pieza y el más próximo a el destacado.

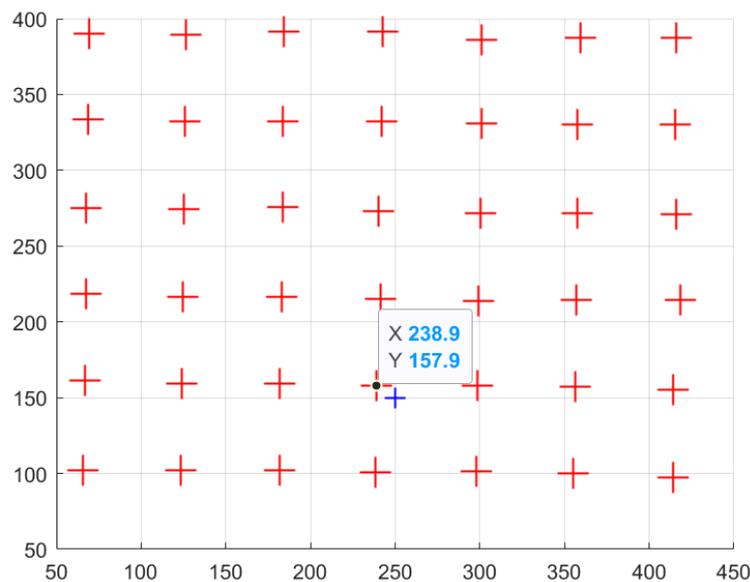


Figura 42: Prueba del ajuste exacto

Con este método se solventa el problema de que la *bounding box* no quede bien ajustada a la pieza o que el centro de la *bounding box* no coincida. Aporta gran precisión al modelo y una considerable robustez ante ligeros cambios, a diferencia de los métodos anteriores.

4.5 EXTRACCIÓN DE LA ALTURA Y DEL ÁNGULO DE LA PIEZA

Como se ha comentado en apartados anteriores, en este sistema la altura se va a tratar de una manera diferente a como se planteó originalmente. En este proyecto sólo se espera recibir una pieza por palé, eso implica una sola altura. Por lo que el robot tendrá que desplazarse siempre a la misma coordenada Z.

El problema entonces se puede enfocar de dos modos, aprovechar el sistema de partida, y ajustarlo a la situación en la que se encuentra de manera que se obtenga la variación indicada o fijar directamente la altura al valor necesario que como se puede ver en la FIGURA X es de 148 milímetros.

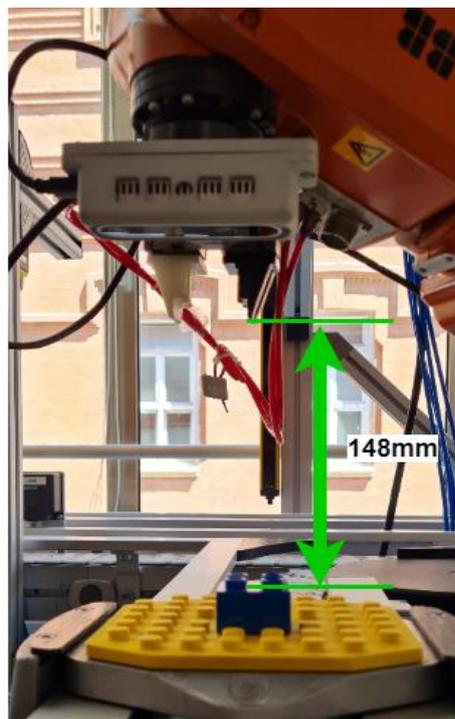


Figura 43: Distancia en el eje Z de la pinza hasta la pieza a recoger

Algo parecido sucede con el ángulo con el que esta rotada la pieza. Para el caso de estudio, el palé se va a bloquear en la misma posición siempre, y la pieza identificada, al ser cuadrada no va a estar rotada en ningún momento. Razón por la que en este caso no es necesario conocer esta característica de la pieza.

Capítulo 5. Acciones del brazo robótico

Este capítulo abarca las tareas que realiza el brazo robótico. Primero se analizará la zona de trabajo para la construcción de figuras. Se tratará la gestión de los palés por la cinta, las prevenciones que se toman para evitar colisiones entre ellos. Por último, se expondrán las trayectorias que toma el robot y el tipo de movimiento que realiza.

Para que el robot se pueda mover de forma automática y continua a la vez que comunicarse con MATLAB, se desarrolló el programa que regirá el comportamiento del robot en el software de RobotStudio [22], sobre el que se irán comentando aspectos a lo largo de este apartado y el siguiente.

5.1 ZONA DE TRABAJO

Una de las primeras tareas que se desarrollaron en este proyecto fue la determinación de los límites del área de trabajo donde se construirán las figuras. El robot tiene un límite de alcance como se puede observar en la Figura 44 y por ello, no puede alcanzar todo el área de construcción de las figuras. De hecho, alcanza una zona muy limitada. En la Figura 45 se puede observar el área de trabajo y destacada el área que alcanza el robot.

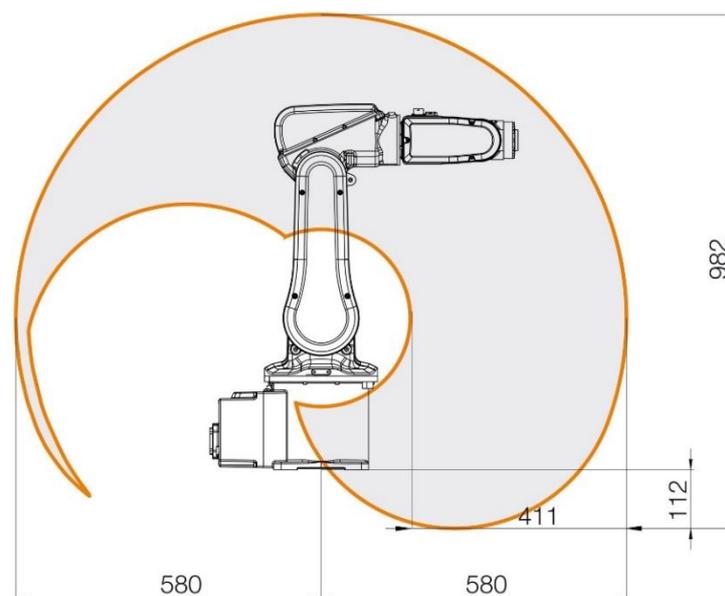


Figura 44: Alcance del brazo robótico IRB 120

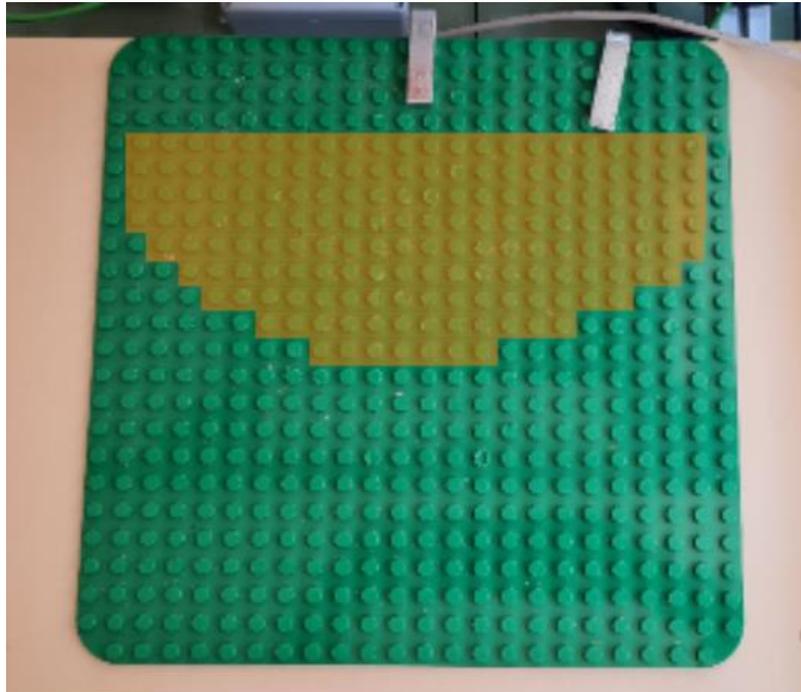


Figura 45: Zonas alcanzables por el robot en el área de trabajo

Como se puede apreciar en la Figura 45, la zona de trabajo queda reducida a aproximadamente un tercio del total. Esto es también debido a que el robot se encuentra suspendido de una estructura y para llegar a la mesa ha de estar prácticamente con todas las articulaciones extendidas, lo que limita mucho el movimiento en la zona.

5.2 MOVIMIENTOS DEL ROBOT

En este proyecto, el robot es el responsable del traslado de las piezas, pero también lo es de posicionarse para detectarlas. El robot ha de moverse sin entrar en ningún momento en una zona prohibida y evitando todas las colisiones.

En el programa se desarrolla la trayectoria del robot siguiendo una ruta de puntos que, debido a la posición del robot y de algunas articulaciones, no han de ser muy distantes entre sí.

Para ir de un punto a otro, el robot puede desplazarse de manera lineal, siguiendo una recta o de forma más libre con el movimiento de los ejes. Al igual que ocurriría con los puntos de partida y de destino, el movimiento queda restringido a la forma lineal debido a que la disposición del robot puede provocar colisiones si se emplean movimientos no lineales o

requeriría de programar la ruta con un número mayor de puntos a menores distancias. Se decidió optar por el uso de movimientos lineales.

El robot se va a mover entre dos secciones principalmente, la zona de bloqueo de palés, dedicada a la recogida de las piezas, y el área de trabajo, dedicada a la colocación de las piezas para la construcción de las figuras.

Para la zona de bloqueo de los palés, el robot tendrá un punto de referencia o reposo desde el que esperará la llegada de nuevos palés y estará ya ubicado para la captura de imágenes con la cámara, que se encuentra en su muñeca junto a la pinza para coger las piezas de lego como se puede ver en la Figura 25, dicho punto se fijó comprobando con la interfaz que la imagen capturada contenía todo el palé y era capaz de detectar una pieza en cualquier ubicación del mismo. La posición en concreto se expone en la Figura 46. Es una posición delicada debido a la proximidad de la pinza con las conexiones entre las articulaciones, pero gracias a una planificación de trayectorias prudente, se ha tenido en cuenta este hecho y el primer movimiento que realiza el robot desde esta posición es para alejarse de la zona de riesgo.

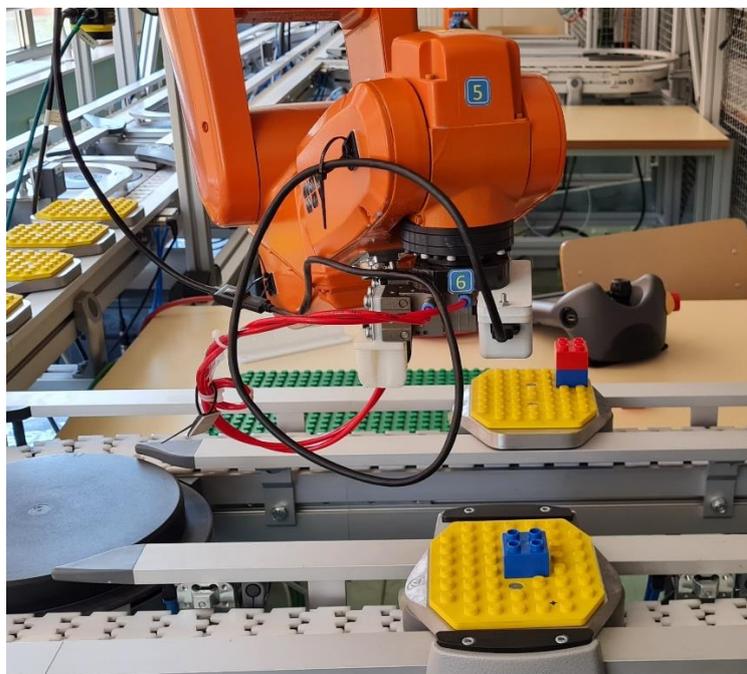


Figura 46: Posición de referencia/reposo frente al palé bloqueado

Desde la posición de reposo, tras haber desviado un palé y bloquearlo el robot espera el mensaje de MATLAB con la información relativa a la ubicación de la pieza, su color y cuantas piezas han sido detectadas ya y si debe aproximarse al palé para recoger la pieza o liberarlo. El siguiente movimiento que hace el robot es para coger una pieza, si no tiene que cogerla, permanecerá estático y liberará el palé. Para desplazarse hacia el palé, lo primero que hace es moverse en el eje Z positivo, es decir, hacia abajo, para alejarse de la zona delicada como se ha expuesto anteriormente. Tras este primer movimiento el robot se desplaza justo encima de la pieza a recoger y se aproxima lentamente hacia dicha pieza para después cerrar la pinza como se ve en la Figura 47.



Figura 47: Recogida de la pieza tras análisis de la imagen

Una vez agarrada la pieza el primer paso es levantarla y alejarse del palé. Tras esto, el robot se desplaza hacia el área de montaje a través de varios puntos. Una vez encima del área de trabajo, el robot procesa dónde debe colocar la pieza y de qué manera debe colocarla. Los puntos en los que se deben colocar las piezas han sido definidos manualmente en un inicio, es decir, se construyó la figura de forma manual y se definieron completamente los puntos de destino.

Para desarrollar la tarea de colocar las piezas en su posición final se desarrolló una función para cada figura *Colocar1* y *Colocar2* para las figuras 1 y 2 respectivamente. En las funciones se detalla la lógica que debe seguir el robot para construir una figura a partir de la información que le proporciona MATLAB sobre el número de piezas ya detectadas y el color de la actual. Para cada figura, es decir, para cada función se han programado tanto en RobotStudio como en MATLAB, al menos 2 formas de construir las figuras dependiendo de las piezas que le lleguen primero al robot, para liberar la menor cantidad de palés con pieza que se aproximen. La decisión de liberar un palé o no se toma en MATLAB y se transmite después al robot como se ha comentado anteriormente. Para la identificación de las piezas se empleó la anotación de la Figura 48 y la Figura 49.

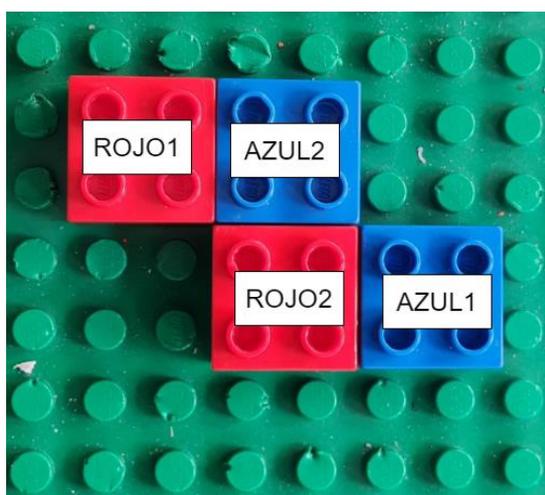


Figura 48: Figura 1 anotada

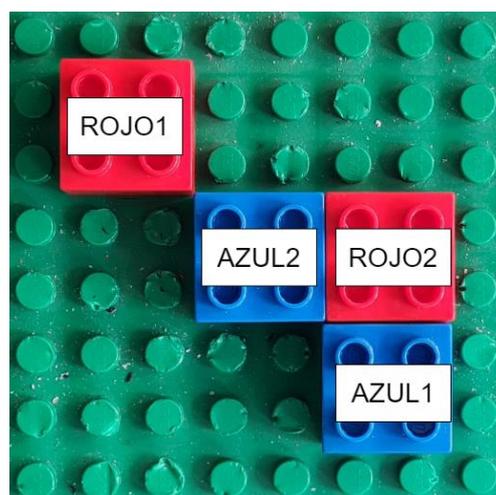


Figura 49: Figura 2 anotada

El caso de que se tenga que liberar un palé se dará, por ejemplo, si en la construcción de la figura 1, expuesta en la Figura 48, el robot recibe 2 palés seguidos con piezas rojas. La primera será colocada de manera exitosa, pero si se coloca la segunda en su posición, la pieza azul próxima a ambos no se podrá colocar debido a que la pinza que lleva la pieza colisionaría con una de las piezas rojas ya colocadas. Debido a esto el palé se liberaría y se tendría que esperar a una de las piezas azules. Esta lógica también está implantada en MATLAB para que sea capaz de indicarle al robot si debe coger la pieza del palé o liberarlo.

Según la pieza que se esté tratando existen dos maneras de colocarla en su sitio. O bien con la misma orientación con la que el robot se ha aproximado al palé, o bien con una rotación de 90°. Esta rotación tiene la razón de que según como se esté montando una figura u otra, puede que la primera forma haga que la pinza colisione con una pieza y no coloque bien la que tiene agarrada o se pare el robot.

Volviendo a poner el ejemplo de la Figura 48, el robot puede colocar las piezas azul y roja de arriba colocándolas con la misma orientación como se puede ver en la Figura 50, pero la siguiente pieza debe ser colocada girando la pinza 90° como se ha comentado antes y se expone en la Figura 51 y la Figura 52, debido a que, si no se hiciese, al colocar la pieza roja de abajo, la pinza colisionaría con la pieza azul.

Entonces, la lógica que sigue la construcción de la figura 1 u la figura 2 se basa en el análisis del color de la pieza que se esté manejando y conocer cuantas piezas de la figura han sido detectadas ya.

Para la primera figura, si la primera pieza detectada es roja se procederá a colocarla en la posición ROJO1, después se rechazará todo palé que lleve una pieza roja debido a que, como se ha comentado antes, no se podría colocar la pieza AZUL2 después. Cuando se detecte la pieza AZUL2, ocurrirá lo mismo que antes, con las piezas azules que lleguen, serán rechazadas para poder colocar la pieza ROJO2. Una vez colocada la pieza ROJO2, se terminará colocando la pieza AZUL1. La segunda forma correspondería con el proceso inverso al descrito, primero AZUL1, luego ROJO2, seguido de AZUL2, y por último ROJO1.

La segunda figura permite una mayor flexibilidad a la hora de su construcción, lo que llevó al caso de que la programación de las formas de construir la figura resultase en hasta tres, con dos de ellas muy similares. Para todas las formas, la primera pieza roja que se detecte corresponderá a ROJO1, si se detecta antes una azul, corresponderá de igual manera a AZUL2, pero tras esta, la primera roja será ROJO1. En este caso como en el anterior existe una excepción, si se detecta una pieza azul (AZUL2), las siguientes azules se rechazarán hasta haber colocado ROJO2. No se da dicho caso al detectar una pieza roja, si se detectan dos piezas rojas seguidas, y no había ninguna antes, la primera corresponderá a ROJO1 y la

segunda a ROJO2. Después solo quedará completar la figura con las piezas azules, primero AZUL2 y después AZUL1.

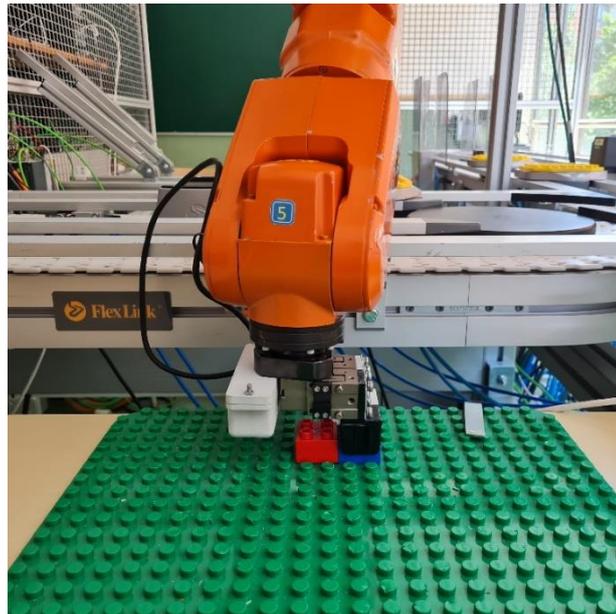
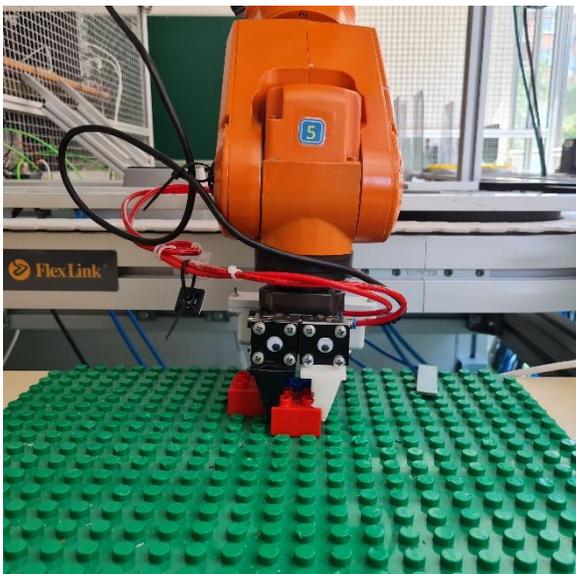
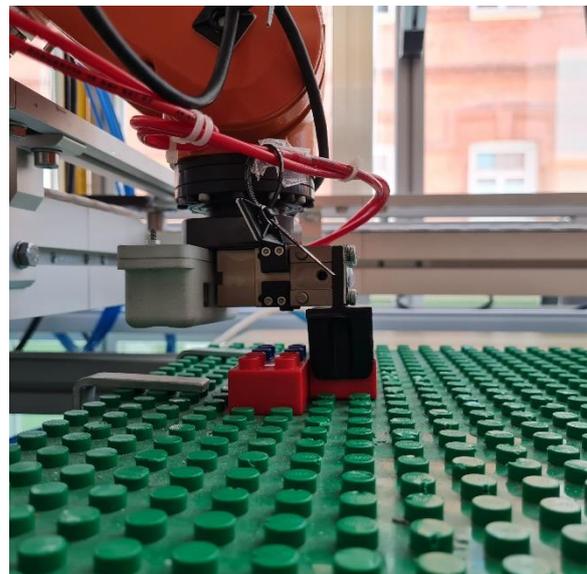


Figura 50: Pieza insertada sin rotación de la pinza



*Figura 51: Pieza insertada con la pinza rotada
90°*



*Figura 52: Pieza insertada con la pinza rotada
90° (vista lateral)*

De esta manera salvamos dicho problema, pero se ha de tener en cuenta que a cada forma de colocar una pieza le corresponde la definición de un punto, es decir, la pieza colocada sin rotación es un punto de destino, y la pieza colocada en la misma posición, pero con otra rotación es otro punto de destino.

Con las piezas ya colocadas se procede a la vuelta por la trayectoria seguida inicialmente hasta la posición de reposo a la espera de un nuevo palé para su análisis.

5.3 GESTIÓN DE LAS CINTAS TRANSPORTADORAS

En los apartados anteriores se ha comentado donde va a trabajar el robot y los movimientos que va a realizar para desplazarse por el área de operación. La última tarea que desarrolla el robot es la de la gestión de las cintas transportadoras y los palés. Esta tarea implica evitar las colisiones entre palés que entran y salen de la zona de desvío, que sólo entre a la zona de desvío un palé o sincronizar correctamente los accionamientos de desviadores y retenedores.

Para conocer la ubicación de los sensores y accionadores de los que se está hablando se adjunta en la Figura 53 un esquema de la planta con la señalización de estos sobre ella.

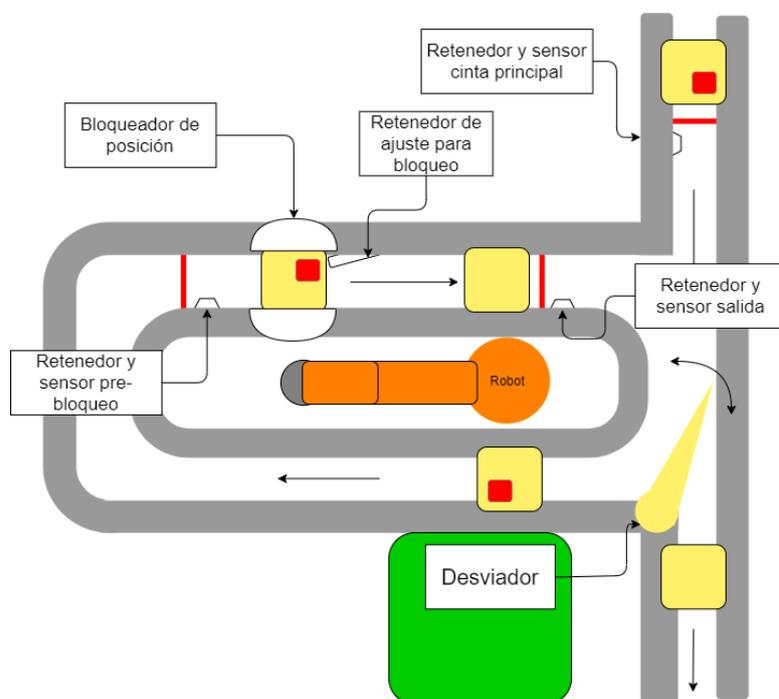


Figura 53: Esquema de los sensores de la planta

5.3.1 RETENEDORES Y SENSORES

Los retenedores son dispositivos situados entre las placas guía de las cintas y la propia cinta. En esta ubicación, según la presión aplicada sobre ellos despliegan u ocultan una pestaña que bloquea el paso de los palés al aproximarse. En esta planta existen tres tipos de retenedores diferentes que se desarrollan a continuación.

El primero de ellos, expuesto en la Figura 54 y la Figura 55, hace que el palé quede justo encima del retenedor y no pueda avanzar más allá de dicha posición. Este tipo se emplea en el retenedor de la cinta principal, en el de pre-bloqueo y en el de salida. Cabe destacar que este tipo de retenedores siempre van acompañados de un sensor que detecta la presencia del palé justo antes del retenedor.

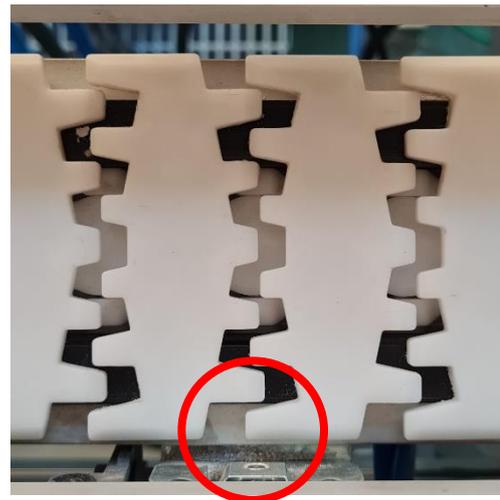
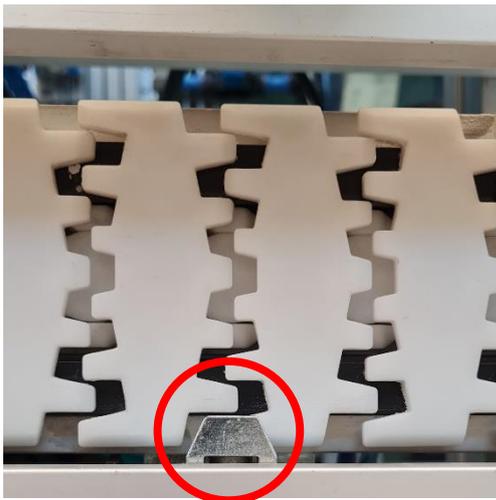


Figura 54: Retenedor cinta principal expuesto

Figura 55: Retenedor cinta principal recogido

Por otro lado, el retenedor de ajuste para bloqueo, debido a la forma que presenta y expuesta en la Figura 56 y la Figura 57 tiene la diferencia de que el palé no queda por encima de él como se puede apreciar en la Figura 58, colisiona directamente con él.

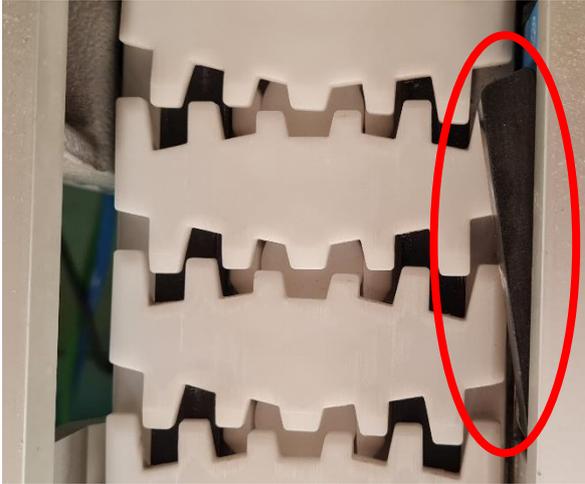


Figura 56: Retenedor de ajuste expuesto

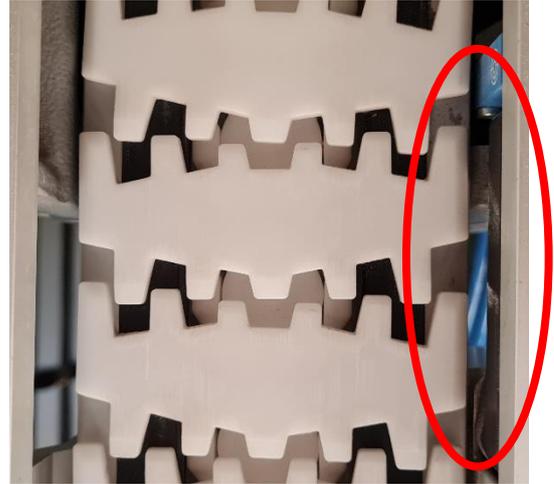


Figura 57: Retenedor de ajuste recogido

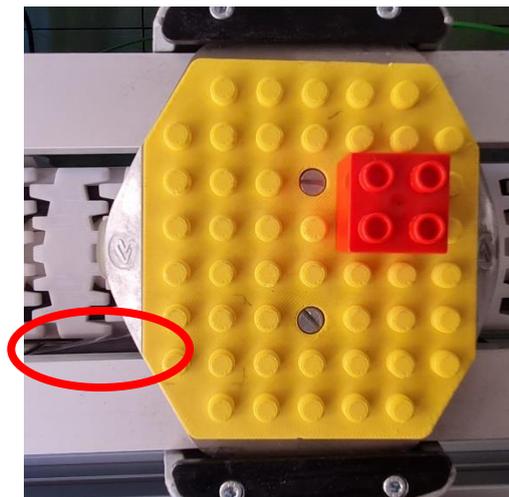


Figura 58: Retenedor de ajuste parando palé

El ultimo dispositivo retenedor es el empleado para la inmovilización de la pieza en la misma ubicación siempre. La diferencia con los anteriores es que este está ubicado encima de las placas guía e inmoviliza directamente el palé a ambos lados de la cinta impidiendo posibles variaciones que podían ocurrir con los anteriores que solo bloqueaban el paso a un lado e indirectamente al otro. En la Figura 59, y la Figura 60 se observa la forma en la que se elevan los dos retenedores simultáneamente. Estos coincidirán con los agujeros destacados en la Figura 61.

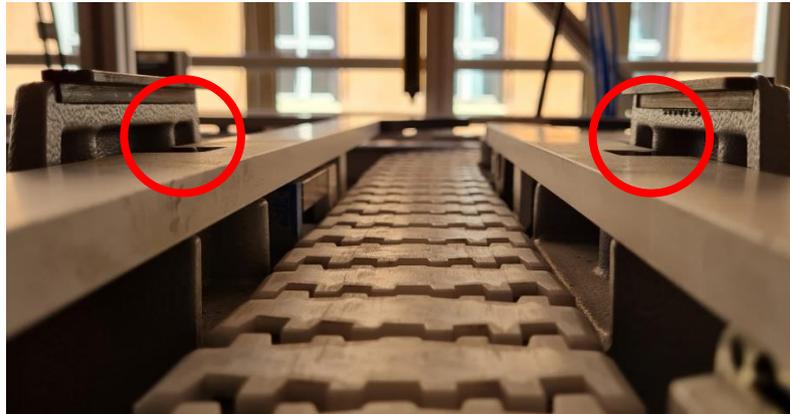


Figura 59: Bloqueadores recogidos

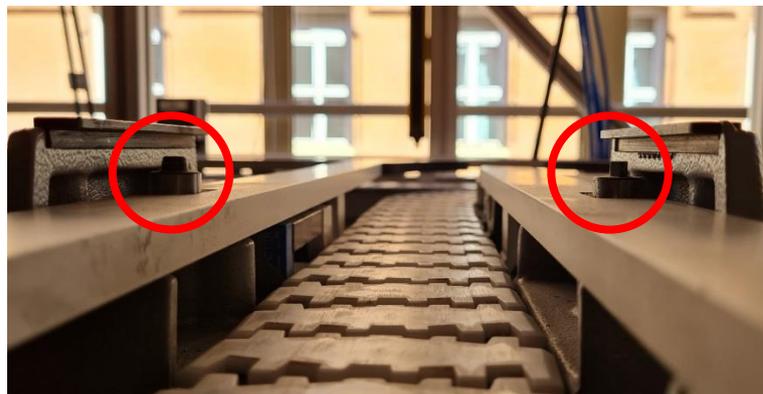


Figura 60: Bloqueadores expuestos

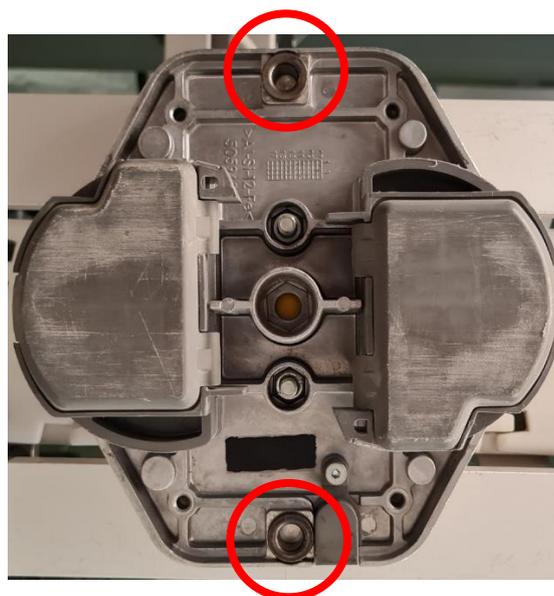


Figura 61: Vista inferior de palé con agujeros para bloqueo resaltados

5.3.2 DESVIADOR

El siguiente elemento es el responsable de poder cambiar la trayectoria de un palé en la cinta principal a la cinta que rodea al robot. La estructura se expone en la Figura 62 y la Figura 63.

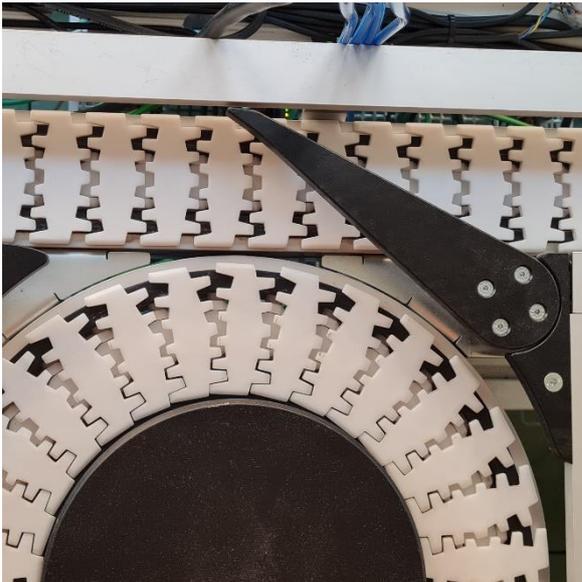


Figura 62: Desviador en activo



Figura 63: Desviador desactivado

5.4 PROGRAMACIÓN

Todos los actuadores mencionados anteriormente se pueden controlar desde el robot como salidas digitales, es decir, activándolos y desactivándolos con ‘1’ y ‘0’. Para cada uno de ellos, se creó una función que permite su control de una manera más sencilla.

En este proyecto, lo que no se ha controlado ha sido la velocidad de las cintas. Se fijó al 40% tanto la velocidad de la cinta principal como la de la propia del robot y se ajustaron los tiempos de activaciones y desactivaciones de los actuadores respecto de esta.

También se ha de tener en cuenta que, para todo mensaje recibido por el robot desde MATLAB, se devuelve una respuesta corta indicando que se ha recibido el mensaje.

De esta manera, la lógica con la que se programó en el robot se explica a continuación.

Al iniciar el programa, el robot se desplaza hasta la posición de la captura de imágenes y espera hasta conectarse con MATLAB, por otro lado, se activa el retenedor de salida de palés a la cinta principal, se desactiva el bloqueador y el retenedor de ajuste para liberar un palé que estuviese bloqueado en un inicio se es que había. Además, el desviador se pone en posición de no desvío, el retenedor de la cinta principal se mantiene oculto. Una vez conectado con MATLAB, se mantiene a la espera de recibir un mensaje que indique el inicio de construcción de una figura.

Cuando llega el mensaje de inicio de construcción de una figura, se activa el retenedor de la cinta principal y espera hasta que el sensor que acompaña a este detecta el palé. Cuando se detecta un palé, se desactiva el retenedor de la cinta principal por un instante sólo para evitar el paso de más de un palé y se pone el desviador en posición de desvío. Este proceso se expone visualmente en la Figura 64, la Figura 65.

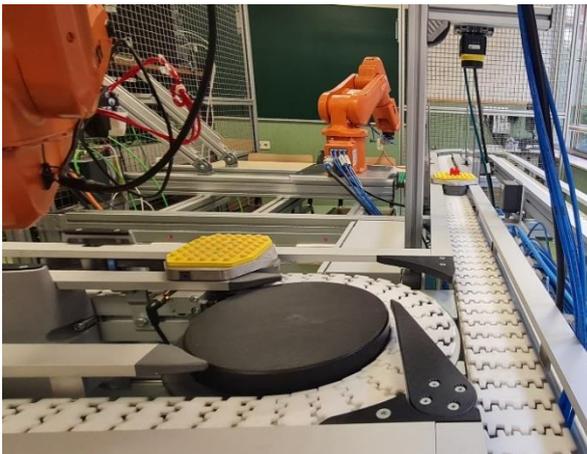


Figura 64: Aproximación de palé al retenedor principal

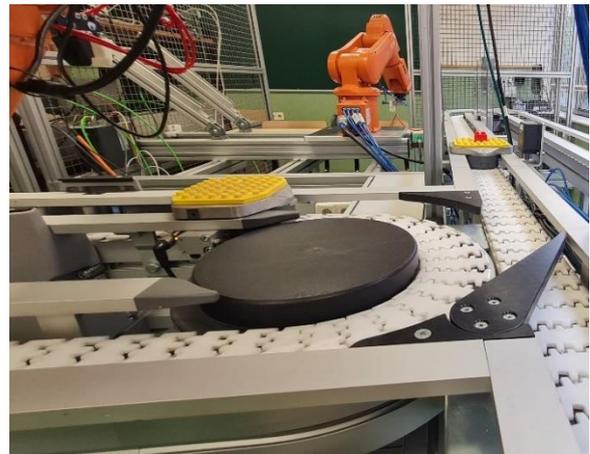


Figura 65: Desvío de palé

Una vez que el palé alcanza la cinta propia del robot, el desviador se cambia de posición de nuevo y se desactiva el retenedor de salida para que el palé que estuviera esperando incorporarse a la cinta principal así lo haga, por otro lado, se activa el retenedor pre-bloqueo. Una vez incorporado el palé a la cinta principal, se desactiva el retenedor principal y se libera la cinta. Durante este proceso, el palé desviado en un inicio alcanza el sensor pre-bloqueo y se desactiva su retenedor para activar el retenedor de ajuste. Tras un tiempo prudencial en el que el palé colisiona con el retenedor de ajuste, se activa el bloqueador y se comunica a

MATLAB que debe comenzar el procesamiento de imágenes y la captación de coordenadas. En la Figura 66, y la Figura 67 vemos este proceso.

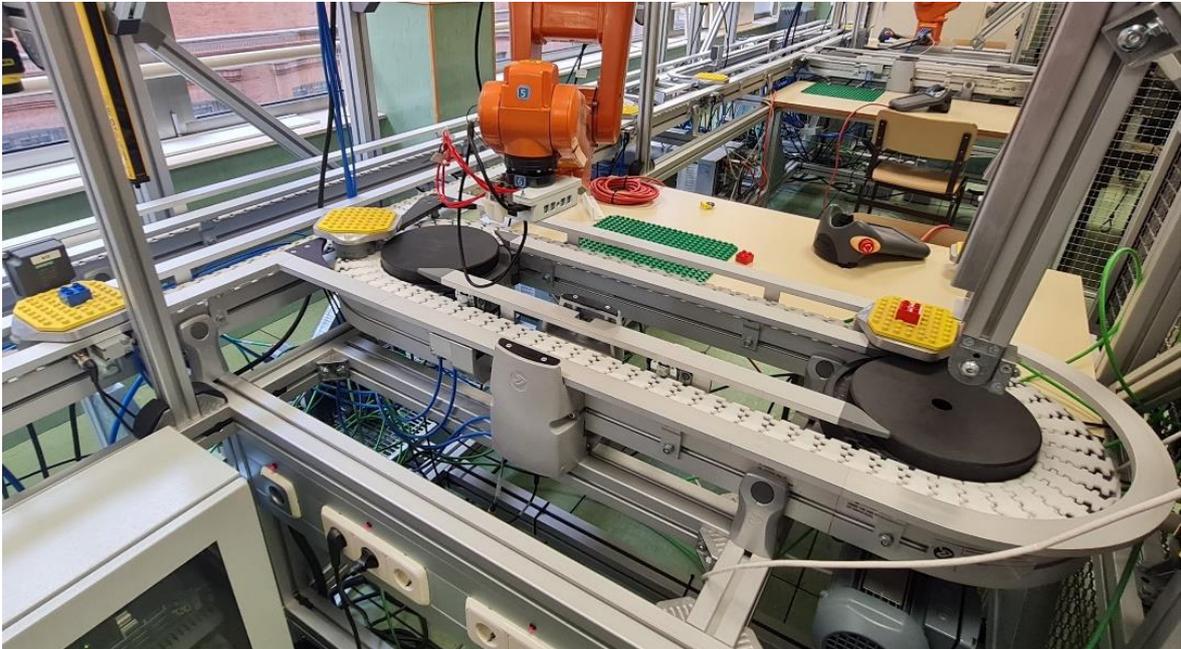


Figura 66: Incorporación de palé bloqueado a la cinta principal

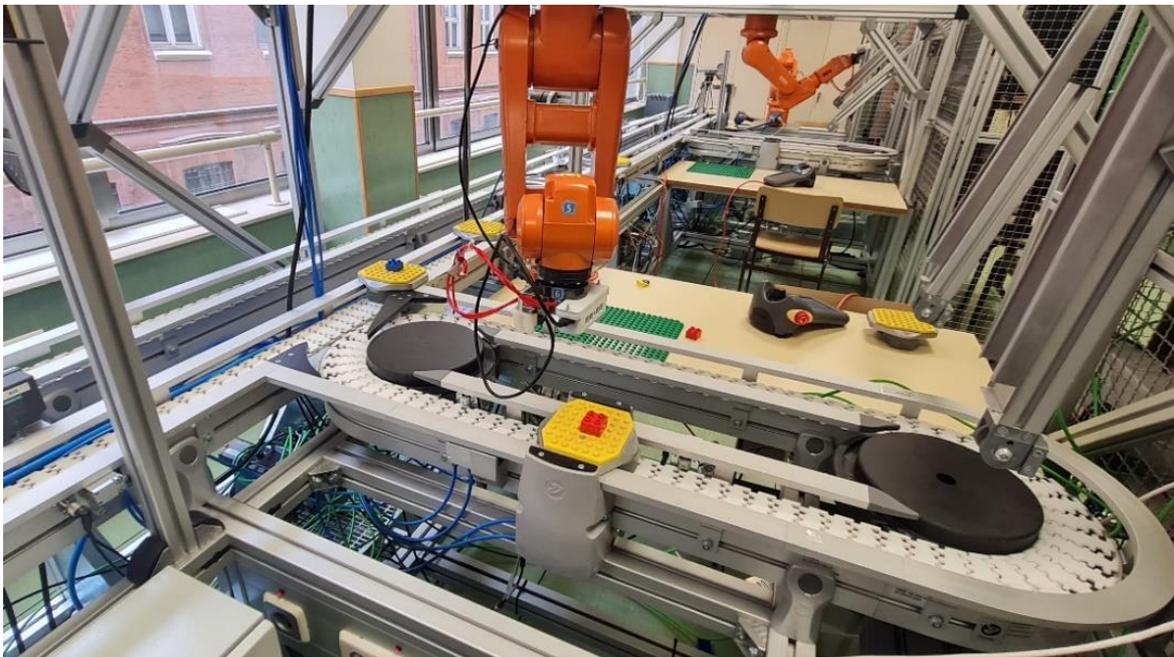


Figura 67: Llegada del palé desviado a la posición de bloqueo

Una vez procesada la imagen, según las indicaciones de MATLAB, el robot liberará el palé o se aproximará a la pieza para cogerla y desplazarse al área de construcción de figuras. Si ha tenido que coger la pieza, durante el traslado, se liberará el palé de la posición de bloqueo y se mantendrá a la espera en la posición del retenedor de salida. La colocación de las piezas se desarrolla en mayor detalle en el apartado 5.2. Una vez colocada, el robot volverá a la posición de toma de imágenes y, si la figura no se ha terminado de construir, esperará hasta la llegada de un nuevo palé activando el sensor principal y reanudando el proceso.



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

CAPÍTULO 5: ACCIONES DEL BRAZO ROBÓTICO

Capítulo 6. Resultados

Este proyecto no tiene unos resultados numéricos como tal. El objetivo principal era conseguir construir las figuras de LEGO empleando el robot, la cámara y las cintas transportadoras conjuntamente a partir del del sistema de visión artificial desarrollado en [5]

Se realizaron numerosas pruebas a lo largo del proyecto y actualmente la tarea se completa con un 100% de éxito. Por lo tanto, se puede concluir que el objetivo se cumplió y el sistema es operativo. El operario solo tiene que escoger la figura que desea construir. Tras esto el robot, MATLAB y la cámara trabajaran conjuntamente para construirla en el área de destino y el operario solo tiene que supervisar el montaje de esta.

Sin embargo, se va a plantear el análisis de tiempos de procesado y ejecución tanto de la aplicación de MATLAB como de la gestión del robot. Para la medida de estos se han empleado funciones propias del software de MATLAB y de RobotStudio.

6.1 TIEMPOS MATLAB

En el caso de MATLAB se midieron tres tiempos distintos: el tiempo de la función *app_picture*, que se desarrolló en [5] y es la responsable de la clasificación y detección de piezas en las imágenes, el tiempo que se tarda en enviar la información al robot desde que se obtiene toda la información necesaria para; y el tiempo total del procesado de un palé, desde que el robot manda el mensaje comunicando que ya está en posición para la captura de imágenes hasta que se le envía la información precisa, es decir que este último tiempo medido contiene a los otros dos.

En la Figura 68 se muestra el diagrama de cajas de la distribución de los tiempos de la función *app_picture* (la presencia de ciertos *outliers* dificulta el análisis del grueso de los datos, por lo que para su entendimiento se despreciaran. En la Figura 69 se muestran los mismos datos despreciando dichos *outliers*), la Figura 70 y la Figura 71 son las correspondientes al tiempo de envió de coordenadas al robot y del procesado de la pieza

	Media (s)	Mínimo (s)	Máximo (s)	Mediana(s)	Desviación típica(s)
app_picture	0.03781422	0.028617	0.1876	0.03063	0.027288945
Envío Robot	1.94694222	1.0827	3.6449	1.8598	0.475831691
Procesado Total	13.6249111	6.7497	25.5919	13.2369	3.703150175

Tabla 4: Estadísticos de los tiempos obtenidos

Observando la distribución de tiempos de *app_picture* y sus estadísticos, se puede concluir que los *outliers* son casos excepcionales que no tienen mucho peso como se puede observar en los estadísticos. A pesar de existir un tiempo máximo muy superior a la mediana, la media no se ve muy afectada, por lo que es de suponer que se pueden despreciar dichos datos, esto lo confirma la desviación típica que no es muy elevada. Por lo demás, el grueso de los tiempos se centra alrededor de 31 milisegundos.

En el caso de los tiempos de envío de coordenadas al robot hay que tener en cuenta que se incluye la transformación de coordenadas del sistema de la cámara al sistema del robot. También presenta algún *outlier*, pero de igual forma que en el caso de *app_picture* no afecta en gran medida a la media que se mantiene próxima a la mediana y como se puede observar analizando la desviación típica. En este caso los tiempos se concentran alrededor de los 2 segundos.

Por último, los tiempos de procesado total presentan una distribución similar a los de *app_picture* presentan valores superiores a la mediana, pero no afectan en gran medida a la media, como se deduce también por la desviación típica. Ambas se mantienen próximas a los 13.5 segundos.

6.2 TIEMPOS ROBOT

Para el caso del robot, se midieron los tiempos entre las distintas fases del montaje de una figura. El inicio se marca con la llegada del primer palé al retenedor principal después de haber ordenado la construcción de una figura. Para cada palé que se desvía se mide el tiempo del momento en el que se desvía, cuando llega a la posición de bloqueo, cuando se coge la pieza, cuando se deja y cuando vuelve el robot a la posición de origen. En este caso se plantea un análisis de las figuras con la llegada de palés con las piezas en el orden correspondiente a una de las formas de construir la figura, esto quiere decir que no se incluirán palés vacíos

o con piezas que no se vayan a coger en los ensayos. Esto es debido a que a partir del resto de tiempos se pueden extrapolar los tiempos que corresponderían a la inclusión de este, debido a que no implica movimiento del robot. En la Figura 72 se expone de manera visual donde se medirán los tiempos para cada figura.

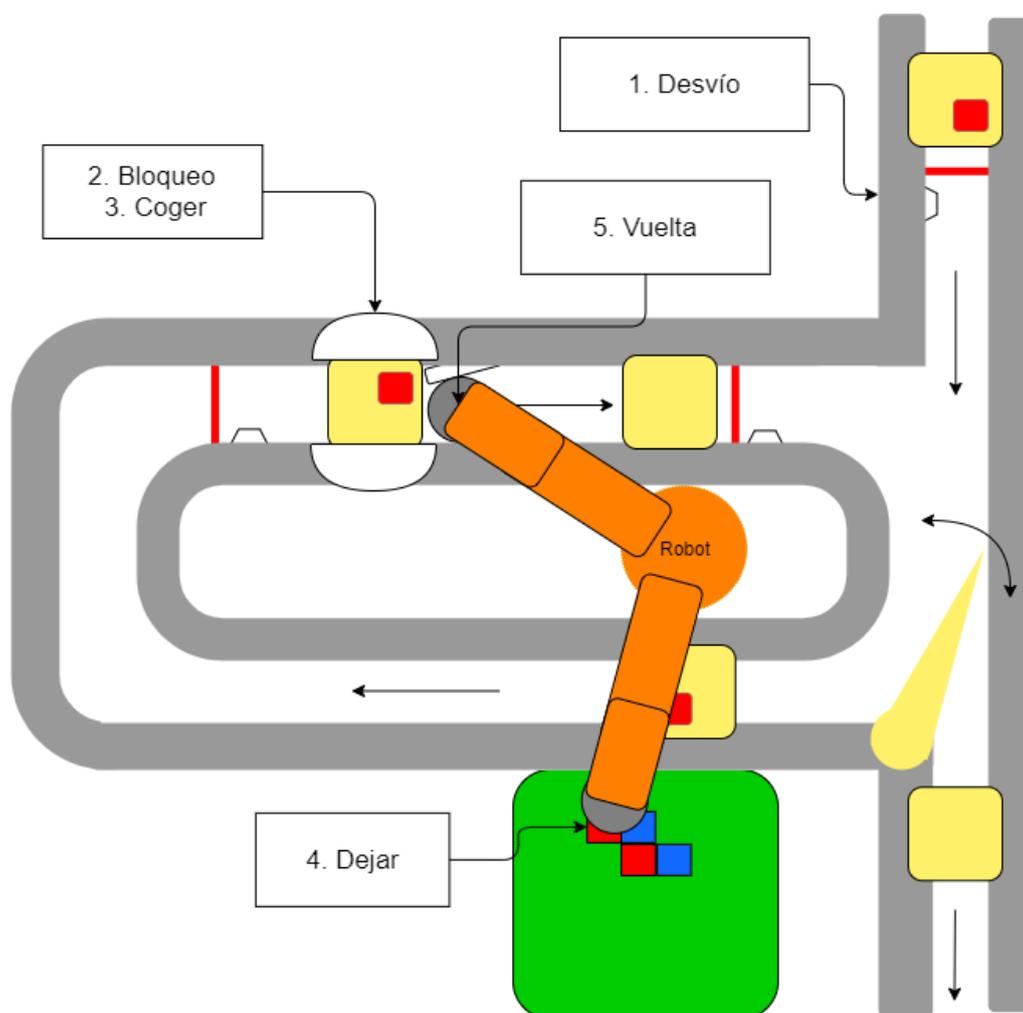


Figura 72: Esquema de los puntos de medida de tiempos del robot

Como se ha comentado anteriormente, cada figura tiene 2 formas de construirse. se realizaron 5 ensayos para cada forma de cada figura midiendo los tiempos. Los tiempos detallados de los ensayos se encuentran en el Anexo IV. Por otro lado, al ser los tiempos dependientes de la velocidad de las cintas se realizaron los ensayos correspondientes a la Tabla 5 y la Tabla 6 con la velocidad fijada al 40%, en cambio, para los ensayos de la Tabla

7 y la Tabla 8 se aumentó un 10% (se reajustaron tiempos de espera en la programación del robot) En este apartado se exponen los estadísticos destacados a modo de resumen. Para cada forma de construir una figura se expone la media, la desviación típica y el máximo y mínimo. Además, los tiempos se desglosan en el absoluto desde el inicio de la pieza y en el parcial desde la medida anterior.

	MEDIA		Desviación típica		Mínimo		Máximo	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	03:26.19	00:16.62	02:44.14	00:20.61	00:00.00	00:00.00	08:39.66	01:13.07
Bloqueo	04:11.05	00:44.86	02:44.09	00:00.29	00:44.68	00:44.00	09:24.25	00:45.09
Coger	04:39.97	00:28.92	02:48.21	00:10.25	01:00.22	00:15.13	10:17.64	00:53.39
Dejar	05:04.13	00:24.16	02:49.22	00:04.83	01:22.46	00:20.47	10:39.17	00:32.40
Vuelta	05:24.38	00:20.24	02:50.58	00:05.18	01:39.14	00:16.05	10:57.53	00:31.61

Tabla 5: Estadísticos de la construcción de la primera figura con la primera forma

	MEDIA		Desviación típica		Mínimo		Máximo	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	03:20.40	00:23.49	02:38.70	00:17.65	00:00.00	00:00.00	07:31.77	00:52.24
Bloqueo	04:05.58	00:45.17	02:38.76	00:00.21	00:45.02	00:45.01	08:16.92	00:45.92
Coger	04:27.99	00:22.41	02:41.04	00:05.80	00:58.55	00:13.53	08:36.43	00:38.62
Dejar	04:52.23	00:24.24	02:44.71	00:04.76	01:20.35	00:21.33	09:08.67	00:32.25
Vuelta	05:11.61	00:19.38	02:49.04	00:05.47	01:36.62	00:16.10	09:37.30	00:28.64

Tabla 6: Estadísticos de la construcción de la segunda figura con la primera forma

	MEDIA		Desviación típica		Mínimo		Máximo	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	02:58.05	00:16.41	02:17.40	00:13.99	00:00.00	00:00.00	06:12.64	00:38.89
Bloqueo	03:34.57	00:36.52	02:17.31	00:00.25	00:36.43	00:36.10	06:48.79	00:36.98
Coger	03:51.12	00:16.55	02:16.84	00:00.82	00:53.84	00:15.21	07:05.20	00:18.25
Dejar	04:19.41	00:28.29	02:22.48	00:07.09	01:14.84	00:20.64	07:37.12	00:37.79
Vuelta	04:38.81	00:19.40	02:26.62	00:05.38	01:31.49	00:15.82	08:05.67	00:28.64

Tabla 7: Estadísticos de la construcción de la primera figura con la segunda forma

	MEDIA		Desviación típica		Mínimo		Máximo	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	03:00.36	00:18.55	02:24.94	00:13.15	00:00.00	00:00.00	06:43.50	00:48.38
Bloqueo	03:36.97	00:36.61	02:24.87	00:00.29	00:36.50	00:36.34	07:20.00	00:37.59
Coger	03:56.61	00:19.65	02:26.03	00:01.32	00:54.28	00:17.45	07:39.97	00:21.86
Dejar	04:20.70	00:24.09	02:26.84	00:04.47	01:15.63	00:19.99	07:59.96	00:31.69
Vuelta	04:39.97	00:19.27	02:28.22	00:05.14	01:32.11	00:15.57	08:16.40	00:28.23

Tabla 8: Estadísticos de la construcción de la segunda figura con la segunda forma

De las tablas anteriores, es destacable el valor del máximo de la fila “Vuelta”, debido a que este es el peor tiempo en el que se ha completado la construcción de la figura. Para las dos primeras tablas se observa que la diferencia es de algo menos de minuto y medio. En cambio, para las otras dos, esta diferencia es más reducida.

De los tiempos parciales podemos extraer que son muy similares entre sí casi todos, salvo por el caso de la fila de “Bloqueo” que difiere de las dos primeras tablas a las siguientes debido a que se cambió la velocidad de las cintas transportadoras como se ha comentado antes.

De los tiempos que se obtienen si se incluyese un palé vacío en el proceso, este intervendría en los sectores de “Desvío” y de “Bloqueo”, podemos suponer que, al no ir a coger la pieza, sus tiempos parciales para “Dejar” y “Vuelta” será 0 y para “Coger” coincidirá con los valores de los tiempos de procesado total de una pieza en MATLAB, expuestos en la Tabla 4.

Por último, cabe destacar que estos tiempos se han obtenido operando con el robot en modo manual, si se hubiese trabajado en modo automático, los tiempos serían menores.

Capítulo 7. Conclusiones

Como se ha comentado anteriormente, los objetivos de este trabajo de final de grado no radican únicamente en los números. Se buscaba la integración de un sistema de visión artificial optimizado en la planta que constituirían el robot, las cintas transportadoras, la cámara y el ordenador.

El objetivo principal de este proyecto se logró con mucho éxito. Para poner en marcha todo el trabajo realizado, un operario solo debe iniciar el robot, arrancar la aplicación de MATLAB, establecer la comunicación entre ambos y finalmente, elegir la figura candidata a ser construida. Una vez completado todo este proceso, el operario tiene una última labor: supervisar la fabricación de la pieza y la gestión de los palés en el movimiento de la cinta transportadora.

Por otro lado, se analizaron distintos métodos para la integración del sistema de visión artificial desarrollado por Ignacio Ortiz de Zúñiga Mingot. En un inicio, se partió desde un enfoque flexible, con herramientas sencillas y transformaciones simples. Después una gran labor de captación de información y comprensión de las redes disponibles, se consiguió un sistema personalizado para el robot, con gran precisión y robustez para este proyecto en concreto.

Sin embargo, hay un problema persistente en estos momentos del proyecto. Este consiste en la calibración inicial de todos los puntos del palé, para conseguir que el robot sea capaz de coger la pieza con exactitud y sin ocasionar daños en la misma ni en el brazo robótico.

Respecto a la gestión de los palés, resulta interesante señalar la posibilidad de aplicar una estructura similar para todos los robots de la planta. Además, se ha trabajado siempre con el robot en modo manual, para evitar percances imprevistos y controlar la ejecución continuamente. Cabe destacar que, si se llevase a cabo el proyecto en modo automático, los movimientos se realizarían de una manera más rápida, lo que también mejoraría los tiempos de ejecución de toda la planta en su conjunto.

Para enriquecer el proyecto, se ha desarrollado la lógica para la construcción de las figuras de diferentes maneras. Para poder implementar algún cambio en alguna de las estructuras establecidas, se debe adaptar y programar el código correspondiente.

Finalmente, analizando los objetivos cumplidos, se puede concluir que el trabajo se ha desarrollado con éxito y que se puede implantar en el resto de los robots industriales realizando los ajustes pertinentes. Además, este robot puede llevar a cabo otras funciones provechosas para la universidad. Una de las ideas reflexionadas, sería la adaptación del sistema para programarlo como un robot colaborativo, aunque el modelo de robot empleado en este proyecto no se pueda emplear por seguridad, el sistema se podría adaptar a un modelo viable. Por otro lado, también sería útil la implementación del robot de una forma didáctica, para así poder exponer cómo manejar la planta o cualquier otro sistema a su disposición.

Capítulo 8. Futuros desarrollos

La integración del sistema de visión artificial del que se partía en la planta de la minifábrica de ICAI se ha desarrollado con éxito. Sin embargo, existen aspectos que se podrían mejorar para dotar de una mayor flexibilidad al sistema o mejorar los tiempos de ejecución.

- Mejora del sistema de visión artificial. Dotar de una mayor precisión al sistema de visión puede hacer que se obtenga un mejor resultado y se pueda flexibilizar el sistema global para su implantación en otros robots con uno de los métodos del apartado 4.3, aquí descartados.
- Tratar con más de una pieza en los palés. Si se optimiza a este punto los tiempos de ejecución pueden mejorar, ya que se incluyen más posibilidades de que se tenga que coger una pieza del palé, lo que reduciría el número de palés con piezas innecesarias desviados
- Optimización de la lógica empleada y de las comunicaciones. Es uno de los aspectos que más tiempo consume, por lo que, si se optimiza este apartado, puede desembocar en una gran mejora
- Diseño de un nuevo sistema de agarre. Si bien las pinzas proporcionan un sistema robusto para la captura de objetos, estorban en la misma tarea. Un sistema que solo haga contacto con la parte superior de las piezas puede ser mucho más eficiente y evitar tener que desarrollar formas en las que la orientación de la pinza no sea un problema a la hora de colocar las piezas (o de cogerlas si tienen una forma no cuadrada).
- Inclusión de piezas de diferentes formas en el sistema de visión artificial. Así se expandirían las posibilidades para crear nuevas figuras.
- Migración a un nuevo lenguaje de programación más rápido. Las comunicaciones entre el robot y el ordenador no requieren precisamente de MATLAB y lo mismo ocurre con el ordenador y la cámara, si se migrase el sistema a un nuevo lenguaje más rápido se optimizaría el tiempo de procesado de las imágenes y se reduciría el global de construcción de las figuras.

- Cambiar la lógica que sigue MATLAB para descartar un palé o no. Si un palé lleva una pieza que no se necesita en ese preciso instante será liberado y no se cogerá la pieza. Un posible cambio es que se coja la pieza y se lleve a una posición temporal para colocarla en su sitio cuando proceda.

Como resumen, en este proyecto, la optimización del tiempo no era un objetivo fundamental, pero futuros desarrollos de este pueden incluirlo. Además, partiendo de esta base se debe de tratar de flexibilizar para su fácil adaptación a un nuevo robot en una nueva posición.

Capítulo 9. Referencias

- [1] M. Cotteleer y B. Sniderman, «Forces of change: Industry 4.0» Diciembre 2017. Disponible: https://www2.deloitte.com/content/dam/insights/us/articles/4323_Forces-of-change/4323_Forces-of-change_Ind4-0.pdf. [Último acceso: 4 Diciembre 2020].
- [2] C.-Y. Tsai, C.-C. Wong, C.-J. Yu, C.-C. Liu y T.-Y. Liu, «A Hybrid Switched Reactive-Based Visual Servo» Marzo 2015. Disponible: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6920015>. [Último acceso: 21 Enero 2021].
- [3] S. Soltan, A. Oleinikov, M. F. Demirci y A. Shintemirov, «Deep Learning-Based Object Classification and Position Estimation Pipeline for Potential Use in Robotized Pick-and-Place Operations» 18 Agosto 2020. Disponible: https://www.mdpi.com/2218-6581/9/3/63?type=check_update&version=1. [Último acceso: 27 Diciembre 2020].
- [4] ABB, «IRB 120» Disponible: <https://new.abb.com/products/robotics/es/robots-industriales/irb-120>. [Último acceso: 10 Enero 2021].
- [5] I. Ortiz de Zúñiga Mingot, «Optimización de un sistema de visión artificial de un robot industrial» Trabajo de Fin de Grado, ICAI.
- [6] «Realsense D435». Disponible: <https://www.intereasense.com/depth-camerad435/>. [Último acceso: 28 Diciembre 2020].
- [7] S. Hutchinson, G. D. Hager y P. I. Corke, «A Tutorial on Visual Servo Control» 5 Octubre 1996. Disponible: <https://ieeexplore.ieee.org/document/538972>. [Último acceso: 25 Enero 2021].
- [8] A. Krizhevsky, I. Sutskever y G. E. Hinton, «ImageNet Classification with Deep Convolutional Neural Networks» Diciembre 2012. Disponible: <https://dl.acm.org/doi/10.5555/2999134.2999257>. [Último acceso: 15 Enero 2021].
- [9] K. Simonyan, A. Zisserman, «Very Deep Convolutional Networks for Large-Scale Image Recognition» 10 Abril 2015. Disponible: <https://arxiv.org/Absoluto/1409.1556v6>. [Último acceso: 23 Enero 2021].
- [10] «AlexNet – ImageNet Classification with Deep Convolutional Neural Networks» Disponible: <https://test.neurohive.io/en/popular-networks/alexnet-imagenet-classification-with-deep-convolutional-neural-networks/>
- [11] W. Nash, T. Drummond, N. Birbilis, «A review of deep learning in the study of materials degradation» 15 Noviembre 2018. Disponible: https://www.researchgate.net/publication/328966158_A_review_of_deep_learning_in_the_study_of_materials_degradation
- [12] «R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms» Disponible: <https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e>

- [13] C. Hernandez, M. Bharatheesha, W. Ko, H. Gaiser, J. Tan, K. Van Deurzen, M. de Vries, B. Van Mil, J. Van Egmond, R. Burger, M. Morariu, J. Ju, X. Gerrmann, R. Ensing, J. Van Frankenhuyzen, M. Wisse, «Team Delft's Robot Winner of the Amazon Picking Challenge 2016» 1 Noviembre 2017. Disponible: https://link.springer.com/chapter/10.1007/978-3-319-68792-6_51. [Último acceso: 4 Febrero 2021].
- [14] A.P. del Pobil, M. Kassawat, A. J. Duran, M.A. Arias, N. Nechyporenko, A. Mallick, E. Cervera, D. Subedi, I. Vasilev, D. Cardin, E. Sansebastiano, E. Martinez-Martin, A. Morales, G.A. Casañ, A. Arenal, B. Goriatcheff, C. Rubert and G. Recatala, «UJI RobInLab's Approach to the Amazon Robotics Challenge 2017» 16 Noviembre 2017. Disponible: <https://ieeexplore.ieee.org/document/8170448>. [Último acceso: 5 Febrero 2021].
- [15] J.P. Saut, D. Sidobre, «Efficient models for grasp planning with a multi-fingered hand» 26 Agosto 2011. Disponible: <https://www.sciencedirect.com/science/article/Absoluto/pii/S0921889011001515>. [Último acceso: 4 Febrero 2021].
- [16] J. Shi, G.S. Koonjul, «Real-Time Grasping Planning for robotic Bin-Picking and Kitting Applications» 2 Abril 2017. Disponible: <https://ieeexplore.ieee.org/document/7893003>. [Último acceso: 4 Febrero 2021].
- [17] A. Zeng, S. Song, K. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo, N. Fazeli, F. Alet, N. C. Dafle, R. Holladay, I. Morona, P. Q. Nair, D. Green, I. Taylor, W. Liu, T. Funkhouser, A. Rodriguez, «Robotic Pick-and-Place of Novel Objects in Clutter with Multi-Affordance Grasping and Cross-Domain Image Matching» 30 Mayo 2020. Disponible: <https://arxiv.org/pdf/1710.01330.pdf>. [Último acceso: 4 Febrero 2021].
- [18] K. Saenko, B. Kulis, M. Fritz, T. Darrell, «Adapting Visual Category Models to New Domains» 2010. Disponible: https://link.springer.com/chapter/10.1007/978-3-642-15561-1_16. [Último acceso: 4 Febrero 2021].
- [19] F. Li, Y. Liang, T. Li, Y. Feng, S. Chen, , «Design of a robot end effector with measurement system for precise pick-and-place of square objects» 2020. Disponible: <https://www.sciencedirect.com/science/article/pii/S2351978920314864>. [Último acceso: 4 Febrero 2021].
- [20] Y. Zhang, B. K. Chen, X. Liu, Y. Sun, «Autonomous Robotic Pick-and-Place of Microobjects» 1 Febrero 2010. Disponible: <https://ieeexplore.ieee.org/document/5342467>. [Último acceso: 5 Febrero 2021].
- [21] S. Wade-McCue, N. Kelly-Boxall, M. McTaggart, D. Morrison, A.W. Tow, J. Erskine, R. Grinover, A. Gurman, T. Hunn, D. Lee, A. Milan, T. Pham, G. Rallos, A. Razjigaev, T. Rowntree, R. Smith, K. Vijay, Z. Zhuang, C. Lehnert, I. Reid, P. Corke, J. Leitner, «Design of a Multi-Modal End-Effector and Grasping System How

- Integrated Design helped win the Amazon Robotics Challenge» 19 Junio 2018.
Disponible: <https://arxiv.org/Absoluto/1710.01439>. [Último acceso: 5 Febrero 2021].
- [22] «RobotStudio 2020.4» Disponible:
<https://new.abb.com/products/robotics/robotstudio/downloads>. [Último acceso: 10 Enero 2021].
- [23] «MATLAB wrapper for Intel Realsense», Disponible:
<https://github.com/InteReaSense/ibreasense/tree/master/wrappers/matab> . [Último acceso: 28 Diciembre 2020].
- [24] «Objetivos de desarrollo sostenible» Disponible:
<https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>.



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

CAPÍTULO 9: REFERENCIAS

Anexo I Desarrollo de la aplicación de MATLAB

A lo largo de la memoria se ha comentado que es lo que tenía que hacer el usuario para poder trabajar con la aplicación de MATLAB, pero sin explicar cómo se desarrolló. Este anexo tiene dicho propósito, explicar con detalle dicha aplicación, su interfaz y funcionalidades.

El punto de partida se encuentra en la aplicación desarrollada por Ignacio Ortiz de Zúñiga Mingot que se expone en el apartado 4.1 de esta memoria. Desde ahí fue evolucionando hasta llegar al resultado final que se expone en la Figura A.1. 1 a continuación. Se explicará detalladamente las diferencias con el sistema de partida y los nuevos elementos integrados.

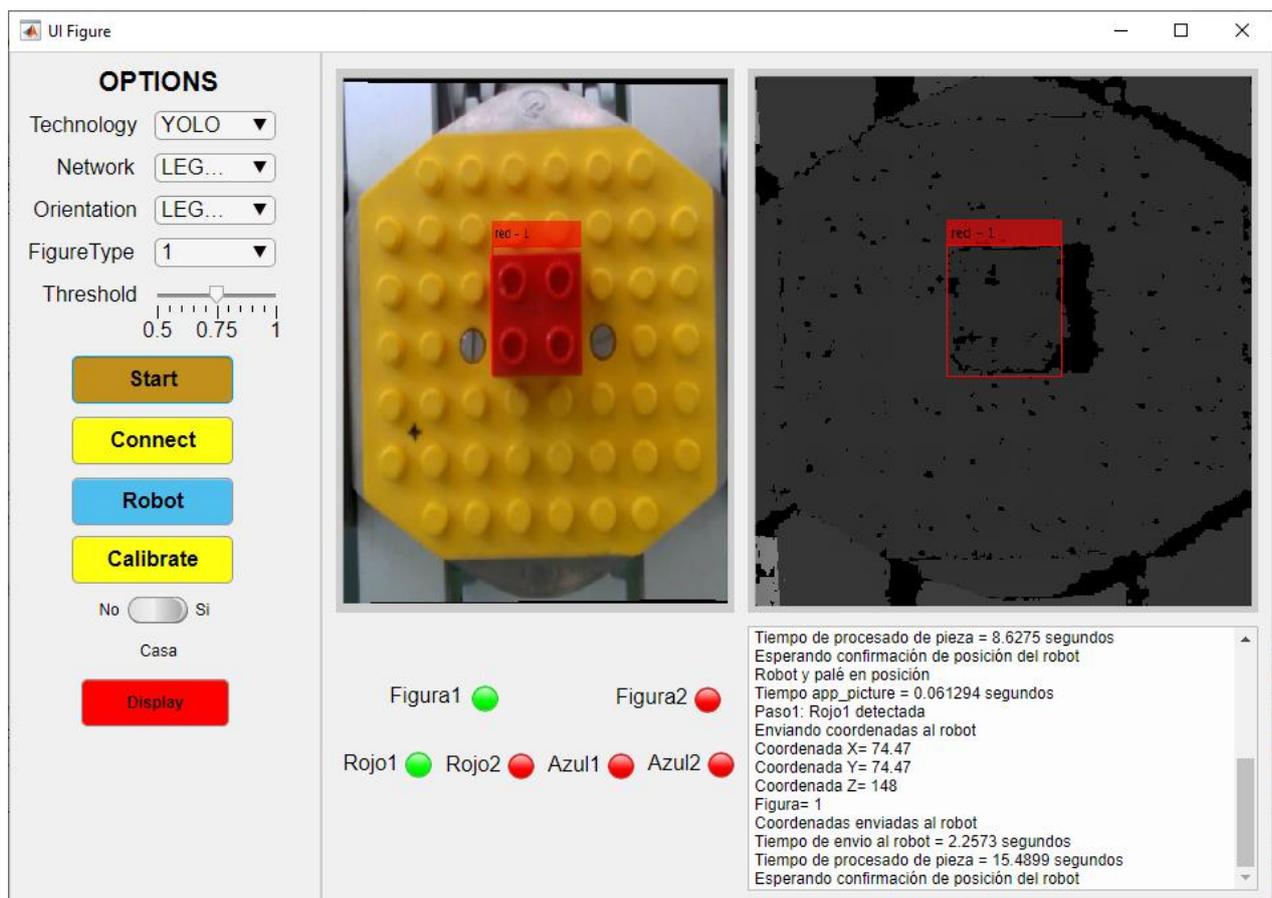


Figura A.1. 1: Disposición de los elementos de la aplicación de MATLAB

I. IPANEL IZQUIERDO

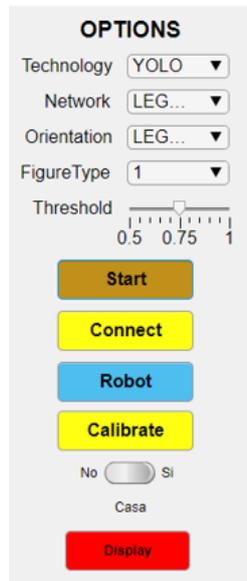


Figura A.1. 2: Panel izquierdo de la interfaz actual

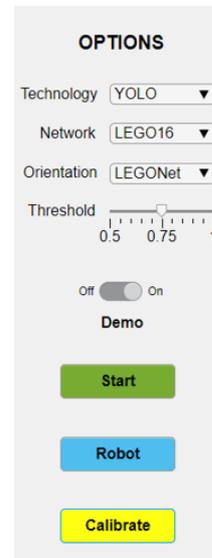


Figura A.1. 3: Panel izquierdo de la interfaz de partida

Este panel contiene todos los elementos con los que puede interactuar el usuario. Primero se comentarán los elementos que se conservan de la Figura A.1. 3:

- Desplegables “Technology”, “Network”, “Orientation”: estos desplegables, sirven para seleccionar el tipo de red neuronal que se desea emplear para la localización de la figura, su clasificación y la detección de la orientación respectivamente.
- *Slider threshold*: este elemento sirve para limitar el intervalo de confianza con el que se detectan y localizan las piezas.
- Botón *Start*: botón que iniciaba el antiguo sistema de visión artificial. Actualmente se ha modificado para que sirva como botón de inicio de la construcción de las figuras una vez establecidos los parámetros de los desplegables y el *slider*.
- Botón *robot*: botón cuya función es mandar los datos obtenidos del análisis de imágenes al robot. En el antiguo sistema no se llegó a mandar al robot, en vez de eso, se mostraba por pantalla dicha información. Actualmente se ha desarrollado para que cumpla la función del envío de los datos al robot.
- Botón *calibrate*: botón empleado para calibrar las imágenes obtenidas por la cámara.

Como nuevos elementos se pueden encontrar:

- Desplegable *FigureType*: Este desplegable es el que se emplea para la elección de la figura que se desee construir.
- Botón *Connect*: botón empleado para establecer la conexión entre el ordenador y el robot a través del protocolo TCP/IP.
- Interruptor *Casa*: interruptor que permite el trabajo sólo con la cámara simulando el comportamiento del robot.
- Botón *Display*: botón que muestra en pantalla la posición del interruptor *Casa*.

I.II PANEL DERECHO

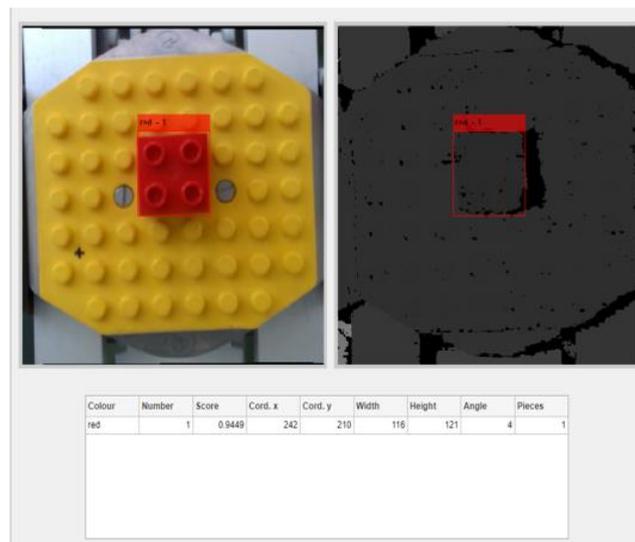
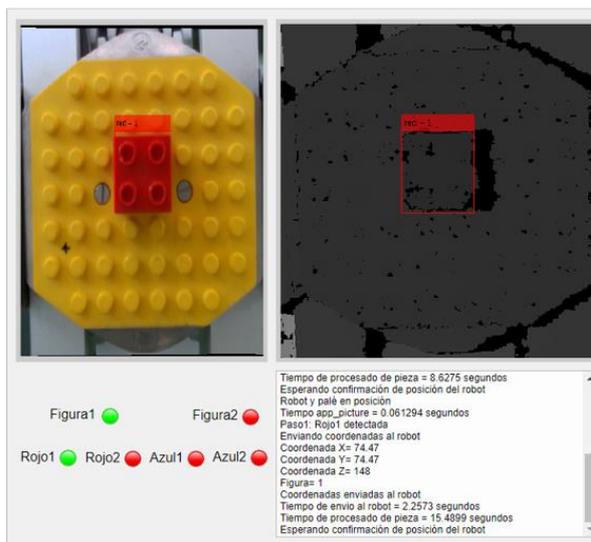


Figura A.1. 4: Panel derecho de la interfaz actual

Figura A.1. 5: Panel derecho de la interfaz de partida

Del mismo modo que en el caso anterior, se han mantenido ciertos elementos. Estos son los dos recuadros superiores que actúan como pantallas para mostrar imágenes, o de las figuras que se pueden construir o bien de la pieza analizada en el momento.

La parte inferior del panel se ha sustituido completamente. A un lado se presenta un conjunto de lámparas para indicar el estado de ciertas variables. La fila superior indica el tipo de figura

que se ha seleccionado y que se va a construir, y la fila inferior las piezas de esta que se han detectado ya y las que faltan por detectar.

Además, en la parte derecha se puede encontrar una consola de texto que informa sobre detalles del proceso en cuestión. Indica el inicio y fin de la construcción de las figuras, que pieza se ha detectado y las coordenadas a las que se va a enviar, si se está esperando al robot y a que llegue un palé y, además, ofrece información sobre los tiempos del proceso.

Anexo II Carcasa de la cámara

Como se ha podido observar en distintas partes de la memoria, la cámara se instala en la muñeca del robot con una carcasa protectora. El diseño de esta se heredó de otros proyectos y consiste en dos piezas distintas, la propia carcasa de la cámara, y el adaptador de la carcasa a la muñeca del robot.

A lo largo del curso y durante el transcurso de este proyecto, el adaptador se quedaba instalado en la muñeca del robot, pero la cámara y la carcasa de la misma se retiraban por seguridad. Debido a un accidente ocasionado por un alumno durante una clase, el adaptador se rompió, quedando totalmente inutilizado.

Por todo lo comentado anteriormente, al tener la necesidad de volver a imprimir el adaptador de la cámara, se optó por rediseñarlo para así optimizar su tamaño y tener más libertad de movimiento con la cámara instalada. Cabe destacar que, al rediseñar el adaptador mencionado, se tuvo que realizar el mismo proceso con la carcasa de la propia cámara para ajustar con precisión las uniones entre las dos piezas.

A continuación, se muestra en la Figura A.2 1 y la Figura A.2 2 la comparación entre los diseños antiguos y los nuevos. Se consiguió reducir el largo en 1cm y con ello el ancho manteniendo las proporciones precisas.

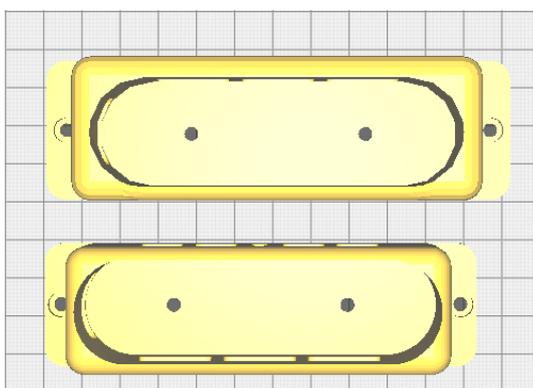


Figura A.2 1: Comparación carcasa de la cámara

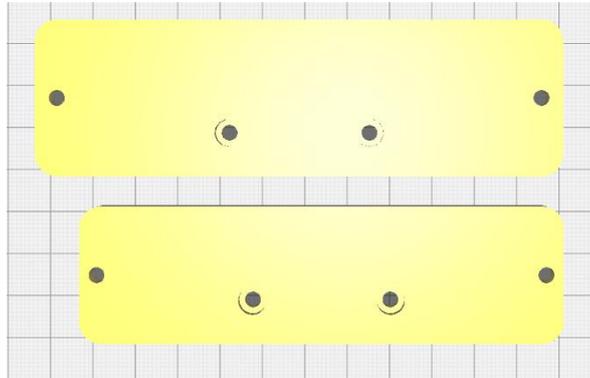


Figura A.2 2: Comparación Adaptador cámara-robot

Anexo III Alineación con los ODS

Vivimos en una época crítica, en la que el uso de los recursos energéticos y materiales ha de ser responsable. En la industria esto ha de estar muy presente ya que es uno de los principales consumidores. La Industria 4.0 es el escenario idóneo para impulsar estos objetivos [24] y marcar nuevos para el avance continuo hasta el punto del desarrollo sostenible.

Este proyecto encaja en el contexto del objetivo 9: “Industria, innovación e infraestructuras”, que fomenta la industrialización inclusiva y sostenible; y del objetivo 12: “Producción y consumo responsables”, con el que se busca garantizar modalidades de consumo y producción responsables. Esto es debido a los motivos expuestos a continuación.

El uso de sistemas de visión artificial y la aplicación de la robótica en la industria fomenta el uso de los recursos de una manera eficiente. Además, su aplicación no se limita a la industria de producción.

En las plantas de reciclaje se puede aplicar este mismo sistema para la correcta clasificación de los residuos, facilitando su reciclaje. De esta manera serviría como un nuevo integrante en la economía circular.

De igual manera, puede aplicarse este sistema en un entorno de trabajo colaborativo para facilitar las tareas correspondientes al operario con el que trabaje.

Así pues, esta tecnología entra en juego en la nueva industria abriendo las puertas hacia la producción sostenible y eficiente, así como a la extensión de la vida útil de los productos y su reintegración en el ciclo de producción.



COMILLAS
UNIVERSIDAD PONTIFICIA

ICAI

ANEXO III

Anexo IV Tablas de tiempos de los procesos del robot

Este anexo es necesario debido a que no era viable poner todos los tiempos obtenidos en su capítulo correspondiente, por lo que un desglose más detallado de estos se expondrá aquí.

Figura1 Forma1								
Ensayo1								
	ROJO1		AZUL2		ROJO2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	02:46.03	00:48.83	04:56.00	00:19.14	08:39.66	01:13.07
Bloqueo	00:44.68	00:44.68	03:30.56	00:44.53	05:40.00	00:44.00	09:24.25	00:44.59
Coger	01:14.34	00:29.66	03:56.11	00:25.55	06:22.93	00:42.93	10:17.64	00:53.39
Dejar	01:36.48	00:22.14	04:16.66	00:20.55	06:54.98	00:32.05	10:39.17	00:21.53
Vuelta	01:57.20	00:20.72	04:36.86	00:20.20	07:26.59	00:31.61	10:57.53	00:18.36
Ensayo2								
	ROJO1		AZUL2		ROJO2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	02:08.68	00:01.00	04:15.30	00:23.01	06:26.65	00:03.00
Bloqueo	00:45.02	00:45.02	02:53.60	00:44.92	05:00.30	00:45.00	07:11.64	00:44.99
Coger	01:29.24	00:44.22	03:14.43	00:20.83	05:23.93	00:23.63	07:40.53	00:28.89
Dejar	01:50.88	00:21.64	03:35.55	00:21.12	05:56.32	00:32.40	08:02.27	00:21.74
Vuelta	02:07.68	00:16.80	03:52.29	00:16.74	06:23.65	00:27.33	08:19.74	00:17.47
Ensayo3								
	ROJO1		AZUL2		ROJO2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	02:06.16	00:27.02	04:13.74	00:25.32	06:21.63	00:01.00
Bloqueo	00:45.09	00:45.09	02:51.22	00:45.06	04:58.81	00:45.07	07:06.63	00:45.01
Coger	01:00.22	00:15.13	03:11.57	00:20.35	05:20.71	00:21.90	07:26.90	00:20.27
Dejar	01:22.46	00:22.24	03:32.05	00:20.47	05:52.86	00:32.15	07:48.38	00:21.48
Vuelta	01:39.14	00:16.68	03:48.42	00:16.37	06:20.63	00:27.76	08:05.14	00:16.76
Ensayo4								
	ROJO1		AZUL2		ROJO2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	02:10.05	00:21.78	04:15.33	00:16.67	06:39.79	00:06.00
Bloqueo	00:44.87	00:44.87	02:55.06	00:45.01	05:00.31	00:44.98	07:24.81	00:45.02
Coger	01:09.53	00:24.66	03:21.49	00:26.43	05:34.19	00:33.88	07:55.81	00:31.00
Dejar	01:31.65	00:22.12	03:42.62	00:21.13	06:06.45	00:32.25	08:17.35	00:21.54
Vuelta	01:48.27	00:16.62	03:58.67	00:16.05	06:33.78	00:27.34	08:34.44	00:17.08

Tabla A.4. 1: Tiempos de ensayos de la figura 1 con la forma 1

Figura1 Forma1								
MEDIAS								
	ROJO1		AZUL2		ROJO2		AZUL1	
	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial
Desvío	00:00.00	00:00.00	02:17.73	00:24.66	04:25.09	00:21.04	07:01.93	00:20.77
Bloqueo	00:44.91	00:44.91	03:02.61	00:44.88	05:09.86	00:44.76	07:46.83	00:44.90
Coger	01:13.33	00:28.42	03:25.90	00:23.29	05:40.44	00:30.58	08:20.22	00:33.39
Dejar	01:35.37	00:22.03	03:46.72	00:20.82	06:12.65	00:32.21	08:41.79	00:21.57
Vuelta	01:53.07	00:17.71	04:04.06	00:17.34	06:41.16	00:28.51	08:59.21	00:17.42
Desviación típica								
	ROJO1		AZUL2		ROJO2		AZUL1	
	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial
Desvío	0	0	0.0002192	0.0002274	0.0002386	4.481E-05	0.0007593	0.0004043
Bloqueo	2.097E-06	2.097E-06	0.0002164	2.795E-06	0.0002327	5.9E-06	0.0007569	2.401E-06
Coger	0.0001402	0.0001405	0.000238	3.638E-05	0.0003345	0.0001132	0.0009163	0.0001634
Dejar	0.0001373	3.091E-06	0.0002366	4.116E-06	0.0003333	1.713E-06	0.000916	1.345E-06
Vuelta	0.0001414	2.328E-05	0.0002578	2.231E-05	0.0003565	2.404E-05	0.0009234	7.997E-06
Mínimo								
	ROJO1		AZUL2		ROJO2		AZUL1	
	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial
Desvío	00:00.00	00:00.00	02:06.16	00:01.00	04:13.74	00:16.67	06:21.63	00:01.00
Bloqueo	00:44.68	00:44.68	02:51.22	00:44.53	04:58.81	00:44.00	07:06.63	00:44.59
Coger	01:00.22	00:15.13	03:11.57	00:20.35	05:20.71	00:21.90	07:26.90	00:20.27
Dejar	01:22.46	00:21.64	03:32.05	00:20.47	05:52.86	00:32.05	07:48.38	00:21.48
Vuelta	01:39.14	00:16.62	03:48.42	00:16.05	06:20.63	00:27.33	08:05.14	00:16.76
Máximo								
	ROJO1		AZUL2		ROJO2		AZUL1	
	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial
Desvío	00:00.00	00:00.00	02:46.03	00:48.83	04:56.00	00:25.32	08:39.66	01:13.07
Bloqueo	00:45.09	00:45.09	03:30.56	00:45.06	05:40.00	00:45.07	09:24.25	00:45.02
Coger	01:29.24	00:44.22	03:56.11	00:26.43	06:22.93	00:42.93	10:17.64	00:53.39
Dejar	01:50.88	00:22.24	04:16.66	00:21.13	06:54.98	00:32.40	10:39.17	00:21.74
Vuelta	02:07.68	00:20.72	04:36.86	00:20.20	07:26.59	00:31.61	10:57.53	00:18.36

Tabla A.4. 2: Estadísticos de los tiempos de la figura 1 con la forma 1 desglosados por pieza

Figura2 Forma1								
Ensayo1								
	ROJO1		AZUL2		ROJO2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	02:21.07	00:29.02	04:51.63	00:47.59	07:31.77	00:52.24
Bloqueo	00:45.15	00:45.15	03:06.16	00:45.09	05:36.64	00:45.01	08:16.92	00:45.15
Coger	01:14.01	00:28.86	03:26.34	00:20.18	06:01.62	00:24.98	08:36.43	00:19.51
Dejar	01:35.78	00:21.77	03:47.73	00:21.40	06:23.43	00:21.81	09:08.67	00:32.24
Vuelta	01:52.06	00:16.27	04:04.04	00:16.31	06:39.53	00:16.10	09:37.30	00:28.64
Ensayo2								
	ROJO1		AZUL2		ROJO2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:43.63	00:07.01	04:03.61	00:38.72	06:23.42	00:34.22
Bloqueo	00:45.02	00:45.02	02:28.80	00:45.17	04:48.71	00:45.10	07:08.54	00:45.12
Coger	00:58.55	00:13.53	02:47.25	00:18.45	05:11.49	00:22.78	07:30.46	00:21.92
Dejar	01:20.35	00:21.80	03:08.59	00:21.33	05:32.93	00:21.44	08:02.62	00:32.16
Vuelta	01:36.62	00:16.27	03:24.90	00:16.31	05:49.20	00:16.27	08:31.26	00:28.64
Ensayo3								
	ROJO1		AZUL2		ROJO2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	02:12.72	00:34.45	04:42.30	00:40.02	06:54.35	00:26.01
Bloqueo	00:45.23	00:45.23	02:57.78	00:45.06	05:27.39	00:45.09	07:39.55	00:45.20
Coger	00:59.75	00:14.52	03:24.36	00:26.58	05:50.53	00:23.14	08:18.17	00:38.62
Dejar	01:21.52	00:21.77	03:45.97	00:21.60	06:12.07	00:21.54	08:50.41	00:32.24
Vuelta	01:38.27	00:16.75	04:02.28	00:16.31	06:28.34	00:16.27	09:19.04	00:28.64
Ensayo4								
	ROJO1		AZUL2		ROJO2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	02:07.52	00:22.01	04:15.04	00:24.01	06:19.42	00:20.62
Bloqueo	00:45.15	00:45.15	02:52.69	00:45.17	05:00.14	00:45.11	07:05.34	00:45.92
Coger	01:07.36	00:22.21	03:13.39	00:20.70	05:21.09	00:20.95	07:26.98	00:21.64
Dejar	01:29.24	00:21.88	03:34.72	00:21.33	05:42.44	00:21.35	07:59.23	00:32.25
Vuelta	01:45.51	00:16.27	03:51.03	00:16.31	05:58.80	00:16.36	08:27.52	00:28.29

Tabla A.4. 3: Tiempos de ensayos de la figura 2 con la forma 1

Figura2 Forma1								
MEDIAS POR PIEZA								
	ROJO1		AZUL2		ROJO2		AZUL1	
	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial
Desvío	00:00.00	00:00.00	02:06.24	00:23.12	04:28.14	00:37.58	06:47.24	00:33.27
Bloqueo	00:45.14	00:45.14	02:51.36	00:45.12	05:13.22	00:45.08	07:32.59	00:45.35
Coger	01:04.92	00:19.78	03:12.84	00:21.48	05:36.18	00:22.96	07:58.01	00:25.42
Dejar	01:26.72	00:21.81	03:34.25	00:21.41	05:57.72	00:21.54	08:30.23	00:32.22
Vuelta	01:43.12	00:16.39	03:50.56	00:16.31	06:13.97	00:16.25	08:58.78	00:28.55
Desviación típica								
	ROJO1		AZUL2		ROJO2		AZUL1	
	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial
Desvío	0	0	0.000186	0.0001376	0.000261	0.0001141	0.0003882	0.00016
Bloqueo	1.008E-06	1.008E-06	0.0001856	6.446E-07	0.0002606	5.208E-07	0.0003859	4.43E-06
Coger	8.345E-05	8.321E-05	0.0002081	4.091E-05	0.0002747	1.913E-05	0.0004012	0.0001026
Dejar	8.349E-05	5.625E-07	0.000209	1.485E-06	0.0002767	2.29E-06	0.0004014	4.517E-07
Vuelta	8.219E-05	2.768E-06	0.000209	6.682E-09	0.0002758	1.241E-06	0.0004026	2.016E-06
Mínimo								
	ROJO1		AZUL2		ROJO2		AZUL1	
	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial
Desvío	00:00.00	00:00.00	01:43.63	00:07.01	04:03.61	00:24.01	06:19.42	00:20.62
Bloqueo	00:45.02	00:45.02	02:28.80	00:45.06	04:48.71	00:45.01	07:05.34	00:45.12
Coger	00:58.55	00:13.53	02:47.25	00:18.45	05:11.49	00:20.95	07:26.98	00:19.51
Dejar	01:20.35	00:21.77	03:08.59	00:21.33	05:32.93	00:21.35	07:59.23	00:32.16
Vuelta	01:36.62	00:16.27	03:24.90	00:16.31	05:49.20	00:16.10	08:27.52	00:28.29
Máximo								
	ROJO1		AZUL2		ROJO2		AZUL1	
	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial	ABSOLUTO	Parcial
Desvío	00:00.00	00:00.00	02:21.07	00:34.45	04:51.63	00:47.59	07:31.77	00:52.24
Bloqueo	00:45.23	00:45.23	03:06.16	00:45.17	05:36.64	00:45.11	08:16.92	00:45.92
Coger	01:14.01	00:28.86	03:26.34	00:26.58	06:01.62	00:24.98	08:36.43	00:38.62
Dejar	01:35.78	00:21.88	03:47.73	00:21.60	06:23.43	00:21.81	09:08.67	00:32.25
Vuelta	01:52.06	00:16.75	04:04.04	00:16.31	06:39.53	00:16.36	09:37.30	00:28.64

Tabla A.4. 4: Estadísticos de los tiempos de la figura 2 con la forma 1 desglosados por pieza

Figura1 Forma2								
Ensayo1								
	AZUL1		ROJO2		AZUL2		ROJO1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:49.24	00:17.01	03:45.80	00:24.01	05:58.65	00:17.01
Bloqueo	00:36.57	00:36.57	02:25.66	00:36.42	04:22.30	00:36.50	06:35.08	00:36.43
Coger	00:54.54	00:17.97	02:44.40	00:18.74	04:42.25	00:19.95	06:56.94	00:21.86
Dejar	01:15.96	00:21.42	03:05.48	00:21.08	05:13.86	00:31.61	07:18.69	00:21.75
Vuelta	01:32.22	00:16.26	03:21.79	00:16.31	05:41.64	00:27.78	07:34.95	00:16.26
Ensayo2								
	AZUL1		ROJO2		AZUL2		ROJO1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:53.10	00:20.99	03:48.90	00:19.54	06:13.48	00:27.53
Bloqueo	00:36.50	00:36.50	02:29.66	00:36.56	04:25.53	00:36.63	06:49.94	00:36.46
Coger	00:54.72	00:18.22	02:49.47	00:19.81	04:46.41	00:20.88	07:11.50	00:21.56
Dejar	01:16.54	00:21.82	03:12.54	00:23.07	05:17.72	00:31.31	07:33.13	00:21.63
Vuelta	01:32.11	00:15.57	03:29.36	00:16.82	05:45.95	00:28.23	07:49.64	00:16.51
Ensayo3								
	AZUL1		ROJO2		AZUL2		ROJO1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:56.41	00:23.96	03:59.52	00:26.78	06:43.50	00:48.38
Bloqueo	00:36.61	00:36.61	02:34.00	00:37.59	04:35.86	00:36.34	07:20.00	00:36.50
Coger	00:54.49	00:17.88	02:53.98	00:19.98	04:55.63	00:19.77	07:39.97	00:19.97
Dejar	01:16.38	00:21.89	03:16.62	00:22.64	05:27.32	00:31.69	07:59.96	00:19.99
Vuelta	01:32.45	00:16.07	03:32.74	00:16.12	05:55.12	00:27.80	08:16.40	00:16.44
Ensayo4								
	AZUL1		ROJO2		AZUL2		ROJO1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:52.50	00:20.00	03:51.03	00:24.79	06:13.63	00:26.76
Bloqueo	00:36.83	00:36.83	02:29.12	00:36.62	04:27.42	00:36.39	06:50.36	00:36.73
Coger	00:54.28	00:17.45	02:48.23	00:19.11	04:47.84	00:20.42	07:11.12	00:20.76
Dejar	01:15.63	00:21.35	03:09.30	00:21.07	05:19.20	00:31.36	07:32.86	00:21.74
Vuelta	01:32.50	00:16.87	03:26.24	00:16.94	05:46.87	00:27.67	07:49.50	00:16.64

Tabla A.4. 5: Tiempos de ensayos de la figura 1 con la forma 2

Figura1 Forma2								
Media								
	AZUL1		ROJO2		AZUL2		ROJO1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:52.81	00:20.49	03:51.31	00:23.78	06:17.31	00:29.92
Bloqueo	00:36.63	00:36.63	02:29.61	00:36.80	04:27.78	00:36.47	06:53.85	00:36.53
Coger	00:54.51	00:17.88	02:49.02	00:19.41	04:48.03	00:20.25	07:14.88	00:21.04
Dejar	01:16.13	00:21.62	03:10.99	00:21.97	05:19.52	00:31.49	07:36.16	00:21.28
Vuelta	01:32.32	00:16.19	03:27.53	00:16.55	05:47.40	00:27.87	07:52.62	00:16.46
Desviación típica								
	AZUL1		ROJO2		AZUL2		ROJO1	
	ABS	PAR	ABS	PAR	ABS	PAR	ABS	PAR
Desvío	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Bloqueo	1.641E-06	1.641E-06	3.961E-05	6.187E-06	6.699E-05	1.493E-06	0.0002179	1.5701E-06
Coger	2.098E-06	3.711E-06	4.569E-05	6.739E-06	6.474E-05	5.788E-06	0.0002088	9.8422E-06
Dejar	4.762E-06	3.173E-06	5.482E-05	1.208E-05	6.556E-05	2.16E-06	0.0001995	9.9522E-06
Vuelta	2.145E-06	6.217E-06	5.392E-05	4.568E-06	6.519E-05	2.85E-06	0.0002001	1.8581E-06
Mínimo								
	AZUL1		ROJO2		AZUL2		ROJO1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:49.24	00:17.01	03:45.80	00:19.54	05:58.65	00:17.01
Bloqueo	00:36.50	00:36.50	02:25.66	00:36.42	04:22.30	00:36.34	06:35.08	00:36.43
Coger	00:54.28	00:17.45	02:44.40	00:18.74	04:42.25	00:19.77	06:56.94	00:19.97
Dejar	01:15.63	00:21.35	03:05.48	00:21.07	05:13.86	00:31.31	07:18.69	00:19.99
Vuelta	01:32.11	00:15.57	03:21.79	00:16.12	05:41.64	00:27.67	07:34.95	00:16.26
Máximo								
	AZUL1		ROJO2		AZUL2		ROJO1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:56.41	00:23.96	03:59.52	00:26.78	06:43.50	00:48.38
Bloqueo	00:36.83	00:36.83	02:34.00	00:37.59	04:35.86	00:36.63	07:20.00	00:36.73
Coger	00:54.72	00:18.22	02:53.98	00:19.98	04:55.63	00:20.88	07:39.97	00:21.86
Dejar	01:16.54	00:21.89	03:16.62	00:23.07	05:27.32	00:31.69	07:59.96	00:21.75
Vuelta	01:32.50	00:16.87	03:32.74	00:16.94	05:55.12	00:28.23	08:16.40	00:16.64

Tabla A.4. 6: Estadísticos de los tiempos de la figura 1 con la forma 2 desglosados por pieza

Figura2 Forma2								
Ensayo1								
	ROJO1		ROJO2		AZUL2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:48.49	00:17.01	03:42.19	00:23.01	05:30.54	00:00.95
Bloqueo	00:36.68	00:36.68	02:25.12	00:36.63	04:18.86	00:36.67	06:07.26	00:36.72
Coger	00:54.00	00:17.32	02:40.88	00:15.75	04:35.82	00:16.96	06:23.69	00:16.42
Dejar	01:15.18	00:21.17	03:02.91	00:22.03	05:13.53	00:37.71	06:56.31	00:32.62
Vuelta	01:31.49	00:16.31	03:19.18	00:16.27	05:29.59	00:16.06	07:24.94	00:28.64
Ensayo2								
	ROJO1		ROJO2		AZUL2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	02:01.40	00:29.77	04:10.42	00:38.89	06:12.64	00:16.68
Bloqueo	00:36.43	00:36.43	02:38.13	00:36.73	04:46.68	00:36.26	06:48.79	00:36.15
Coger	00:54.68	00:18.25	02:53.34	00:15.21	05:02.53	00:15.85	07:05.20	00:16.41
Dejar	01:15.32	00:20.64	03:15.50	00:22.16	05:39.63	00:37.10	07:37.12	00:31.92
Vuelta	01:31.63	00:16.31	03:31.53	00:16.03	05:55.96	00:16.33	08:05.67	00:28.55
Ensayo3								
	ROJO1		ROJO2		AZUL2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:58.40	00:26.48	03:42.00	00:11.77	05:46.12	00:17.19
Bloqueo	00:36.73	00:36.73	02:34.98	00:36.58	04:18.53	00:36.53	06:23.10	00:36.98
Coger	00:54.52	00:17.79	02:51.50	00:16.52	04:34.98	00:16.45	06:38.51	00:15.41
Dejar	01:16.10	00:21.58	03:13.52	00:22.02	05:12.53	00:37.55	07:10.62	00:32.11
Vuelta	01:31.92	00:15.82	03:30.23	00:16.71	05:28.93	00:16.40	07:38.52	00:27.90
Ensayo3								
	ROJO1		ROJO2		AZUL2		AZUL1	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	02:08.52	00:36.86	04:15.52	00:33.99	06:12.52	00:09.93
Bloqueo	00:36.62	00:36.62	02:44.96	00:36.44	04:51.63	00:36.11	06:48.62	00:36.10
Coger	00:53.84	00:17.22	03:01.50	00:16.54	05:07.94	00:16.31	07:04.96	00:16.34
Dejar	01:14.84	00:21.00	03:24.62	00:23.12	05:45.73	00:37.79	07:37.10	00:32.14
Vuelta	01:31.66	00:16.82	03:41.53	00:16.91	06:02.59	00:16.86	08:05.63	00:28.53

Tabla A.4. 7: Tiempos de ensayos de la figura 2 con la forma 2

Figura2 Forma2								
MEDIAS POR PIEZA								
	Pieza1		Pieza2		Pieza3		Pieza4	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:59.20	00:27.53	03:57.53	00:26.92	05:55.46	00:11.19
Bloqueo	00:36.62	00:36.62	02:35.80	00:36.60	04:33.93	00:36.39	06:31.94	00:36.49
Coger	00:54.26	00:17.64	02:51.80	00:16.01	04:50.32	00:16.39	06:48.09	00:16.15
Dejar	01:15.36	00:21.10	03:14.14	00:22.33	05:27.86	00:37.54	07:20.29	00:32.20
Vuelta	01:31.67	00:16.32	03:30.62	00:16.48	05:44.27	00:16.41	07:48.69	00:28.41
Desviación típica								
	Pieza1		Pieza2		Pieza3		Pieza4	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	0	0	9.614E-05	9.542E-05	0.0002077	0.0001399	0.0002404	8.779E-05
Bloqueo	1.514E-06	1.514E-06	9.544E-05	1.397E-06	0.0002049	2.938E-06	0.0002362	4.97E-06
Coger	4.668E-06	5.481E-06	9.817E-05	7.459E-06	0.000201	5.289E-06	0.0002376	5.689E-06
Dejar	6.175E-06	4.52E-06	0.0001031	6.114E-06	0.0002003	3.568E-06	0.0002348	3.46E-06
Vuelta	2.088E-06	4.725E-06	0.0001058	4.647E-06	0.000203	3.864E-06	0.0002356	3.931E-06
Mínimo								
	Pieza1		Pieza2		Pieza3		Pieza4	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	01:48.49	00:17.01	03:42.00	00:11.77	05:30.54	00:00.95
Bloqueo	00:36.43	00:36.43	02:25.12	00:36.44	04:18.53	00:36.11	06:07.26	00:36.10
Coger	00:53.84	00:17.22	02:40.88	00:15.21	04:34.98	00:15.85	06:23.69	00:15.41
Dejar	01:14.84	00:20.64	03:02.91	00:22.02	05:12.53	00:37.10	06:56.31	00:31.92
Vuelta	01:31.49	00:15.82	03:19.18	00:16.03	05:28.93	00:16.06	07:24.94	00:27.90
Máximo								
	Pieza1		Pieza2		Pieza3		Pieza4	
	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial	Absoluto	Parcial
Desvío	00:00.00	00:00.00	02:08.52	00:36.86	04:15.52	00:38.89	06:12.64	00:17.19
Bloqueo	00:36.73	00:36.73	02:44.96	00:36.73	04:51.63	00:36.67	06:48.79	00:36.98
Coger	00:54.68	00:18.25	03:01.50	00:16.54	05:07.94	00:16.96	07:05.20	00:16.42
Dejar	01:16.10	00:21.58	03:24.62	00:23.12	05:45.73	00:37.79	07:37.12	00:32.62
Vuelta	01:31.92	00:16.82	03:41.53	00:16.91	06:02.59	00:16.86	08:05.67	00:28.64

Tabla A.4. 8: Estadísticos de los tiempos de la figura 2 con la forma 2 desglosados por pieza