ISSN: 0974-5823 Vol. 7 No. 6 June, 2022

International Journal of Mechanical Engineering

Pick &Place Task Implementation of a Scara Manipulator via Robot Operating System and Machine Vision

Adem Candemir [1]

[1] Etkin Medical Devices.

Fatih Cemal Can [2]

[2] Mechatronic Engineering, İzmir Katip Çelebi University, Turkey.

Abstract - The need for efficient automation methods in the industry has triggered rapid development in the field of robotics. Yet, despite the several robotic solutions available in the field, the majority of current industrial robots do not use the Robot Operating System (ROS) with computer vision, and they almost all have limitations in terms of autonomously correcting errors during their tasks. Important considerations when developing a robot are the recognition of the objects to be handled and the selection of the appropriate methodology for picking&placing tasks for these objects. This paper develops a method that is flexible enough for object recognition and pick&place applications in the industry. The main purpose of this study is to create a serial manipulator with three degrees of freedom and to move this manipulator from the positions of objects of predetermined color with the help of machine vision to the specified station by using the Ros framework. Our method consists of 3 steps:1-appropriate design in Solidworks,2-object recognition (open source OpenCV), and 3-control algorithm with Ros. The proposed approach can be valuable in the field of robotics and can be potentially applied in multiple sectors such as in the field of agriculture for harvesting etc.

Index Terms— *Machine Vision, MoveIt, Scara Manipulator, ROS.*

INTRODUCTION

Scara is formed from the initials of the words Selective Compliance Assembly Robotic Arm. This robot was developed after 1970 by the Japanese Industrial Consortium and a group of researchers at Yamanashi University in Japan[1]. Scara type robot is a type of robot with very high speeds, best repetition ability, high precision, and accuracy rates. The end effector is designed to move vertically so that the other axes of motion are parallel to the ground. This makes SCARA a good choice for tasks where workpiece access is top-down, such as packaging applications, electronic component typesetting, and 3D printing applications. Scara has 3 degrees of freedom with a serial chain such as (RRR) or (RRP)structure. The simple picture of Scara is given in Fig.1.

Willow Garage of the United States has developed a service robot named PR2 (Personal Robot 2)[2]. The PR2 is the first robot to be equipped with ROS, its two arms are 7-DOF redundant manipulators. Depending on ROS, PR2 can independently complete different types of complex tasks, such as opening doors, etc. This robotics work also drives the application of Robot Operating Systems (ROS) in robotics research. The research on robot arm robots has developed rapidly in recent years. For the collaborative operation of dual-arm robots, motion planning and control are very important research areas[3]. Several studies show the simulation of a robotic arm for manipulating objects [4],[5]. In [6] an indoor object handling application is used. It has a 7 degrees of freedom lightweight robotic arm and a commercial depth sensor. To control the trajectory of the robotic arm, they tested different planners on Moveit! (3rd party ROS node for motion planning). Another study using an integration of Ros and Moveit was carried out by Sergio Hernandez-Mendez et al [7].

This paper aims to complete the design, mechanic analysis, machine vision, and Ros packages of a 3 dof Scara manipulator. Robot Arm (Scara) can be designed with many different types of cad programs. For this study, 3D computer-aided design software Solidworks was used.

Kinematic analysis and dynamic analysis of planar structures have been studied by many researchers with analytical, numerical, and experimental methods according to the type of problem. Most of the publications on kinematic and dynamic behaviors of planar structures are based on classical theory. The kinematic modeling and analysis of a robot manipulator define the relationship between the links that make up this kinematic chain and the joints that are the connecting elements. In order to define this relationship, the Denavit Hartenberg convention or its variations, which is one of the most popular methods and is known to be used by several authors and robot-related publications, is used. The successive screw displacement method is an alternative representation to this classic approach. In this paper, the conventional Denavit Hartenberg method is used for kinematic analysis.

There are two well-known formulations generally used for dynamic analysis in the field of robotics. With this; 1-Recursive Newton-Euler and 2- Euler-Lagrange. The torque equations of the joint motors were found by examining the robot dynamics with dynamic analysis. In the dynamic analysis of the robot, Recursive Newton-Euler was preferred. The effect of gravity is ignored for planar manipulators.

Scara robot is a microcontroller and computer-based mechatronic system that detects the object,after that picks that object from the source location, and places it at the desired location. For detection of an object, a hardware Logitech c270 camera was preferred, and as software, an open-source OpenCV library was used.

Path planning, real-time control, and all the kinematic, dynamic solutions result was simulated with the help of Ros. Path planning was done with the Ros framework to avoid hard transitions during pick and place operations.

In this study, we designed and implemented a 3- DOF manipulator to achieve the movement of objects. In addition, we designed the parts of the manipulator by using Solidworks. The output STL files were used to print the manipulator parts, and they were exported to ROS to create the Unified Robot Description Format (URDF) files for visualization in Rviz, and Gazebo environments. Both the motion planning and the manipulator were visualized in RViz and Gazebo together while the physical manipulator was executing the plan(see Fig.2)used to move and grasp the object.

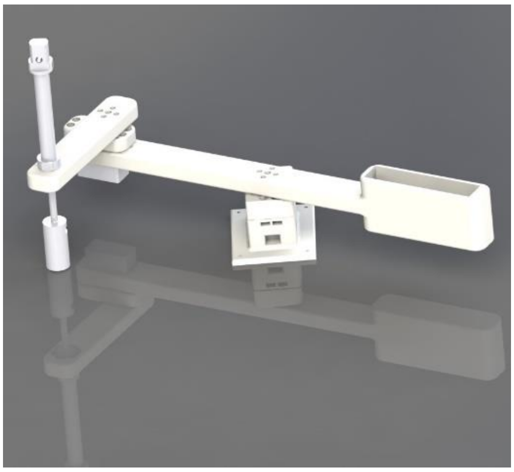


Fig.1 Scara Manipulator

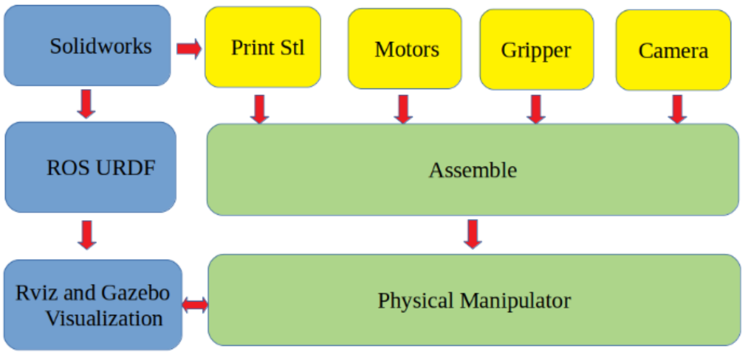


Fig.2 Schematic of Design

DESIGN OF THE MANIPULATOR

Fig. 4a it is shown the manipulator technical drawing, the manipulator has 3 degrees of freedom; the manipulator length with the end-effector included is 25 cm. Fig. 4b shows the assembled manipulator. Licensed software Solidworks Cad program was used to design the serial manipulator which has 3 degrees of freedom. Additionally, with the help of Solidworks export as an urdf plugin, it gives outputs urdf and a part of Ros packages to visualize the physical robot in Ros.

The manipulator components are shown In Fig. 4a. The first one is the manipulator's base link, it allows the connection of other parts of the manipulator to the ground. There is the first link, which is connected to the base link and also where the first actuator is located. The second link is connected to this link, where the second actuator is also located, and the end effector where the single-acting pneumatic cylinder and electromagnet are located at the endpoint. Base link is made of wood, manipulator links are manufactured from PLA 3D printer material. The end-effector in the last part is taken from the market as a single-acting cylinder ready.

Hi-wonder HTS-35H bus servo motors provide movement for each revolute joint of the manipulator. HTS-35H bus servo motors allow us to acquire their speed, temperature, shaft position, and voltage. These motors are daisy-chained together on a communication bus and connected to a PC's USB port by a USB to Hiwonder bus linker adapter.

As a result of the dynamic analysis, the motors with the closest torque values to the appropriate torque values were selected. Since the actuator of the first link moved the entire manipulator, the torque value was higher. In the second link, a lower torque actuator was preferred to reduce the weight and give the proper movement. A single-acting cylinder and electromagnet in the end effector part, the lightest but stroke-sufficient (50 mm) pencil cylinder in the market, and an electromagnet with a carrying capacity of 3 kg were used. In addition, a motor driver adapter, relay for airflow control, power supply to provide power, and Arduino Mega2560 microcontroller to control all of these are used. Logitech c270 is used to recognize objects according to colors and to perform image processing.

MECHANICAL ANALYSIS

The mechanical Analysis of the manipulator consists of three parts such as kinematic, jacobian, and dynamic analysis. In order to start these analyses, we first define the Denavit Hartenberg parameters. This table is shown in TableI.

Table -I. Denavit Hartenberg Parameters

Axes	α_i	a_i	d_i	Q_i	**Initial Q**
1	0	a_1	0	Q_1	0
2	0	a_2	0	Q_2	0

Denavit Hartenberg parameters are considered as twist angle(αi), link length (a_i), offset (d_i), and joint angles(Q_i).Twist angles and offsets are zero for the Scara manipulator. Link lengths are constant (a1=150mm, a2=100mm). Joint angles are variables and they will be controlled using actuator and Ros control packages.

A. Kinematic Analysis

Transformation matrices should be calculated before forward and inverse tasks. The General Transformation formula and matrix are shown in equations (1) and (2), respectively.

$$ {}^{i-1}_{i}A = T(Z,d).T(Z,Q).T(x,a_i).T(x,\alpha) \tag{1} $$

$$ {}^{i-1}_{i}A = \begin{bmatrix} C(Q_i) & -C(\propto_i).S(Q_i) & S(\propto_i).S(Q_i) & a_i.C(Q_i) \\ S(Q_i) & C(\propto_i).C(Q_i) & -S(\propto_i).C(Q_i) & a_i.S(Q_i) \\ 0 & S(\propto_i) & C(\propto_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

where $C(Q_i)$, $S(Q_i)$, $C(\propto_i)$, and $S(\propto_i)$ refers $\cos(Q_i)$, $\sin(Q_i)$, $\cos(\propto_i)$,and $\sin(\propto_i)$,respectively.

Transformation matrix of link1 wrt. base link is written as follows,

$$ {}^{0}_{1}A = \begin{bmatrix} \cos(Q_1) & -\sin(Q_1) & 0 & a_1.\cos(Q_1) \\ \sin(Q_1) & \cos(Q_1) & 0 & a_1.\sin(Q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3} $$

Transformation matrix of link2 wrt. link1 is written as follows,

$$ {}^{1}_{2}A = \begin{bmatrix} \cos(Q_2) & -\sin(Q_2) & 0 & a_2.\cos(Q_2) \\ \sin(Q_2) & \cos(Q_2) & 0 & a_2.\sin(Q_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4} $$

Transformation matrix of link2 wrt. base link is calculated by multiplying matrixes which are given in equations (3),and (4).result matrix is found as follows,

$$ {}^{0}_{2}A = \begin{bmatrix} \cos(Q_{12}) & -\sin(Q_{12}) & 0 & a_2.\cos(Q_{12}) + a_1.\cos(Q_1) \\ \sin(Q_{12}) & \cos(Q_{12}) & 0 & a_2.\sin(Q_{12}) + a_1.\sin(Q_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5} $$

Forward kinematics for a manipulator takes input joint angles and calculates the Cartesian position and orientation of the end effector. By looking at the matrix above, we see that it expresses end-effector position and orientation relative to the base link. From this knowledge, we can write the forward kinematic equations of the manipulator like below.

$$X = a_2.\cos(Q_{12}) + a_1.\cos(Q_1) \quad (6)$$

$$Y = a_2.\sin(Q_{12}) + a_1.\sin(Q_1) \quad (7)$$

Inverse kinematics takes as input the Cartesian end-effector position and orientation and calculates joint angles. Inverse kinematic calculation is shown step by step at below.

Given the final position of the robot P=(Px, Py). Find Q_1, Q_2, for the scara robot. The final ${}^0_2A$ matrix is given above. To find Q_2,we square and sum Px and Py, we can get an expression in Q_2 as shown.

$$P_x^2 + P_y^2 = a_1{}^2 + a_2{}^2 + 2 * a_1 * a_2 * \cos(Q_2) \quad (8)$$

$$\cos(Q_2) = \frac{P_x^2 \mp P_y^2 - a_1{}^2 - a_2{}^2}{2 * a_1 * a_2}, \sin(Q_2) = \sqrt[2]{1 - \cos(Q_2)^2} \quad (9)$$

$$Q_2 = \mp cos^{-1}\left(\frac{P_x^2 + P_y^2 - a_1{}^2 - a_2{}^2}{2 * a_1 * a_2}\right) \quad (10)$$

This is really just the derivation of the law of cosines which we can also use to find Q_2,

$$a_1{}^2 + a_2{}^2 - 2 * a_1 * a_2 * \cos(180 - Q_2) = P_x^2 + P_y^2 \quad (11)$$

$$\cos(180 - Q_2) = \frac{P_x^2 + P_y^2 - a_1{}^2 - a_2{}^2}{-2 * a_1 * a_2} \quad (12)$$

$$\cos(Q_2) = \mp \frac{P_x^2 + P_y^2 - a_1{}^2 - a_2{}^2}{2 * a_1 * a_2} \quad (13)$$

Q_2 from above To solve for Q_1 we wrote two equations in two unknowns cos(Q_1),sin(Q_1)), and we solve for the followings,

$$a_1.\cos(Q_1) + a_2.\cos(Q_{12}) = P_x \quad (14)$$

$$a_1.\sin(Q_1) + a_2.\sin(Q_{12}) = P_y \quad (15)$$

$$Q_1 = atan_2(a_2.\sin(Q_2) * P_x + (a_1 + a_2.\cos(Q_2)) * P_y \;, (a_1 + a_2.\cos(Q_2)) * P_x - a_2.\sin(Q_2) * P_y) \quad (16)$$

B. Singularity Analysis

To find the singularity points of the manipulator, the jacobian matrix is written. If the determinant of a matrix is equal to zero, that matrix cannot be inverted. A Jacobian matrix whose determinant is zero is said to be singular. The Jacobian matrix is singular at some values of the joint variables. At the points where the Jacobian matrix is singular, the robot loses its degree of freedom and at these points where it loses, the matrix cannot be used in robot control. 2*2 jacobian matrix for Scara manipulator is written as follows,

$$J = \begin{bmatrix} -(a_1.\sin(Q_1) + a_2.\sin(Q_{12})) & a_2.\sin(Q_{12}) \\ a_1.\cos(Q_1) + a_2.\cos(Q_{12}) & a_2.\cos(Q_{12}) \end{bmatrix} \quad (17)$$

C. Dynamic Analaysis

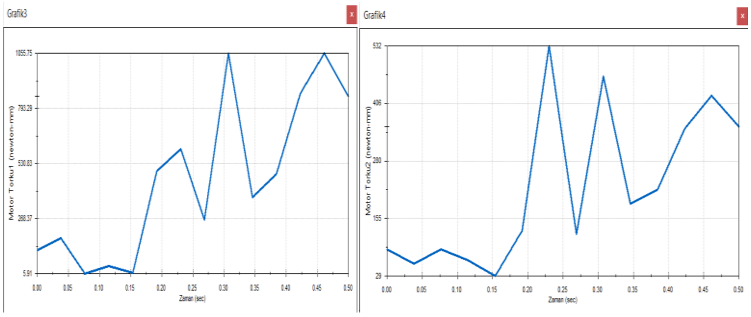
Dynamic analysis of serial manipulators is discussed in the 9th chapter of Lung-Wen Tsai's Robot Analysis book[8]. By using the equations of Newton Euler method in this chapter, torque values suitable for the joints of the Scara robot are found. Actuator selections are made considering these torque values.

The torque equations written respectively to the joints are given below.

$$T1 = \left[\left(\frac{1}{3} * m1 + m2\right) * a_1{}^2 + m2 * a_1 * a_2 * \cos(Q_2) + \frac{1}{3} * m2 * a_2{}^2\right] * \ddot{Q}_1 + \left(\frac{1}{2} * m2 * a_1 * a_2 * \cos(Q_2) + \frac{1}{3} * m2 * a_2{}^2\right) * \ddot{Q}_2 - m2 * a_1 * a_2 * \sin(Q_2) * \left(\dot{Q}_1 * \dot{Q}_2 + \frac{1}{2} * \dot{Q}_2{}^2\right) \quad (18)$$

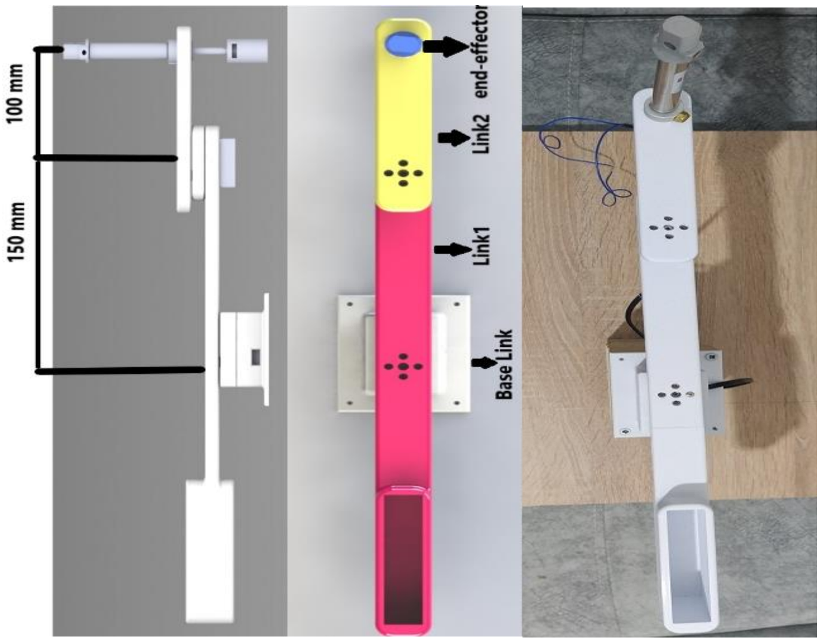
$$T2 = \left(\frac{1}{2} * m2 * a_1 * a_2 * \cos(Q_2) + \frac{1}{3} * m2 * a_2{}^2\right) * \ddot{Q}_1 \quad + \frac{1}{3} * m2 * a_2{}^2 * \quad \ddot{Q}_2 + \frac{1}{2} * m2 * a_1 * a_2 * \sin(Q_2) * \dot{Q}_1{}^2 \qquad (19)$$

The torque of actuator1 and actuator2 is described in equation 18 and equation 19, respectively. The masses of the link1 and link2 are measured using electronic scale. The result is m1= 150gr and m2=175gr.Torque equations are solved by using 500 gram payload. Plots of the torque1 and torque2 are shown in Fig. 3. Torque1 does not exceed the actuator torque that is given 35kg*cm(3.43 N*m) by motor manufacturer.



(a) (b)

Fig 3. Plots a) Torque1, b)Torque2



(a) (b)

Fig 4. a) Solidworks Design b)Assembled Manipulator

SOFTWARE

Ros framework was chosen for the simultaneous operation of the physical robot and the simulation robot. ROS packages that can be used to plan and execute motion trajectories for a manipulator in simulation and real-life. These packages were tested under ROS Melodic and Ubuntu 18.04 and it works perfectly. A ROS system consists of nodes communicating with other nodes using the publish/subscribe messaging model. Many nodes can exist within a ROS system. Nodes are processes that can perform computation. For example; a node receives the image from the camera, a node processes the image, and a node can display the image. In order for these nodes to be aware of and communicate with each other, a ROS Master is needed. ROS Master allows the ROS nodes to find and talk to each other. Nodes communicate by publishing topics and subscribing to these topics.

The next parts explain each of the packages used in the manipulator and can be seen in Fig. 5.

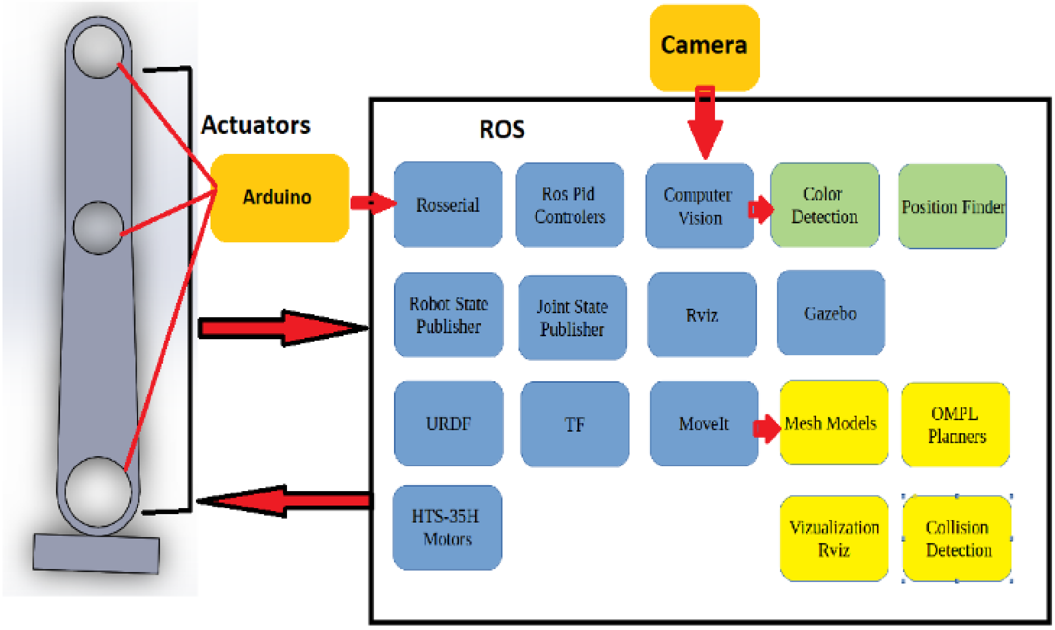


Fig.5 Real Manipulator and and Software in Ros Frame.

A. Ros Packages of Manipulator

The physical robot must be equal to the Kinematic manipulator model. ROS has a robot model that includes important packages that help to build the 3D robot models in the Ros frame. The Unified Robot Description Format (URDF) is used inside this package. The URDF is an XML specification, which describes the manipulator model. This file includes inertia matrix, collision detection, visual properties, transmission, reduction rate, actuator, gazebo physical properties, sensor plugin, and needed controllers.

The URDF model uses five different language elements, which are links, joints, transmission, sensor plugin, and controller plugin. For a link; it is possible to specify the size, shape, color, collision, and inertia matrix of it. The joint is the connection elements of two different links. There are different types of joints, which are continuous, revolute, prismatic, fixed, floating, and planar.

Fig. 6a is shown an image from Rviz displaying the manipulator model. ROS has two packages that can be included inside the model of the robot, the packages are joint_state_publisher and the package robot_state_publisher. These packages make relationships between links to control the manipulator's joints. Basically, the robot_state_publisher publishes the tf transforms of your robot based on its URDF file. You can also publish the joint states using the joint_state_publisher or do it yourself. Either way, the robot_state_publisher uses this information to calculate the forward kinematics of your robot.Fig. 6b shows TF.

B. Path Planning Algorithms with Moveit!

MoveIt is an open-source robotic manipulation platform that allows you to develop complex manipulation applications using ROS. MoveIt has lots of different types of planners. In this paper, the OMPL(Open Motion Planing Library) package is preferred.

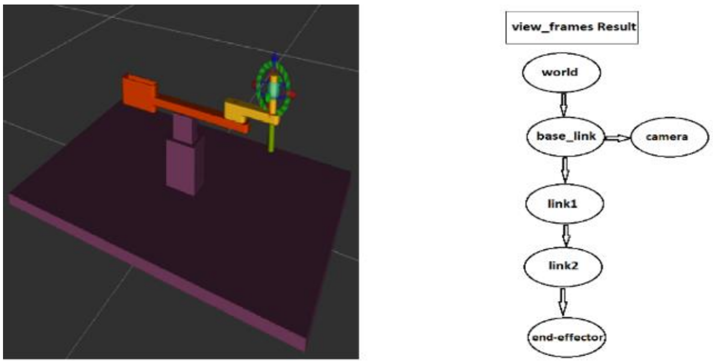


Fig.6 a)Manipulator Displayed in Rviz b) Ros view_frames and rqt_tf_tree Results

MoveIt permits users to interact with motion planners in an easy manner. The basic task of the MoveIt! the system is to provide the necessary paths for the links of a manipulator to put the end effector in a given place. The manipulator uses Moveit plugin to apply kinematics by the KDL solver. These packages can be tested in the gazebo simulation tool and the real manipulator.

VISION-BASED OBJECT RECOGNITION AND PRECISE POSITIONING

Image Processing done by using "opencv" library in Pyhton. This library allows users to do many things with the camera, such as recognizing the shape, color and etc. Here, an image processing technique called Color Detection and Segmentation is used. Color detection is a technique of detecting any color in a given range of HSV (hue saturation value) color space. Image segmentation is the process of partitioning digital image and labeling every pixel, where each pixel having the same label shares certain characteristics. Steps are shown to detect objects of determined color using OpenCV (see Fig.7).

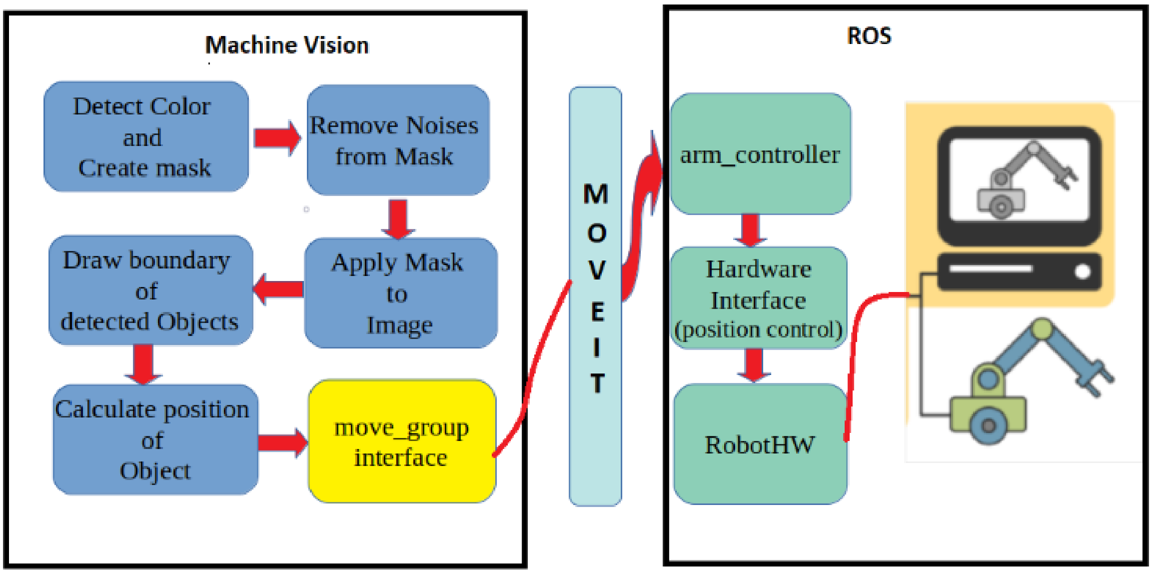


Fig.7 Schematic of Machine Vision and Entire Process

In this study, the blue object was chosen as the predetermined color in the experimental studies. For machine vision HSV (hue, saturation, value) model is used which is an alternative representation of the RGB color model. We write a Python script that gives a chance to users to use sliders to adjust the HSV min and max values (see Fig.8). Adjust the min and max slide bars in the track bars window until you get the desired object alone and the rest is black. We take the corresponding HSV values and used these in our main algorithm.

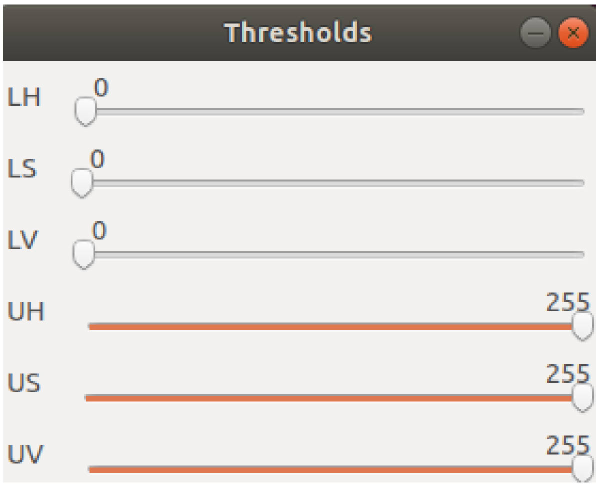


Fig.8 Hsv Trackbars Window

EXPERIMENT

For the physical manipulator and the robot in the simulation environment to behave equally (position and orientation) as a result of kinematic solutions, the transformation tree has been adjusted to be equal.

In the Fig.9 the environment in which the manipulator is located is shown. Since the machine vision algorithm works during the task, the object to be grasped can be left in a random position.

The position of the object is known with the help of machine vision. In Fig.9 the environment in which the manipulator is located as shown. For this work, lots of planning algorithms from OMPL are tried.

While determining the object's position by machine vision and placing it in the target position, it is shown as the physical manipulator Fig.10(1-12)A in the manipulation experiment, while the manipulator (1-12)B in the simulation.

Fig.10(1-6)A shows the real-world experimental environment of the physical manipulator while Fig.10(1-6)B shows the simulation environment which is Rviz. The position of the manipulator in the initial state is shown in the physical manipulator Fig.10(1A) and the manipulator in the simulation is shown in Fig.10(1B). The position of the object with the predetermined color is determined by machine vision and this information is sent to the motion planner as the target position, and the planner moves the manipulator to the target position. The execution of the trajectory can be seen in Fig. 10(1-6)A for the physical manipulator and in Fig.10(1-6)B for the visualized manipulator. When the manipulator comes to the Fig.10 (6A) position, the air is pressed into the pneumatic cylinder and the electromagnet becomes active at the same time. This part is the pre-pick phase. The cylinder in the end effector goes up and down very quickly, picking up the object by magnetizing it. After taking the object, the cylinder is deactivated and the object is taken up with the help of the spring in the single-acting cylinder. The electromagnet is active until the manipulator end-effector reaches the target position which is shown Fig.10(11A). When the end effector reaches the position shown in Fig.10(12A), the object is brought closer to the ground by first giving air to the pneumatic cylinder, and then the electromagnet is deactivated and the object is released.

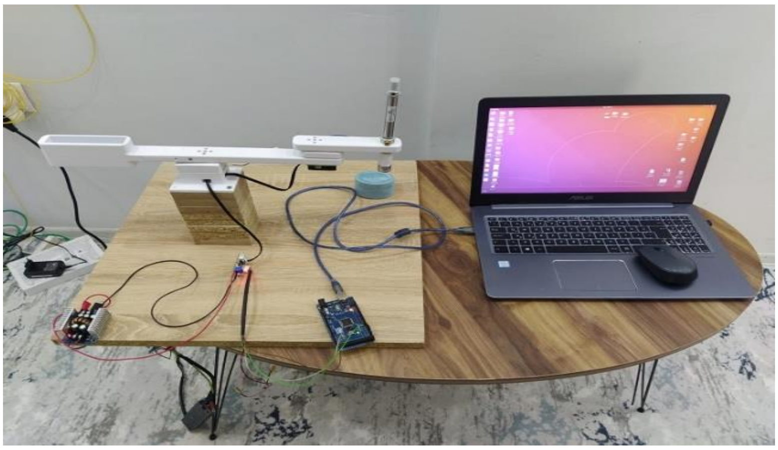


Fig.9 Experimental Setup

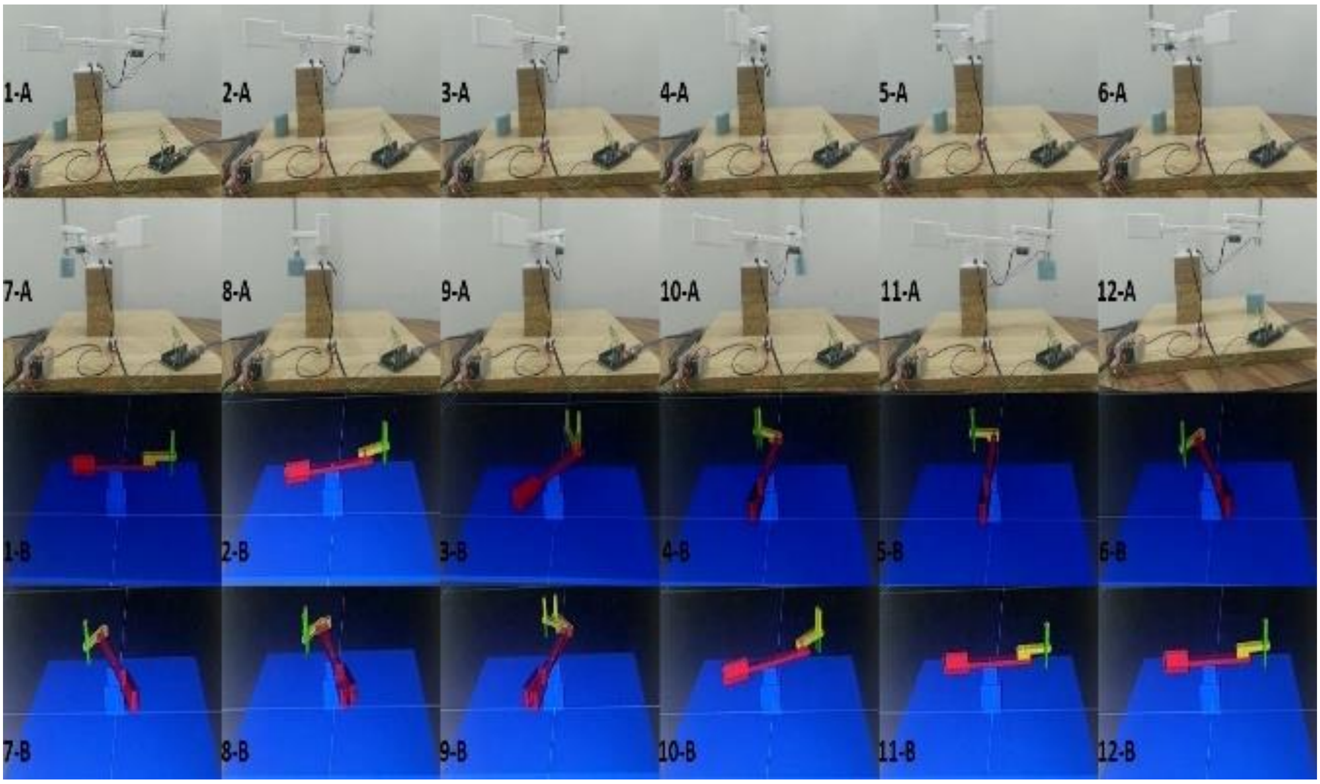


Fig.10 Entire Process of This Study which Mentioned Experiment Section

CONCLUSIONS

In this study, pick&place task implementation of a Scara Manipulator via ROS and machine vision were presented. A Physical manipulator, machine vision, control algorithms, and simulation studies have been combined with the Ros framework and its packages. With the help of this integration, it was possible to visualize the state of the physical manipulator in Rviz during the pick&place task. 3rd party software MoveIt was used to calculate the trajectory and move the object from the position determined by image processing to the predetermined station position. Trajectory tries and pick&place studies performed in the simulation fully matched the joint angles and end-effector position values calculated in the kinematic analysis. The method that we developed in this paper is easy and flexible enough for object recognition and pick&place applications for industry. In future work, this study's outputs can be implemented for real-world problems in the industry field.

REFERENCES

[1] Daş,M.T., Dülger,L.C., "SCARA Tipi Robotun Programlanabilir Mantk Denetleyicisiyle (PLC) Kısmi Hareket Denetimi, ",11.Ulusal Makina Teorisi Sempozyumu., vol.4 , pp. 129-136,September 2003

[2] Cohen B, Chitta S, Likhachev M, et al. "Search-based Planning for Dual-arm Manipulation with Upright Orientation Constraints," IEEE International Conference on Robotics and Automation (ICRA)., pp. 3784–3790, June 2012 .

[3] Yan L, Yang YM, Xu WF, et al. "Dual-arm Coordinated Motion Planning and Compliance Control for Capturing Moving Objects with Large Momentum, " 25th IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). , pp. 7137-7144, January 2019.

[4] W. Qian, Z. Xia, J. Xiong, Y. Gan, Y. Guo, S. Weng, H. Deng, Y. Hu, and J. Zhang, "Manipulation task simulation using ros and gazebo," in Robotics and Biomimetics (ROBIO), IEEE International Conference on. , pp. 2594–2598 , April 2015.

[5] C. Zito, R. Stolkin, M. Kopicki, and J. L. Wyatt, "Two-level rrt planning for robotic push manipulation," IEEE/RSJ International Conference on Intelligent Robots and Systems., pp. 678–685, December 2012.

[6] C. Militaru, A.-D. Mezei, and L. Tamas, "Object handling in cluttered indoor environment with a mobile manipulator," in Automation, Quality and Testing, Robotics (AQTR) 2016 IEEE International Conference on., pp.1-6 , June 2016.

[7] Hernandez-Mendez, S., Maldonado-Mendez, C., Marin-Hernandez, A., Rios-Figueroa, H. V., Vazquez-Leal, H., & Palacios-Hernandez, E. R. "Design and implementation of a robotic arm using ROS and MoveIt!," In 2017 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC),, pp. 1-6, January 2018.