# EE366L/CE366L

# Introduction to Robotics Lab

# Lab Report 02


## Members

Khuzaima Ali Khan (ka06466)

Syed Jahania Shah (ss06864)

### Teachers

Dr Basit Memon

### Research Assistant

Ahmed Ali Mustansir

Haseeb Khan

# Chapter 03: Sensing and actuation for the arm

## Task 3.1:

**Task 3.1** Understanding functioning of system (10 points)

(a) [*] Identify and list all the sensors and actuators in our complete robotic system illustrated in Figure 3.1.

### Number of Sensors: 02

In our complete robotic system, we have two sensors. One is camera, which is used to input the image of the object, we are targetting.

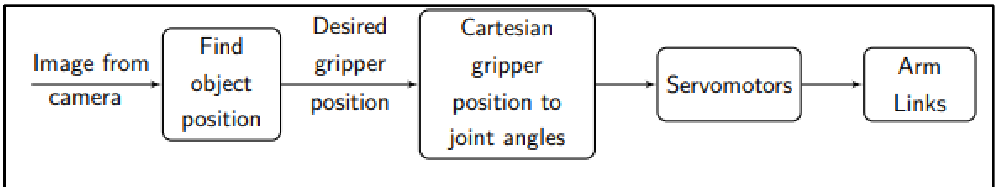



Figure 3.1: Block diagram of our vision-based pick and place robot

Another, we have position sensor which is used to measure the rotational position of the output shaft of the servo motor, we are using in Figure 3.1 above.

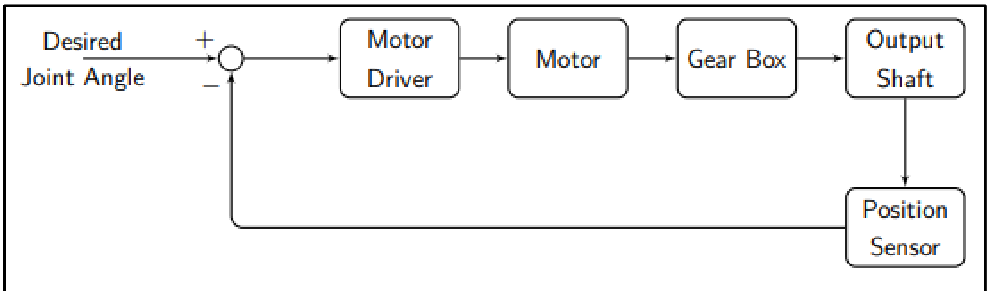



Figure 3.2: Block diagram of a servomotor

Figure 3.2 shows the inside mechanism of the servo motor, which is the closed loop feedback control system.

### Actuator: Motor

In our robotic system, Servo motor is working as an actuator in our control system. More specifically, Motor in figure 3.2 is the actuator of the system that is working inside the servomotor.

(b) [*] Figure 3.3 suggests that a potentiometer is being used as the shaft position sensor. Research and describe how can a potentiometer be used for this purpose.

### Explanation:

A potentiometer can be used as a shaft position sensor in a servo motor by measuring the voltage across the potentiometer's terminals and using that information to determine the position of the servo motor's shaft. This is typically done by connecting the potentiometer's terminals to the servo motor's control circuit, and using the voltage measurement to control the position of the servo motor's shaft via pulse width modulation. The potentiometer's position can be correlated to the angle of the servo motor's shaft by calibration. The servo motor's control circuit uses this information to drive the motor to the desired position.

In robotics arm, the potentiometer is mounted on the servo motor's shaft so that as the shaft rotates, the potentiometer's resistance also changes. This change in resistance can then be used to determine the position of the shaft and thus the position of the robotic arm. This method is used and prefered due to cost effectivity.

Additionally, a robotic arm may use multiple potentiometers, one for each joint, to sense the position of each segment of the arm. This allows for a more accurate measurement of the arm's position.

## Task 3.2:

**Task 3.2** Motor Specifications (10 points)

This task is about familiarizing ourselves with the Dynamixel reference manual (`https://emanual.robotis.com/docs/en/dxl/ax/ax-12a/`) and some relevant specifications included in it.

(a) Find the angle rotation limits, resolution (see the definition below), speed limit, and torque limit[a] of AX-12A servo.

| | |
|---|---|
| **Angle Rotation Limits** | **0 to 300** (degree) (See attached figure 3.2.1) |
| **Resolution** | **0.29** (degree) (See definition below) |
| **Speed Limit** | **0 to 144** (rpm) (0 to 1023 units where 1 unit is equal to 0.111rpm) |
| **Torque Limit** | **0 to 1.5** (N.m) (Stall torque) |

## Angle Rotation Limits:

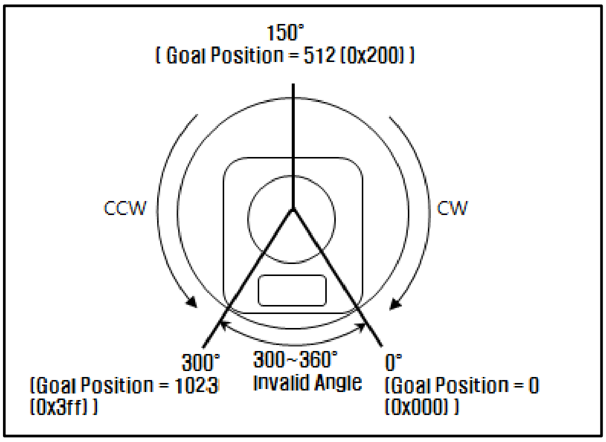



**Figure 3.2.1:**

### Resolution:

This specification represents the smallest incremental joint motion that can be produced and sensed by the robot. It means if we input angle less than 0.29 degree, the joint motion will not be executed. A robotic system's resolution depends on its sensing capabilities.

### Speed Limit:

$$\mathbf{1023}\ \boldsymbol{units} * \mathbf{0.111}\ \boldsymbol{rpm} = \mathbf{113.553}\ \approx \mathbf{114\ rpm}$$

> (b) [*] Will this motor resolution limit the possible Cartesian resolution of the end-effector? If yes, why?

### Explanation:

YES! The motor resolution will limit the possible cartesian resolution of the end effector. The resolution of a servo motor is an important factor that can limit the precision of a robotic arm's movement. It is typically measured in terms of the smallest angular increment that the servo motor can move its shaft. A high-resolution servo motor can move its shaft with a high degree of precision, allowing for precise movement of the robotic arm's joints, which in turn leads to a high precision of the end effector's movement in Cartesian space. On the other hand, a low-resolution servo motor will not be able to move its shaft as precisely, which will limit the precision of the end effector's movement.

### Task 3.3:

> **Task 3.3** Getting to know the camera (0 points)
>
> Download `Intel RealSense Viewer` tool from Canvas to verify that your camera is working and to explore the various parameters. If you enable both the RGB and depth streams, you shall see live videos for both where the depth stream represents different depths in different colors. Hover over any pixel in the depth image and you shall see the depth value in meters at the bottom. Explore the different processing filters available for each stream. Details of filters are provided at [6].

The software Intel Realsense Viewer has been installed on the computer, and has been explored as well.

### Task 3.4:

> **Task 3.4** Image Manipulation in MATLAB (60 points)
>
> Complete at least modules 1-4 of the 'Image Processing OnRamp' (`https://matlabacademy.mathworks.com/details/image-processing-onramp/imageprocessing`) and provide your 'Progress Report' as submission for this task.

The first four modules of Image Processing OnRamp course has been completed and progress reports of both members have been attached.

## Task 3.5:

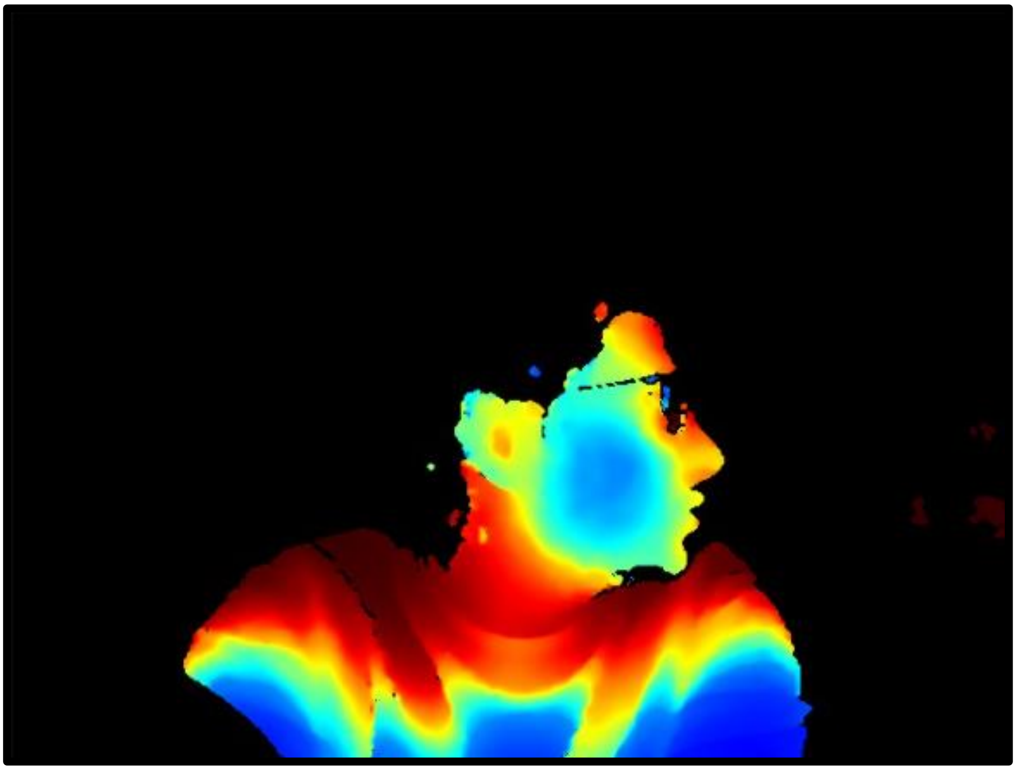
| Task 3.5 | Extract color and depth images (10 points) |
| --- | --- |

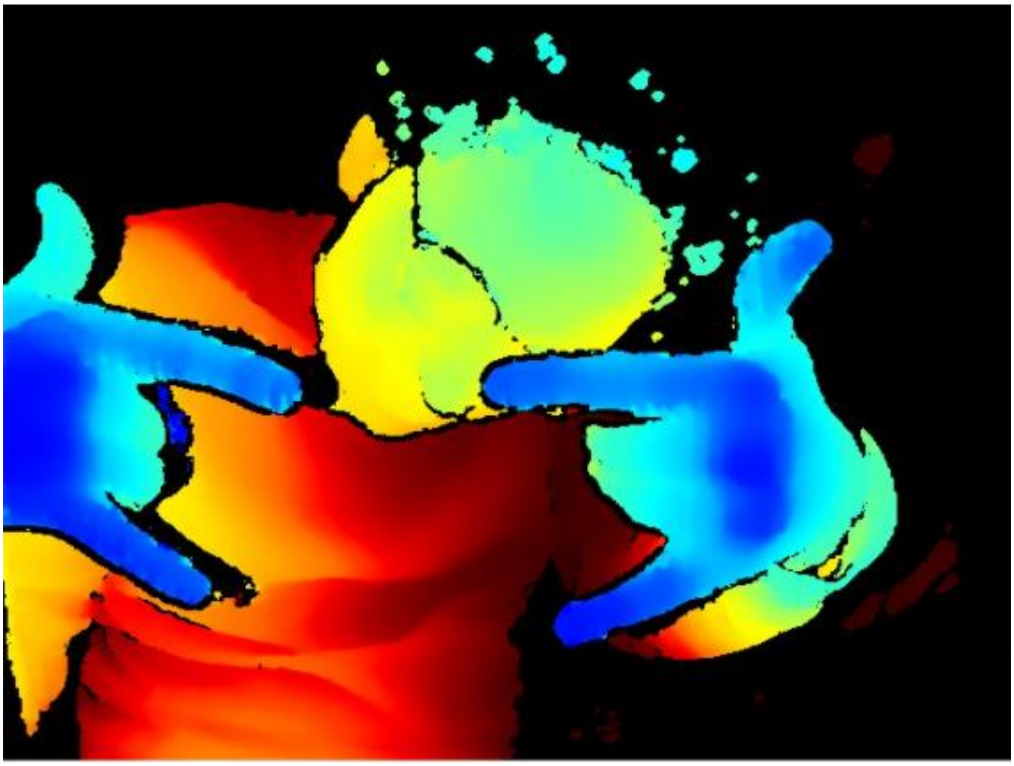
The MATLAB function `depth_example()`, located in the folder where the SDK is installed, provides the code to extract a depth image. Verify that the code works and make sense of the provided code.

Modify this code so that it also extracts a color frame and displays it. You can do so by using the function `get_color_frame()`. Don't forget to reference the appropriate class instance and format the received frame before displaying.

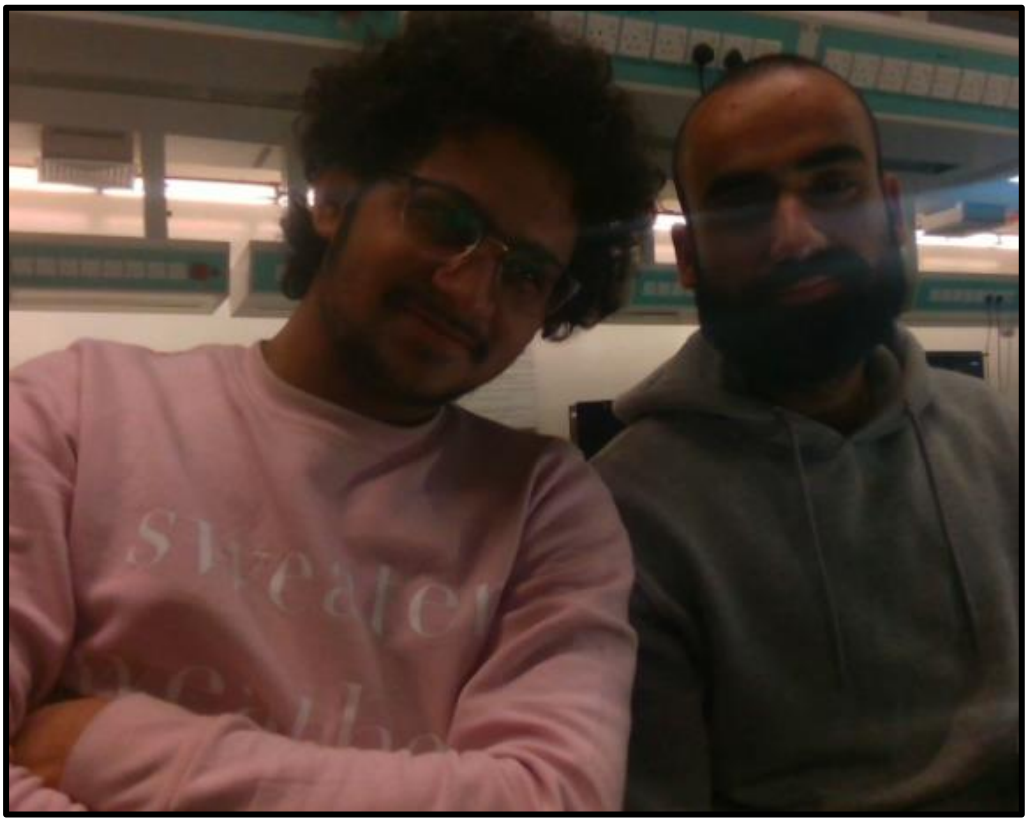
## Observation and Exploration:

The Matlab function depth-examples() has been explored and read multiple time to see how it works. This function serves for the depth frame of the image. Some example of this frame has been attached below.





To get the color frame of the image, we have used get_color_frame() funtion where previously get_depth_frame() was called. Also, in color frame, we don’t need to colorize the image as it is not required in color frame since the image is already colorized. The following picture has been retrived from get_color_frame().



**Matlab Code:**

```
function depth_example()
    % Make Pipeline object to manage streaming
    pipe = realsense.pipeline();
    % Make Colorizer object to prettify depth output
    colorizer = realsense.colorizer();

    % Start streaming on an arbitrary camera with default settings
    profile = pipe.start();

    % Get streaming device's name
    dev = profile.get_device();
    name = dev.get_info(realsense.camera_info.name);

    % Get frames.
    % the camera time to settle
    for i = 1:5
        fs = pipe.wait_for_frames();
    end

    % Stop streaming
    pipe.stop();

    % Select depth frame
    %depth = fs.get_depth_frame();
    % Colorize depth frame
    %color = colorizer.colorize(depth);

    % Get actual data and convert into a format imshow can use
    % (Color data arrives as [R, G, B, R, G, B, ...] vector)
    %data = color.get_data();
    %img = permute(reshape(data',[3,color.get_width(),color.get_height()]),[3 2 1]);


    % Select depth frame
    color = fs.get_color_frame();

    % Get actual data and convert into a format imshow can use
    % (Color data arrives as [R, G, B, R, G, B, ...] vector)
    data = color.get_data();
    img = permute(reshape(data',[3,color.get_width(),color.get_height()]),[3 2 1]);

    % Display image
    imshow(img);%,img2,"montage");
    title(sprintf("Colorized depth frame from %s", name));
end
```

## Task 3.6:

| Task 3.6 | Reflection (10 points) |
| --- | --- |
| Based on your experiences with the robot arm in the current and previous session, write a brief note outlining your takeaways and any unanswered questions about the vision-based pick and place robotic system that we intend to develop as our first project. | |

## Reflection:

First of all, let us say that the content for the robotics lab is multidimensional. We have to study multiple things together and then apply them to one specific task. In the last two labs, we have built upon the mechanical structure of the robotics arm such as number of joints, links, degree of freedom, number of motors, their resolution, sensors and many more. We have found the accuracy and repeatability of the arm then comparing it with the industrial arm. The few take aways from comparing are;

1. The accuracy and repeatability of an industrial robotic arm are affected by factors such as the resolution of the servo motors, the mechanical design of the arm, and the quality of the control circuit and position sensor.
2. To achieve high accuracy and repeatability, it is important to use high-resolution servo motors and design the robotic arm with a high degree of mechanical accuracy.
3. The use of high-quality control circuit and position sensor can further enhance the accuracy and repeatability of the robotic arm's movement.

Although, we have seen that the accuracy and repeatability of industrial arms are very high as compared to our lab arm. This could be because we don't need this much accuracy and repeatability in lab.

We have also learnt about how high-resolution servo motor affect the resolution of the end effector in cartesian space in this lab.

Still have curiosity about knowing, how we are going to find out the position of location from RGB image of the space in real world.

MathWorks® | *Training Services*

# Progress Report

**Name:** Jahania Shah

**Course:** Image Processing Onramp

**Progress:** **70%** complete (as of 23 January 2023)

## Chapters

1. Introduction **100%**
2. Images in MATLAB **100%**
3. Image Segmentation **100%**
4. Preprocessing and Postprocessing Techniques **100%**
5. Classification and Batch Processing **0%**
6. Conclusion **0%**

Release: R2022b | Language: English