EE 122: Introduction to Control Systems

Problem Set #3: System Properties

Name: ______________ **Submission Date:** ______________

Problem 1 Given a second-order dynamical system with transfer function

$$G(s) = \frac{Y(s)}{U(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where ω_n is the natural frequency and ζ is the damping ratio (this is pronounced "zeta"), we can obtain the response of the system to a step input by computing the inverse Laplace transform of the transfer function multiplied by $(1/s)$ since the Laplace transform of a unit step input is $1/s$. Here are the specific steps (your task is to fill in the missing details that are clearly marked in this derivation):

Start by finding the partial fraction decomposition of $Y(s) = G(s)U(s) = G(s)/s$:

$$Y(s) = G(s)U(s) = \frac{\omega_n^2}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} = \frac{a_1}{s} + \frac{a_2 s + a_3}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

In partial fraction decomposition, when we have a quadratic term in the denominator, we use a linear expression (i.e., $a_2 s + a_3$) in the numerator for that term.

(a) [**25 points**] Find the constants a_1, a_2, a_3 in terms of ω_n and ζ.

Once you have the partial fraction decomposition, you can find the inverse Laplace transform of each term separately. But for the second term, we have the quadratic equation in the denominator which is not as straightforward to invert. To handle this, we can use the "completing the square" method. Here's how that works. Write the denominator:

$$s^2 + 2\zeta\omega_n s + \omega_n^2 = (s + \zeta\omega_n)^2 + \omega_n^2 - \zeta^2\omega_n^2 \tag{1}$$

(b) [**5 points**] Verify that the left-hand side of equation (1) is equal to the right-hand side.

Let us define $\omega_d = \omega_n\sqrt{1-\zeta^2}$. This will simplify some of the expressions, and we give it a name: the damped natural frequency.

Now that we have completed the square, we can rewrite the second term of the partial fraction decomposition as:

$$\frac{a_2 s + a_3}{(s + \zeta\omega_n)^2 + \omega_d^2} = \frac{a_2(s + \zeta\omega_n) + (a_3 - a_2\zeta\omega_n)}{(s + \zeta\omega_n)^2 + \omega_d^2}$$

which can be separated into two fractions:

$$= \frac{a_2(s+\zeta\omega_n)}{(s+\zeta\omega_n)^2+\omega_d^2} + \frac{a_3 - a_2\zeta\omega_n}{(s+\zeta\omega_n)^2+\omega_d^2}$$

$$= a_2 \cdot \frac{s+\zeta\omega_n}{(s+\zeta\omega_n)^2+\omega_d^2} + (a_3 - a_2\zeta\omega_n) \cdot \frac{1}{(s+\zeta\omega_n)^2+\omega_d^2}$$

(c) [**25 points**] Using the Laplace transform tables, find the inverse Laplace transform of each of the two fractions above to get the final time-domain step response of the system. Your final answer should be in terms of ω_n, ζ, and t.

(d) [**25 points**] Using your result from part (c), derive expressions for the following step response metrics in terms of ω_n and ζ:

- Peak time
- Maximum overshoot
- Approximate settling time with 2% criterion
- Steady-state value
- Steady-state error (for a unit step input)

[turn to next page]

Problem 2 [20 points] A torsional drivetrain can be modeled as two rotating inertias (motor side and load side) connected by a flexible shaft. This leads to a fourth order transfer function from an applied motor torque (input) to the load-side angular speed (output) given as:

$$G(s) = \frac{b_1 s + b_0}{s^4 + a_4 s^3 + a_3 s^2 + a_2 s + a_1}$$

where the constants capture aggregate physical effects.

Using MATLAB or Python, simulate the step response of this drivetrain model and plot the angular speed (output). Your goal is to find parameter values that yield a step response with the following characteristics:

- Rise time: less than 0.5 seconds
- Maximum overshoot: less than 10%
- Settling time (2% criterion): less than 5 seconds
- Steady-state error: close to zero

You can run simulations using trial-and-error to find suitable parameter values.

Hint: In MATLAB, you can use the following code to set up the system and plot the step response:

```
% Define the transfer function parameters
b1 = ...; % Set this value
b0 = ...; % Set this value
a4 = ...; % Set this value
a3 = ...; % Set this value
a2 = ...; % Set this value
a1 = ...; % Set this value

% Create the transfer function
num = [b1, b0];
den = [1, a4, a3, a2, a1];
sys = tf(num, den);

% Plot the step response
step(sys);
title('Step Response of Torsional Drivetrain Model');
xlabel('Time (seconds)');
ylabel('Angular Speed (rad/s)');
```

In Python, you can use the following code:

```python
import numpy as np
import matplotlib.pyplot as plt
from control import TransferFunction, step_response

# Define the transfer function parameters
b1 = ...  # Set this value
b0 = ...  # Set this value
a4 = ...  # Set this value
a3 = ...  # Set this value
a2 = ...  # Set this value
a1 = ...  # Set this value

# Create the transfer function
num = [b1, b0]
den = [1, a4, a3, a2, a1]
sys = TransferFunction(num, den)

# Plot the step response
t, y = step_response(sys)
plt.plot(t, y)
plt.title('Step Response of Torsional Drivetrain Model')
plt.xlabel('Time (seconds)')
plt.ylabel('Angular Speed (rad/s)')
plt.grid()
plt.show()
```

Include your code and the resulting plot with your submission, along with the parameter values you found that meet the specified criteria. You can automate the computation of the step response metrics to speed up your parameter search — manual trial-and-error will be very slow. We will expect to see your processs of trial-and-error / automation in your code. Simply using the "magical numbers" indicates that you got the parameters without understanding the system behavior.

Here's an initial set of parameters to get you started (these will likely not meet the criteria):

$$a_1 = 400, \quad a_2 = 200, \quad a_3 = 50, \quad a_4 = 10, \quad b_0 = 400, \quad b_1 = 40$$