ECE 3300
MATLAB Assignment 6

In this assignment we explore how to obtain and use dynamic system models in MATLAB. Our focus is particularly on linear time-invariant system models. These models are a part of MATLAB's Control System Toolbox. (MATLAB calls these representations *numeric model objects*.) Some of the capabilities described here we have already built out of low-level commands; here, we look at what these higher-level commands can do. The commands have many options; here we simply present the basic forms of the commands, because the "default" settings are adequate for our purposes. Although these commands can be used with both continuous-time and discrete-time systems, our description and focus is on continuous-time systems.

Signal Models

`[x,t]=gensig(type,per,dur,c)` generates commonly used test signals. Here, `type` is one of `'sin'`, `'square'`, or `'pulse'`, `per` is the fundamental period of the signal, `dur` is the duration of the signal, and `c` is the time between samples.

System Models

Consider the linear time-invariant system with transfer function

$$H(s) = \frac{2s(s-1)^2}{(s+2)^2(s+1)} = \frac{2s^3 - 4s^2 + 2s}{s^3 + 5s^2 + 8s + 4}$$

`sys=zpk([0 1 1],[-2 -2 -1],[2])` stores this system in the variable `sys`. The first argument is a list of zeroes, the second argument is a list of poles, and the final argument is the "gain", the additional scaling factor k. Complex-valued zeroes and poles are permitted; if the numerator were $(s^2 + 2s + 2) = (s + 1 + i)(s + 1 - i)$, then `z=[-1-i -1+i]`.

`sys=tf([2 -4 2 0],[1 5 8 4])` alternatively stores the system in "transfer function" format; the first argument is a list representing the numerator polynomial and the second argument is a list representing the denominator.

Time-Domain Analysis

`[h,t]=impulse(sys)` determines the impulse response $h(t)$ of the system `sys` and stores the result in the variables `h` and `t`. It chooses its own values for t. `impulse(sys)` instead plots the impulse response.

`[g,t]=step(sys)` determines the step response $g(t)$ of the system `sys` and stores the result in the variables `g` and `t`. It also chooses its own values for t. `step(sys)` insead plots the step response.

`y=lsim(sys,x,t)` plots the response of the system `sys` to an arbitrary signal defined via `x` and `t` and stores the output in `y`. (The output time values are the same as the input time values.) The command plots input and output signals on a single graph. Often this command is used in conjunction with `gensig`.

`stepinfo(sys)` determines the following data with regards to the step response of the system `sys`: (1) The rise time, defined as the time from when signal goes from 10% to 90% of its steady-state value. (2) The settling time, defined as the time at which the signal is thereafter within 2% of its steady-state value. (3) The settling minimum and maximum, defined as the minimum and maximum values of the output once the response has risen. (4) The overshoot and undershoot, defined as the percentage overshoot and undershoot relative to the settling value. (5) The peak and peak time, defined the maximum absolute value of the ouptut and the time this value occurs.

`lsiminfo(y,t)` determines the following data with regards to the signal defined by `y` and `t`: (1) The settling time, defined as the time to which the signal remains within 2% of its final value.

(2) The minimum and maximum values of $y(t)$ and the times these values occur.

Frequency-Domain Analysis

`bode(sys,{a,b})` produces Bode magnitude and phase plots of the system `sys` from $\omega = a$ to $\omega = b$.

`pzplot(sys)` produces a pole-zero diagram of the system `sys`.

PROBLEM STATEMENT

Consider the system

$$H(s) = \frac{2(s^2 + 6s + 10)(s + 2)}{(s^2 + 6s + 13)^2}$$

1. Represent the system in MATLAB using `tf`. Use `impulse` to determine the impulse response $h(t)$ and plot the result. *Hint:* you will need to expand out the denominator first.
2. Factor $H(s)$ by hand, using complex values if needed. Represent the system in MATLAB using `zpk`. Use `impulse` to determine the impulse response $h(t)$. Plot both this answer and the previous one on a single plot to verify that they are the same. Scale the plot appropriately.
3. For the impulse response (using either version, since they should be the same), use `lsiminfo` to determine the settling time and the minimum and maximum values of $h(t)$.
4. Determine the step response $g(t)$ using `step` and plot it. Use `stepinfo` on it to determine the rise time, settling time, minimum and maximum, and overshoot and undershoot.
5. Use `gensig` to create a 100-sample per second 12-second duration periodic square wave with period 3 seconds. Let this be the input to the system. Determine the output signal via `lsim` and plot it.
6. Determine the Bode magnitude and phase plots using `bode` from $\omega = 10^{-1}$ to $\omega = 10^3$.
7. Produce the pole-zero diagram using `przplot`.