EE 122: Introduction to Control Systems

Problem Set #2: Modeling Dynamical Systems

Name: ________________ **Submission Date:** ________________

Problem 1 As discussed during the lectures, modeling the system that we are interested in controlling is the first and a necessary task because without a model, we cannot design controllers or analyze system performance. You are familiar with differential equation models. In this problem, your goal is to derive state-space representation of dynamical systems starting from an ODE and relate the transfer function representation to the state-space representation.

(a) [**25 points**] An ordinary differential equation (ODE) representation of a single-input single-output (SISO) linear time-invariant (LTI) dynamical system can be given as follows (same as equation 3.7 in FBS 2nd Edition textbook):

$$\frac{d^n y(t)}{dt^n} + a_1 \frac{d^{n-1} y(t)}{dt^{n-1}} + a_2 \frac{d^{n-2} y(t)}{dt^{n-2}} + \ldots + a_{n-1} \frac{dy(t)}{dt} + a_n y(t) = u(t)$$

For this ODE, assuming zero initial conditions, derive the transfer function representation $G(s) = \frac{Y(s)}{U(s)}$ of the system by taking the Laplace transform of both sides of the ODE.

(b) [**25 points**] Solve Exercise 3.1 in FBS (2nd Edition) textbook, which is based on the equation above.

(c) [**25 points**] For n = 2, i.e., for a second-order system, get the parameter values that are consistent with the parameters you identified in the Lab 1 experiment for the pendulum setup that was assigned to you. To do this, write the transfer function as in part (a) above and compare with the transfer function used in the lab to find out the values for a_1 and a_2. Then, write the state-space representation numerically as well (following your result from part (b)). Compute the eigenvalues of the system matrix A in the state-space representation and the roots of the denominator of the transfer function. Comment on your results.

(d) [**25 points**] Implement the second-order system you derived in part (c) in MATLAB or Python to create a `System` object. In MATLAB, you can use the `ss` function and in Python control, you can use the `control.ss`[1] method. Using the `step` function in MATLAB or

[1] Documentation reference for state-space models: https://python-control.readthedocs.io/en/latest/generated/control.ss.html

`control.step_response`[2] method in Python control, plot the step response of the system. Label your axes and include a title for your plot. Comment on the characteristics of the step response (rise time, settling time, overshoot, steady-state value, etc.) based on the parameters of your system.

[2] Documentation reference for step response: https://python-control.readthedocs.io/en/latest/generated/control.step_response.html