B.Tech

Project Report

Semester (IV)

Control of UR5 Robotic Arm To Follow a Moving Object

Students:

Mukesh Sharma

(B16ME015)

Bhaskar Vijay

(B16ME005)

Mentor:

Dr. Suril V. Shah

Assistant Professor

Department of Mechanical Engineering

Indian Institute of Technology, Jodhpur

Department of Mechanical Engineering

20^{th} April 2018

Acknowledgement

We are very grateful to Assistant Professor Dr Suril V. Shah, our faculty mentor, who have been enthusiastic and helpful in guiding us through this project. We avail this opportunity to acknowledge the support of our Head of Department, Dr Kaushal Desai for having a belief in us. We thank our seniors – Sudhir Yadav, Kartik and Subham for acquainting us with robotic Laboratory equipments. At last we thank our institute for providing us the opportunity to take forward this project.

1 Abstract

The project is about UR-5 Robotic Arm mounted with eye in hand camera to follow a moving object using visual servoing technique. Initially, we started with understanding of Teach pendant and UR-script. Then after learning safe use of UR-5 through Teach pendant and UR-script we established ROS Environment. We were working on Forward and Inverse Kinematics of robotic arm. We worked on detecting object (ball) in a image along with its center. We also worked on P3DX controlling wirelessly. The code for detecting the object center and diameter of ball in image was written in Matlab for understanding. ROS publisher and subscriber were made to take data from UR-5 and to publish data.

2 Introduction

Since 18th century when industrial revolution started, automation has been the major force when trying to rationalise the production process and in this process there came an important role that has been played by robotic manipulators. Actually, robotic manipulator is any mechanical device that can be programmed to perform a wide variety of applications. Although the Universal Robots are safe to work with there still remains the problem with the ROS drivers on one side while operating UR-5 and using visual servoing technique with monocular camera to determine depth. Here one of the traditional method of determining the depth based on the simple rule that the area of the object projected decreases with the increase in distance has been used to obtain the depth using eye in hand camera. Along with modern universal drivers has been used for operation of UR-5.

2.1 Objective

The aim of the project is to control UR-5 Robotic manipulator to follow a moving target using visual servoing technique with eye in hand camera.

2.2 Motivation

In current scenario, there is widespread need of performing activities with versatility, repeatability and efficiency in factories and industries that goes beyond the human reach. To carry such tasks there is need of device which can do do manipulations to complete desired tasks. Some of these activities includes painting, carrying load from origination to destination, pick and place of objects, automated welding, sorting of objects, follow a moving target within workspace, etc. There came the motivation to work on UR-5 robotics manipulator because of its widespread application in industrial and research field.

2.3 UR-5 Manipulator Arm

UR-5 is a robotics manipulator capable of manipulating or moving object 3 dimensions. It consists of six joints all Revolute in nature and therefore is capable of changing all six degree of freedom of an object. Among six revolute joints three first three joints counted from the base are responsible for the position of the end effector and rest three are responsible for the orientation of the end effector. The payload of this robot is 5 kilogram. It is safe to work with UR-5 as it has very interesting cage free mode or we can say collaborative i.e it stops as it touches the human in this mode or we can say does not harm the human.

2.3.1 Specification Of Arm

UR-5 is manufactured by manufactured by the company Universal Robots a Denmark based company which has provided the teach pendant and the following specifications of the UR-5 arm.

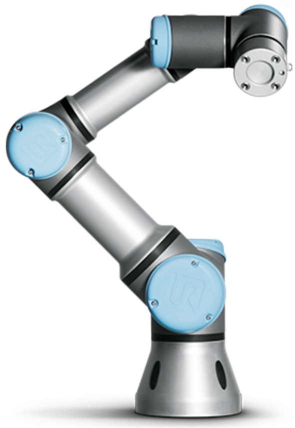

Figure 1: UR-5 Robotic Arm

2.3.2 Teach pendant

Teach pendant is a controller + receiver device used to control ur5. It is an easy to use interface for beginners to learn and understand the features of robot and can be used to program(hardcore) the robot manually.

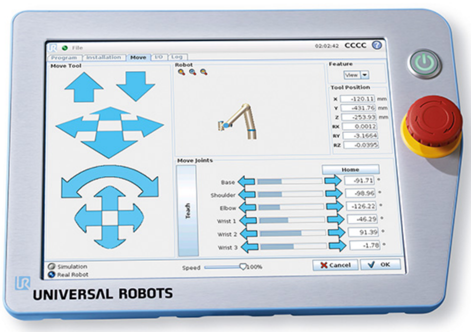

Figure 2: Teach Pendant

2.3.3 Work-space of the Robot

The work space of the UR5 robot extends 850mm from the base joint. It is important to consider the cylindrical volume directly above and directly below the robot base when a mounting place for the robot is chosen. Moving the tool close to the cylindrical volume should be avoided if possible, because it causes the joints to move fast even though the tool is moving slowly, causing the robot to work inefciently and the conduction of the risk assessment to be difcult.

2.4 P3DX Mobile Robot

The Pioneer 3DX (P3DX) is an advanced research robot that is equipped with encoders, high precision sensors like sonars and also on board central processing unit which can be easily used with a monitor, mouse and keyboard and communicates via both ethernet and wirelessly. The P3DX can be interface through ROS drivers and the robot is good for implementing roscode while learning.

2.5 Visual Servoing for object detection

Visual servoing is a vision based robot control technique in which information is extracted from the sensors used and this information is used to control the robot. Usually the control based on visual servoing technique is closed loop feedback control which can be based on any of these three types namely: Image-based, Position/pose-based,Hybrid approach and all these three has different pros and cons.

3 Work done and Result

To reach the final goal of this project the following activities were done in the sequence mentioned below.

3.1 Literature Review

In our opinion the effective way to start this project was going through the literature available about UR-5. So, we stick to our opinion and studied the user manual available for the UR-5 robotic arm in due course of time we came to know about the safety rules, hardware specifications, joint limits, teach pendant and UR-scripts, ROS drivers and the area of applications. And through the study of books: Robot Modeling and Control First Edition by Mark W. Spong, Seth Hutchinson, and M. Vidyasagar, Springer Handbook of Robotics Bruno Siciliano, Oussama Khatib (Eds) and papers: Analytic Inverse Kinematics for the Universal Robots UR-5/UR-10 Arms Kelsey P. Hawkins and Analytic Inverse Kinematics for the Universal Robots UR-5/UR-10 Arms Kelsey P. Hawkin.

3.2 ROS

Robot Operating System is a specially designed meta operating system to control multiple sensors and actuators at the same time. It allows to run robots in simulation. To be able to use ROS one should have good understanding of LINUX terminal commands and also some basic knowledge of PYTHON /C++ language.

For this purpose we intalled Linux operating system and there install ROS kinetic16.04 and did environment setup for ROS. Here we learnt to use ROS meta operating system and simultaneously get readings from different sensors and give output to actuators using multiple terminals.
Then we started lookin for ROS drivers required for starting the UR-5. We got the ROS drivers for the UR-5 but all those we old version and there came problem for publishing and subscribing data at good frequency.So we looked for newer drivers and found one but they needed to be modified for the robot. For the same we build the modern UR driver and modified the files as per our UR-5 requirement.

Earlier there was problem in publishing velocities of joints through laptop use but after correcting the the ROS drivers and making necessary changes we are able to operate UR-5 withour own laptop.

3.3 Networking

3.3.1 P3DX

P3DX contain an inboard computer that can be accessed using a monitor, keyboard and a mouse.First we used the inboard computer and run some codes like subscribing the ros topic and publishing the velocity.But when the P3DX is connected to monitor, it cannot move freely as the moitor is connected to power source. So we used the wireless networking method in which we connect the P3DX and our system(Desktop or Laptop) over the same network.Now, run ros master on our system and export the Internet Protocol(IP) address which is accepted by P3DX as ros master.Once, this procedure is completed, P3DX can be disconnected form the monitor, keyboards and mouse and can be freely controlled by the system publishing the commands.

3.3.2 Configuring UR-5 Hardware

Here UR-5 and system is connected via ethernet cable and with static address method IP address is manually assigned to UR-5 through the teach pendant and other required parameters such as Sub-net Mask and DNS servers are assigned accordingly. Next we created a connection in our system with IP address same as that of UR-5 except the last digit as two systems cannot have the same IP address. After configuring all these the UR-5 robot can be operated through the system or any other system provided they have all the ROS drivers required. If we want to get started quickly with a network router and don't mind the possibility of UR-5's IP address changing we can select DHCP network method but here changing IP address of the robot creates problem.

3.4 Control and lab experiment

3.4.1 P3DX

Here we learned the basics of P3DX so that we can make program to control it.Here this simple differential drive system can take velocity both angular and linear from the user. Programs to control the P3DX using arrow keys of the keyboards was made. Similarly programs to make a circle with desired speed of user, to rotate clockwise, rotate anticlockwise, to trace square of desired length was made. Also to a camera was mounted on the P3DX and it was made folow the ball center using velocity control technique. All these were done so that we have enough confident to make programs for UR-5 as the system is complicated and much expensive.

Control

To make the P3DX follow a moving object detecting the ball center is based on velocity control. The velocity of ball center moving is calculated in the global frame mounted on the top surface of P3DX using image jacobian, and simple proportional controller is used in which the velocity of P3DX is proportional to the error between ball center velocity and P3DX velocity.

3.4.2 UR-5

Forward Kinematics

In case of forward kinematics we possess the joint angles of all the joints, and our task is to determine the position of the end effector or tool center point called (TCP). Initially we assign the local cartesian coordinate system/frame to each of the joints and world coordinate system to the base and obtain the transformation matrix for transforming points from one frame to another. This transformation matrix is obtained using the Denavit-Hartenberg parameters.

Using the definition of the transformation matrix of a robotic arm from [1], [2], the transformation matrix from the base to the end effector is in the form of:

i	a_i	α_i	d_i	θ_i
0	0	0	-	-
1	0	$\frac{\pi}{2}$	0.08916	θ_1
2	0.425	0	0	θ_2
3	0.39225	0	0	θ_3
4	0	$\frac{\pi}{2}$	0.10915	θ_4
5	0	$-\frac{\pi}{2}$	0.09456	θ_5
6	-	-	0.0823	θ_6

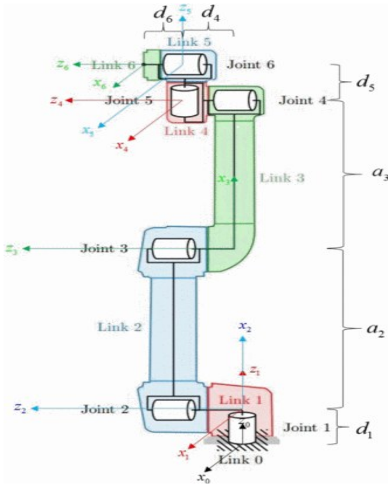

Figure 3: DH Parameters(left), Forward Kinematics(right)

Then the forward kinematics of the robot position would easily be obtained from the fourth column of the 0T n as:

$$p_x = d_5 c_1 s_{234} + d_4 s_1 - d_6 c_1 c_{234} + a_2 c_1 c_2 + d_6 c_5 s_1 + a_3 c_1 c_2 c_3 - a_3 c_1 s_2 s_3$$
$$p_y = d_5 s_1 s_{234} - d_4 c_1 - d_6 s_1 c_{234} - d_6 c_1 c_5 + a_2 c_2 s_1 + a_3 c_2 c_3 s_1 - a_3 s_1 s_2 s_3$$
$$p_z = d_1 - d_6 s_{234} s_5 + a_3 s_{23} + a_2 s_2 - d_5 c_{234}$$

where s_{234} represents the $\sin(\theta_2 + \theta_3 + \theta_4) and c_{234}$ for the $\cos(\theta_2 + \theta_3 + \theta_4) of the same.$

Inverse kinematics

The inverse kinematics problem of the UR-5 is complicated as the three wrist joints are not concurrent and the decouplling technique cannot be used. Since wee need to follow the ball center only, orientation is not taken into account and inverse kinematics of only first three joint need to be solved which is quite simple and a vector is added to the end point of the third link which gives the coordinate of TCP.

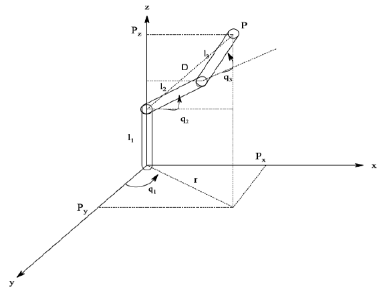

Figure 4: For inverse kinematics

$\theta_2 = atan2(sin(\theta_2), cos(\theta_2))$
$\theta_1 = atan2((k_1 y_n - k_2 x_n),(k_1 x_n - k_2 y_n))$
$\theta_3 = \phi - (\theta_1 + \theta_2)$
where, $k_1 = l_1 + l_2\cos(\theta_2)$, $k_2 = l_2$, $sin(\theta_2)cos(\theta_2) = (x^2 + y^2 - l_1^2 - l_2^2)/2l_1 l_2$, $sin(\theta_2) = p + (1 - \cos^2\theta_2)$,
$x_n = x - l_3\cos(\phi)$ and $y_n = y - l_3\sin(\phi)$

Control

To follow the ball center using camera mounted on the arm, the ball center was acquired and similarly the joint angles of all the six motors were acquired.Now we had the coordinates of the tool center point(TCP) gained using forward kinematics and ball center in world frame obtained using the transformation matrix. Using the inverse kinematics for ball center in world frame we obtained the joint angles required. Now we have the current joint angles and the desired joint angles. Then we use proportional controller in which the error is the difference between the desired joint angles and current joint angles.Based on this error the joint velocity is provided.

error = (desired joint angles) - (current joint angles)
velocity = (Proportionality constant) * error / (desired time)

where desired time is the time in which we want the TCP to reach the ball center.

3.5 Detecting Object center

3.5.1 Detecting ball

To detect an object using camera we will take images at regular interval. The image we get is in RGB format but we need grayscale image for further operation. Since colored image is a 3 dimensional matrix.
Image size = (640*480*3);
Red scale image = (640*480*1)
Green scale image = (640*480*2)
Blue scale image = (640*480*3)
An image consist of pixels building block of an image. Every pixel has a value from 0 to 255 according to color every image Red Scale , Green Scale, Blue Scale has values between 0 to 255.

To convert the image to grayscale we can do the following operation:
Grayscale image = (Red scale image + Green scale image + Blue scale image)/3;

Now you have a grayscale image if we do not have the idea of color of object we can plot histogram of the grayscale image and get the range of values which the object shows in pixel.

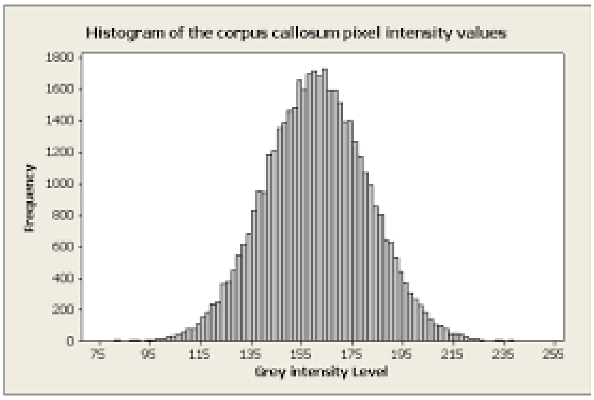

Figure 5: Histogram

Now we need to binarize the image . As we know binary means 0 or 1 so we convert the value of image pixel to either 0 or 255 from [0 ,255] (close interval). Let the pixel value for the ball ranges in X1 to X2 then if we convert all values greater than X1 and less than X2 to 255 and every one else to 0 then we get a binary image. Now the image we get is only black and white in which the the object is shown white And background will be black

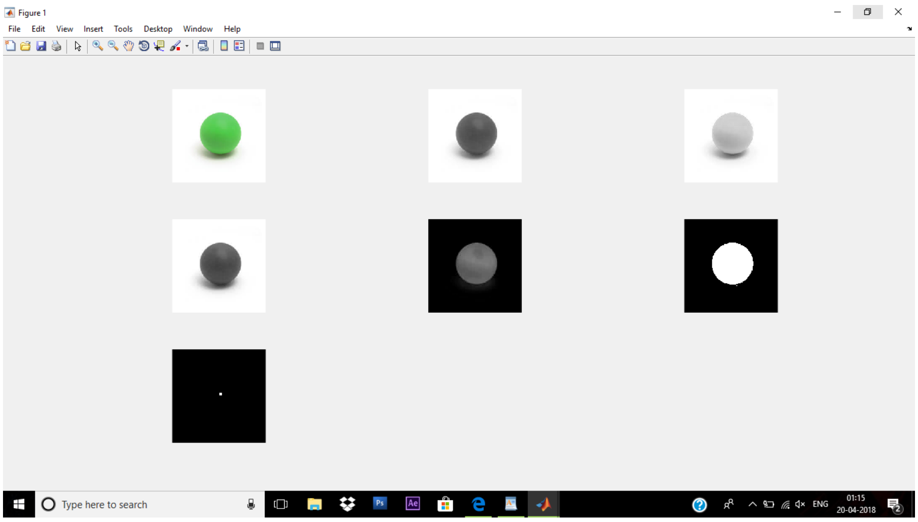

Figure 6: Ball in different color scale with it center

3.5.2 Ball Center Coordinate

The image coordinate are the coordinate of pixel on which the ball center lies. We can get center of ball using basic image processing as discussed in 3.4.1. Using simple for loop we can get center of ball in image coordinate. The image coordinates gives you an idea about location of ball in image plane qualitatively.

Ball center coordinate in image frame can be easily found using a for loop We can also find diameter of ball in image coordinate by finding the maximum length of axis for which the pixels have 255 value.This diameter of course change with change in distance of ball from camera.

3.5.3 Global Coordinate

Global coordinates are the coordinates of object with respect to camera as origin. We can find the center of object in global coordinates by exploiting some basic properties of optics and basic trigonometry.

Let diameter of ball in image coordinate(D_1 and D_2) and D(actual diameter of the ball) are given then:
H_1 = X_1*tan(θ)
H$_2$ = X_2*tan(θ)

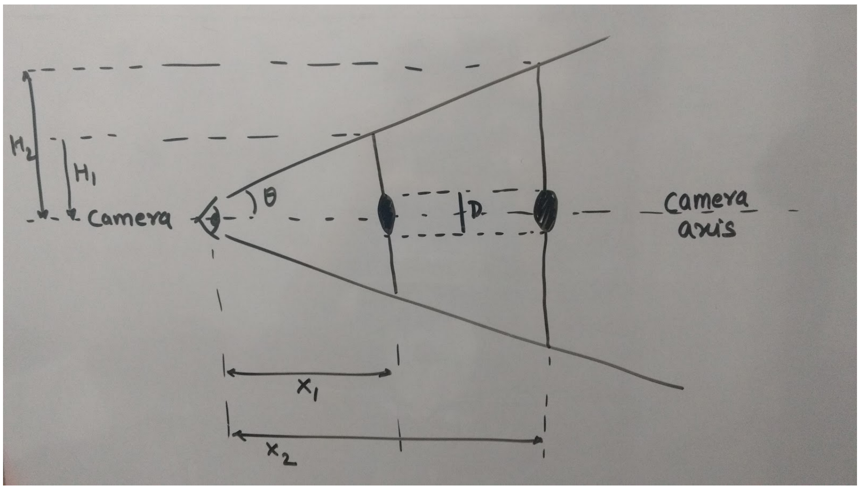

Figure 7: Basic properties of optics and basic trigonometry

$H_1/H_2 = X_1/X_2 = D_2/D_1$
$X_2 = D_1 * X_1/D_2$
Where X_1 is the distance of ball center from camera.

3.5.4 Extending ball center detection to face center detection

To detect the human presence by the arm we used face tracking with the help of opencv. It was just the extension of our work. A simple code in python with help of opencv helped in finding center of bounding box for human face.

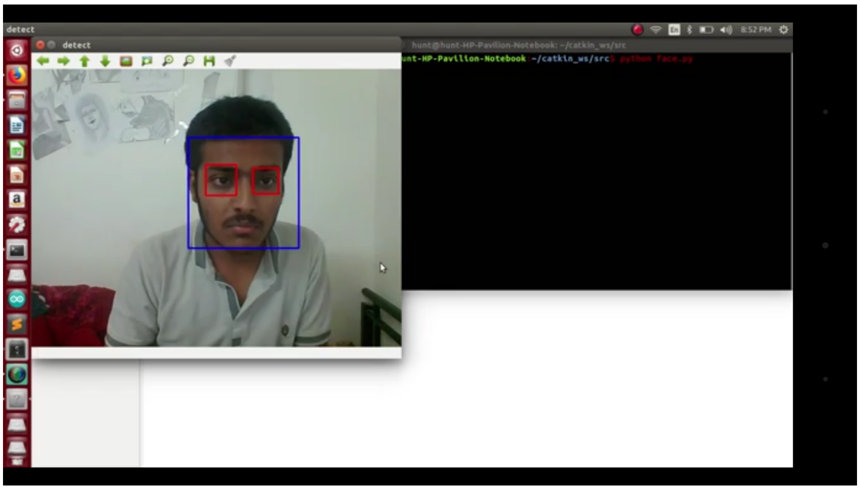

Figure 8: Face center detection

4 Result and Discussion

Finally we are able to subscribe topics and extract the useful data and use to perform manipulations and publish message on the topic. The TCP is able to follow the moving ball center point with the control used over here. The TCP followed the moving ball center with different accelerations and velocities due to the control design. Also we were able to make P3DX follow the moving ball center smoothly.

5 Conclusion and Future Work

Following a moving object using eye in hand camera mounted on the UR-5 TCP has great importance in industry and in research field. Actually this task is quite tough because there comes acceleration into picture when we give some velocity to joints and it accelerates until the limit exceeds. Also there is indescribable singularities which comes in joint space. Morever the control design over here is unable to publish smooth velocities because we have made a different approach that is using position control we provide velocity.

As a part of our future work, we seek to test new control method based on velocity control in which we intend to use jacobian. Also we seek to investigate the singularity region in joint space as our objective is to make the UR-5 TCP movement robust by escaping the singularity space.

References

[1] Robot Modeling and Control First Edition Mark W. Spong, Seth Hutchinson, and M. Vidyasagar [2] Kinematic and Dynamic Modelling of UR5 Manipulator Parham M. Kebria 1 , Saba Al-wais, Hamid Abdi, and Saeid Nahavand [3] Analytic Inverse Kinematics for the Universal Robots UR-5/UR-10 Arms Kelsey P. Hawkins [4]Springer Handbook of Robotics Bruno Siciliano, Oussama Khatib (Eds.)