Intelligent Control Systems Spring 2018 – Homework 2

Due at 4:00 p.m. on February 7, 2018
100 points

This is a Matlab homework assignment. You will create a file called "YOURLASTNAMEhw2.m" and submit that file on Blackboard. Your file will do everything described in the problem statement on the next page. Do not submit any other .m files. Also copy the text in your .m file into a .docx or .pdf file and submit this file that is readable by Blackboard's plagarism software. The grading will be as follows:

- 100 points = correct solution
- 75 points = mostly correct solution
- 50 points = a good attempt, but significant improvements needed
- 25 points = a seriously flawed submission
- 0 points = no submission or failure to follow the submission instructions above

If you submit your homework on time or within your remaining late homework days, you will be given the chance to resubmit your solution if you did not get a score of 100. You will receive feedback and will need to resubmit by the deadline for the next homework assignment.

Please respect the rules on academic misconduct described in the course syllabus.

Problem 1:

Consider a single pendulum with the equation of motion:

$$\frac{ml^2\ddot{\theta}}{3} + \frac{mgl\cos\theta}{2} = \tau$$

where m is the mass of the pendulum, l is the length, θ is the angle from horizontal, and τ is the torque applied at the base of the pendulum. A model of the pendulum is

$$y = \mathbf{w}^\top \phi(\mathbf{x})$$

where the $y = \tau$, $\mathbf{x} = [\theta \;\; \ddot{\theta}]^\top$, $\phi(\mathbf{x}) = \begin{bmatrix} \frac{g\cos\theta}{2} & \frac{\ddot{\theta}}{3} \end{bmatrix}^\top$ and $\mathbf{w} = [ml \;\; ml^2]$.

(a) Bayesian Regression. Load the file "pendulumdata.mat" into your Matlab workspace. This is a different file than the one you used in the interactive session last week. It contains vectors of joint positions, velocities, and accelerations along with torques sampled from the motion of a single pendulum. Compute the parameters using Bayesian regression taking the data one sample at a time. This means you will have the same number of parameter estimates as you have data points. Use a prior estimate of $w_0 = [1 \;\; 1]$ with a prior covariance of $S_0 = 10I$ where I is a 2×2 identity matrix. The sensor variance is $\beta^{-1} = 0.01$.

1. In one figure plot the values of the parameters $\mathbf{w}$ as a function of the number of samples used. Title the figure "Part (a1)".
2. In a second figure plot the square root of the variance of each parameter as a function of the number of samples used. Title the figure "Part (a2)".
3. Make your .m file print in the command window your estimates of m and l. You might use the command `display(['Part (a3):  m = ',num2str(m)])`.

(b) Computed Torque Control. Use computed torque control with no feeback to move the pendulum along the desired trajectory contained in the attached file "desiredtrajecotry.mat". Your simulation should use the parameters $m = 0.6$ kg and $l = 4$ m. Your controller should use the final parameters from part (a). Use a simulation time step of 0.001 s.

1. In one figure plot the desired joint angle and the actual joint angle as functions of time from 0 to 6 s. Make a legend to label the two plots and be sure to label your axes. Title the figure "Part (b1)".
2. Make your .m file print in the command window the root mean squared error of your controller in joint angle tracking in radians. That means at each time step compute the difference between the actual and desired joint angles and square it. Add all the errors over time together and then take the square root. You might use the command `display(['Part (b2):  rms error = ',num2str(rmserror)])`.
3. In a second figure plot an animation of the pendulum with the desired final position of the end of the pendulum clearly marked with a red circle.

(c) Computed Torque Control with Feedback. Use computed torque control with proportional plus derivative feeback to move the pendulum along the desired trajectory contained in the attached file "desiredtrajecotry.mat". Your simulation should use the parameters $m = 0.6$ kg and $l = 4$ m. Your controller should use the parameters from part (a). Tune the proportional and derivative gains until your rms error is below 0.1 rad. Use a simulation time step of 0.001 s.

1. In one figure plot the desired joint angle and the actual joint angle as functions of time from 0 to 6 s. Make a legend to label the two plots and be sure to label your axes. Title the figure "Part (c1)".
2. Make your .m file print in the command window the root mean squared error of your controller in joint angle tracking in radians. That means at each time step compute the difference between the actual and desired joint angles and square it. Add all the errors over time together and then take the square root. You might use the command `display(['Part (c2):  rms error = ',num2str(rmserror)])`.
3. In a second figure plot an animation of the pendulum with the desired final position of the end of the pendulum clearly marked with a red circle.