Laboratory #2
Teleoperation with Keyboard Control of a Simulated Mobile Robot

Objective:

- Use MATLAB to implement a TeleOp (teleoperation) with keyboard to control movement of a simulated mobile robot on a map
- Use Simulink and Stateflow to implement traffic light controller simulation

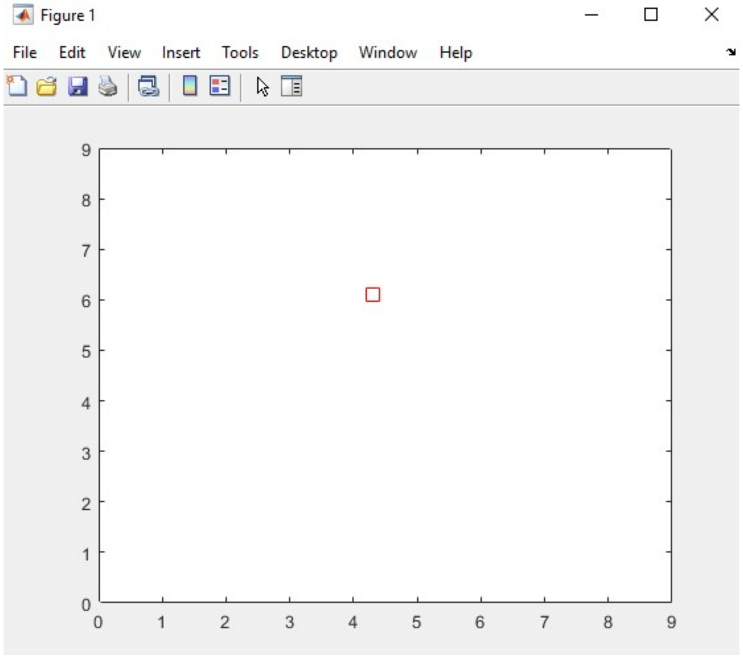
Introduction:

In this week's lab assignments you will be utilizing MATLAB, Simulink, and Stateflow tools to solve problems. These assignments will give you the opportunity to enhance your skills in using these tools for future robotic related problems.

Tasks:

1. TeleOp (teleoperation) to control a simulated mobile robot on a given map

First, you need to create a 10 x 10 map with a small red square object (representing a mobile robot) in the middle of the map. Then you will allow the user to interactively control the movement of the mobile robot using the "wasd" keys on the keyboard. In response to the key inputs of the user the mobile robot needs to move on the map in a 2-dimentional grid, with step resolution of 0.1. The movements are restricted to the four directions: up, down, right, and left. As the mobile robot moves make sure it does remain within the boundaries of the map. A sample view of the map with the mobile robot represented by a small red square object inside is shown in the figure below.



2. Use Simulink and Stateflow to simulate a traffic light controller

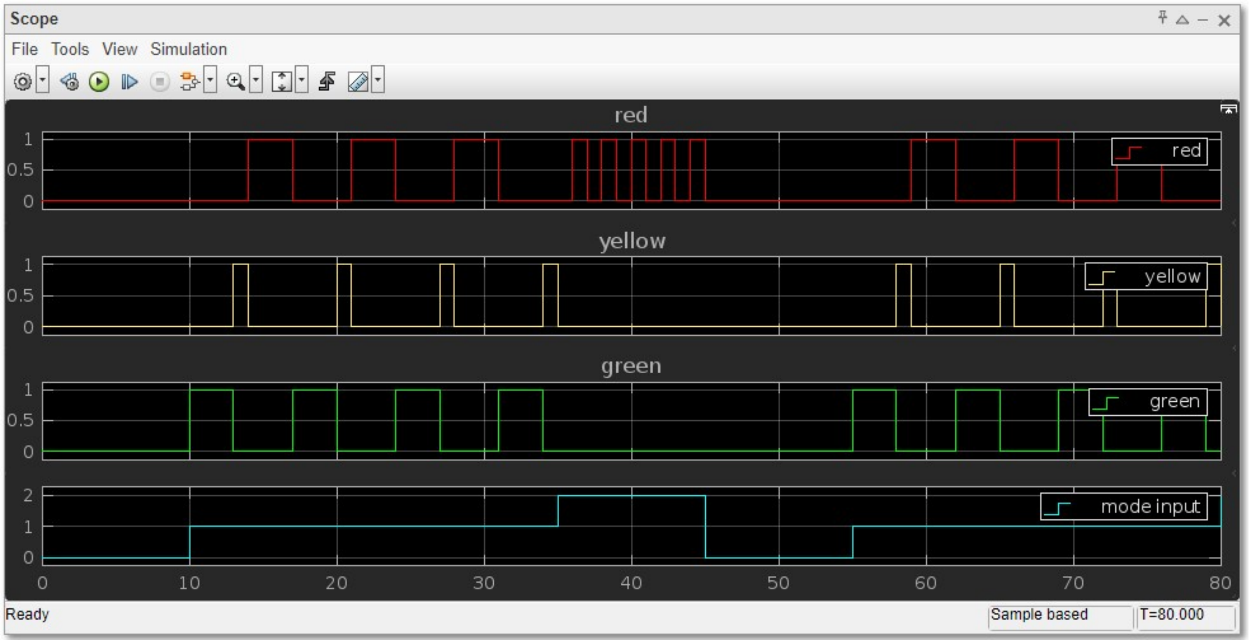
In this assignment you are required to create a model for simulating a traffic light controller that has three operating modes. The three modes behave as follows:

a) Off mode: none of the lights are turned on

b) Normal mode; the traffic lights operate in a normal mode where the three signals sequence as follows:
   i) Green signal is on for 3 seconds, while Yellow and Red are off
   ii) Yellow signal is on for 1 second, while Red and Green are off
   iii) Red signal is on for 3 seconds, while Yellow and Green are off
   iv) Repeat the above sequence until a mode change is detected

c) Override mode: only the red signal toggles on and off every one second, while Yellow and Green are off.

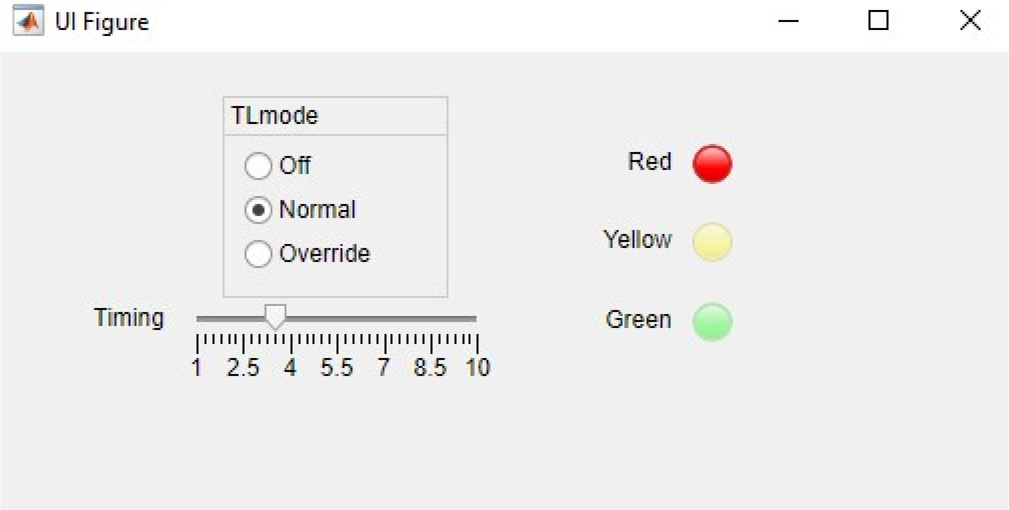
The mode selection input for your model could be provided from a "repeating sequence stair" block from the "sources" Simulink library. The output signals from your traffic light controller model are to be connected to a scope for displaying the waveforms. The expected simulation waveforms are shown in the figure below:



3. Create a GUI based simulator for your traffic light controller (5% extra credit)

MATLAB has tools for creating apps with graphical user interfaces (GUI). If you are interested to explore this tool, create a GUI based interactive simulator for your traffic light controller. If you complete it successfully you can earn up to 5% extra credit on this week's lab grade.

The figure below shows the expected GUI, with the traffic light signals (using colored lamps), the operating mode of the system is controlled using radio buttons within the TLmode block, and the signal timing is controlled using the slider object. Your application should be interactive so the user can control the operation of the traffic light using the input interfaces.



Here is an example you can use as a reference:
https://www.mathworks.com/help/stateflow/ug/sf4ml-lamp-example.html

NOTE: MATLAB provides free online drive (MATLAB Drive), similar to google drive, that you can use for saving your lab files in and using it as a shared media for collaboration by sharing folders among the lab group members.

Deliverables:

Demonstrate your work for both tasks and submit your lab report via Blackboard. Your lab report is expected to follow the format and structure guidelines given in the "Laboratory and Project Report Guidelines" document which is available on Blackboard. It needs to include the title page, table of contents, objectives, your well-commented sources (MATLAB script, Simulink model, and Stateflow diagrams), and output screenshots as applicable for each of the tasks (including the extra credit if you completed it), and conclusion. You also need to upload a zip folder containing your program files -- MATLAB script, Simulink model, Stateflow diagrams, and MATLAB App (if you completed the extra credit).