Final Report Guidelines for Introduction to Robotics Lab

Hazika Farooq*, and Syed Mustafa†
Habib University, Pakistan
Email: *hf06870@st.habib.edu.pk, †mm@st.habib.edu.pk

I. Introduction

***Abstract*—This paper outlines the various stages of operations involved in the pick and place robotic arm. It is an automated material handling system is synchronizing the movement of robotic arm to pick the object. Using an RRR robotic arm such that the arm has 5 motors that serve as actuators, a pinch grasper, and a microcontroller board Arbotix-M at the heart bottom of the arm's base. the project was aimed at building a theoretical and physical model using a system design to make the arm pick and place objects based on color detection strategy and a closed loop control from the camera mounted on top. Allover, construction of a Vision-based Pick and Place Robot via first mapping the forward kinematics, setting up the communication between MATLAB and arm, designing a complete motion control system by detemining the manipulator's Jacobian and finally adding a visual sensing to the system by mounting a camera at the top to have the workspace visible.**

***Index Terms*—Phantom X Pincher, Robot Manipulator, Pick and Place**

II. Introduction

Robotic pick and place automation speed up the process of picking parts up and placing them in new location, increasing production rate. In this project we are designing complete model of pick and place robotic arm. The process variables that are to be controlled in this project are movement of arm, position of arm, etc.
The project includes creating a model. The complete process is controlled by using Arbotix-M microcontroller connected to MATLAB. In this model the object is picked by arm automatically via color detection and place it to desired position. So, the overall objective of this report is to map down the process we've gone through while designing a vision-based pick and place Robot system using the Arbotix-M microcontroller, arm with 3 revolute joints and an end effector as gripper mounted with 2 claws, and MATLAB to communicate to the arm.

Robotic Arm-
An artificial developed structure to ease complex tasks used to pick up heavy loads to do complex tasks with perfection and speed. Robotic manipulator, usually programmable, with similar functions to a human arm.

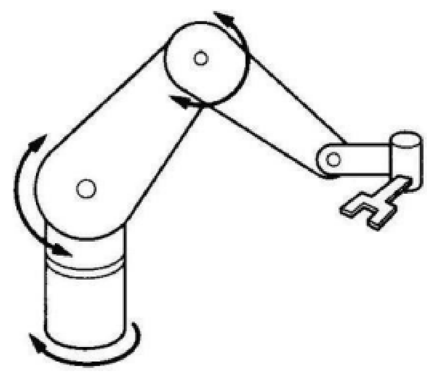


Fig. 1. Caption

Arbotix-M microcontroller
This controller receives higher level instructions from an external processing unit and generates instructions in specific format required for the servomotors (actuators) installed in this arm. In our case, a computer will serve as external processing unit and we'll pass instructions from MATLAB to Arbotix-M.

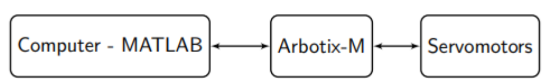


Fig. 2. Data flow between MATLAB and servomotors

Gripper
Sometimes called hand grippers, are primarily used for testing and increasing the strength of the hands; this specific form of grip strength has been called crushing grip, which has been defined as meaning the prime movers are the four fingers, rather than thumb. Robot grippers are the physical interface between a robot arm and work piece. Material handling is one of the benefits So it is important to choose the right type of gripper for your application.
DC motors
A servomotor is a regular DC motor coupled with sensing, to measure the rotational position of the output shaft, and a controller, which uses that position feedback to precisely control the position of the motor. Advanced servos can also control the motor's angular velocity and acceleration. This feedback loop is illustrated in the block diagram.

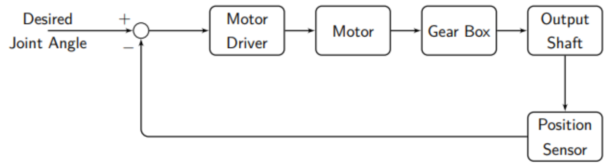


Fig. 3. Block diagram of Servo motor [1]

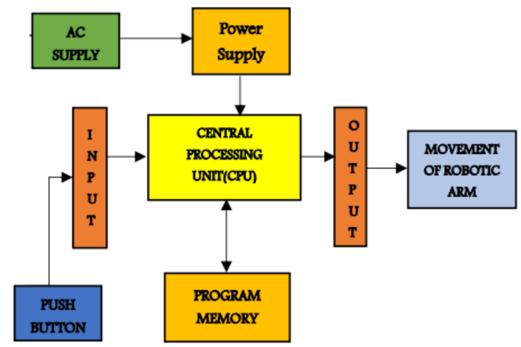
In our project we have used 4 D.C motors for gripper movement, arm up-down movement, rotation of arm. the actuators used here is DYNAMIXEL AX-12A with BAUD RATE: 7,843 [bps] 1 [Mbps] and rpm: 59 [rev/min] at 12V.

Fig. 4. DYNAMIXEL AX-12A Robot Actuator

III. Methodology

The proposed idea of this project is to develop and design Pick and Place Robotic Arm which can be controlled by using MATLAB. The idea was to design an automated system for industrial purpose so that could be able to control it from anywhere and at any time In this section we will provide a complete overview in an organized manner of the methodologies taken throughout the project.

- *Block Diagram*
  The block diagram represents the simplified principle of any project. Here, the block diagram of pick and place robotic arm is as shown in fig. which one simply example the whole working of the robotic arm.



- *Forward Kinematic*
  Using the DH parameters, we can build a model of our manipulator in MATLAB as well. MATLAB treats robots in exactly the same way as we have done in class, i.e. a chain of joints and rigid bodies [2]. We then controlled the robot from MATLAB and physically verify that our forward kinematics computation are correct using the Arbotix library [3]

Fig. 5. Joints of the ARM

- Inverse Kinematics
  the objective of the inverse kinematics mapping is to determine the joint coordinates, given the end-effector position and orientation. This is an absolutely vital step for our pick and place operation. In general, the existence and uniqueness of solution is not certain in inverse kinematics problems. We encountered ambiguity in computations throughout, and we constantly had to choose one solution (based on motor angle limits, continuity, shortest route, obstacle avoidance, etc.). We solved the inverse kinematics problem by obtaining a closed-form expression as they are better for fast and efficient real-time control [4]. We then made functions like findjointAngles and findOptimalSolution with end effector and sum of all joint angles as parameter to the function, we were able to obtained the desired joint angles using the end-effector position.
- *Motion Control System*
  Our next step was to desgin a Motion control system given two locations -pick location *(x1,y1,z1)* and place location *(x2,y2,z2)* from the user. Having received the locations, it will move the arm from its present location to

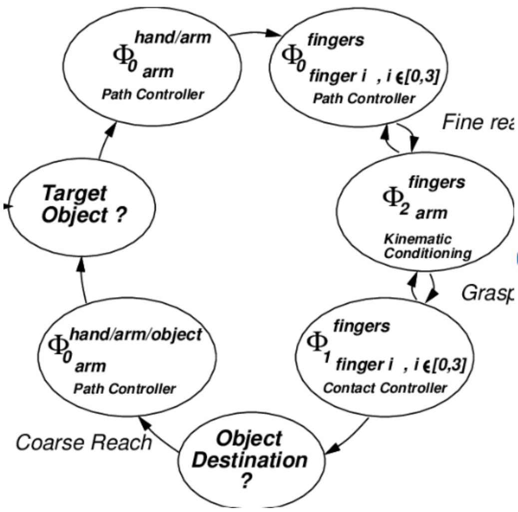


Fig. 6. Finite State Machine

(x1,y1,z1) pick a known object from that location, move to new location *(x2,y2,z2)*, and place it there. To implement this system we had to build certain subsystems, and define a finite state machine. We also determined Manipulator Jacobian which relates the joint velocities to end-effector velocities to bring into account the speed of the manipulator's joints or end-effector. Having determined the Jacobian,we identified the kinematic singularities of this manipulator and observe the behavior of the manipulator near singularities.

- *Visual Sensing*
  We adopted the geometric approach to computer vision for determining the position of the object, We used to use the RGB-D camera, Intel RealSense SR-305 the motion control system developed had to know the location of object to be pick, although, this is not what we want. There we implemented a visual sensing to the system using color detection technique. We placed multiple different colored blocks to the workspace, then implementing the threshold-based techniques to determine the coordinates of pick object. We developed a perception pipeline, the block diagram of vision pipeline is,

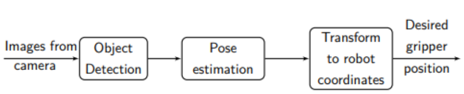


Fig. 7. Block diagram of our vision pipeline [1]

Our preception pipeline helped us detect the object of interest in the camera images, i.e cubes of same colours etc. We estimated the estimated pose of the detected object in camera frame, and then transformed that into robot world frame. the results to which are shown,

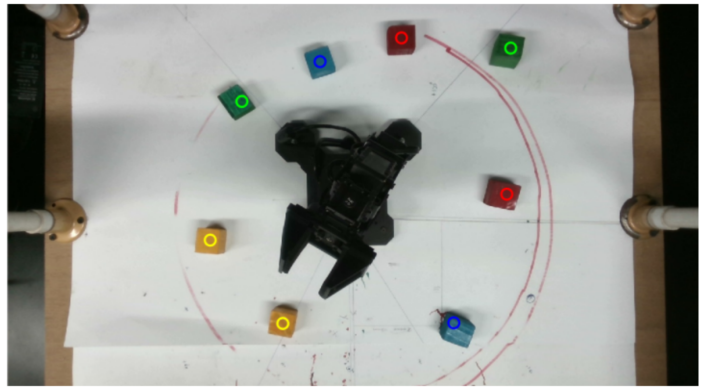


Fig. 8. Object Detection

- *Perception Pipeline*
  Now We complete developing the perception pipeline, and integrate it with the motion control system Using the image processing pipeline above to obtain the coordinates of objects of interest, located in the camera's field of view, in the camera frame. we now transform those coordinates to the robot's base frame, using the perspective projective model of a camera. We know that the coordinates of a point in the image plane are related to the corresponding world coordinates through the camera projection matrix, according to the perspective projective model of a camera. This camera projection matrix includes the intrinsic as well as extrinsic parameters of a camera. Deploying the functions such as determinIntrinsics(), determineExtrinsics() by the data provided by [5] which act as parameters, and depth of the camera, we find the real-world coordinates and make the arm perform vision-based pick and place operation. Now we were able to identify the location of the cube, pick it, and place it at a pre-specified free region in the workspace, using the motion control FSM. We connected the perception pipeline to motion control FSM.
- *Pick and Place*
  Now finally to have our goal accomplished, we will first analysed the singularities as configuration of a manipulator where the rank of the Jacobian is less than its maximum possible rank. At singularities, manipulator loses its abilities to generate velocities in certain directions and consequently its ability to move instantaneously in those directions. finding the conditions at which the Jacobian drops rank was not an easy task, however we leverage form the fact that the Jacobian singularities are an inherent property of the physical manipulator and are unaffected by our choice of the base or end-effector frames. So, we placed the end-effector frame at a position that yields a simpler Jacobian and simplifies our singularity analysis. overall, we had the vision pipeline in place, by making a few change in the pick and place code we made the arm able to perform the tasks given through closed-loop vision feedback.

IV. Results

This section includes quantitative and qualitative results for the pick-and-place operation performance of your system. It can be arranged as follows:

- The arm was able to stack the blocks using the vision pipeline we made and the color detection technique. The code for object detection when building the motion control system stored an array in the pick and place code, and to stack the blocks different pick locations were taken from the image and damping them on the specified placce location. Define what is meant by success for your robotic system and the corresponding metric for performance.
- the other task was accomplsihed using the same object detection cod ein taking the corodinate of different colour blocks and placing the same colred ones together using an array of indices from the color block detected images and the same place location for the same colored blocks.

Sources of Errors

1) we faced alot of hurdles in finding the optimal IK for the arm due to unforeseen arm offsets.
2) During the object detection pipeline via color threshold, the co-ordinates of the middle of the block were difficult to obtain, hence giving a slight offset in the final object coordinates.
3) this offset, when implementing the pick and place operation made it difficult to reach the arm at the right place above the block and making it difficult to grip it perfectly.
4) While implementing th evision based pipeline and merging it with the pick and place code, the optimal solution for Inverse Kinematic was not giving appropriate result therefore, making the logic nullify as a whole and making it difficult to implement the vision based pieline.

V. Conclusion and Future Work

to sum up, we were able to achieve the pick and place operation via implementing the vision and motion pipeline for th erobotic arm. Although, there is still room for the improvement of the vision pipeline, as it has errors. Also, we can implement force control on the motor using a computer torque model or any other approach. Further exploration can be done in trajectory planning and making it completely independent of random colurs and object sizes.

Acknowledgments

We would like to thank Dr. Basit Memon and Mr. Muhammad Ali Mustansir for their continous help and spport throughout all the labs and for pushing us till the end and having us accomplish the end results.

References

[1] Introduction to Robotics Lab Handbook
[2] https://www.mathworks.com/help/robotics/ug/rigid-body-tree-robot-model. html
[3] https://petercorke.com/toolboxes/robotics-toolbox/
[4] Spong, Mark W., Seth Hutchinson, and Mathukumalli Vidyasagar. Robot modeling and control. John Wiley Sons, 2020.
[5] Intel RealSense Depth Camera SR300 Series Product Family Datasheet
[6] H. Sira-Ramirez, "On the sliding mode control of nonlinear systems," *Syst. Control Lett.*, vol. 19, pp. 303–312, 1992.