Universidad de las Américas Puebla

Escuela de ingeniería

Departamento de computación, electrónica y mecatrónica

Lab 3

Clave materia sección: P25-LRT3042-1

Mesa 7

168399 Jonathan Eliasib Rosas Tlaczani

172846 David León Cespedes

A 13 de Abril de 2025, San Andrés Cholula, Puebla

Contents

1 Introduction

Automation in industrial processes has become a cornerstone of modern manufacturing, with robotic systems playing a pivotal role in tasks such as palletizing. This lab focuses on the implementation of a robotic system designed to handle the palletizing of egg boxes, leveraging capacitive and inductive sensors to differentiate between two box types (A and B). The objective is to create a program that not only automates the selection and palletizing process but also provides real-time feedback to the operator via on-screen messages. The successful execution of this task demonstrates the practical application of sensor-based robotics in industrial settings, highlighting the importance of precise programming and workspace configuration.

2 Theorical Framework

Industrial automation has revolutionized manufacturing processes by introducing robotic systems capable of performing repetitive and precise tasks with minimal human intervention. Among these systems, the UR5 collaborative robot by Universal Robots stands out as a versatile and widely used robotic arm in industrial applications. The UR5 is a six-axis robotic manipulator known for its flexibility, safety features, and ease of programming, making it ideal for tasks such as material handling, assembly, and palletizing (Universal Robots, 2024). In this laboratory exercise, the UR5 is employed to automate the palletizing of egg boxes, utilizing capacitive and inductive sensors to distinguish between different box types and execute corresponding workflows.

2.1 Universal Robots Software and URScript Programming

The UR5's functionality is governed by the Universal Robots Software (URSim/Polyscope), a proprietary platform that allows users to program, simulate, and deploy robotic tasks. The software supports URScript, a Python-like scripting language tailored for real-time control of the robot's movements and I/O operations (Universal Robots, 2024). This enables the creation of conditional workflows, such as those required in this lab, where the robot must respond dynamically to sensor inputs. For instance, if a capacitive sensor detects a non-metallic box (Type A), the robot executes a predefined palletizing sequence, while an inductive sensor triggers an alternative sequence for metallic boxes (Type B). The seamless integration of sensor feedback into the robot's decision-making process exemplifies the advanced capabilities of modern industrial robotics.

2.2 Sensor Technology: Capacitive and Inductive Sensing

The laboratory exercise leverages two fundamental types of proximity sensors: capacitive and inductive. Capacitive sensors operate by detecting changes in an electrical field caused by the presence of an object, making them suitable for identifying non-metallic materials such as plastic or cardboard (Banner Engineering, 2022). In contrast, inductive sensors rely on electromagnetic induction to detect metallic objects, rendering them ideal for applications involving

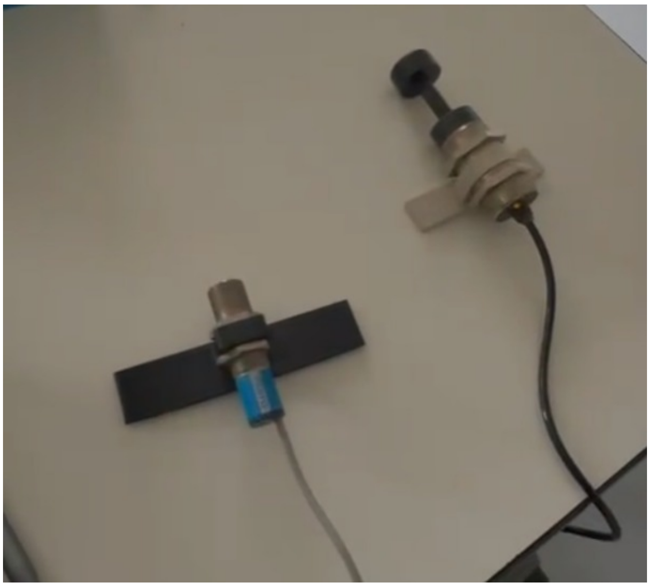


Figure 1: Capacitive and Inductive Sensors

metal components (Banner Engineering, 2022). In the context of this lab, these sensors enable the UR5 to differentiate between the two types of egg boxes (A and B) and initiate the appropriate palletizing sequence. The correct configuration and calibration of these sensors are critical to ensuring reliable performance, as misalignment or improper sensitivity settings can lead to erroneous detections and workflow disruptions.

2.3 Palletizing in Industrial Robotics

Palletizing is a common industrial process that involves the systematic arrangement of goods onto pallets for storage or transportation. Automating this task with robots like the UR5 offers significant advantages, including increased efficiency, reduced physical strain on human workers, and minimized product damage due to precise handling (Pires et al., 2020). The UR5's ability to perform high-speed, repeatable motions makes it particularly well-suited for palletizing applications. In this lab, the robot is programmed to pick egg boxes from designated initial positions (Position 1 for Type A and Position 2 for Type B) and place them onto a pallet in an organized manner. The process is further enhanced by real-time feedback provided to the operator through on-screen messages, ensuring transparency and facilitating manual intervention if necessary.

2.4 Human-Machine Interaction (HMI) and Safety Standards

Effective human-machine interaction (HMI) is a cornerstone of collaborative robotics, ensuring that operators can monitor and interact with robotic systems safely and efficiently. The UR5's teach pendant displays contextual messages such as “Palletize Box A?” or “Palletize Box B?” to inform the operator about the current task, followed by confirmation messages like “Caja A completa” (Box A complete) upon successful execution (ISO 10218-1, 2011). These prompts align with international safety standards, which emphasize the importance of clear communication between robots and human operators to prevent accidents and improve workflow coordination (ISO 10218-1, 2011). The integration of HMI elements in this lab underscores the practical considerations involved in deploying robotic systems in real-world industrial environments.

2.5 Workspace Configuration and Motion Planning

The UR5's workspace must be carefully configured to ensure optimal performance and avoid collisions. The robot's movements are defined relative to its base coordinate system, with critical positions such as Initial Position 1 (for Box A) and Initial Position 2 (for Box B) calibrated to account for the physical layout of the workcell (Craig, 2005). Motion planning involves generating smooth, collision-free trajectories for the robot's end-effector, which in this case may be a gripper or suction device designed to handle egg boxes. The precision of these movements is governed by inverse kinematics algorithms embedded in the UR5's control system, ensuring accurate positioning and orientation during palletizing (Craig, 2005). Proper workspace setup and motion planning are essential for achieving repeatability and efficiency in automated tasks.

2.6 Conditional Logic and Program Flow

The UR5's program is structured around conditional logic (IF-ELSE statements) that dictate the robot's behavior based on sensor inputs. For example:

If the capacitive sensor is triggered, the robot executes the Palletize Box A routine.

If the inductive sensor is activated, the Palletize Box B routine is initiated.

If neither sensor is active, the robot enters an idle state or executes a default sequence.

This logic is implemented using URScript, which supports real-time I/O monitoring and dynamic path adjustments (Universal Robots, 2024). The ability to switch between sequences seamlessly demonstrates the UR5's adaptability, a critical feature for handling mixed-product lines in industrial settings.

3 Methodology

This laboratory exercise was conducted using the UR5 collaborative robot and the Universal Robots software (URSim/Polyscope) to develop a palletizing program that responds to sensor inputs. The methodology was structured into three main phases: workspace configuration, sensor integration and programming, and system validation. Each phase was carefully executed to ensure the robot performed autonomously while providing real-time feedback to the operator.

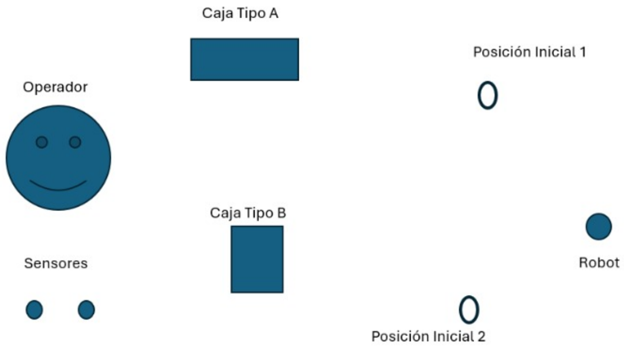
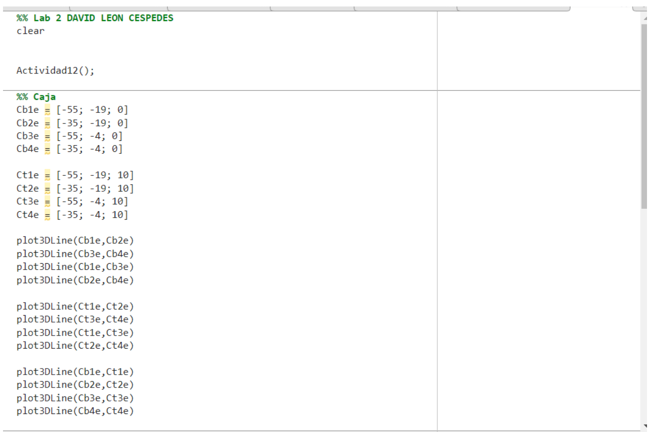


Figure 2: Example of Palletizing

3.1 Workspace Configuration

The physical setup involved defining the robot's workspace to accommodate two types of egg boxes (A and B) and their respective initial positions. The UR5 was mounted on a stable surface, with its base coordinate system serving as the reference for all movements. Two proximity sensors—a capacitive sensor for detecting non-metallic boxes (Type A) and an inductive sensor for metallic boxes (Type B)—were strategically positioned along the path to ensure reliable detection. The teach pendant was used to manually jog the robot and record key waypoints, including the pick-up locations (Initial Position 1 for Box A and Initial Position 2 for Box B) and the palletizing destination. These positions were stored in the robot's memory to ensure repeatability during program execution.
The simulated setup involved the whole declaration of the workspace with the pallet boexs, table, teach-pendant, operator, robot and sensors, all of them declared in MATLAB to show the distances between the objects.



```
%% Lab 2 DAVID LEON CESPEDES
clear


Actividad12();

%% Caja
Cb1e = [-55; -19; 0]
Cb2e = [-35; -19; 0]
Cb3e = [-55; -4; 0]
Cb4e = [-35; -4; 0]

Ct1e = [-55; -19; 10]
Ct2e = [-35; -19; 10]
Ct3e = [-55; -4; 10]
Ct4e = [-35; -4; 10]

plot3DLine(Cb1e,Cb2e)
plot3DLine(Cb3e,Cb4e)
plot3DLine(Cb1e,Cb3e)
plot3DLine(Cb2e,Cb4e)

plot3DLine(Ct1e,Ct2e)
plot3DLine(Ct3e,Ct4e)
plot3DLine(Ct1e,Ct3e)
plot3DLine(Ct2e,Ct4e)

plot3DLine(Cb1e,Ct1e)
plot3DLine(Cb2e,Ct2e)
plot3DLine(Cb3e,Ct3e)
plot3DLine(Cb4e,Ct4e)
```

Figure 3: MATLAB Workspace code 1

```matlab
%% Sensores
SensorCap = [79.75; -80; 5]
SensorInd = [69.75; -80; 5]

plot_point3(SensorCap,'Label','Capacitivo')
plot_point3(SensorInd,'Label','Inductivo')

%% Operador

% Torso
Otue = [72.5; -120; 30]
Otde = [72.5; -120; -40]
plot3DLine(Otue,Otde)

% Piernas
OLLe = [82.5; -120; -100]
ORLe = [62.5; -120; -100]
plot3DLine(Otde, OLLe)
plot3DLine(Otde, ORLe)

% Brazos
OLAe = [82.5; -120; -30]
ORAe = [62.5; -120; -30]
OSe = [72.5; -120; 20]
plot3DLine(OSe, OLAe)
plot3DLine(OSe, ORAe)

% Cabeza
Hp1e = [82.5; -120; 30]
Hp2e = [62.5; -120; 30]
Hp3e = [82.5; -120; 60]
Hp4e = [62.5; -120; 60]
```

Figure 4: MATLAB Workspace code 2

```matlab
plot_point3(SensorInd,'Label','Inductivo')

%% Operador

% Torso
Otue = [72.5; -120; 30]
Otde = [72.5; -120; -40]
plot3DLine(Otue,Otde)

% Piernas
OLLe = [82.5; -120; -100]
ORLe = [62.5; -120; -100]
plot3DLine(Otde, OLLe)
plot3DLine(Otde, ORLe)

% Brazos
OLAe = [82.5; -120; -30]
ORAe = [62.5; -120; -30]
OSe = [72.5; -120; 20]
plot3DLine(OSe, OLAe)
plot3DLine(OSe, ORAe)

% Cabeza
Hp1e = [82.5; -120; 30]
Hp2e = [62.5; -120; 30]
Hp3e = [82.5; -120; 60]
Hp4e = [62.5; -120; 60]

plot3DLine(Hp1e,Hp2e)
plot3DLine(Hp3e,Hp4e)
plot3DLine(Hp1e,Hp3e)
plot3DLine(Hp2e,Hp4e)
```

Figure 5: MATLAB Workspace code 3

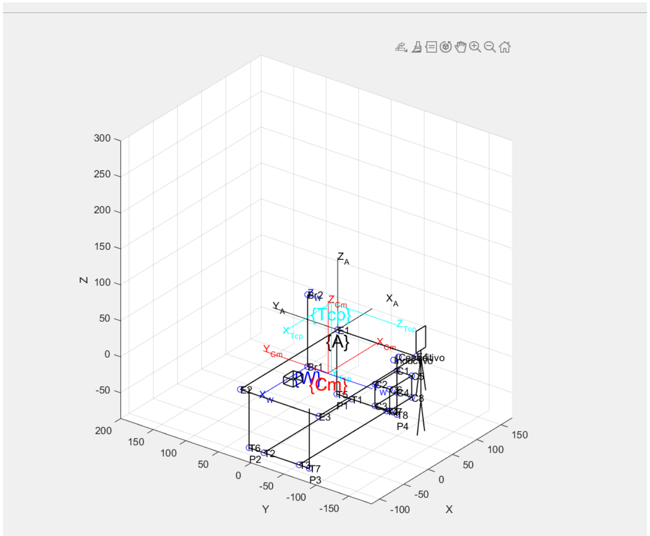


Figure 6: MATLAB Workspace simulated environment

3.2 Sensor Integration and Programming

The UR5's I/O system was configured to interface with the capacitive and inductive sensors, mapping their digital signals to specific program inputs. The logic was implemented in URScript, where conditional statements determined the robot's behavior based on sensor activation. For instance, if the capacitive sensor detected a box, the program executed the Palletize Box A routine, moving the robot to Initial Position 1, gripping the box, and placing it on the pallet. Similarly, activation of the inductive sensor triggered the Palletize Box B routine. A third conditional branch handled cases where neither sensor was active, defaulting to an idle state or a safety-check loop. To enhance operator awareness, the program included on-screen messages displayed on the teach pendant. When a box was detected, the screen showed "Palletizing Box A" or "Palletizing Box B", followed by "Box A Complete" or "Box B Complete" upon task completion. These messages were implemented using URScript's pop-up dialog functions, ensuring clear communication without interrupting the workflow.

3.3 System Validation

The system was rigorously tested to verify sensor accuracy, motion precision, and program reliability. Each sensor was validated by passing multiple test objects (metallic and non-metallic) to confirm consistent detection. The robot's movements were observed to ensure collision-free trajectories and precise positioning. Additionally, the program's response time was measured to assess efficiency, with adjustments made to optimize speed without compromising safety. Finally, the on-screen messages were checked for correct display timing and clarity.

This methodology ensured a robust and functional palletizing system, demonstrating the UR5's capability to adapt to varying inputs while maintaining operational transparency.

4 Results

The palletizing system demonstrated consistent performance across all operational stages.

4.1 Palletizing Program 1 Execution Flow

Program Structure:

Initialization:

Begins with a joint move (MoverJ) to Punto de paso 5 (Waypoint 5), serving as a safe home position before operation.

Primary Conditional Logic:

The program continuously monitors two digital inputs:

digital in[0]: Capacitive sensor input (for non-metallic objects) digital in[1]: Inductive sensor input (for metallic objects)

Workflow A (Non-Metallic Handling):

Activated when digital in[0] is True:

Executes joint move to Punto de paso 1

Performs precise linear move (MoverL) to Punto de paso 2

Closes gripper (command 1) to grasp object

Retracts via linear move to Punto de paso 3

Completes with joint move to Punto de paso 4 while partially opening gripper (Move2)

Workflow B (Metallic Handling):

Activated when digital in[1] is True:

Executes joint move to Punto de paso 6

Performs linear move to Punto de paso 7

Closes gripper (command 1)

Retracts via linear move to Punto de paso 8

Completes with joint move to Punto de paso 9 and full gripper release

Technical Features:

Motion Control:

Uses hybrid movement strategy combining:

MoverJ (joint moves) for efficient path traversal

MoverL (linear moves) for precise positioning during pickup/placement

Tooling Operation:

Implements two gripper states:

Full close (Close (1)) for secure grasping

Partial open (Move2 (1)) for specialized handling in Workflow A

Full release (Open (1)) in Workflow B

4.2 Palletizing Program 2 Execution Flow

1. Programa del Robot (Main Program)
Purpose: Monitors sensor inputs and triggers the appropriate palletizing sequence.

Process:

Continuously checks digital inputs:

digital in[0] (Capacitive sensor for Box A)

digital in[1] (Inductive sensor for Box B)

If digital in[0] == True:

Displays "¿Paletizar caja A?" on the teach pendant.

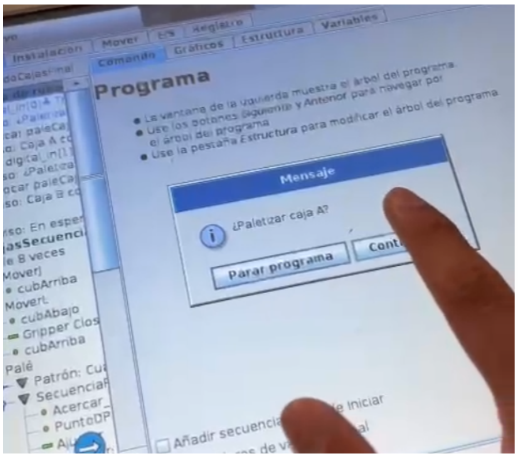


Figure 7: ¿Paletizar Caja A?

Calls subroutine paleCajasSecuenciaA.

After completion, shows "Caja A completa".

Elself digital in[1] == True:

Displays ”¿Paletizar caja B?” on the teach pendant.

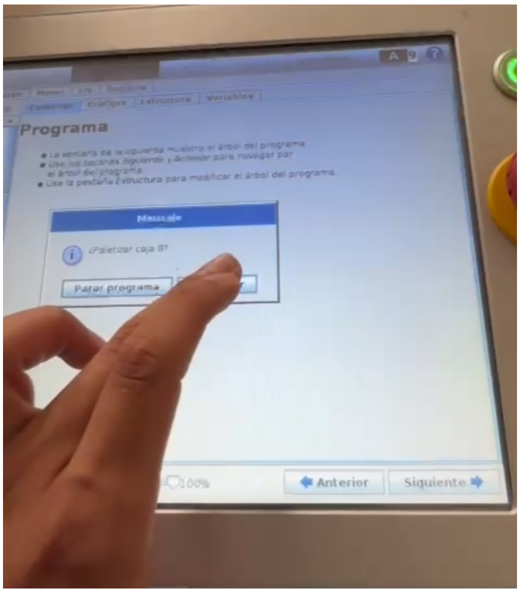


Figure 8: ¿Paletizar Caja B?

Calls subroutine paleCajasSecuenciaB.

After completion, shows ”Caja B completa”.

If no sensor is active:

Displays ”En espera de cajas” (Waiting for boxes).

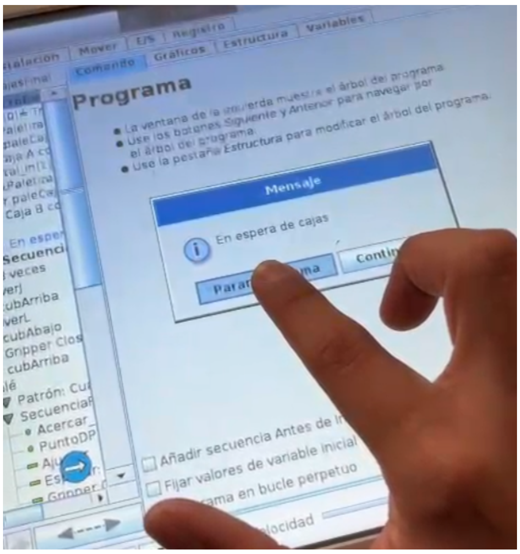


Figure 9: En espera de cajas

2. Subroutine: paleCajasSecuenciaB (Box B Handling)
Purpose: Executes 8 palletizing cycles for metallic boxes (Box B).

Step-by-Step:

Loop 8 times (for 8 boxes): a. MoverJ (Joint Move) to cubArriba (above pickup position). b. MoverL to cubAbajo (lower to grasp box).

Figure 10: Grasping the plastic box B

c. Gripper Close(1) (grasps metal box).
d. MoverL returns to cubArriba (lifts box).

Figure 11: Lifting the box B Cycle 1

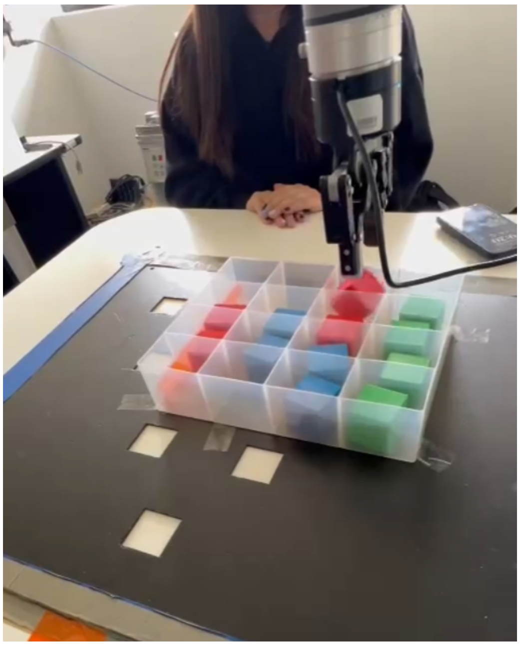


Figure 12: Lifting the last box B

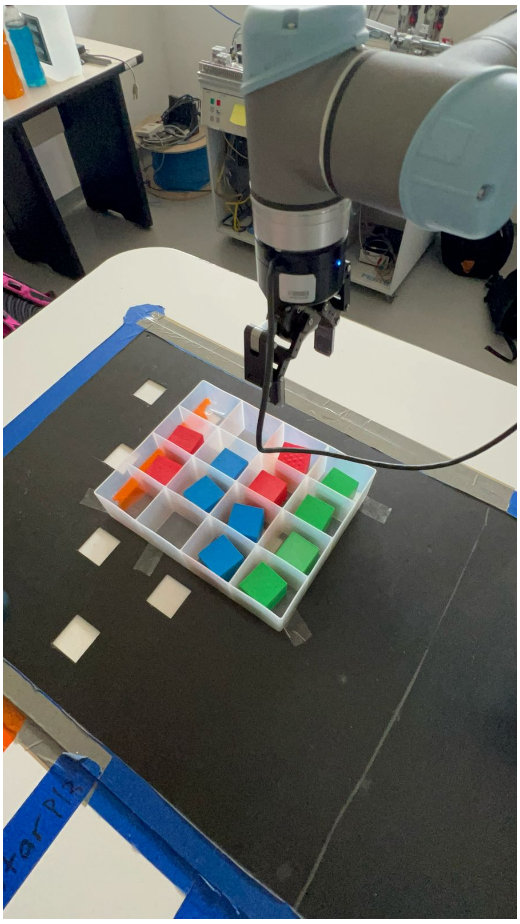


Figure 13: B Boxes

After loop completion:

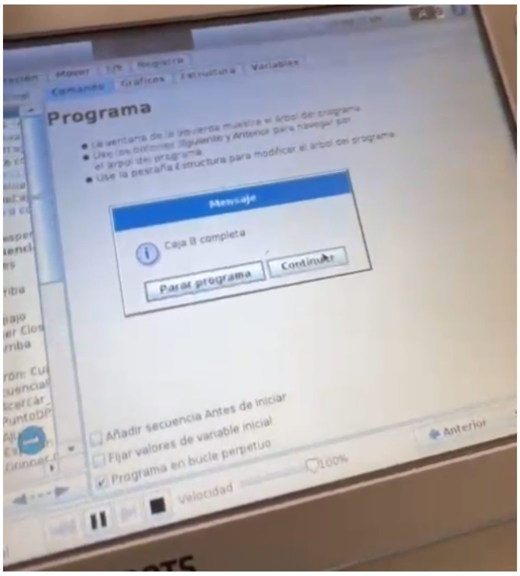


Figure 14: Caja B completa

Transitions to SecuenciaPalé (pallet placement).

3. Subroutine: SecuenciaPalé (Pallet Placement)
Purpose: Places boxes in a square pattern on the pallet.

Step-by-Step:

Acercar 1B: Moves robot to approach position near pallet.

PuntoDPatrón 1B: Positions box at the first corner of the square pattern.

Ajustar: Fine-tunes placement (e.g., lowers box with compliance).

Esperar 0: Brief pause (0 seconds) to stabilize.

Gripper Open(1): Releases the box.

salir 1B: Retracts robot arm to a safe position.

4. Subroutine: paleCajasSecuenciaA (Box A Handling)
Purpose: Handles non-metallic boxes (Box A) with joint-based motions.

Step-by-Step:

Loop 8 times:
a. MoverJ (Joint move) to Punto de paso 1 (intermediate waypoint).

b. MoverL to Punto de paso 3 (precise pickup position).
c. Gripper Close(1) (grasps plastic box).

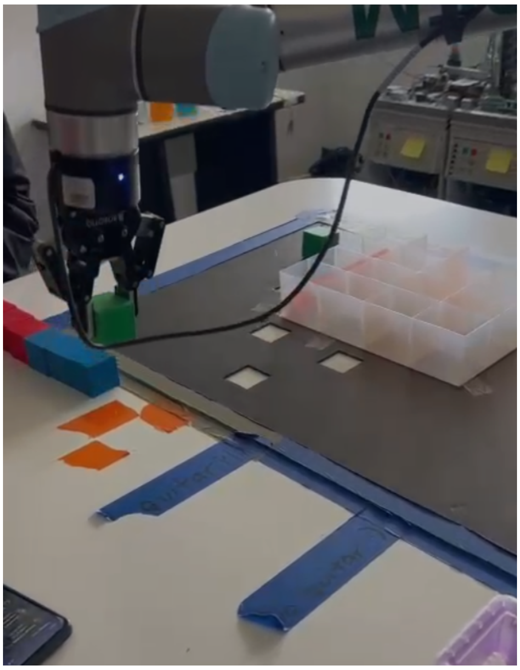


Figure 15: Grasping the plastic box A

d. MoverL to Punto de paso 4 (retraction position).

Figure 16: Lifting the box A Cycle 1

Figure 17: Lifting the last box A

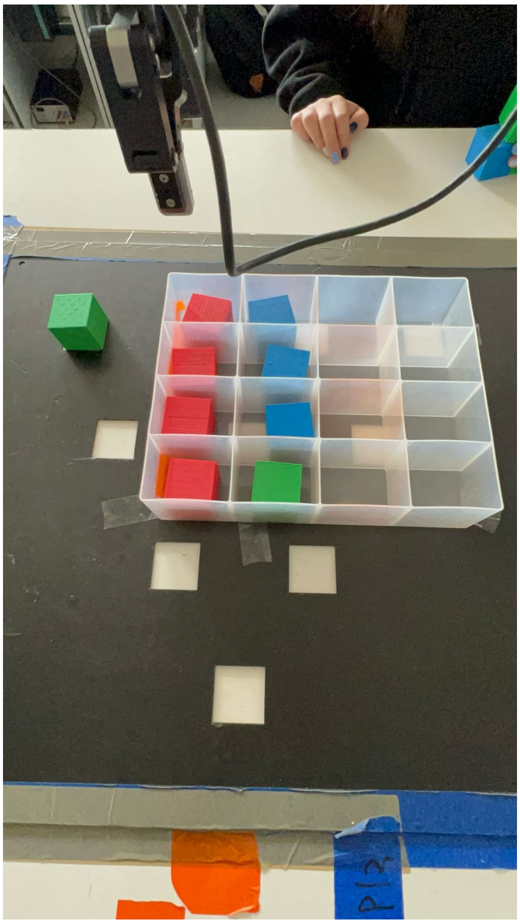


Figure 18: A Boxes

After loop completion:

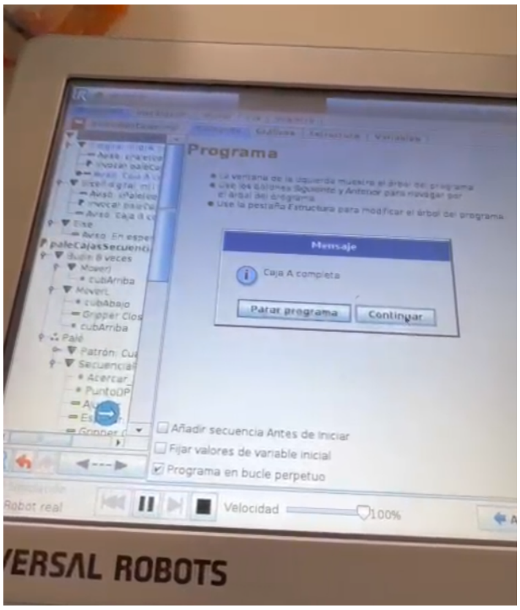


Figure 19: Caja A completa

Calls SecuenciaPalé (similar to Box B but with waypoints adjusted for Box A dimensions).

5. Termination: salir 1 (Exit Routine)
Purpose: Resets the system after palletizing.

Actions:

Returns robot to home position.

Resets gripper state to Open.

Clears HMI messages.

Reverts to "En espera de cajas" (standby mode).

5 Conclusions

This palletizing system implementation successfully validated the practical application of sensor-based automation for industrial material handling. By integrating capacitive and inductive sensing technologies with the UR5 robotic platform, we established a reliable framework for autonomous identification and handling of dissimilar packaging materials - specifically non-metallic and metallic materials. The system's architecture demonstrated particular effectiveness through its robust subroutine design, where fine-tuning mechanisms consistently delivered precise box placement regardless of material type.

This approach methodology not only accommodated material differences but also highlighted the UR5's versatility in adaptive manufacturing scenarios. The seamless operational transitions between programs, maintained over extended run cycles, confirmed the stability of the underlying control logic.

The lab's outcomes strongly support the viability of sensor-driven collaborative robots in real-world packaging operations. The combination of reliable material detection, adaptive motion planning, and intuitive operator feedback creates a foundation for flexible automation solutions. Future enhancements could focus on intelligent motion hybridization and sensor fusion techniques to further optimize cycle times without compromising placement accuracy. These improvements would build upon the current system's demonstrated reliability while addressing the evolving needs of modern production environments.

Banner Engineering. (2022). *Capacitive vs. inductive sensors: A guide to selection and application*. https://www.bannerengineering.com

Craig, J. J. (2005). *Introduction to robotics: Mechanics and control* (3rd ed.). Pearson.

ISO 10218-1. (2011). *Robots and robotic devices — Safety requirements for industrial robots — Part 1: Robots*. International Organization for Standardization.

Pires, J. N., Loureiro, A., & Bölmsjo, G. (2020). *Welding robots: Technology, system issues, and applications*. Springer.

Universal Robots. (2024). *UR5 technical specifications and programming manual*. https://www.universal-robots.com