Universidad de las Américas Puebla

Enginnering

Department of Computing, Electronics, and Mechatronics

Packing line

Cinemática y Dinámica de Robots

164645 Aldair Arios Rojas Velazquez
164467 Jorge Ignacio Serrano Baez
166876 Valeria Adriana De Santos Frutos
168399 Jonathan Eliasib Rosas Tlaczani
175629 José Fabián Molina Enríquez

Mayo 8, 2024. San Andrés Cholula, Puebla

1 Abstract

The design and development of an automated packing line using industrial robots is the main goal of this project. The kinematics and dynamics of the robots on the production line are thoroughly examined using MATLAB and RoboDK.

In automated manufacturing environments, safe and effective robot operation depends heavily on their dynamics and kinematics. This study describes the procedures for modeling and simulating dynamics, which takes into account the forces and moments impacting the robots' motion, as well as kinematics, which covers the geometry and motion of the robots.

The positions and orientations of the robot's actuators are determined based on the spatial coordinates of the end effector through in-depth analyses of forward and inverse kinematics carried out with MATLAB. Furthermore, optimal real-time solutions are obtained by solving singularity problems using sophisticated methodologies.

RoboDK, on the other hand, is used to simulate and validate the behavior of the robots in a virtual production environment. Algorithms for trajectory planning are used to maximize efficiency and reduce cycle times while optimizing robot mobility.

The collected results are reported in detail in the paper, along with comparative assessments of various robot configurations and an assessment of the packing system's accuracy and speed. These findings' practical ramifications are examined, and possible enhancements for further applications are recomended.

2 Introduction

In the field of industrial automation, the incorporation of robotic technologies has marked a milestone in the optimization of production processes. The ability of robotic arms to perform repetitive tasks with millimeter precision and in reduced time has driven their adoption in a wide range of industries, from manufacturing to logistics.

This project focuses on a specific aspect of industrial automation: palletizing. Palletizing is a critical process in the supply chain, where goods are arranged orderly on pallets for efficient storage and transportation. In this context, efficiency and precision are essential to ensure cargo integrity and optimize workflows.

The main objective of this project is to analyze in detail a robotic arm designated for a palletizing line, with the specific task of completing a final pallet containing a total of 20 boxes. This task may seem simple at first glance, but it involves a number of technical and logistical challenges that must be meticulously addressed.

To fully understand the operation of the robotic arm in this context, a comprehensive analysis covering multiple aspects is necessary. One of the key components of this analysis is the study of Denavit-Hartenberg parameters, which describe the geometry and kinematics of the robotic arm in relation to its joints and links.

In addition, an analysis of the direct and inverse kinematics of the robot will be carried out. Forward kinematics determine the position and orientation of the end of the robot (the tool or end effector) based on joint configurations, while inverse kinematics calculates the joint configurations necessary to achieve a desired end position and orientation. of the robot.

To carry out this analysis, advanced modeling and simulation tools will be used, including software such as MATLAB and RoboDK. These tools allow you to visualize the behavior of

the robot in a virtual environment and apply analytical processes learned during the course.

3 Problem statement

In a highly competitive industrial environment, efficiency and precision in production processes are critical aspects that determine the success of a company. In particular, product palletizing is a fundamental task in the supply chain, where the proper arrangement of cargo on pallets can significantly impact logistics efficiency and operational performance.

Manual palletizing, although common in many companies, has limitations in terms of speed, precision and safety. Additionally, as demand and product complexity increases, it becomes increasingly difficult for human operators to maintain the required pace and quality.

To address these challenges, many companies are turning to automation by using robotic arms on their palletizing lines. However, successful implementation of this technology requires a deep understanding of the process requirements, as well as the capabilities and limitations of available robotic systems.

In this context, the need arises to analyze and optimize the performance of a robotic arm intended for a specific palletizing line. The objective is to design an automated system that can complete palletizing efficiently, accurately and safely, meeting established production requirements.

4 Justification and Objectives

Why is it done?

The implementation of an automated palletizing system using a robotic arm has several fundamental reasons:

- Efficiency: Automating the palletizing process allows the speed and precision of the process to be increased, thereby reducing cycle times and improving the overall productivity of the production line.
- Precision: Robotic arms can place boxes or products on the final pallet with millimeter precision, ensuring orderly and safe disposition of the load.
- Cost Reduction: By decreasing reliance on human labor, labor costs associated with manual palletizing can be reduced while minimizing errors and waste.
- Safety: Automating palletizing reduces workers' exposure to hazardous conditions, such as lifting heavy loads or working in potentially hazardous environments.

What is the purpose?

The main purpose of implementing an automated palletizing system is to improve the efficiency and accuracy of the production process. By using a robotic arm to palletize products, the aim is to optimize cycle times, guarantee the quality of the final pallet and reduce the risks associated with manual handling of loads.

Additionally, the automated system can easily adapt to changes in demand or the types of products to be palletized, providing flexibility and responsiveness to market needs.

What problems does it solve?

Implementing an automated palletizing system solves several problems that may arise in the production process, including:

- Inefficiency in the manual palletizing process, which can result in long cycle times and low levels of productivity.
- Human errors, such as incorrectly placing boxes on the final pallet, can lead to damaged or wasted products.
- Risks to worker health and safety associated with manual handling of heavy loads or exposure to hazardous environments.
- Limitations in the ability to adapt to changes in demand or product requirements, which can hinder the efficiency and flexibility of the production line.

5 Theorical framework

In industrial automation using the UR5 robotic arm from Universal Robots. This 6-axis robot is versatile and collaborative, designed for a wide range of industrial tasks, including automated palletizing.

The UR5 is characterized by its precision and ease of programming, making it suitable for applications such as palletizing. Its kinematics are modeled using Denavit-Hartenberg parameters, allowing its movements to be accurately calculated.

Simulation of the UR5 is carried out using specialized software such as RoboDK, allowing the robot's behavior to be modeled and optimized in a virtual environment before its implementation in real production.

6 UR5 Robot

The UR5 robotic arm, developed by Universal Robots, represents an outstanding innovation in the field of industrial automation. This 6-axis collaborative robot has gained popularity due to its versatility, precision, and ease of integration into a variety of work environments.

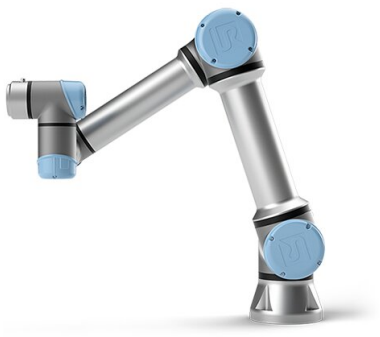

Figure 1: UR5 Robot

The UR5 is designed to perform a wide range of tasks in industrial and collaborative environments, from assembly and material handling to palletizing and packaging. With a payload of up to 5kg and repeatability accuracy of ±0.1mm, the UR5 offers precise and reliable handling capability.

Its modular design and intuitive user interface allow for quick and easy setup, reducing downtime and making it easy to adapt to different applications. Additionally, the UR5 is equipped with a number of advanced safety features, such as integrated sensors and collision detection technology, which enable safe and efficient interaction with human operators.

In the context of automated palletizing, the UR5 offers an effective solution for the orderly and efficient arrangement of products on pallets. Its ability to pick up, transport and place objects with millimeter precision makes it an invaluable tool for optimizing production processes and improving productivity in the industry.

To model and simulate the behavior of the UR5, Denavit-Hartenberg parameters are used, a standard convention in robotics that describes the geometry and kinematics of the robot. This information allows the robot's movements to be accurately calculated and its performance optimized in different applications.

Simulation of the UR5 can be performed using specialized software, such as RoboDK, which provides advanced tools to model, simulate and optimize the robot's behavior in a virtual environment. This capability allows engineers and designers to test and refine their applications before deploying them to the real production environment, reducing development costs and time.

7 Denavit-Hartenberg method

The Denavit-Hartenberg (DH) method is a convention used to describe the geometry and kinematics of robotic systems, especially articulated robotic arms. This method provides a

systematic framework for modeling and calculating the movements of robots, allowing their actions to be precisely planned and controlled.

The DH method is based on the assignment of reference systems to each joint and link of the robot, which allows describing the position and orientation of each element in relation to the previous one. For each joint, four parameters are defined: , a, d and , which represent the angle of rotation around the anterior axis (alpha), the length of the link (a), the length of the displacement along the anterior z-axis (d) and the angle of rotation around the previous z axis (teta).

The DH convention establishes a series of rules to assign these parameters to each joint and link of the robot, which allows calculating the homogeneous transformation between adjacent reference systems and, therefore, determining the position and orientation of the robot end in space. of work.

The DH method is widely used in robotics due to its simplicity and effectiveness. It allows robots to be modeled systematically and accurately, facilitating design, path planning and motion control. Furthermore, the DH method can be implemented in a computationally efficient manner, making it suitable for use in real-time systems.

8 Forward kinematics

Forward kinematics is a fundamental concept in robotics that describes how to determine the position and orientation of the robot's end (also known as the end-effector) in the workspace based on the robot's joint configurations.

In a robotic system, joints are the connection points between the different links of the robot, and can be rotary or prismatic. Direct kinematics allows the position and orientation of the robot's end to be calculated in three-dimensional space in relation to a reference coordinate system, usually fixed to the base of the robot.

To calculate direct kinematics, homogeneous transformation matrices are used, which are 4x4 matrices that represent the position and orientation of one reference system in relation to another. These matrices are used to represent the geometric transformations between the different coordinate systems associated with the joints and links of the robot.

The forward kinematics process involves computing a chain of homogeneous transformation matrices that describe the position and orientation of the robot end based on the joint configurations. This information is used to determine the trajectory of the robot's end in the workspace, allowing its movements to be planned and controlled accurately and efficiently.

9 Inverse kinematics

Inverse kinematics is an essential concept in robotics that describes how to determine the joint configurations of a robot to achieve a specific position and orientation of the robot's end in the workspace.

In a robotic system, inverse kinematics involves the calculation of the joint configurations necessary to achieve a desired position and orientation of the robot end, from specified spatial coordinates. That is, while forward kinematics calculates the position of the end of the robot

given the state of the joints, inverse kinematics does the opposite: it determines the state of the joints required to reach a target position and orientation.

The calculation of inverse kinematics can be complex and is highly dependent on the geometry and configuration of the robot. Generally, mathematical techniques, such as trigonometry and analytical geometry, are used to solve the equations that describe the relationship between the robot end coordinates and the joint configurations.

It is important to note that inverse kinematics may lead to multiple solutions, implying that there may be more than one set of joint angles that lead to the same result for the position and orientation of the end effector. In some cases, additional constraints are applied to select the best solution, such as minimizing joint movement or avoiding singular configurations that may cause control issues.

Once the joint configurations are determined, commands can be sent to the robot controller to move the joints to the calculated positions. This allows the robot to reach and maintain the desired position and orientation in the workspace, which is essential for performing precise and controlled tasks.

10 Denavit-Hartenberg

We needed to follow the Denavit Hartenberg steps learned in the course for calculating the parameters of the table.

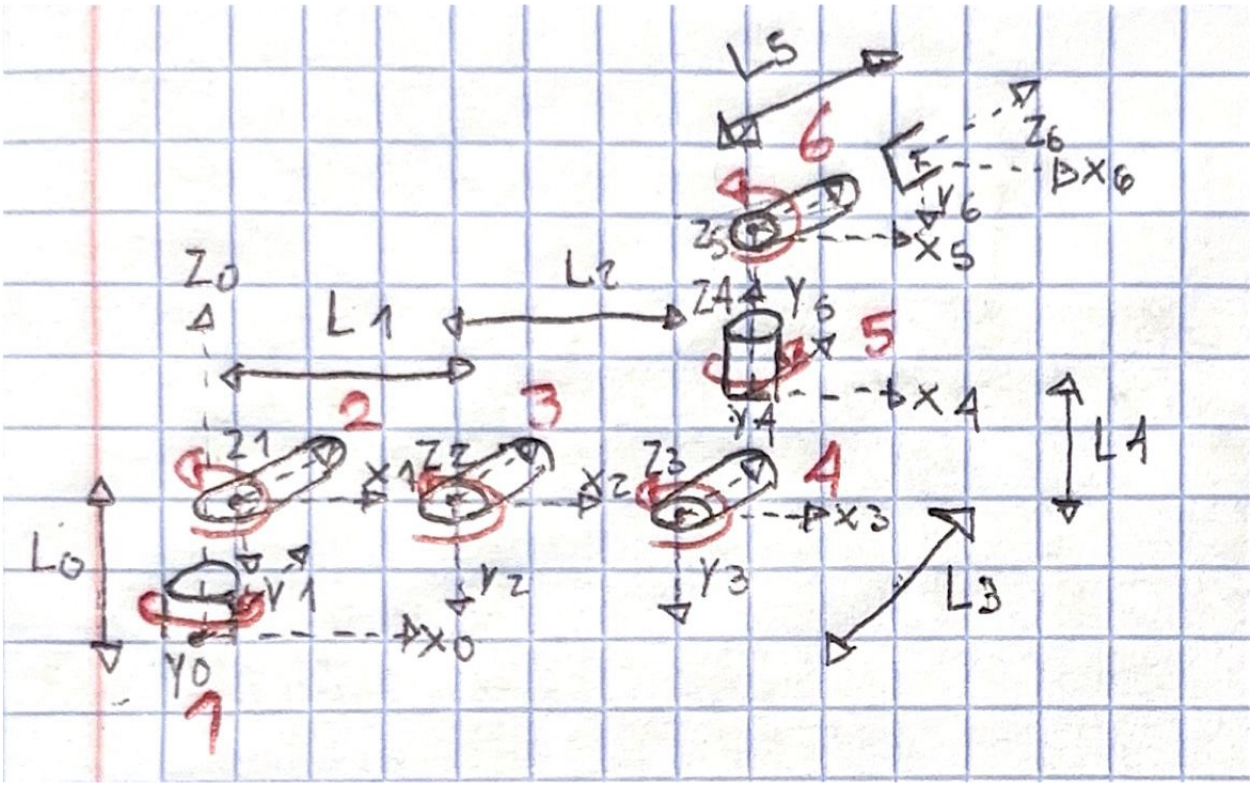

Figure 2: DH Steps

After doing the representation of the joints, links and axes as we can see in figure 2, we

filled the table as shown in figure 3.

Rs, Ys	Link	θ	d	a	α
S0-S1	1	q1	L0	0	90º
S1-S2	2	q2	0	L1	0
S2-S3	3	q3	0	L2	0
S3-S4	4	q4	L3	0	90º
S4-S5	5	q5	L4	0	-90º
S5-S6	6	q6	L5	0	0

Figure 3: Denavit-Hartenberg Table.

11 Forward kinematics

For the forward kinematics we use the DH table for calculating the homogeneous transformation matrix and also for making a 3D representation of the UR5 robot in matlab.

```
syms q1 q2 q3 q4 q5 q6
L0=5
L1=5
L2=5
L3=5
L4=5
L5=5
aTb=trotz(q1)*transl(0,0,L0)*transl(0,0,0)*trotx(pi/2)
bTc=trotz(q2)*transl(0,0,0)*transl(0,0,L1)*trotx(0)
cTd=trotz(q3)*transl(0,0,0)*transl(0,0,L2)*trotx(0)
dTe=trotz(q4)*transl(0,0,L3)*transl(0,0,0)*trotx(pi/2)
eTf=trotz(q5)*transl(0,0,L4)*transl(0,0,0)*trotx(-pi/2)
fTg=trotz(q6)*transl(0,0,L5)*transl(0,0,0)*trotx(0)
%%step 15
T=aTb*bTc*cTd*dTe*eTf*fTg
T=simplify(T)
q1=deg2rad(0)
q2=deg2rad(0)
q3=deg2rad(0)
q4=deg2rad(0)
q5=deg2rad(0)
q6=deg2rad(0)
T1=eval(T)
angles= tr2rpy(T1,'deg')
%%using toolbox
L(1)=Link([0 L0 0 pi/2 0])
L(2)=Link([0 0 L1 0 0])
L(3)=Link([0 0 L2 0 0])
L(4)=Link([0 L3 0 pi/2 0])
L(5)=Link([0 L4 0 -pi/2 0])
L(6)=Link([0 L5 0 0 0])
cilin=SerialLink(L,'name','UR5')
figure
cilin.teach
Q=[q1 q2 q3 q4 q5 q6]
T2=cilin.fkine(Q)
```

Figure 4: Forward Kinematics Code

This is the step by step of the code:

1. Symbolic Variables Declaration: Define symbolic variables ‘q1‘ through ‘q6‘ representing joint angles.

2. Link Length Definition: Set the lengths of the robot links ‘L0‘ to ‘L5‘. We inserted a lenght of 5 so we can see the joints in a better way.

3. Homogeneous Transformations Calculation: Calculate the homogeneous transformation matrices for each joint using rotation and translation functions (‘trotz‘ for rotation about z-axis, ‘transl‘ for translation, ‘trotx‘ for rotation about x-axis).

4. Total Transformation Matrix Composition: Combine individual transformations to obtain the total transformation matrix ‘T‘.

5. Matrix Simplification: Simplify the total transformation matrix ‘T‘.

6. Joint Angle Assignment: Set joint angles in radians using ‘deg2rad‘.

7. Evaluation of Transformation Matrix: Evaluate the transformation matrix ‘T‘ with the specified joint angles.

8. Roll, Pitch, Yaw Calculation: Compute Roll, Pitch, Yaw angles from the transformation matrix ‘T1‘ using the ‘tr2rpy‘ function.

9. Robotics Toolbox Usage: Define Denavit-Hartenberg (DH) parameters for each link.

10. SerialLink Object Creation: Create a ‘SerialLink‘ object ‘cilin‘ representing the robot using the defined DH parameters.

11. Robot Visualization: Display the robot using the ‘teach‘ method of the ‘cilin‘ object.

12. Joint Configuration Definition: Define joint configuration using the joint angles.

13. Forward Kinematics Calculation: Calculate forward kinematics (end-effector pose) using the robot object and specified joint configuration.

After running the code we could see the 3D simulation of the robot in Matlab as in figure 5.

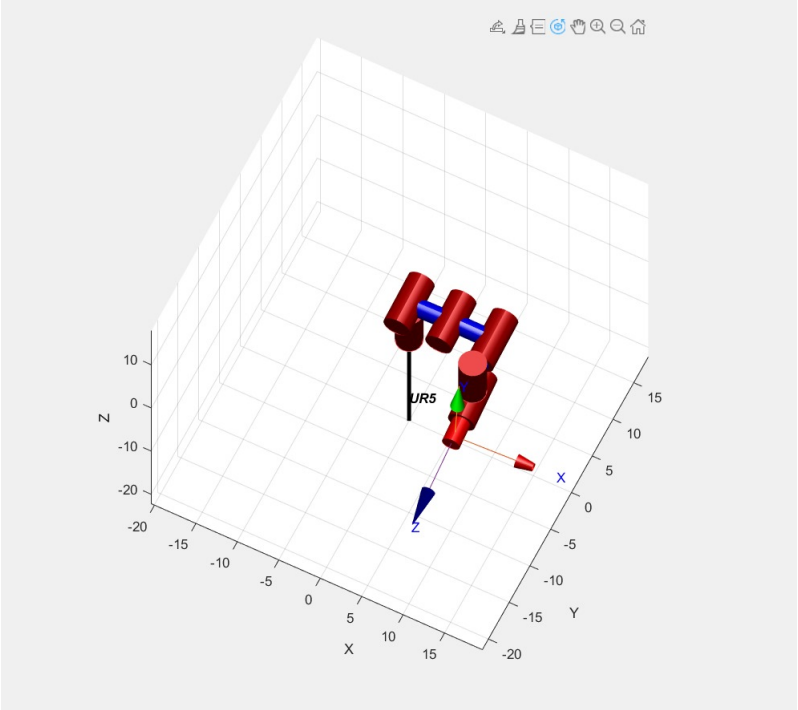

Figure 5: Simulation in Matlab of the UR5 Robot.

Also we could observe the transformation matrix as we could see in figure 6.

```
>> ur5_fk
>> ur5_fk
>> T

T =

[  cos(q6)*(sin(q1)*sin(q5) + cos(q2 + q3 + q4)*cos(q1)*cos(q5)) - sin(q2 + q3 +↙
q4)*cos(q1)*sin(q6), - sin(q6)*(sin(q1)*sin(q5) + cos(q2 + q3 + q4)*cos(q1)*cos(q5)) ↙
- sin(q2 + q3 + q4)*cos(q1)*cos(q6),   cos(q5)*sin(q1) - cos(q2 + q3 + q4)*cos(q1) ↙
*sin(q5), 15*sin(q1) + 5*cos(q5)*sin(q1) - 5*cos(q2 + q3 + q4)*cos(q1)*sin(q5) +↙
5*cos(q2 + q3)*cos(q1)*sin(q4) + 5*sin(q2 + q3)*cos(q1)*cos(q4)]
[- cos(q6)*(cos(q1)*sin(q5) - cos(q2 + q3 + q4)*cos(q5)*sin(q1)) - sin(q2 + q3 +↙
q4)*sin(q1)*sin(q6),   sin(q6)*(cos(q1)*sin(q5) - cos(q2 + q3 + q4)*cos(q5)*sin(q1)) ↙
- sin(q2 + q3 + q4)*cos(q6)*sin(q1), - cos(q1)*cos(q5) - cos(q2 + q3 + q4)*sin(q1) ↙
*sin(q5), 5*cos(q2 + q3)*sin(q1)*sin(q4) - 5*cos(q1)*cos(q5) - 5*cos(q2 + q3 + q4) ↙
*sin(q1)*sin(q5) - 15*cos(q1) + 5*sin(q2 + q3)*cos(q4)*sin(q1)]
[                              cos(q2 + q3 + q4)*sin(q6) + sin(q2 + q3 +↙
q4)*cos(q5)*cos(q6),                                  cos(q2 + q3 + q4)*cos(q6) ↙
- sin(q2 + q3 + q4)*cos(q5)*sin(q6),                          -sin(q2 + q3 + q4) ↙
*sin(q5),↙
5 - 5*sin(q2 + q3 + q4)*sin(q5) - 5*cos(q2 + q3 + q4)]
[↙
0,↙
0,                                               0,↙
1]

>>
```

Figure 6: Transformation Matrix.

We can see the results for variable T1 of the code in figure 7.

```
T1 =

     1     0     0     0
     0     0    -1   -20
     0     1     0     0
     0     0     0     1
```

Figure 7: Evaluation of the transformation matrix with specified joint angles.

The results for variable angles can be observed in figure 8.

```
angles =

    90     0     0
```

Figure 8: Roll, Pitch and Yaw angles from the transformation matrix.

Q results can be observed in figure 9.

```
Q =

     0     0     0     0     0     0
```

Figure 9: Joint Configuration.

Finally the results of variable T2 can be observed in figure 10.

```
T2 =
     1     0     0    10
     0     0    -1   -10
     0     1     0     0
     0     0     0     1
```

Figure 10: Forward Kinematics using the robot object.

12 Inverse kinematics

These expressions are derived from the FK matrix and represent each element of the rotation matrix (r1 to r9) and the position (px, py, pz). They are expressed in terms of trigonometric functions of the joint angles.

Forward Kinematics Transformation matrix can be observed in (1).

$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

The following expressions were obtained with (1):

$$r_{11} = C_1 C_{234} C_5 C_6 + C_6 S_1 S_5 - C_1 S_{234} S_6 \tag{2}$$

$$r_{21} = C_{234} C_5 C_6 S_1 - C_1 C_6 S_5 - S_1 S_{234} S_6 \tag{3}$$

$$r_{31} = C_5 C_6 S_{234} + C_{234} S_6 \tag{4}$$

$$r_{12} = -C_1 C_{234} C_5 S_6 S_1 S_5 S_6 C_1 C_6 S_{234} \tag{5}$$

$$r_{22} = -C_{234} C_5 S_1 S_6 + C_1 S_5 S_6 C_6 S_1 S_{234} \tag{6}$$

$$r_{32} = -C_5 S_{234} s_6 + C_{234} C_6 \tag{7}$$

$$r_{13} = -C_1 C_{234} S_5 + C_5 S_1 \tag{8}$$

$$r_{23} = -C_{234} s_1 S_5 C_1 C_5 \tag{9}$$

$$r_{33} = -S_{234} S_5$$

(10)

$$p_x = -c_1c_{234}s_5d_6 + c_5s_1d_6 + c_1s_{234}d_5 + s_1d_4 + c_1c_{23}a_3 + c_1c_2a_2$$

(11)

$$p_y = -c_{234}s_1s_5d_6c_1c_5d_6 + s_1s_{234}d_5c_1d_4 + c_{23}s_1a_3 + c_2s_1a_2 \tag{12}$$

$$p_z = -s_{234}s_5d_6c_{234}d_5 + s_{23}a_3 + s_2a_2 + d_1 \tag{13}$$

The following solutions were obtained with (11-13) and present analytical solutions for both, the FK and the IK. The complete IK solution is shown next:

$$A_1 = p_x d_6 r_{13} \tag{14}$$

$$B_1 = d_6 r_{23} p_y \tag{15}$$

A1 and B1 are computed based on the position of the end-effector and the orientation of the first joint. Theta1 is calculated using the atan2 function, and there are two possible solutions for it. First joint angle can be observed in (16).

$$\theta_1 = atan2(A_1, B_1)$$

(16)

$$+ - atan2(\sqrt{A_1^2 + B_1^2 - d_4^2}, d4) \tag{17}$$

$$c_5 = s_1 r_{13} c_1 r_{23} \tag{18}$$

$$s_5 = \sqrt{(s_1 r_{11} c_1 r_{21})^2 + (s_1 r_{12} c_1 r_{_{2)^2}}}$$

(19)

C5 and S5 are calculated based on the first three elements of the rotation matrix. Theta5 is calculated using atan2, and there are two possible solutions.

Fifth joint angle can be observed in (20).

$$\theta_5 = + - atan2(s_5, c_5) \tag{20}$$

Theta6 is computed based on the fourth and fifth joint angles and the last two elements of the rotation matrix.

Sixth joint angle can be observe in (21).

$$\theta_6 = atan2(\frac{c_1 r_{22} s_1 r_{12}}{sign(s5)}, \frac{s_1 r_{11} c_1 r_{21}}{sign(s5)}) \tag{21}$$

$$A_{234} = c_1 r_{11} + s_1 r_{21} \tag{22}$$

$$B_{234} = c_5 c_6 r_{31} \tag{23}$$

A234 and B234 are computed based on the first joint angle and the rotation matrix elements. Theta234 is calculated using atan2, and there are two possible solutions.

A2, B2, C2, D2, and E2 are calculated based on the position of the end-effector and the orientation of the first three joints. Theta2 is computed using atan2, and there are two possible solutions.

A3 and B3 are computed based on the position of the end-effector and the orientation of the first four joints. Theta3 is calculated using atan2.

Theta4 is calculated as the difference between the combined angle of the second, third, and fourth joints and theta3.

Second, third, and fourth join angles can be observed in (24).

$$\theta_{234} = atan2(B_{234} s_6 A_{23} - 4, c_5 c_6 A_{234} + s_6 r_{31}) \tag{24}$$

$$A_2 = c_1 p_x + s_1 p_y \tag{25}$$

$$B_2 = p_z d_1 \tag{26}$$

$$C_2 = 2a_2(d_1 p_z d_5 c_{234} d_6 s_5 s_{234}) \tag{27}$$

$$D_2 = 2a_2(d_5 s_{234} d_6 s_5 c_{234} c_1 p_x s_1 p_y) \tag{28}$$

$$E2 = a_3^2 a_2^2 d_5^2 + A_2(2d_5 s_{234} 2d_6 s_5 c_{234} A_2) B_2(2d_5 c_{234} + 2d_6 s_5 s_{234} + B_2) d_6^2 s_5^2 \tag{29}$$

Second joint (30).

$$\theta 2 = atan2(C_2, D_2) \tag{30}$$

$$+ - atan2(\sqrt{C_2^2 + D_2^2 - E_2^2}, E_2) \tag{31}$$

$$A_3 = \frac{c_2 B_2 s_2 A_2 + c_{34} d_5 + s_{34} s_5 d_6}{a_3} \tag{32}$$

$$B_3 = \frac{c_2 A_2 + s_2 B_2 s_{34} d_5 + c_{34} s_5 d_6 a_2}{a_3} \tag{33}$$

Third joint (34).

$$\theta_3 = atan2(A_3, B_3) \tag{34}$$

Fourth joint (35).

$$\theta_4 = \theta_{34} \theta_3 \tag{35}$$

13 Results

The objective of the project was to mimic a Culture Media Tube akin to one portrayed in Figure 11. At first the project needed the building of two lines of production for palletizing, for instance. two lines of palletizing. Every pallet was to have 20 boxes, the tiers of which are organised in the form of the 5 boxes in each tier. As an added challenge, the project called for the expansion to four production lines, with a new configuration: 5 meets and 8 boxes at each level.

Figure 11: Example of the packing line solicited

In Figure 11, a visual representation of the creation of the four production lines is provided, offering insight into their layout and arrangement.

To successfully execute this project, several components were indispensable:

- 4 UR5 robots, each equipped with its corresponding Vacuum gripper comprising 4 cups.

- 4 pallets, crucial to be able to accommodate all the requested boxes
- 8 conveyors, serving as the conveyor belt system to facilitate the movement of items along the production lines.
- A total of 120 boxes were required to populate the pallets across the various levels of the production lines.
- Additionally, 4 pedestals were necessary to support and stabilize the robots involved in the packing process.

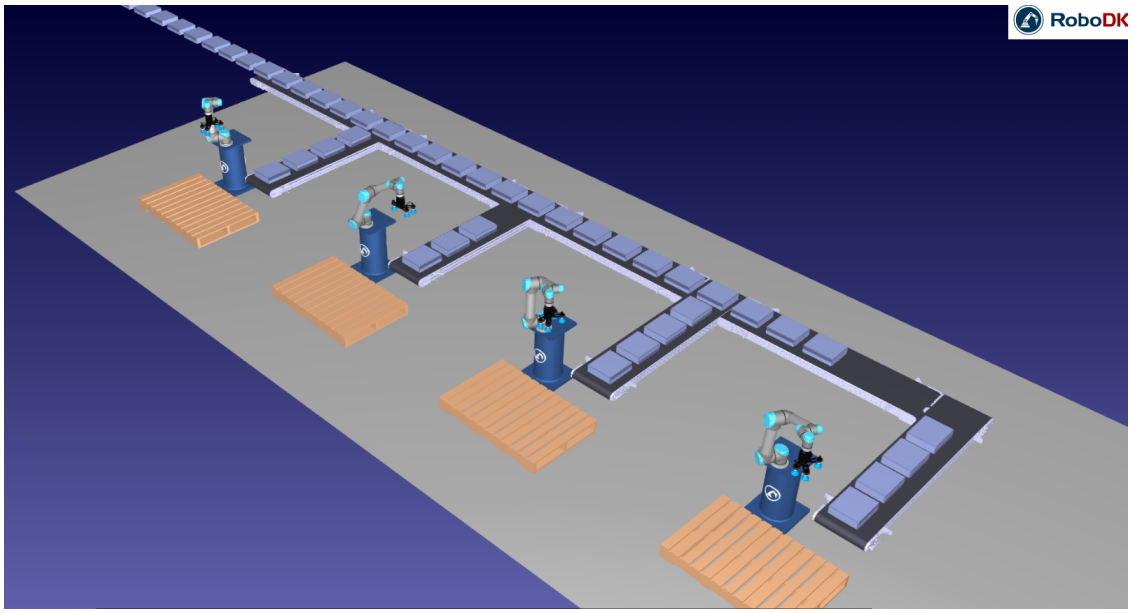

Figure 12: Final packing line in Robo DK

The project was accomplished by following all the process steps including, assembling components, configuring production lines according to the previously stated criteria, and finally ensuring the ability to copy the added features and efficiency of the referenced Packing Line, and exceed the challenge of accommodating two more packing line during the production.

The first step was to begin by adding the components of a single packing line, which included a UR5 robot with its gripper, a pallet, and a pedestal. Once these components were positioned as required, the next step was to create the necessary targets for the robot.

Initially, fourth primary targets were established:

- Home target: This designated the initial position of the robot, serving as its starting point for future operations by other users.
- Safety target: Here, a position was assigned to the robot where it wouldn't be obstructed by any objects, whether it had a box or not. It was crucial to ensure that the robot could reach this position safely without the risk of colliding with other objects as we can see in the Fig 13.

- Approach target: This target was positioned close to where the boxes would be picked up as we can observe in Fig 14.
- Take target: a target assigned for the robot to grasp the box, as illustrated in Figure 15.

Additionally, targets within the pallet were created to define the positions where the boxes would be placed. With 5 tiers of 8 boxes each, a total of 40 targets needed to be created, taking into account the dimensions of the boxes (12 x 10 inches) to prevent collisions or overlapping when arranging them.

Once the robot successfully grasped a box (with the assistance of an attach operation), it would proceed to the approach position to lift the box and then to the safety position. From there, it would be capable of placing the box onto the pallet (releasing it with the help of a detach operation) at the previously assigned positions

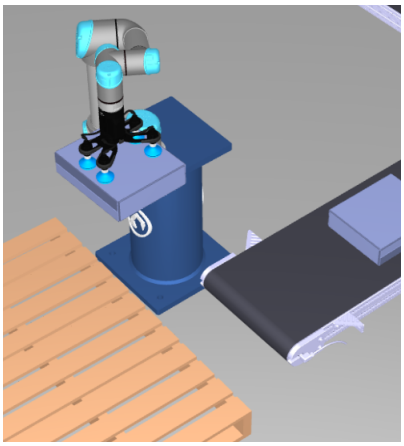

Figure 13: Safety target

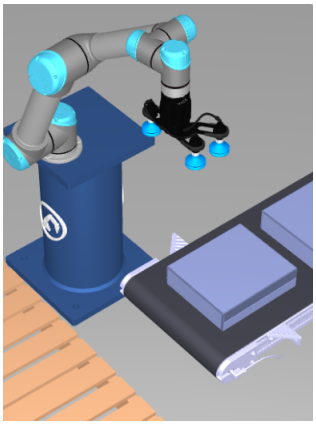

Figure 14: Approach target

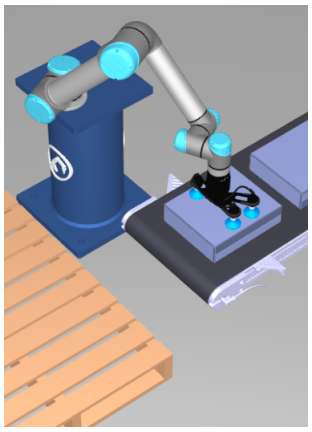

Figure 15: Take target

The next step is to create the program with the respective movements of the robot, including both angular and linear motions. We'll assign the movements following the logic that the robot should start at the safety position, then move to the approach position, proceed to the take position to grasp the box, lift it back to the approach position, then return to safety. From there, it will move to the first target of the first box on the pallet.

To ensure smooth operation and prevent issues, the robot should move from left to right and from the top of the pallet downwards. In the first column of the pallet, two boxes will be placed, followed by three boxes in the second and third columns (when viewed from above). Once the robot reaches the position for the first box, we'll assign a detach operation to release it. These movements will be repeated 40 times to complete the first packing line.

One we finish the first packing line, we have to do the same fourth more times to accomplish the full packing production. The final result is showing in the Fig 16.

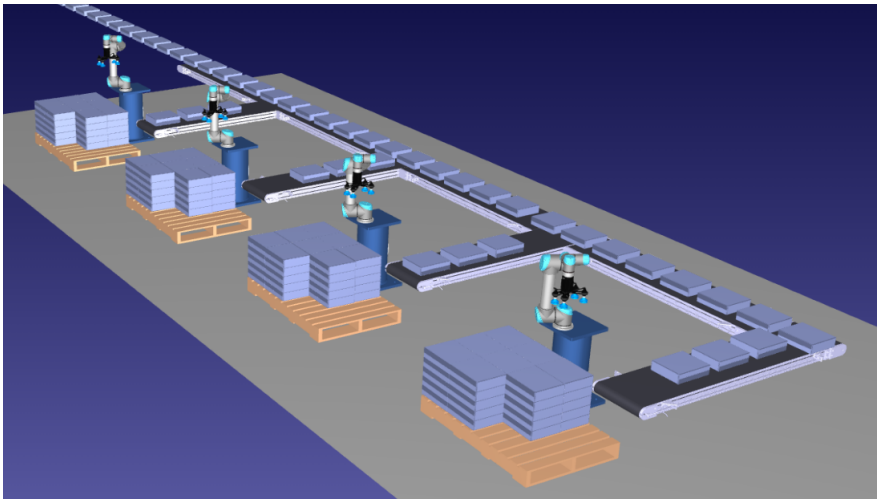

Figure 16: Result of the packing production

14 Conclusion

Overall, for this automation project, the combination of RoboDK and MATLAB was instrumental in the in-depth analysis of the robot kinematics and dynamics by doing the wrapped code for the Mat deployment platform. The process of meticulous modeling, simulation and analysis finally led us to an insight that was impressive enough to figure out how the robotic system ran and would yield the results.

MATLAB was used to mimic the actual process of forward as well as inverse kinematics; the real actuators on the robots could be placed properly. Singularity problems were exactly depicted by correspondence mechanisms, ensuring real-time answers with accuracy. Another prominent RoboDK feature is the simulation, which enables the user to create a realistic environment of a cybernetic entity for the testing of trajectory planning algorithms and the optimization of robot behavior that helps to makes the cycle times shorter.

Comparative assessments of different robot configurations provide what conditions certain functionalities for the optimum performance of the system which in turn provide informed decisions during design and optimization. Such statistical components were important because they showcased how fast and precise the system was and therefore provided us with the means to report its effectiveness.

In a more general way, the results will yield additional data that can be applied in the process of the line's programming and designing the fitting packaging guidelines. Demonstrating a high adaptability to existing production patterns and the ability to continuously increase process yield with RoboDK and MATLAB should allow manufacturers achieve this goal.

It is the entirety of this research that brings forth the machination of how advanced computational devices act as enablers to the industrial process hence, enabling future research into automation and robotic applications.

15 Bibliography

1 Craig, J. J. (2005). Introduction to Robotics: Mechanics and Control (3rd ed.). Prentice Hall.

2 Siciliano, B., Khatib, O. (Eds.). (2016). Springer Handbook of Robotics. Springer.

3 MathWorks. (n.d.). MATLAB Documentation. https://www.mathworks.com/help/matlab/

4 IEEE Transactions on Robotics. (n.d.). https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=8

5 RoboDK. (n.d.). RoboDK Documentation. https://robodk.com/doc/en/