EML 6351 Spring 2020

Simulation Project 6 - Due April 14 by Midnight.

Instructions:

For this simulation project collaboration with colleagues and any available resource is encouraged. However, any sharing of electronic files with any other individual is considered a violation of the honor code. You may use any simulation software you desire, but Matlab is highly recommended. You are expected to turn in (on-line only) a zipped file containing a written report, which includes all plots and mathematical derivations. You are also required to provide an executable code (meaning a *.m file, Simulink file or other file that can be examined to validate that the results from your report match your simulation code). **A sample Simulink implementation of the NN is provided online.**

Model:

For this simulation project, use the model of a two-link rigid revolute robot manipulator given by

$$\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} = \begin{bmatrix} p_1+2p_3c_2 & p_2+p_3c_2 \\ p_2+p_3c_2 & p_2 \end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} + \begin{bmatrix} -p_3s_2\dot{q}_2 & -p_3s_2\left(\dot{q}_1+\dot{q}_2\right) \\ p_3s_2\dot{q}_1 & 0 \end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}$$

$$+\begin{bmatrix} f_{d1} & 0 \\ 0 & f_{d2}\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix} \tag{1}$$

where $p_1, p_2, p_3, f_{d1}, f_{d2}$ are unknown positive scalar constants (obviously the simulation file has known values for these parameters, but they can not be used in the controller/adaptation law), $s_2 = sin(q_2)$, $c_2 = cos(q_2)$, $\tau_1(t)$ and $\tau_2(t)$ denote the control torque inputs on the first and second joint respectively, and $q(t)$, $\dot{q}(t)$, $\ddot{q}(t)$ represent the angular position, velocity and acceleration of the about the joints of the robot, respectively, where the subscripts denote which joint. For this simulation, only $q(t)$ and $\dot{q}(t)$ are assumed to be measurable. For this simulation, use the following desired trajectory

$$\begin{aligned} q_{d1} &= cos(0.5t) \\ q_{d2} &= 2cos(t). \end{aligned}$$

Assignment:

1. (25 points) Implement a neural network-based controller with a typical continuous feedback control law for the dynamics. Turn in the code in such a manner that it can be implemented/verified that your simulation produces these results.

2. (25 points) Implement a neural network-based controller with a discontinuous sliding-mode feedback control law for the dynamics. Turn in the code in such a manner that it can be implemented/verified that your simulation produces these results.

3. (25 points) Implement a neural network-based controller with a continuous RISE feedback control law for the dynamics. Turn in the code in such a manner that it can be implemented/verified that your simulation produces these results.

4. (25 points) Provide a typed report that covers the following sections.

   (a) Simulation section including (FOR EACH CONTROLLER):

      1. List control gains used and their values
      2. Tracking error plot for each link
      3. Control input plot for each link (link 1 max torque is 250Nm, link 2 is 30Nm)
      4. Plot of the adaptive estimates
      5. Plot of the error between the actual function that is approximated and the NN function approximation

   (b) Discussion section that describes the following points:

      1. Differences in tuning the control gains/adaptation gains between these controllers and with a standard adaptive controller
      2. Performance of the tracking error for each controller
      3. Ease/difficulty in implementation versus previous conrol methods developed in class.
      4. For the above three discussion topics, be sure to compare and contrast the different results.