3rd International Conference on
Robotics Automation and Non-Destructive Evaluation,
Chennai, India, 23 April 2022.
https://doi.org/10.13180/ RANE.2022.23.04.11

MODELLING AND SIMULATION OF DUAL ARM ROBOT FOR MATERIAL TRANSFER APPLICATIONS

SURYA P and AROCKIA SELVAKUMAR AROCKIA DOSS
School of Mechanical Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India
*E-mail: surya.p2020@vitstudent.ac.in, *arockia.selvakumar@vit.ac.in*

OSCAR FERNANDO AVILES
Mechatronics Engineering, Nueva Granada Military University (UMNG), Bogota, Colombia
E-mail: oscar.aviles@unimilitar.edu.co

Abstract. Dual-arm industrial robot is an upcoming technique; many industries started developing dual-arm robots for the future. Dual-arm robot can be used in complex assembling and efficient than two separate robots. Most of the pick and place Dual-arm robots operate by using two arms individually. This research work is done to make the robot smart to pick and place object by either one or two arms based on its size. This paper shows the modeling and simulation of a Dual Arm robot that can identify and pick objects in front of it by object detection program in python. If object size is less than 100mm, robot use one arm, if size is more than 100mm two arms are used together. The forward and inverse kinematic calculation of the robot is done in MATLAB and Simulation of robot is done is V-Rep. The resulting simulation is obtained by forming communication between all three nodes Python, MATLAB and V-Rep.

Keywords: Robotic Arm, Dual Arm Manipulator, Kinematics, Image Processing, Object Detection.

1. Introduction

Nowadays development in technology is rapidly growing with the human needs. The innovation of robots boosted the development of technology to next level [1-3]. Robot is a machine that work automatically or semi-automatically based on the command given to it. Robot is an interdisciplinary device that can sense the surroundings and act based on it [4-5]. Many attempted were made to replicate human as a robot, dual arm robot is one of result of it. Dual arm robot has two individual arms; complex task that can't be done by single arm robot can be done with the dual arm robot [6]. In this paper, a dual arm robot is modeled and simulated to match the requirement 3 platforms are used for different operations, V-Rep for Simulation, Python for object detection and MATLAB [7-8] for Kinematics calculations. These 3 platforms communicate each other through as shown in Figure 1. There are many techniques are used to identify the object but they were costly and required many cameras to capture object to find its shape and size. In this single camera with mapping technique is used to identify the size of objects.

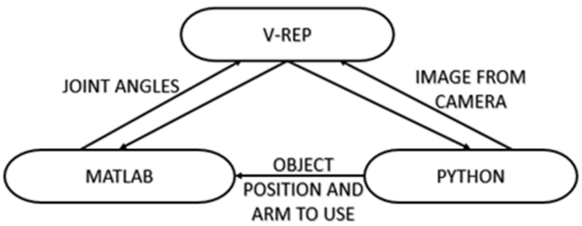


Figure 1. Interaction between 3 platforms.

When the simulation started, the camera in the robot sends an Image taken from the V-Rep to Python, in Python the image was processed and the object shape, height and position are identified and sent to MATLAB, in MATLAB the position is used to find the joint angles of the robot and the joint angles are send to the V-Rep Software the robot pick and place the object to required position. The working flow of the robot program is show in Figure 2, in this if the object size is less than 100mm than one arm is used to pick the object either left or right arm by checking the status of the arm like serial manipulator and if the object size is greater than 100mm than both the arms is used to move the arm, since the robot use both arms, it acts as a parallel manipulator. Forward and inverse kinematic are done in MATLAB by iteration method.

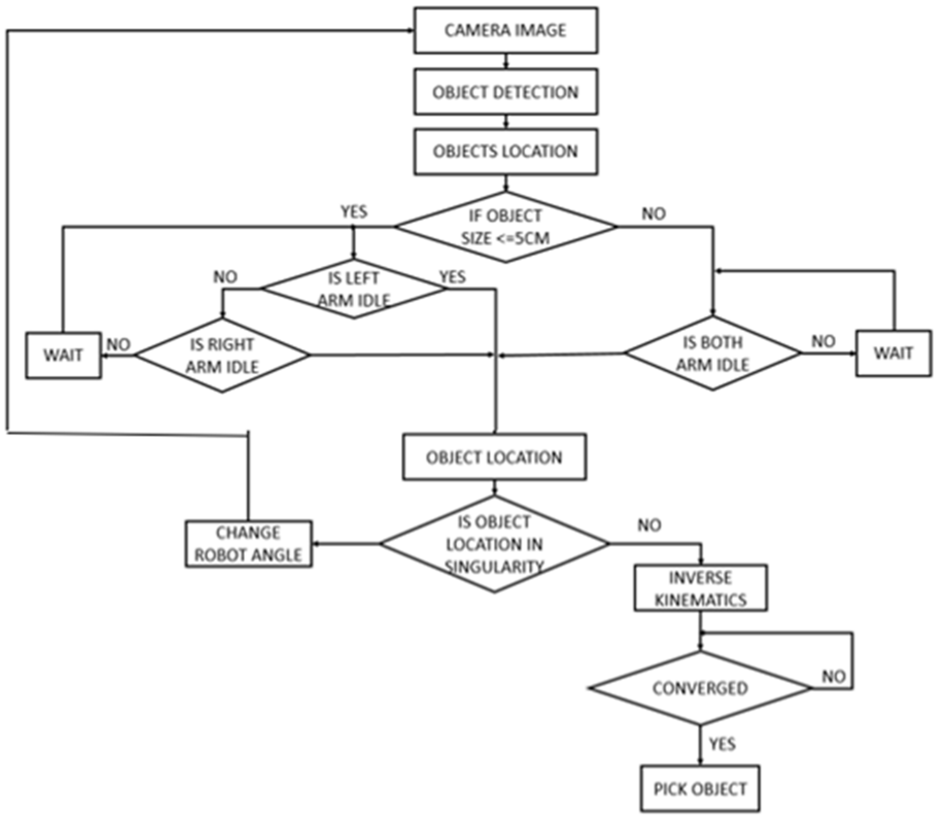
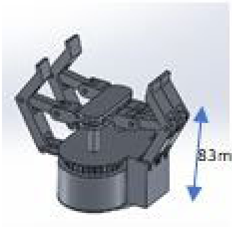


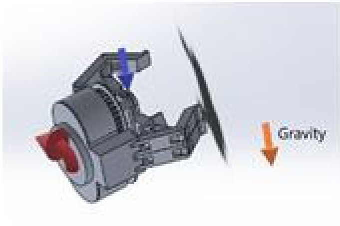
Figure 2. Working Flow Diagram.

2. Modelling of Robot

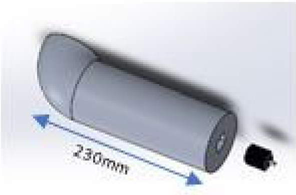
The Robot was modeled and Assembled in SOLIDWORKS. Since this robot is a replica of a Human arm, the link lengths are assumed to be the length of arm of an average human. Similar to human, the Robot has two arms; each arm has 5 Links and 4 Joints. The Robot was modeled from Gripper to body to providing accommodation for the motor in the link for running the previous links. The gripper mechanism was modeled so that it can hold object of various sizes. The gripper has 4 fingers, 2 fingers are static and 2 fingers are movable which can rotate up to 180^{o}, the movable fingers are attached to pair of gears with are driven by a small motor, a piston is attached to fingers, which controls the gripping as shown in Figure 3(a). After modeling the body, using the motion analysis the body was analyzed in SOLIDWORKS by providing a payload of 5N and the Motor is given to run the gripper by 10RPM as the result of the analysis the required torque to move the body is found as shown in Figure 3(b). By comparing the torque with the catalogue, the motor is selected.



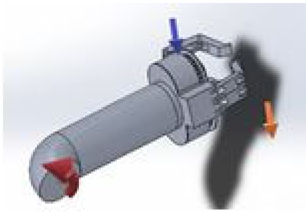
(a)



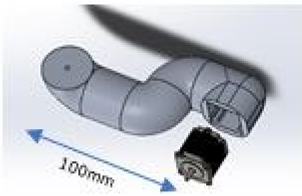
(b)



(c)

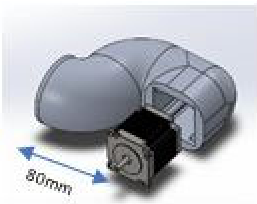


(d)

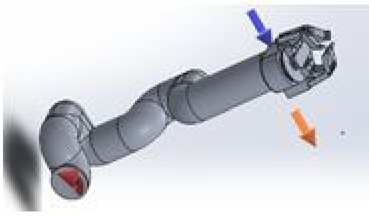


(e)

(f)

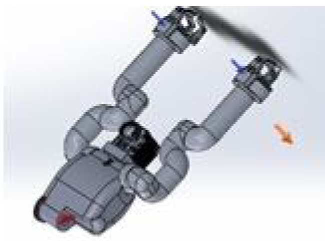


(g)

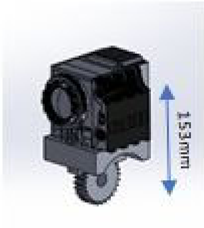


(h)

(i)

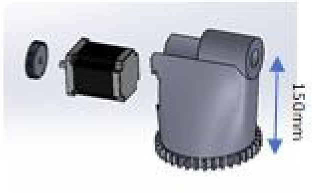


(j)

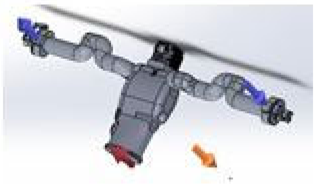


(k)

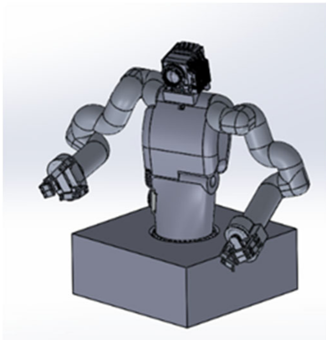
(l)



(m)



(n)



(o)

Figure 3. (a) Gripper Design, (b) Torque Analysis, (c) Arm Design, (d) Arm Torque Analysis, (e) Limb Design, (f) Limb Torque Analysis, (g) Shoulder Design, (h) Shoulder Torque Analysis, (i) Upper Body Design, (j) Upper Body Torque Analysis, (k) Camera Design, (l) Camera Torque Analysis, (m) Lower Body Design, (n) Lower Body Torque Analysis, (o) Complete Assembly of the Robot.

The arm is the next link after the gripper, this link was modelled by considering the accommodation of motor for the previous link i.e., gripper. One end of the arm has the motor accommodation as shown in Figure 3(c) and the other end has a coupling to attach to the motor for its driving. This link is also analysed by attaching the gripper and arm together and same payload of 5N at the end and Torque required was found and motor was selected based the torque as shown in Figure 3(d). The Limb, Shoulder and Upper body of the robot was also modeled in the similar manner as shown in the Figure 3(e)-2(j). The Camera was used in the robot for object detection through image processing; the camera was fitted into a camera mount. The camera mount has a large gear which helps in precise movement of camera; the camera mount also has a separate motor. The camera mount is also analyzed in similar manner as previous links shown in Figure 3(k)-(l). The Lower body and base are the two links which needs high torque since these supports the whole body of the robot, these links are modeled and analyzed in same way as the other due to the high torque requirement gears are used, which are modeled along with the link as shown in Figure 3(m)-(n). The complete assembly is shown in Figure 3(o).

After assembling all the parts, the assembly was converted to URDF format and imported in V-Rep Software for simulation. The Table 1 shows the list of all torque found based on the Gear ratio used and the motor selected from the catalogue.

Table 1. List of Calculated Torque for the selected Motor.

	Joint Torque		**Gear Ratio**	**Required Torque N-m**	**Motor**	
	N-mm	**N-m**			**Name**	**Torque (N-m)**
Gripper	42	0.042	-	0.042	NEMA 8	1.6
Limb	20622	20.622	-	20.622	NEMA 23	320
Arm	134451	134.451	-	134.451	NEMA 23	320
Shoulder	1308150	1308.15	-	1308.15	NEMA 23	1500
Camera	24	0.024	01:03	1.6	NEMA 8	1.6
Upper Body	8123512	8123.512	03:07	3481.505	NEMA 34	3800
Lower Body	26251120	26251.12	01:04	6532.78	NEMA 34	7200

3. Object Detection

The object detection is used to get the size and position of the object in front of the robot, the image is taken from the camera i.e.., Vision sensor in the V-Rep software and it was sent to the Python software. In the python program the Image was processed by separating the images into two separate images by masking technique one for the height and other for the shape of the object, by measuring the pixel we get the dimensions of object in pixels [9] as shown in Figure 4. To get the size in cm a reference object of known size is provided at the left most of the table which is seen in the image of from the vision sensor, in image processing the pixels/cm ratio was found by comparing the reference object to its pixels and it is used to find the shape and size of the object.

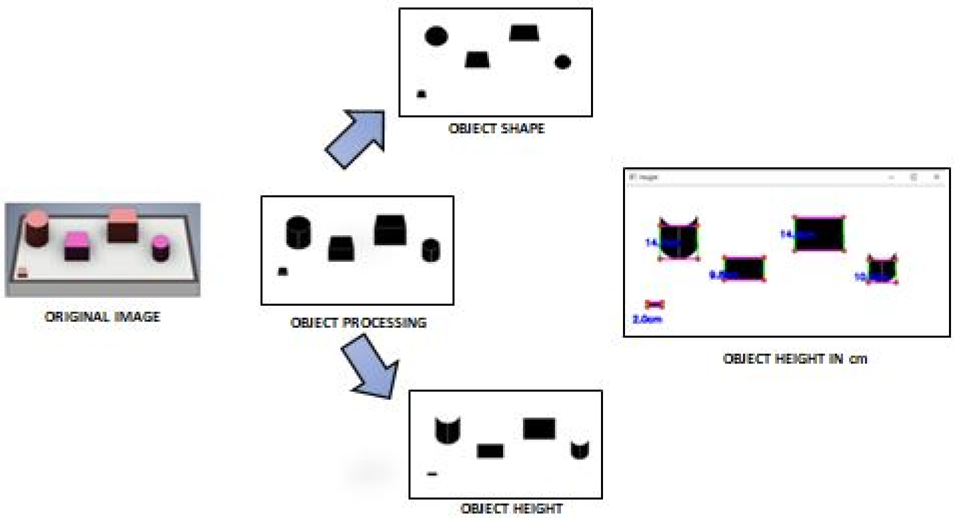


Figure 4. Object Size Detection.

In real-time camera vision the object view is perspective, i.e.., a rectangle show as a trapezoid so mapping technique is used to map the image [10] and pixels/cm as a function of height of object and it is used to find the dimensions and position of the objects [11] [12] as shown in Figure 5 and 6.

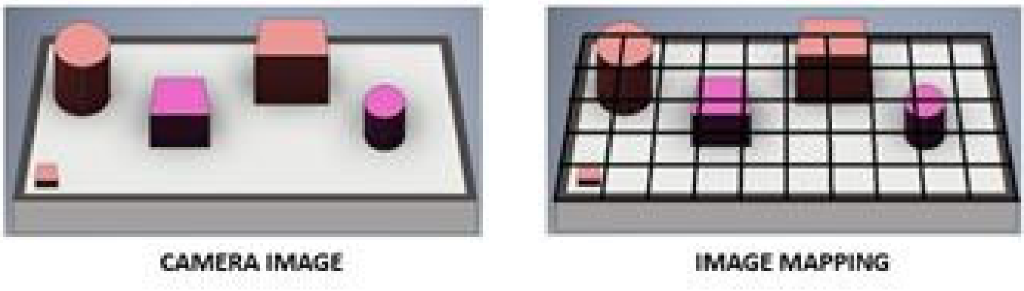


Figure 5. Camera Image Mapping.

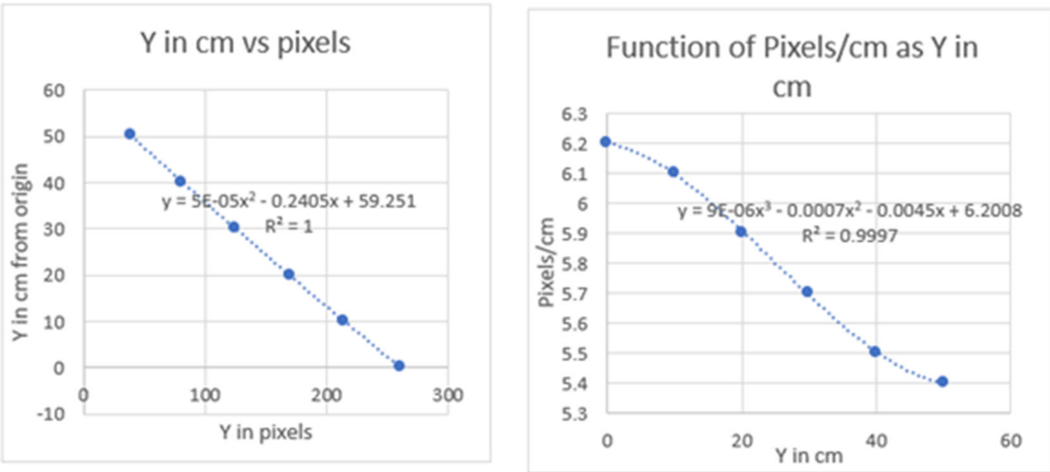


Figure 6. Graph forming relationship between Height and Pixels/cm.

Table 2. Pixels/cm Calculation

Distance (cm)		Pixels		Pixels/cm
X	Y	X	Y	
0	0	326	261	6.2
0	10	326	214	6.1
0	20	326	170	5.9
0	30	326	125	5.7
0	40	326	81	5.5
0	50	326	39	5.4

4. Kinematics

The Kinematics calculations are for the configuration diagram of the robot in the Figure Coordinates are marked based on the Denavit-Hardenberg rules and from it Denavit-Hardenberg parameter table is formed and the parameters of the D-H table is substituted in the generalized homogenous matrix Eq. (1). to get homogeneous matrix for each link as shown in Table 3. These homogeneous matrixes are multiplied together to get the homogeneous matrix of the robot.

Table 3. Denavit–Hartenberg parameters

θ	α	r	D
$\theta_{1L}+90°$	90°	0	200
$\theta_{2L}+90°$	0°	245	0
θ_{3L}	−90°	0	136.071+40
$\theta_{4L}-90°$	−90°	250	0
θ_{5L}	0°	270	0
θ_{6L}	90°	30	0

θ	α	r	D
$\theta_{1R}+90°$	90°	0	200
$\theta_{2R}+90°$	180°	245	0
θ_{3R}	90°	0	136.071+40
$\theta_{4R}-90°$	−90°	−250	0
θ_{5R}	0°	−270	0
θ_{6R}	90°	−30	0

Where

θ - Rotation about Z_{N-1} so X_{N-1} matches the X_N axis.

α - Rotation about X_N so Z_{N-1} matches the Z_N axis.

r - Distance between the axis in X_N direction.

d - Distance between the axis in Z_{N-1} direction.

If we substitute the joint angle of each joint into the homogeneous matrix will give as the end-effector position of the robot. But the output matrix is a 4x4 matrix we have to convert it into 6x1 matrix so it can be used in Inverse Kinematics by using Eq. (2).

$$T_N^{N-1} = \begin{bmatrix} \cos\theta_N & -\sin\theta_N\cos\propto_N & \sin\theta_N\sin\propto_N & r_N\cos\theta_N \\ \sin\theta_N & \cos\theta_N\cos\propto_N & -\cos\theta_N\sin\propto_N & r_N\sin\theta_N \\ 0 & \sin\theta_N & \cos\theta_N & d_N \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \Delta\theta_1 \end{bmatrix} + R_1^0 \begin{bmatrix} 0 \\ 0 \\ \Delta\theta_2 \end{bmatrix} + R_2^0 \begin{bmatrix} 0 \\ 0 \\ \Delta\theta_3 \end{bmatrix} + \cdots \quad (2)$$

The inverse kinematics by closed-form method is very difficult for this application so [13], iterative methods are used, we have used Newton-Rapson method. In Newton-Rapson method an initial joint angle is given as input and for the angle inverse kinematics is done by multiplying inverse Jacobian to it and end-effector position is found and the difference between the target position to end-effector position is found and used in forward kinematics and the angle is added to the initial angle, this process is repeated till the difference is negligible. For the Inverse Kinematics Jacobian matrix is found and since the Jacobian matrix is not a square matrix it can't be inversed so the following methods are used to find the Jacobian inverse.

$$\text{Pseudo Inverse} = (J^TJ)^{-1}J^T \quad (3)$$

$$\text{Single Value Decompostion} = [U, S, V] = SVD(J) = V * S^{-1} * U^T \quad (4)$$

$$\text{Damped Least Square} = J^T(J^TJ)^{-1} + \lambda^2 I \quad (5)$$

A MATLAB function is written to find both forward and inverse kinematics of the robot. If the robot uses one of its arms i.e.., object size is less than 100mm than inverse kinematics is found for that arm. If the robot requires two arms i.e.., object size is greater than 100mm than the inverse kinematics is found by for each increment in angle of one arm whole inverse kinematics of other arm is found and used to run the robot.

5. Conclusion and Future Scope

The robot is successfully modelled in SOLIDWORKS and Exported to V-Rep and interfacing is formed between V-Rep to MATLAB and Python as shown in Figure 7. The object detection is programmed and Python and robot Kinematics are programmed in MATLAB and the simulation runs successfully. Future work includes Dynamic calculation of robot, Load analysis of links.

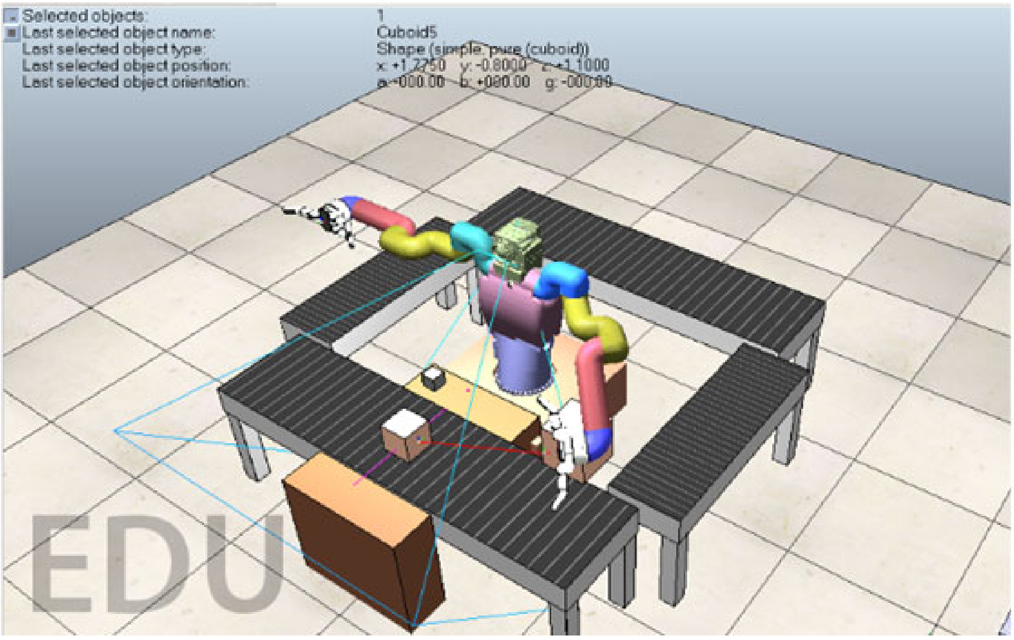


Figure 7. V-Rep Simulation Model.

References

1. Arockia Doss A.S., Mishra B., Mohammed S., Lingampally P.K., Short M. (2021) Robotic Arm for Biomedical Applications. In: Hussain C.M., Di Sia P. (eds) Handbook of Smart Materials, Technologies, and Devices. Springer, Cham. https://doi.org/10.1007/978-3-030-58675-1_38-1.
2. Jalamkar, D., and Arockia Selvakumar A. Krishnakumar. "Implementation of Internet of things in a mobile Humanoid robot: A base for future of IOT enabled robotics applications." *International Journal of Engineering & Technology* 7.4.36 (2018): 386-389.
3. Ahmed, AS Faraz, V. Sudharsan, and Arockia Selvakumar Arockia Doss. "Real-Time Motion Control of a Humanoid Robot Using Deep Learning." *Journal of Physics: Conference Series*. Vol. 2115. No. 1. IOP Publishing, 2021.
4. A. A. Mohammed and M. Sunar, "Kinematics modeling of a 4-DOF robotic arm," 2015 International Conference on Control, Automation and Robotics, 2015, pp. 87-91, doi: 10.1109/ICCAR.2015.7166008.
5. Khairidine Benali, Francois Brethe, François Guérin, Marc Gorka. Dual arm robot manipulator for grasping boxes of different dimensions in a logistics

warehouse. *2018 IEEE International Conference on Industrial Technology (ICIT)*, Feb 2018, Lyon, France. ⟨10.1109/ICIT.2018.8352167⟩. ⟨hal-01794589⟩.

6. Chandan, G M et al. "Real Time Object Detection and Tracking Using Deep Learning and OpenCV." *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)* (2018): 1305-1308.
7. Tarwadi, Parag J., and Arockia Selvakumar Arockia Doss. "Vision based Pick and Place Robot for Sorting Objects Using MATLAB." *Applied Mechanics and Materials*. Vol. 592. Trans Tech Publications Ltd, 2014.
8. Jitendra Tarwadi, P., Selvakumar Arockia , A., Corrales, R., & Juan, A. (2017). Manipulation and Path Planning for KUKA (LWR/ LBR 4+) Robot in a Simulated and Real Environment. *Journal of Automation, Mobile Robotics and Intelligent Systems*, *11*(3), 15–21. https://doi.org/10.14313/JAMRIS_3-2017/24.
9. T. F. Abaas, A. A. Khleif, and M. Q. Abbood, "Inverse Kinematics Analysis and Simulation of a 5 DOF Robotic Arm using MATLAB", *alkej*, vol. 16, no. 1, pp. 1–10, Mar. 2020.
10. Chao, Zhiqiang & Wang, Fei & Zhang, Chuanqing & Li, Huaying. (2017). Inverse Kinematics Solution and Verification of 4-DOF Hydraulic Manipulator. Journal of Physics: Conference Series. 916. 012022. 10.1088/1742-6596/916/1/012022.
11. Peter Corke., A Robotics Toolbox for MATLAB. IEEE Robotics &Automation Magazine (1996).
12. Tarwadi, Parag Jitendra, A. Arockia Selvakumar, Ramon Corrales, and Antonio Juan. "Manipulation and path planning for KUKA (LWR/LBR 4+) robot in a simulated and real environment." Journal of Automation, Mobile Robotics and Intelligent Systems: 15-21 (2017).
13. A. Arockia Selvakumar, A numerical approach for 3 PRS parallel manipulator, International Journal of Engineering and Technology (UAE), 7, 90–94, DOI: 10.14419/ijet.v7i4.10.20712 (2018).