Assignment2-TicTacToe

Start Assignment

Due Nov 14 by 11pm **Points** 100 **Submitting** a text entry box or a file upload

Problem Description- : Create a Robotic Tic Tac Toe Player

Overview: In this assignment, you are to connect your robotics-GUI created in your project1 to actual hardware and make the system move and perform some basic tasks that can be commanded through the GUI in order to play a game of TicTacToe (https://www.youtube.com/watch?v=5n2aQ3UQu9Y)

(https://www.youtube.com/watch?v=5n2aQ3UQu9Y)

. Each Team will create a 10-15 slide presentation along with a common video link of the project to turn in. You will turn in the presentation, Matlab code (both .m and mlapp files).

Software: You are to create a MATLAB TicTacToe **class** that has as its properties: a robot variable, a hardware (arduino_device) variable, any variables pertaining to the game (e.g. matrix of free and open positions), etc.. The methods of the class should be functions like DrawGrid(), MoveRobotTo (Square), DrawX (Square), DrawO (Square), CheckForWin(), ComputeNextMove(), PenUp, PenDonw, and any others that you might need for your robot. You should first test the functionality of your robot on your robot simulation before using the hardware. You should be able to test all the functionality of your robot using the created class and then use the class in your GUI.

For your testbed, create the following steps:

0. Make sure your robot arm has either a rotary joint on the end-effector (or use the 3D printed rack and pinion system) and place a marker (or pencil, don't use a pen) on the end-effector. Plastic sheets with erasable markers work well. You may consider converting your robot model to make your last joint a prismatic joint. This may help an uneven board or a sagging robot.

1. Be able to draw a TicTacToe grid. You can use the move ee code developed in project 1 to record movements or use an equation of a line to draw the xy points.

2. Be able to move the robot ee to any square of the board. These should be hard-coded joint values.

3. Be able to move your marker up or down.

2. Once at a square, your user interface should allow the robot to draw an X or an O. Here you will create deltas that can be applied to the current location (via fkine) to create a small O or an X. Use the ikine function(s) to calculate the joint angles for small increments from the center position to create an X or an O and set them to the hardware. Experiment with all the ikine variations and select what works for your robot. Explore and use some of the interpolation functions for creating smooth movements if needed.

3. Create a game. From your user interface, allow the user to select a square to place an X. The robot will then write an X at the specified location making sure that the spot is available. Then the robot will select a spot to place an O from the remaining spaces available and write an O. This could be done randomly. Then the game will continue. At each step, the robot will check to see if there is a winner. If so, the GUI will display a winning message and the game can be restarted. You can use your move ee function to record the movements needed for an x or o and store the delta values. Then you can use the same delta values to draw from any square.

4. Extra Credit Opportunities:

Extra credit (10 points): Create an intelligent system that can win (or tie) against a human.

Extra Credit (10 Points) Demonstrate, via Socket code, that the game can be played from a remote location. For this, you will need two laptops on the same hub. You will need to study how the socket command works in Matlab.

Some Helpful notes:

* Note that the project has portions that can be done in parallel. So, don't get stuck. Move on to the next piece of the project.

* Again, all this can be tested in simulation first (you can even draw the X and O) in simulation before testing this on hardware. Don't expect the X's and O's to be perfect as there is some error in your robot arm.

* **Here is a link to a previous project. (https://www.youtube.com/watch?v=mPRT38cophE)**

* Please look at these two sites to make sure you have the correct PWM values for your servos for the servo commands:

S-81: **https://www.servocity.com/hs-81 (https://www.servocity.com/hs-81)**
HS-311: **https://www.servocity.com/hs-311-servo (https://www.servocity.com/hs-311-servo)**

Project Requirements:

The main requirement is that you explore the robotics toolkit and the microprocessor and robot hardware features and concepts discussed in class. In the introduction, describe the

inverse kinematics theory, optimization routines and different methods (iterative and closed-form) used in robotics. Include advantages and disadvantages.

Key points for grading (general): (1) level of understanding of the robotics toolkit and various robotics related concepts (2) discussion of the various issues and properties of robots; and (3) presentation and discussion of results.

Presentation Requirements:

Your presentation should have sections, introduction, problem description, numerical solutions, discussion and conclusions etc. as appropriate. Use pictures and diagrams as needed. The key is to demonstrate your understanding of how to use and program with the robotics toolkit. Explain also how these simple robotics techniques could be used in a variety of scientific and engineering fields. In your presentation, include key snippets of the MATLAB code, your GUI design, and any **relevant graphs and illustrations to support your results**, discussion, and conclusion. Presentation style (clarity, etc.) is a major component of a presentation.

Summarize your experience and experiments along with your ideas on where and how such a toolkit and system might be useful.

Also include a video link (youtube) that shows your robot operating along with the GUI.

For the video, you must clearly demonstrate the following to receive **full** points:
1. extending your robot arm - on the page / off the page
2. showing the drawing of an X and an O
3. move to a square of the user's choosing
4. show one iteration of the game

NOTE: You will be critically evaluated on how **clearly** your video demonstrates your robot arm's ability to do steps 1-4.