Exploring Emulation Based Digital Controllers

Josh Cohen

April 2020

Designing the Analog Controller

For a plant with a transfer function of $P = \frac{1}{s^2+0.2s+1}$ we have to design an analog controller that meets the following requirements:

1. zero steady-state error
2. rise time < 0.1 seconds
3. settling time < 1.0 seconds
4. overshoot $< 25\%$

Upon examining the open loop poles it was apparent that a simple P controller would not suffice to meet requirements; however, upon experimenting with different PID controllers one was found to meet system requirements. The controller took the form $C = Kp + Ki + Kd$ with

$$Kp = 100, Ki = \frac{1}{s}, Kd = \frac{20s}{\tau s + 1}, \tau = 0.001$$

The closed loop system that incorporated this controller and the plant had the resulting response to a step input:

1. steady-state error $= 0$
2. rise time $= 0.0723$ seconds
3. settling time $= 0.536$ seconds
4. overshoot $= 12.4\%$

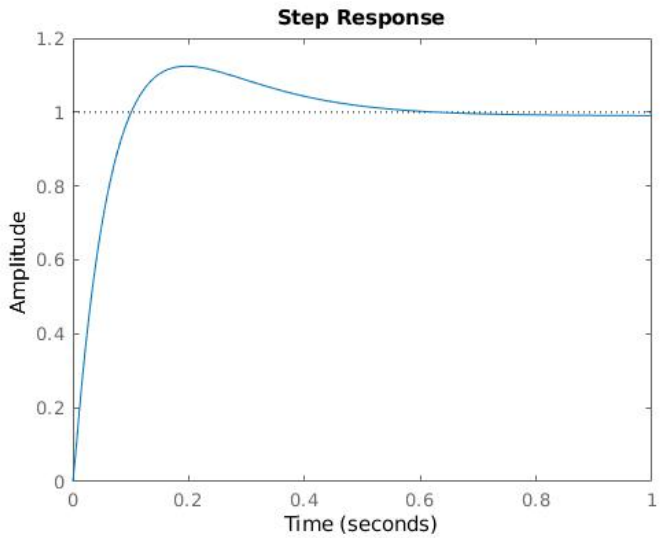

Figure 1: Step Response of Analog Controller

Calculate Digital Controller Emulations

We implemented 7 different emulation paradigms for our analog controller each at 2 differing sampling periods. Each emulated analog controller was digitized once for a sample period of T = 0.1 seconds and once for T = 0.01 seconds.

We we list the frequency ranges for which the digital emulation well matched the analog controller response in both frequency and magnitude. The results were as follows (all units in Hz):

T	step invar.	ramp invar	matched	tustin	tustin prewarp	backward diff	forward diff
0.01	$10^{-4} \rightarrow 10^{-1}$	$10^{-4} \rightarrow 10^{+1}$	$10^{-4} \rightarrow 10^{+1}$	$10^{-4} \rightarrow 10^{+2}$	$10^{-4} \rightarrow 10^{+2}$	$10^{-4} \rightarrow 10^{+2}$	$10^{-4} \rightarrow 10^{+1.75}$
0.1	$10^{-4} \rightarrow 10^{-2}$	$10^{-4} \rightarrow 10^{+0.25}$	$10^{-4} \rightarrow 10^{0}$	$10^{-4} \rightarrow 10^{+1}$	$10^{-2} \rightarrow 10^{+1}$	$10^{-4} \rightarrow 10^{+1}$	$10^{-4} \rightarrow 10^{0}$

Below are the plots of the 3 digital emulations that demonstrated the closest fit to the analog controller over the phase and gain response. As can be seen in the tustin prewarp, forward difference, and backward difference emulations, respectively Figure 2, Figure 3, Figure 4, the phase and gain responses are matched well over almost the entire period of the digital emulator.

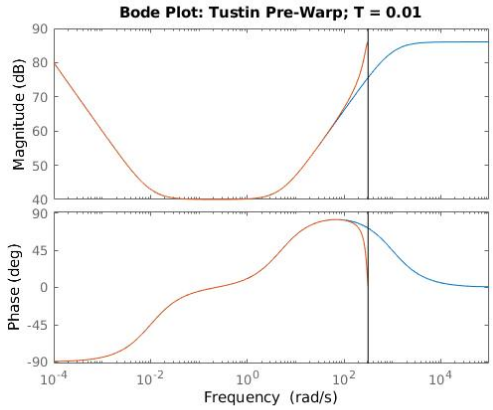

Figure 2: Tustin Prewarp vs Analog Bode Plot

Simulation of Step Response of Digital Emulator for Varying Sample Times

Here we built a simulink model to simulate the response of our digital controller to step inputs. We then examined how the emulated digital controller responded to varying the sample time parameter.

A. $T \to 0$
   As to be expected, when T tended toward 0, i.e. as we sampled with smaller and smaller sampling periods, the digital controller began to behave more and more like the analog controller it was emulating. This can be readily seen in the plot of the step invariance emulators response to a step input as we reduce T from 0.005 to 0.00001 as in Figure 5. As the preiod becomes smaller and smaller its response becomes closer and closer to the teal line that is the analog response.

B. $T \to \infty$ [while meeting specifications]
   Here we vary T by increasing it until our step response no longer matches specifications. The sample period listed here is the last sample taken which corresponds to the first response that does not meet specifications. The code will be attached below. Figure 6 is an example of the resulting step response plots.

T	step invar.	ramp invar	matched	tustin	tustin prewarp	backward diff	forward diff
	0.005	0.023	0.023	0.043	0.045	0.025	0.025

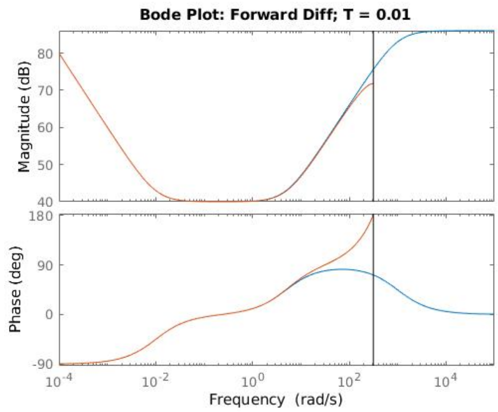

Figure 3: Forward Bode vs Analog Bode Plot

C. $T \to \infty$ [while stable]
Here we increase T until we lose stability in the response to a step input. The sample period here in which the last stable sample was taken. Figues 7 and 8 show plots off increasing the sampling time to unstabilty for both the ramp emulator and the tustin pre-warp digital controller emulator.

T	step in-var.	ramp in-var	matched	tustin	tustin prewarp	backward diff	forward diff
	0.0095	0.070	0.075	0.0975	0.125	0.070	0.075

What is the largest sample period I can use and what is the corresponding emulation method?

Overshoot seems to be the limiting factor of choosing the sample period to meet our specific specifications for this plant. The tustin and tusin pre-warp controllers seem to give the larges tsample period while still meeting specifications, I would use the simple tustin method for its mathematical intuitivenss and ability to be solved by hand with a sampling period of T = 0.043.

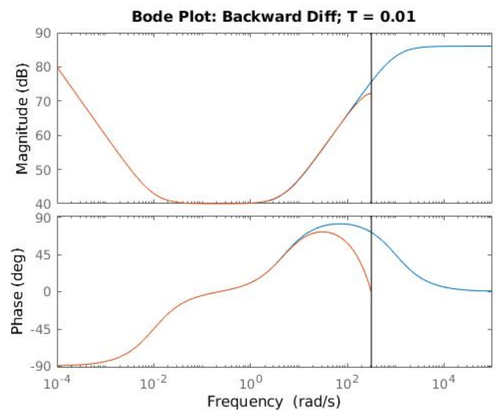

Figure 4: Backward Difference vs Analog Bode Plot

Code

```
function gen_emulator_plots(C, H, tstop, T, name, type)
close all
[yH, tH] = step(H, tstop);
linewidth = 2;

plot(tH,yH, 'LineWidth', linewidth + 5)
legend_list = ["analog controller"];
assignin('base', 'tstop', tstop);

hold on;

params_met = 1;

it_counter = 0;

while(params_met)
    D = c2d(C,T,type);
%      wc = 20.7; %assuming radians here
%      D = c2d(C,T, c2dOptions ('Method','tustin','PrewarpFrequency
    ',wc));
%      z = tf('z', T);
% %      back diff
%      s = (z-1)/(T * z);
% %      forward diff
%      s = (z-1)/(T);
%      Kp = 100;
%      Ki = 1 / s;
%      Kd = 20 * s;
%      D = Kp + Kd + Ki;

    D = minreal(D);
    assignin('base', 'D', D);
    out = sim("analog_emulators");
    t = out.ScopeData(:,1);
```

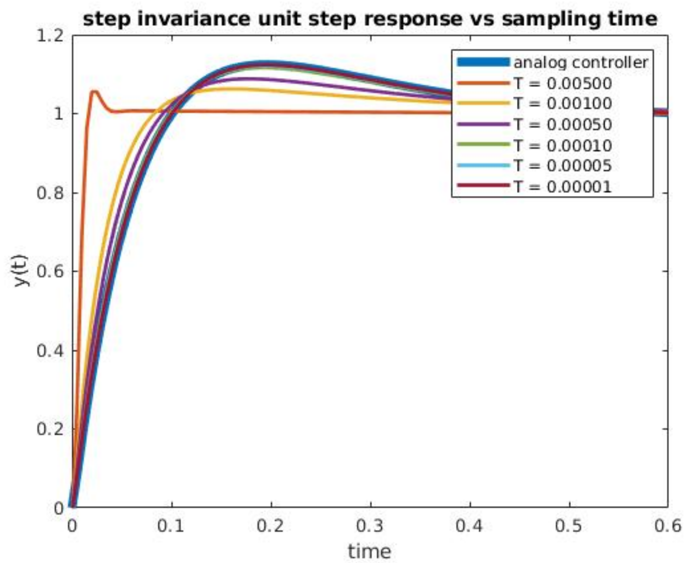

Figure 5: Plotting the step response for the step invariance emulation method for smaller and smaller time steps

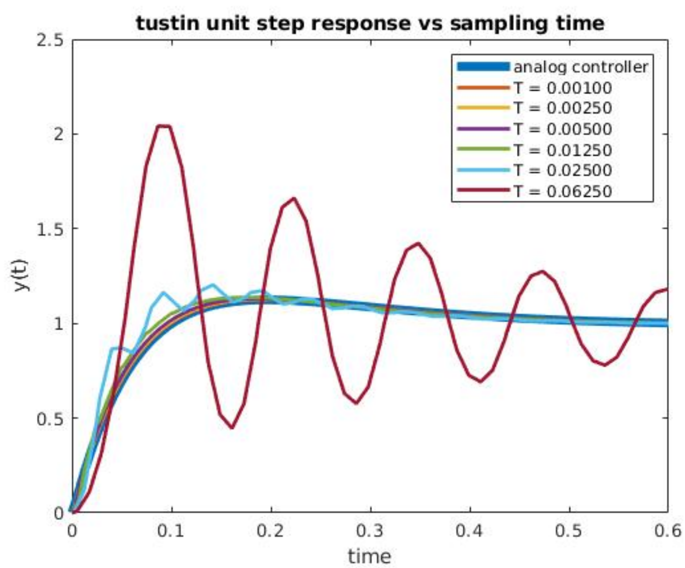

Figure 6: Plotting the step response for the tustin emulation method wit increasing timestep until response specification not met

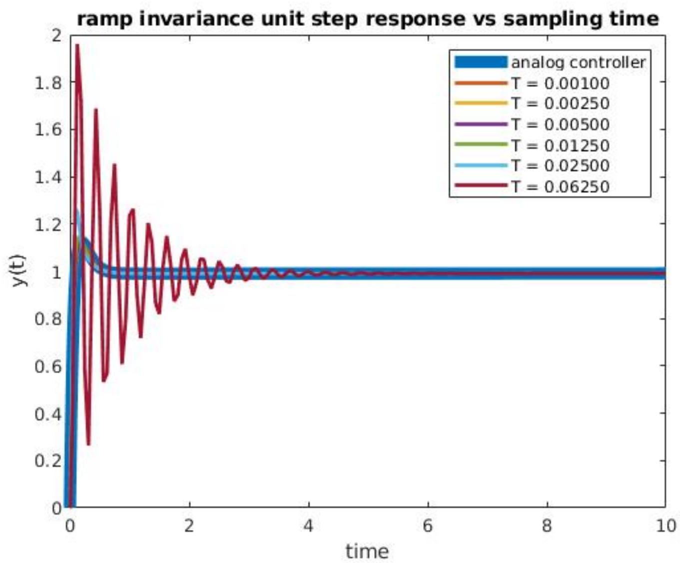

Figure 7: Plotting the step response for the ramp invariance emulation method with increasing sampling time until response becomes unstable

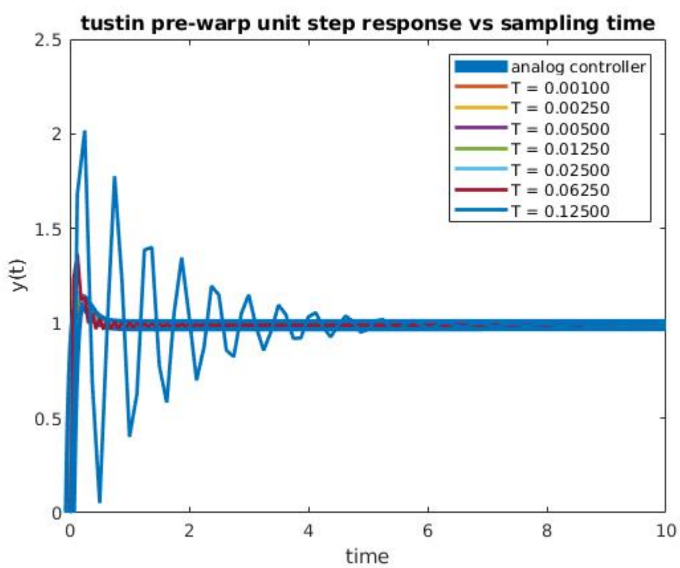

Figure 8: Plotting the step response for the tustin prewarp emulation method with increasing timestep until response becomes unstable

```
    y = out.ScopeData(:,2);
    respinfo = stepinfo(y,t, 'SettlingTimeThreshold', 0.01)


    T_prev = T;
    if (respinfo.SettlingTime > 1.0 || respinfo.RiseTime > 0.1 ||
    respinfo.Overshoot > 25)
    %if (respinfo.SettlingTime > tstop - 0.6 * tstop)
          params_met = 0;
          %break;
    else
        if (mod(it_counter, 2) == 0)
            T = T * 2.5;
        else
            T = T * 2;
        end
    end
    it_counter = it_counter + 1;
    plot(t,y, 'LineWidth', linewidth)
    legend_list(end+1) = sprintf("T = %0.5f", T_prev);
    fprintf(legend_list(end))
    fprintf("\n");

%      if (T < 0.00001)
%          break;
%      end

    if (T> 10)
        break;
    end

end


legend(legend_list)
title( name + " unit step response vs sampling time")
xlabel("time")
ylabel("y(t)")
hold off
```