MTRN 4230 Robotics
Ass 2: Robot System Integration
Group 10: The Power Rangers

Due Date: 21/04/2015

Tutor: Sam Binns

Kevin Kadino z3397298
Ian Yonowiharjo z3402188
Minh Khoa Tran z3414279
Mohammad Iqbal Aziz z3317424
Soo Bin Han z3330656
Roni Permana Saputra z5005883

Introduction

Assignment 1 is about programming the robot to move as directed by the user. The user can control the robot by click of button on the GUI. It involved us to use Robot Studio and Matlab to code to connect between the robot, I/O (input and output) and computer. This short report is about our group's project management practices such as how we planned out to control the robot using ABB and Matlab. It also details which tasks where done by whom, difficulties we had and how we can improve for the next assignment.

ABB control

In the beginning of planning the project, it was determined that a number of tasks is required by the ABB control. The three tasks include;

- Task1 is controlling of the movement of the robot.
- Task2 is getting the user command from the Matlab (GUI).
- Task 3 is sending back all the status of the system which includes the end-effector position, each joint position and the IOs system's status.
- Task 4 is controlling the inputs, outputs and speed.

The tasks mentioned above had been done by Ian, Minh and Kevin to make codes to comunicate with GUI/Matlab using Robot studio.

- The CAD image and Robot cell surrounding area was done first by Ian.

TCP Protocols

From Matlab --> Rapid

1. Send the mode of the movement 0 to 3 in strings data type.
2. Send the data corresponds to the mode:

- In **linear mode**
  '[X,Y,Z]' in string format.
  X = x-coordinate, Y = y-coordinate, Z = z-coordinate
- In **joint mode**
  '[J_1,J_2,J_3,J_4,J_5,J_6]' in strings datatype.
  Where J_1,J_2,J_3,J_4,J_5,J_6 are the joint angles.
- In **I/O mode** and speed mode
  '[DO_1, DO_2, DO_3,DO_4,DO_5,DO_6]' in strings datatype.
  DO_1 = Vacuum (0/1)
  DO_2 = Conv Run (0/1)
  DO_4 = Conv Direction (right/left)
  DO_5 = speed in mm/s
  DO_6 = 0

From Rapid --> Matlab

1. Rapid sends all the current status in one line as a string, separated by square brackets.
   '[X,Y,Z][J_1,J_2,J_3,J_4,J_5,J_6][DI_1, DO_1, DO_2,DO_3,DO_4]\0A'
   X = x-coordinate
   Y = y- coordinate
   Z = z-coordinate
   J_1,J_2,J_3,J_4,J_5,J_6 = joint angles
   DI_1 = Input signal (0/1)
   DO_1 = Vacuum (0/1)
   DO_2 = Solenoid (0/1)

DO_3 = Conv Run
DO_4 = Conveyor Direction (forward/backward)

where \0A represents a new line in RAPID language.

2. Then Matlab process the data further to extract this string into three different variable; position, joints and I/O (double)

Specs

The maximum size of the string is 80.

GUI Design

For the GUI designing, we planned checklist for starting the robot for it to not fail. The designing task was entrusted to be done by Roni, Iqbal and Soo Bin. In the GUI figure, we designed the layout according to the marking criteria. It was made to do the following tasks:

- Moving the end effector in linear mode or joints mode
- Moving the robot's joints.
- Sending the speed specs to change the movement speed of the robot.
- Set/unset the outputs, including conveyor run/stop, conveyor direction CW/CCW, solenoid valve on/off and vacuum pump on/off.
- Pause & Resume button to stop and restart the robot movement
- Displaying the instantaneous position of robot in textboxes
- Preview video feed from both cameras.
- Stop the video feed.
- 'ClicknGo' for each of conveyor and table camera, to select co-ordinates on the video to move the robot to that co-ordinates.
- Capture, to take a snapshot image and detect the location of chocolates and boxes
- Table to display the status of the chocolate, pose, reachable, pickable, flavour and position.
- Plot the position and orientations of any chocolate and box
- Button to connect with the robot studio and update all current robot position, joints and output/input status
- Pause & Resume button to stop and restart the robot movement
- Button to move the robot to calibrate position.

We included timeout function to deal with sending errors signal.

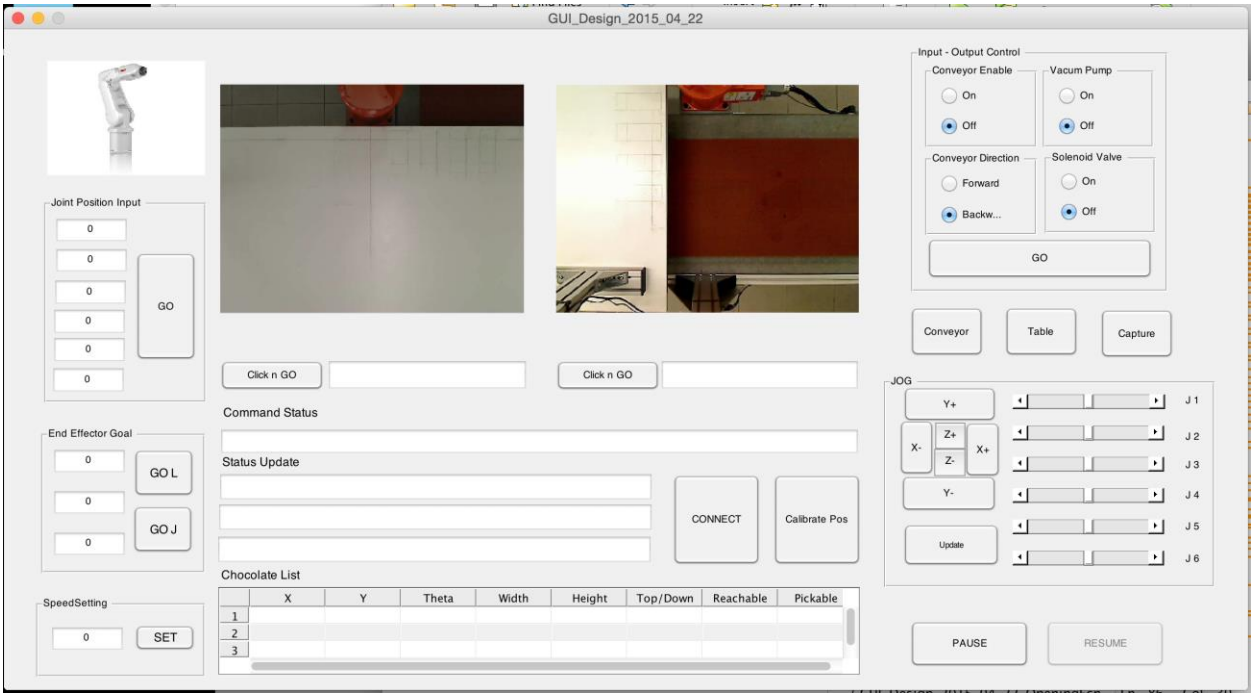

Figure 1: Group 10's GUI design

Using Github

Github was used to share and store codes as a back up between group members. It enable members to edit codes, and know who added the code and fixed the code.

We used Github over gitbucket as it is free but the disadvantage is that it can be seen by other users on the web.

Difficulties

Members responsible for GUI had difficulties in displaying both video at once on GUI. Members responsible for ABB had difficulties in connecting tasks. The Error tasking was also a big issue for us to solve and very important task to be completed for the next assignment. Overall, the most difficult task was to connecting the ABB and GUI using TCP socket.

Methods to Improve Future Assignment.

We can improve our assignment for the next assignment by having previous research on Robot Studio, practicing more on GUI production and Robot Studio. We can be more effiective by better time management such as getting many tasks done pre-hand to due date. We can also do more research on TCP and test the codes before the demonstration date on the real robot.

Conclusion

Overall, we managed to control the virtual robot using GUI. Including, different type of movement such as linear and joint movement. We sent the I/O signal from the GUI to Robot Studio. We designed the GUI in a way that every task required to be done are clicked by buttons.