31377 Final Project Assignment (January 2022)

1. Data collection and Neural Network training

1.1 Data collection

-- Watch the video collect_data.mp4 ----

- Place the robot in-front of the webcam so that on-off switch of the robot is in the left-hand side of you when you are looking at the webcam.
- Then give -90 degrees of rotation for the X angle towards the webcam. (See figure 1.) Throughout the whole process, keep the X angle as this way.

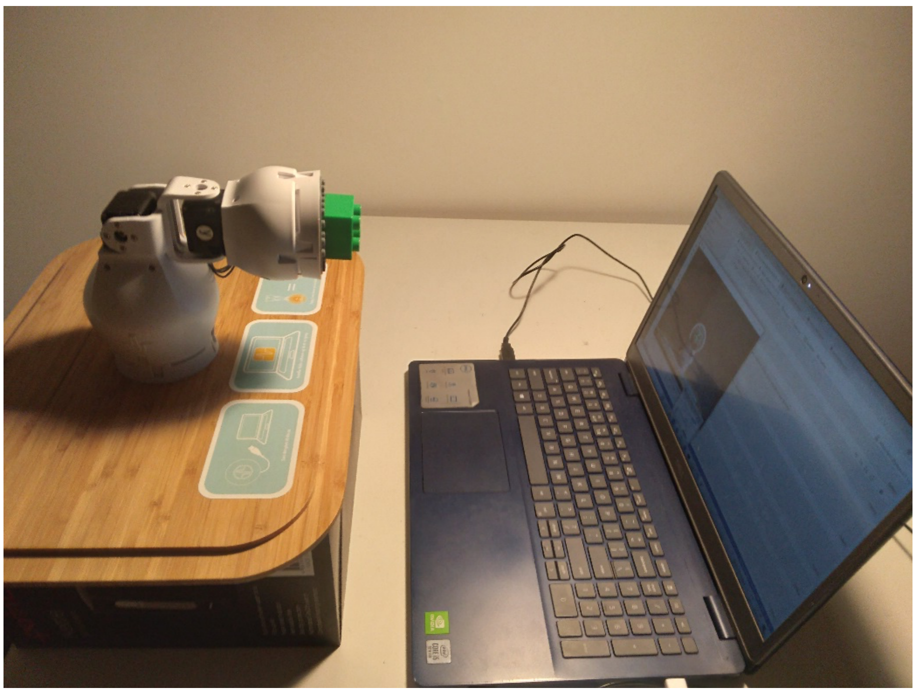


Figure 1

- Open the notebook *project_guidance.ipynb* and follow the procedure to calibrate the camera and to detect and initialize the fable robot.
- Since the robot needs to rotate around Y axis, check whether the camera can detect the end effector with the green box from all positions (that the end effector can be positioned by rotating around Y axis).
- Then, use the “collectData” function in the notebook for data collection. Here, the command to rotate the robot around Y axis has to be given step by step so that “desired_angle_change” is the amount of rotation in one step. As shown in the video *final_project_guidance.mp4*, you can see that, first, the robot rotates in anticlockwise direction and then in clockwise direction.

- Save the Y angle coordinate related to each position into an excel file. Example csv file with already collected data called *angles_1.csv* is already provided. You should collect more data into a different csv file called *angles_2.csv*.
- Also, at each position, save the location of the green box in camera image as X and Y coordinates into a separate csv file. Example csv file with already collected data called *xycoordinates_1.csv* is already provided. You can collect more data into a separate csv file called *xycoordinates_2.csv*.
- If you collect data in two different configurations (for example changing the camera angle and position, changing the location of the robot), for those two configurations save the data into different csv files. That is for example, for the configuration 5 the files should be *xycoordinates_.5csv* and *angles_5.csv* and for the configuration 6, the files should be *angles_6.csv* and *xycoordinates_6.csv*. Then use APIs given in the notebook to read csv files and calculate the error and save those errors in separate lists which can be merged into three main lists called *angle_error_array*, *x_coord_error_array* and *y_coord_error_array*.

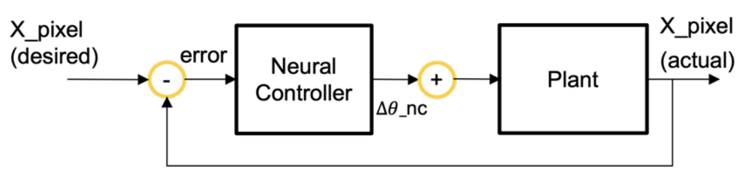
- Try different values for "desired_angle_change'' and dump the results to *angles_2.csv* and *xycoordinates_2.csv* appropriately. The provided csv files contain the results for angle changes such as 1, 2, 3, 5, 7, 13, 15, 17, 19, 30, 50, 70 and 80.

1.2 Training Phase of the Neural Network

Follow the guidelines given in notebook *project_guidance.ipynb* to train the neural network.

---

2. NN Testing for object location



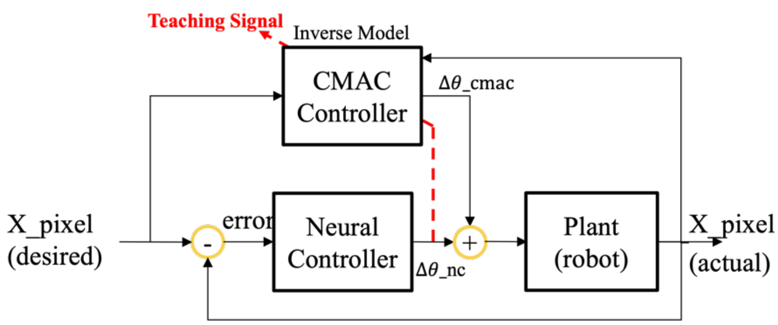
During the testing phase, you should maintain the same configuration of the camera and Fable robot used in the previous training phase. The goal is to test the Neural Network to detect both a target object and the green Lego block on the Fable end effector.

2.1 Control Loop with the Neural Network and <u>without</u> the CMAC (cerebellar controller)

- Complete the *ControlLoopWithNNWithoutCMAC()* function in notebook.
- Follow the following instructions to locate an object with the Fable robot by executing the *ControlLoopWithNNWithoutCMAC* as the control loop.
    - Get an object without green and paste a green color sticker/anything which can be removed later.

- Put that object at a desired position in-front of the camera. The object position must be on the arc of the half circle that robot can touch/reach.
- Then, detect that target object with the webcam and save the target coordinates.
- After this, remove the green color sticker from the target object.
- Start the test/control loop with the green box connected to the end effector of the Fable robot. The end effector will eventually reach the target coordinates. The neural network is driven by the error (coordinates of the target **in camera image** - coordinates of the end effector **in camera image**). The robot is driven by joint angles (In this case it is Y pos because X pos is fixed for the Fable)

2.2 Control loop with Neural Network and with CMAC (cerebellar controller)



- Complete the *ControlLoopWithBothNNandCMAC* function in the notebook.
- Follow the following instructions to locate an object with the Fable robot by executing the *ControlLoopWithBothNNandCMAC* as the control loop.
  - Get an object without green and paste a green color sticker/anything which can be removed later.
  - Put that object at a desired position in-front of the camera. The object position must be on the arc of the half circle that robot can touch/reach.
  - Then, detect that target object with the webcam and save the target coordinates
  - After this, remove the green color sticker from the target object.
  - Start the test/control loop with the green box connected to the end effector of the Fable robot. The end effector will eventually reach the target coordinates. The neural network is driven by the error (coordinates of the target **in camera image** - coordinates of the end effector **in camera image**). The robot is driven by joint angles (In this case it is Y pos because X pos is fixed for the Fable)

(Optional - for the brave ones) 2.3 Upgrade the system

- Try to use two active joints and report the modifications/challenges and the advantages/disadvantages
- What would you change into your system to let it operate in torque control (setTorque)?
- Use the Adaptive Filter instead of the CMAC. What is the advantage/disadvantage?