ECEn 483 / ME 431

Practice Final - Winter 2025

Authors: Professor R. W. Beard and M. D. Killpack

Name: ____________________________

Open book, open notes/slides, open code that was written by you, open J drive, open Python, MS Word, and SymPy. **In terms of AI, you may not use ChatGPT, Co-Pilot or any other similar tools.**

Closed email, internet (except for access to Python or MATLAB documentation, and Learning Suite), and other forms of communication. All helper code provided is in the Python language only. However, you may use MATLAB for certain aspects if you find it helpful.

Work all problems. Unless directed otherwise, write solutions on this document, then scan and submit this in addition to your code and the Word file.

Draw a box around your final answer.

Note that Alt-Printscreen copies the contents of the selected window to the clipboard.

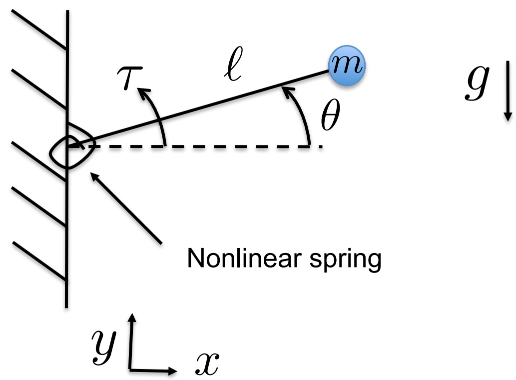
Part 1	________/ 20
Part 2	________/ 20
Part 3	________/ 25
Part 4	________/ 35
Total	________/ 100

Part 1. Equations of Motion - Simulation Model (20 points)

The figure below shows a point mass connected to a massless rod which is connected to the wall with a nonlinear spring and damper. The potential energy for the nonlinear spring is given by

$$V_{\text{spring}} = \frac{1}{2}k_1\theta^2 + \frac{1}{4}k_2\theta^4.$$

The joint has friction which we will model as viscous friction with damping coefficient of $b = 0.1$. The physical parameters of the system are $g = 9.8$ meters per second, $\ell = 0.25$ meters, $m = 0.1$ kg, $k_1 = 0.02$, and $k_2 = 0.01$.
The input torque τ is limited to ± 3.0 Newton-meters.



1.1 Using the configuration variable $q = \theta$, find the kinetic energy of the system.

1.2 Find the potential energy for the system.

1.3 Find the Lagrangian $L = K - P$.

1.4 Find the generalized forces.

1.5 Derive the equations of motion using the Euler-Lagrange equations.

Part 2. Design Models (20 points)

For this section, use the following file to implement the simulation: `practice_final_part_2.py`.

The objective of this part is to use the equations of motion to find the appropriate design models that will be used to design the feedback control strategies.

2.1 Suppose that the objective is to linearize the system around the equilibrium angle θ_e, which may or may not be zero. Find the associated equilibrium torque τ_e so that θ_e is an equilibrium of the system.

2.2 Create a controller that places a constant torque constant torque of τ_e on the physical system. Set the initial conditions to $\theta(0) = \dot{\theta}(0) = 0$ to verify that the equilibrium force is correct, assuming that $\theta_e = 0$ degrees. **Insert a plot of the output of the system with initial condition $\theta(0) = \theta_e = 0$ and an input of τ_e in the associated Word document.**

2.3 Linearize the model found in Part I around the equilibrium (θ_e, τ_e).

2.4 Find the transfer function of the linearized model when $\theta_e = 0$.

2.5 Find a state-space model for the system linearized around θ_e, u_e when $\theta_e = 0$. For your states use $x = (\tilde{\theta}, \dot{\tilde{\theta}})^\top$.

Part 3. PID Control (25 points)

For this section, use the following file to implement the simulation: `practice_final_part_3.py`, `controllerPID.py`. The sampling rate for the controller is $T_s = 0.01$.

3.1 Using the transfer function derived in Problem 2.4, draw the block diagram for the system using PD control, where the derivative gain multiples the angular rate and not the derivative of the error.

3.2 Derive the transfer function from the reference input θ_r to the angle θ.

3.3 Find the proportional gain k_p such that the control input τ saturates at $\tau_{\max} = 3$ Newton-Meters when a step of 20 degrees is placed on the system.

3.4 If the desired closed loop characteristic polynomial is given by

$$\Delta_{cl}^{d}(s) = s^2 + 2\zeta\omega_n s + \omega_n^2,$$

find the natural frequency ω_n and the derivative gain k_d so that the actual transfer function equals the desired transfer function, when $\zeta = 0.9$. Report these values below:

$\omega_n =$

$k_d =$

3.5 Using a dirty derivative, implement PD control where the input θ_r is a square wave with an amplitude of ± 20 degrees and a frequency of $\omega = 0.1$ Hertz. **Insert a plot in the Word file that shows both θ_r and θ for 20 seconds of simulation.**

3.6 Include 10% uncertainty in your model parameters, and then add an integrator to remove the steady state error. **Insert a plot in the Word file that shows both θ_r and θ for 20 seconds of simulation.**

3.7 Insert a copy of the control code (`controllerPID.py`) in the Word document.

Part 4. State Space Control (35 points)

For this section, use the following files to implement the simulation: `practice_final_part_4.py`, `controllerStateSpace.py`. The sampling rate for the controller is $T_s = 0.01$.

The objective of this part is to design a state space feedback controller in stages (first with only full state feedback and an integrator state, then with an observer and disturbance observer, and then using LQR to find the gains for the controller).

4.1 Find the feedback gain K that places the poles at the locations found in Part 3 and the integrator gain k_i so that the pole of the integrator is at -10.

$K =$

$k_i =$

4.2 For the same reference input as in Part 3, tune the controller to get good performance in code, assuming you have access to the full state. **Then insert a plot of the step response of the system (θ and θ_r), for the state space controller with an integrator.**

4.3 Now find the observer gains so that the poles of the observation error, i.e., the eigenvalues of $A - LC$, are five times the eigenvalues of $A - BK$.

$L =$

4.4 Add a disturbance observer, where the pole of the disturbance observer is $p_{dist} = -1$ and report the new L matrix (called L_2 in our notes).

$L_2 =$

4.5 Now implement the observer-based controller (estimating $\hat{x}$ and $\hat{d}$). **Insert a plot of the step response of the system (θ and θ_r), for the complete observer based controller in the Word document.**

4.6 Also, insert a plot of the estimation error and disturbance estimation error in the Word document.

4.7 Finally, find a new set of gains K and k_i using the LQR method. Tune it to have faster performance (even if this requires more torque) than your previous response in 4.2. **Insert a copy of the response in the Word document.** Report the values for Q and R that you used below:

$Q =$

$R =$

4.8 Insert a copy of the final control code (`controllerStateSpace.py`) in the Word document.