Nat. Volatiles & Essent. Oils, 2021; 8(5): 3189-3199

NVEO
Natural Volatiles &
Essential Oils

Design And Simulation Of Vision Based Robotic Arm Using ROS

[1]Vishwa Priya V, [2]Srinivasan V, [3] Dakshineshkar N R, [4]Thamesh N R, [5]Sanjayakumaran S, [6]Vikram R

[1]Robotics and Automation Engineering
PSG College of Technology
Coimbatore, India
vvp.rae@psgtech.ac.in

[2]Robotics and Automation Engineering
PSG College of Technology
Coimbatore, India
srinisri1636@gmail.com

[3]Robotics and Automation Engineering
PSG College of Technology
Coimbatore, India
dakshineshkar007@gmail.com

[4]Robotics and Automation Engineering
PSG College of Technology
Coimbatore, India
thameshsakthi@gmail.com

[5]Robotics and Automation Engineering
PSG College of Technology
Coimbatore, India
kumaransanjay268@gmail.com

[6]Robotics and Automation Engineering
PSG College of Technology
Coimbatore, India
vikramraja.ur@gmail.com

Abstract

A multistage technique for simulating a vision-based 6-DOF (Degrees of Freedom) robotic arm manipulator is described in this research. The major goal of this project is to connect a vision system with a 6-DOF robotic arm to expand the capabilities of the combined camera-robot system. Since the robotic arm does not yet have an integrated vision system, a camera has been mounted above the robot's workspace to accomplish the desired results. This system was modelled in ROS (Robot Operating System), and the main issues are developing a suitable sequence of activities, implementing adequate communication between camera and robot, and integrating with the system components in ROS.

***Keywords**—6DOF, vision, ROS, Gazebo, Rviz, Camera, image processing, center of mass, image subtraction, Homogeneous Transformation, Vision integration.*

I. Introduction

Robot vision guiding is rapidly evolving, and robotic vision systems are already widely used in a wide range of automated manufacturing and packaging operations. Advanced vision technology provides savings, increased quality, dependability, safety, and productivity. Parts are identified, navigated, and organised using robot vision. Vision applications frequently deal with locating and positioning an item for robotic handling or inspection prior to performing an application. Vision-guided robots have the potential to replace multiple mechanical tools with a single robot station. The 24 percent year-on-year growth in robot vision systems indicates that firms in the manufacturing industry have begun to recognise the numerous benefits of utilising this technology. A combination of vision algorithms, calibration, and cameras provide vision capability.

A. *Problem Statement*

A robotic arm without a vision-based sensor, requires fixed workspace as it is not flexible and it requires additional programming. This project is to implement a vison based robotic arm for dynamic workspace, which performs autonomous operations like pick and place.

B. *Objective*

The goal of this article is to use a vision-based robotic arm to decrease human effort and mistakes in material handling systems. A comprehensive vision system in a robotic arm may take into account variations in their working environment. Parts do not have to be displayed in the same order. When industrial robots are outfitted with modern visual systems, they become significantly more active.

II. LITERATURE SURVEY

A. *Literature Review*

Taryudi et al. [1] developed a method for Object posture estimation, which is a critical component of a vision-based object handling system employing an industrial robot manipulator. This paper describes how to use a stereo vision system to estimate 3D (3 dimensional) item position and orientation in order to pick up and place the object targeted at any spot inside the workspace. A calibrated stereo camera is utilized to take pictures of the item on the left and right cameras in order to complete this assignment. The unique object characteristic is then extracted, and the item's 3D location and orientation are computed using an image processing method. Finally, the gripper-equipped end effector of the robot arm will pick up the object targeted by the object pose estimate output and deposit it in the appropriate spot.

Rajesh Kannan Megalingam et al [2] developed and simulated a six-degree-of-freedom (DOF) robotic arm that may be utilized for search and rescue activities in disaster-stricken areas. A user-friendly algorithm designed for keyboard-based interaction with the 6 DOF robotic arm is also described. RVIZ was used to build and visualize the 6-DOF articulated robotic arm, while Moveit was utilized as a control interface with the Robot Operating System (ROS). The joint control mechanism prevents us from utilizing the joint by manipulating the arm with the end effector. The approach described in this study is less costly and easier to regulate. The first three degrees of freedom (DOF) in the proposed 6 DOF robotic arm

Leszek Baranowski et al [3] used the Robot Operating System to describe the notion of combining a three-dimensional image sensor device with a robotic arm manipulator. The location of the user's arm is obtained with the help of a 3D camera. OpenNI Kinect package is used for person detection and tracking. The software enables the user to interface with digital gadgets (camera). Following a successful startup, the tracking script returns the axes coordinates of each portion of the user's body. These coordinates are sent to the Kinova arm's joint. As a result, the arm gripper moves in accordance with the human arm motion of any one hand.

G.S. Sandhu et al. [4] investigated low-cost solutions for real-time pick and place activities. A modular strategy was employed to complete this model, in which it was broken down and its essential pieces – sensing/perceptual capabilities, 3D image processing/manipulation, and robotic manipulators – were integrated using the Robot Operating System (ROS). The sensory capabilities were provided via a depth sensor in the form of a Microsoft Kinect, which was utilized to perform segmentation and clustering procedures using the OpenNI drivers and Point Cloud Libraries. The system was able to distinguish and extract individual items from 3D point clouds using these techniques.

Sergio Hernandez Mandez et al. [5] have created an interface for a robotic arm that includes a motion planner and a 3D viewer (RViz). Their arm's primary function is to manipulate and hold items in order to move them from one area to another utilizing a Robot Operating System (ROS). An open library motion planner that deals with restrictions, i.e., collision detection among arm joints and objects, is utilized to do the mentioned work. A 3D viewer called RViz is also utilized to visualize the arm's movements. The 3 degrees of freedom (DOF) arm was built in FreeCAD, and the output files required to print the arm were exported at the same time to provide the Unified Robot Description Format (URDF) files required by the planner and viewer. In their project, force sensors are also utilized to guarantee safe and secure holding.

B. Outcome of the Literature Review

The literature survey has bought effective results and has helped in understanding the requirements clearly. The outcome includes

- Different techniques used for Object Pose Estimation.
- Control of Robotic Arm using ROS.

C. Patent Search

Process for gripping an object by means of a robot arm equipped with a camera (CA2511861A1)

An intelligent approach for determining the position of an item on successive photos is produced by optimally merging points and pictures by one movement consistent with the movement of the camera that took the two images. The article's position might be computed in this manner. The points that can be coupled in apply to belong to the item, even if they must be obtained automatically, the background of the image often having a lesser variety of unusual points that cannot be paired with this movement.

Method and apparatus for single image 3D vision-guided robotics (US8095237B2)

A technique of identifying and directing three-dimensional objects in order to allow robotic manipulation of an item with changeable location and orientation utilizing a sensor array, which is a collection of one or more sensors capable of creating a single picture.

ROS-based mechanical arm grabbing method and system (WO2018137445A1)

A completely mechanical arm grasping method and machine based on ROS. An implementation method consists of the following steps: configuring the use environment of a camera on a higher computer, then positioning the camera above or laterally above an item to be grabbed and acquiring a photograph of the item to be grabbed that includes a positioning mark; inputting the photograph to the higher computer; analyzing the photograph of the camera and appearing information processing via the use of a specific set of rules to gain spatial pose records. The lower computer receives and parses the movement records queue sent by the upper computer, and then drives a mechanical arm to perform a grasping operation in accordance with a predefined action.

III. METHODOLOGY

The methodology as shown in Fig 1 starts with CAD modeling of the robot arm followed by URDF (Universal Robot Description Format) conversion. The URDF is imported in the ROS Move-it assistant software where the addition of joint state controllers and kinematics planners are provided. After the completion of the Move-it assistant setup, the Move-it configuration file along with the generated URDF is visualized in Gazebo which is a simulation environment. Finally, using image processing, the 2D position of the object is found and sent to the Moveit Commander, which moves the end effector of the robot to the position where the object is located for pick and place application.

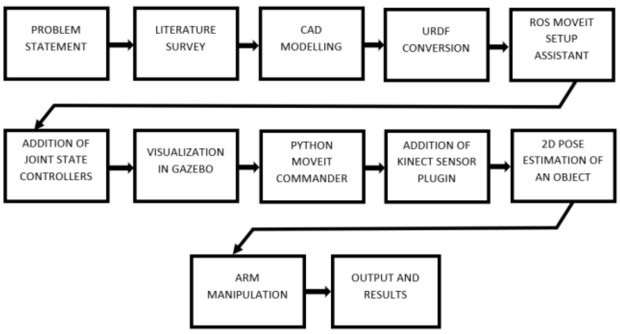

Fig 1 Block Diagram

A. *Robot Description*

SolidWorks software, which is used for 3D solid modelling, was utilised to develop the robotic arm as shown in Fig 2. The designed robot has six degrees of freedom. The reason for developing a 6-DOF arm is that ROS KDL Kinematic Planner requires at least 6 DOF for trajectory planning; however, if less than 6 joints are available, IK fast planner can be used; but even so, the implementation of IK fast planner increases complexity; thus, the 6-DOF arm is designed to eliminate the difficulty. The arm's total reach is 720 mm, and the end effector may be changed to match the user's demands.

General specifications

No of DOF: 6 DOF

REACH: 720mm

Link1: 335mm, Link2: 135mm, link3 125mm, link4 165mm
MATERIALS: Carbon fiber (Links), aluminum (Actuators) which is specified in solid works model

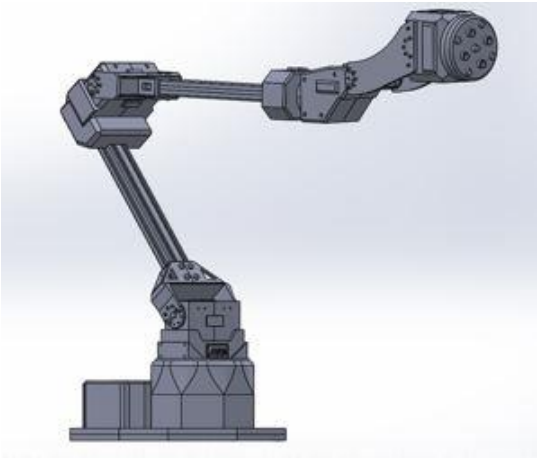

Fig 2 CAD Model of the Robot

B. ROS Moveit Configuration

ROS move it configuration: - The robotic arm employed in this development is a six-axis articulated robot. The Solid-works URDF exporter plugin generates the URDF, allowing us to select the joint type and axis. The revolute kind of joint is utilized in this instance.

The plugin also allows to add a parent-child link relationship. Following the development of the URDF, it is exported to ROS Moveit Assistant for the inclusion of the Self-Collision matrix and ROS Controllers as shown in Fig 3 and 4 respectively.

The robot arm's links and joints are specified in the setup assistant's planning group. In the Setup assistance, the KDL kinematic solver is also selected to complete the configuration setup.

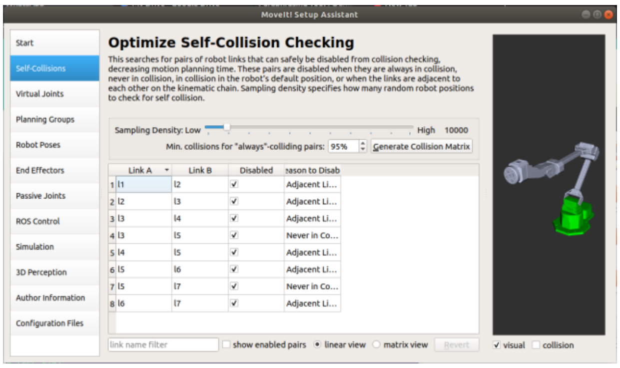

Fig 3 Generating Collision Matrix

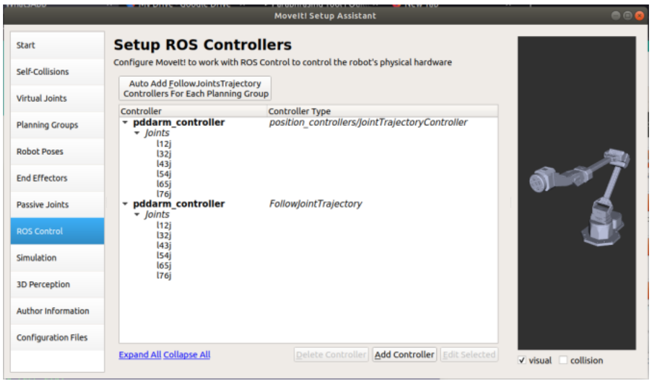

Fig 4 ROS Controllers

IV. IMPLEMENTATION

A. Design Architecture

After designing and spawning the robotic arm in gazebo environment, it is interfaced with Python Moveit commander which controls the robotic arm according to the goal position by fetching object position and the current position of arm, moveit commander plans a path in a such a way to goal position with the shortest path. Object position coordinates are determined with the help of camera feedback from the gazebo environment, which is placed above the robot workspace.

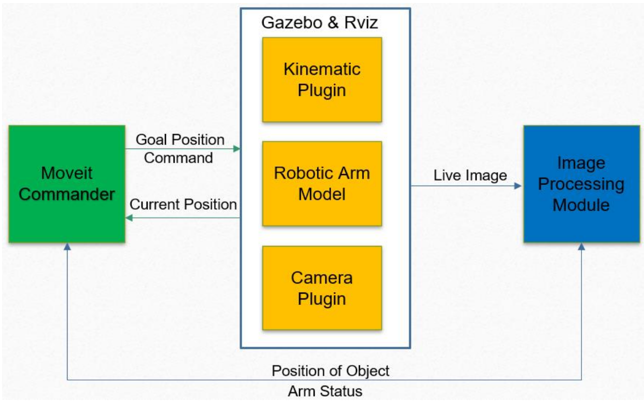

Fig 5 Schematic Diagram

Object's 2d position in robot workspace is estimated using some image processing techniques such as image subtraction, image thresholding, center of mass method. After the pose estimation, the object position is sent to python moveit commander for arm execution and in return it sends feedback on goal position execution whether success or failed.

B. Camera Setup

Camera is a main device, which acts as a vision to robot. Camera is needed to place in such a way that it can cover maximum field of view without affecting resolution. Here the camera is placed above the workspace at a distance of 1.5 meter which covers 1.72 * 1.2 meters.

Pixel per meter is (Distance) as shown in Fig 6 and Fig 7 measured by using the formula pixel per meter = width or height / number of columns or rows, where distance is measured manually by placing a ruler or spawning an object of known size which is a trial-and-error method. Here pixel per ratio is 1.72 meter. With these requirements, position of

the object can be estimated in meter with respect to camera coordinates as shown in Fig 7.

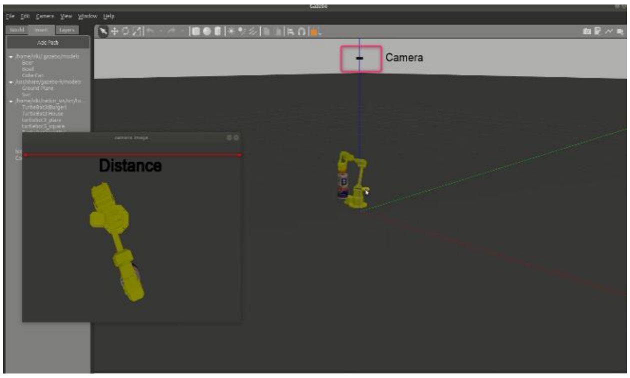

Fig 6 Camera Image

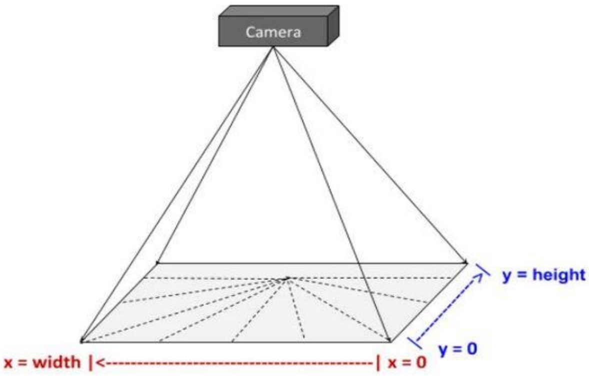

Fig 7 Camera Image Frame Coordinates

C. Image Processing

Main goal of the project is achieved through image processing, where the object's position in the robot workspace is found for performing autonomous pick and place operations. Image subtraction, image thresholding, center of mass method is used for performing image processing. Image subtraction is an easiest way of separating the object or finding the differences of two image.

In image subtraction two image is taken as input which is background image as shown in Fig 8 and another is foreground image or live image as shown in Fig 9, then foreground image or live image is subtracted with background image to obtain difference image which will be an object as shown in Fig 10.

Fig 8 Background Image

Fig 9 Foreground Image

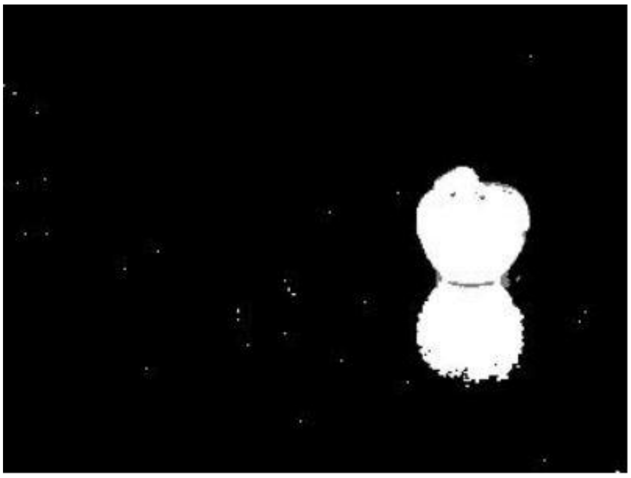

Fig 10 Difference Image

After obtaining the difference image, it is further processed for image thresholding techniques to remove the noises in an image for accurate positioning of objects. Finally center of mass method is followed for finding the object pose in an image, this method gives the exact row and column location of the center point of an object which is present in the given image. Below given figure 7 shows the implementation of center of mass method.

	C1	C2	C3	C4	
R1	0	0	0	0	
R2	0	0	0	0	
R3	0	255	255	255	
R4	0	0	0	0	
ADD	0	255	255	255	
MUL	*1	*2	*3	*4	
Total	**0**	**510**	**765**	**1020**	**2295**

= 2295 / Sum of All Numbers in a matrix = 2295 / 765 = 3
Column Location = 3 * pixel ratio

Fig 11 Example of an Image Matrix

In the above shown Fig 11, consider the rows and columns as an object image matrix where 255 represents pixel value which is considered as object and 0 represents background. For finding the object center point location in the column of an image matrix, all the corresponding column pixel values should be added and to be multiplied with corresponding column number and finally these values are summed up and divided by sum of all numbers in an image matrix, which is 2295/765 = 3 as shown in the above Fig 11.

The obtained column location value can be multiplied with pixel ratio to get x coordinate of an object in meter or centimeter which is found in camera setup process. The same thing can be applied for rows of an image matrix to find row location which is y coordinate. Z is the object height which can be measured manually or assumed to be fixed value. These obtained coordinates are with respect to camera coordinates, which needed to be converted relative to the manipulator frame coordinates, which will be explained below.

D. Camera to Manipulator Coordinate

If we knew the coordinates of the object relative to manipulator frame, we can then use python moveit commander (inverse kinematic plugin) to command the robot to move end effector to that particular object location. So, homogeneous transformation matrix can help us to convert camera coordinates frame relative to the manipulator base frame. It is a 4*4 matrix that allows us to find the position of point in manipulator frame M given the position of a point in camera frame N, which is shown below in the Fig 12, Fig 13 and Fig 14.

$$\begin{bmatrix} x_m \\ y_m \\ z_m \\ 1 \end{bmatrix} = homgen_{mn} \begin{bmatrix} x_n \\ y_n \\ z_n \\ 1 \end{bmatrix}$$

$$\text{homgen\_m\_n} = \begin{bmatrix} \text{rot\_mat\_m\_n} & & \text{disp\_vec\_m\_n} & \\ & & & \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig 12 Matrix (M,N) **Fig 13 Matrix**

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \begin{bmatrix} rotmat_c^0 & rotmat_c^0 & rotmat_c^0 & disp_c^0 \\ rotmat_c^0 & rotmat_c^0 & rotmat_c^0 & disp_c^0 \\ rotmat_c^0 & rotmat_c^0 & rotmat_c^0 & disp_c^0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Fig 14 Homogeneous Matrix

The Fig 15 shown below is the frame of camera (Xc ,Yc) and frame of robot (Xo,Yo).

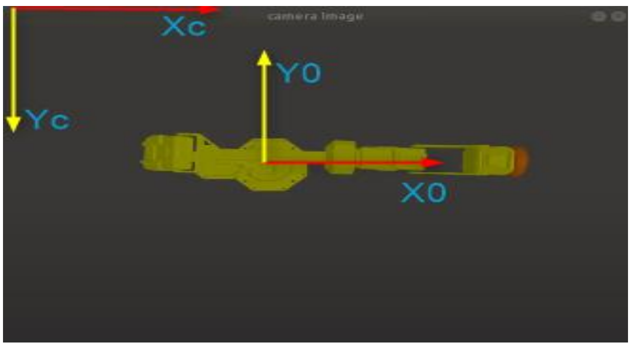

Fig 15 Camera and Robot Frame

To match the robot frame with camera frame Xo must be rotated 180 degrees in clockwise direction, so the rotation matrix for X axis is shown below in Fig 16. And also, the robot frame should be moved to particular distance in both Xo and Yo axis to get matched with camera frame (Xc, Yc) which is known as displacement vector as shown in below image Fig 17. We put them together to get the following homogeneous transformation matrix as shown in Fig 18.

R180 Xo	1	0	0
	0	Cos 180	-Sin 180
	0	Sin 180	Cos 180

Fig 16 Rotation Matrix

Displacement Vector (in Meter)	Xo	-0.862
	Yo	0.645
	Zo	0
	w	1

Fig 17 Displacement Vector

Xo		1	0	0	-0.862		Column Location(Xc)
Yo		0	Cos 180	-Sin 180	0.645		Row Location(Yc)
Zo	=	0	Sin 180	Cos 180	0	*	0.1
Wo		0	0	0	1		1

Fig 18 Final Homogeneous Matrix

We can now convert a point in the camera frame to a point in the robot frame.

V. RESULTS AND DISCUSSION

Following the completion of the arm design and implementation vision system, various missions were assigned to test its capacity of performing assigned task. For this robotic arm has been spawned in the gazebo environment and random object has been placed inside the robot workspace one by one. The object position coordinates has been successfully estimated with accuracy of 0.02 meters in X and 0.018 meters in Y axis. The following below Fig 19, Fig 20, Fig 21 shows the pick and place operation executed successfully without any difficulty.

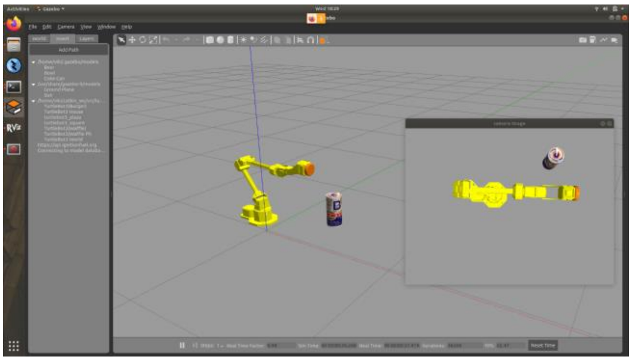

Fig 19 Object Placed in Workspace

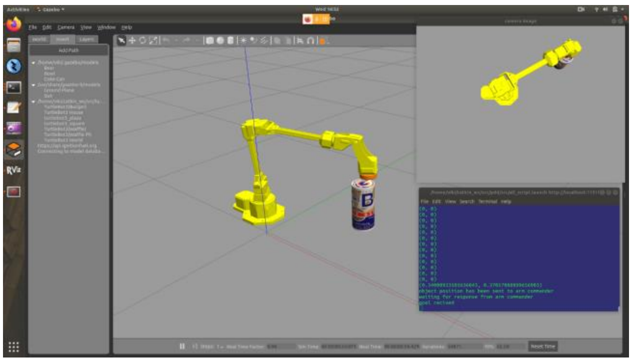

Fig 20 Object Pose Estimated (Pick Pose)

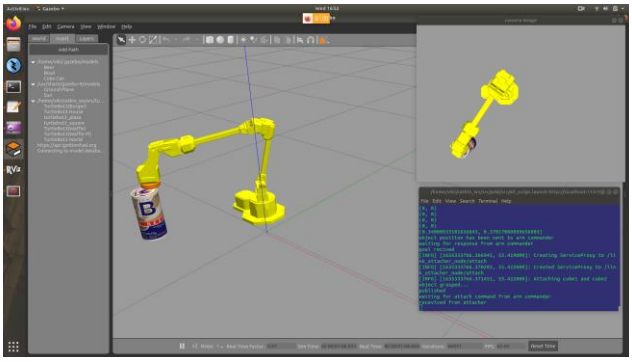

Fig 21 Placing in Predefined Position

And then multiple objects were placed at robot workspace but the center of mass method failed estimate the multiple object position. For this contour detection techniques can be used for multiple object detection.

VI. CONCLUSION

The robotic arm's integration with the vision-based system has been completed successfully. The built program analyses the image collected by the camera and transmits data about the object's location to the Moveit! Command controller. The system has been successfully tested on rectangle, square, and form objects. The system may be extended to accommodate changing environments, such as segregating moving objects in the workspace by understanding the relative speed of the robotic end effector and the moving object.

REFERENCES

1. Taryudi and M. Wang, "3D object pose estimation using stereo vision for object manipulation system.", 2017 International Conference on Applied System Innovation (ICASI).
2. R. K. Megalingam, N et al, "Keyboard-Based Control and Simulation of 6-DOF Robotic Arm Using ROS.", 2018 4th International Conference on Computing Communication and Automation (ICCCA), 2018.
3. Leszek Baranowski et al, “Integration of Vision System and Robotic Arm under ROS.”, 23rd International Conference ENGINEERING MECHANICS, 2017.
4. G.S. Sandhu, C. McGinn, and K. Kelly, “Investigation into a Low-cost Robotic Vision System.”, Conference Proceedings, 2015.
5. S. Hernandez-Mendez et al, "Design and implementation of a robotic arm using ROS and MoveIt.", 2017 IEEE International Autumn Meeting on Power Electronics and Computing .