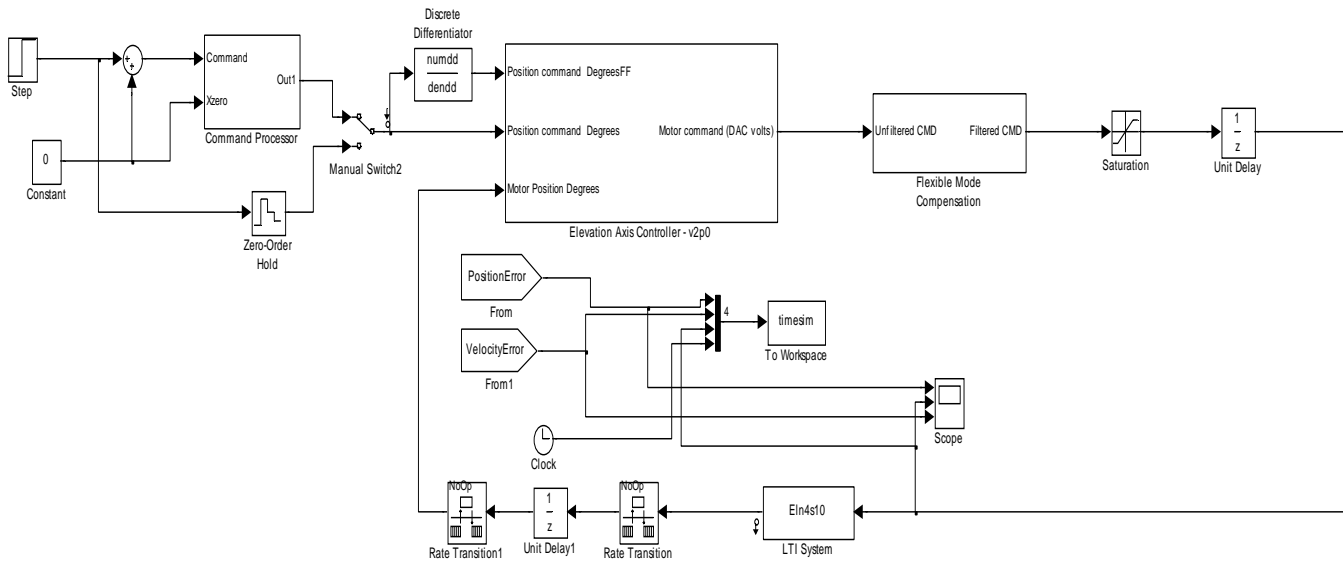


Delays in the MMT Elevation Controller in Simulation

May 18, 2007
D. Clark

The suspect *du jour* for problems getting the VxWorks controller to work at the MMT has been identified as 100 μ S delays in getting tape encoder counts from the LM628 controller chips. At today's servo meeting, it was agreed that I could put together a simulation to investigate this.

Below is the simulation diagram for doing this work. Note that there is *already* a unit delay in the controller output going to the telescope model block. This unit delay is usually set to one servo tick, or 1mS, to prevent Simulink from having difficulties in simulation due to the algebraic loop formed by the dependence of the controller inputs on outputs from the model. The loop is broken by deliberately introducing a delay, which forces the model output to be 0.0 for the first tick, and allowing values for the entire loop to be properly computed. I inserted another delay, and two rate transitions, on the telescope model output. I had to change the simulation loop iteration rate to 100 μ S ticks to accommodate the delays in the new unit delay block. This does not affect the computations for the controller loop, as every block in the diagram has its sample rate set to 1kHz.



Next, I iteratively simulated the model's output with a variable feedback delay of 100 to 1000 μ S with this Matlab code:

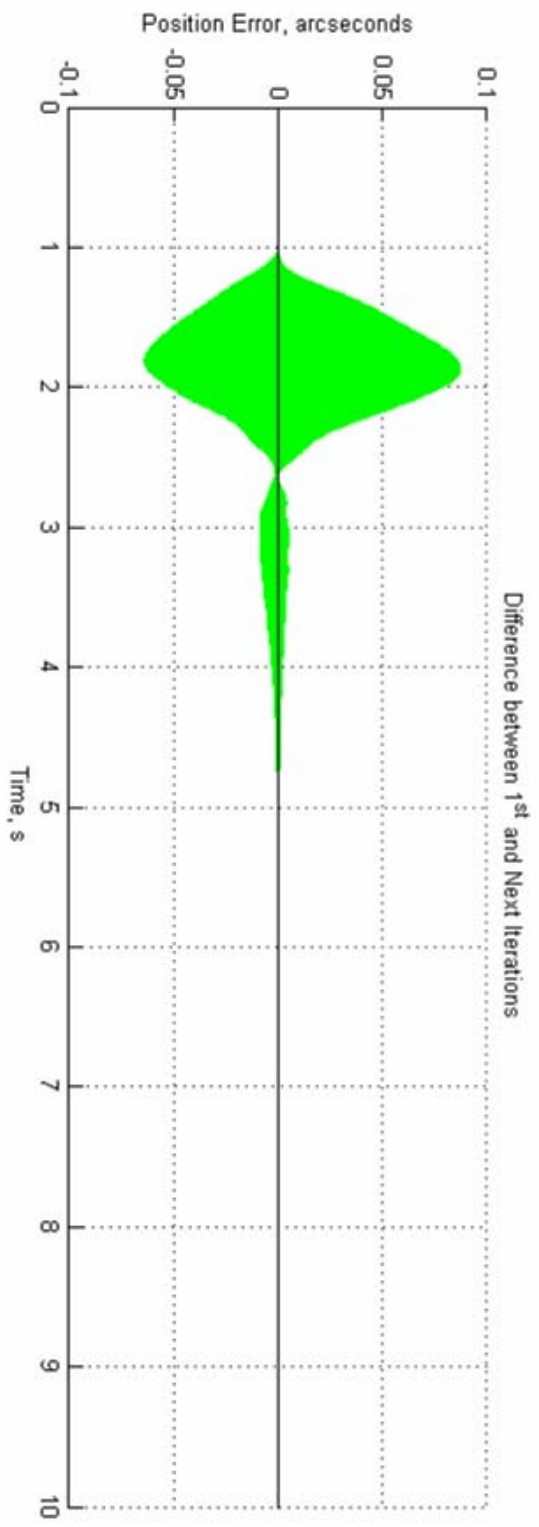
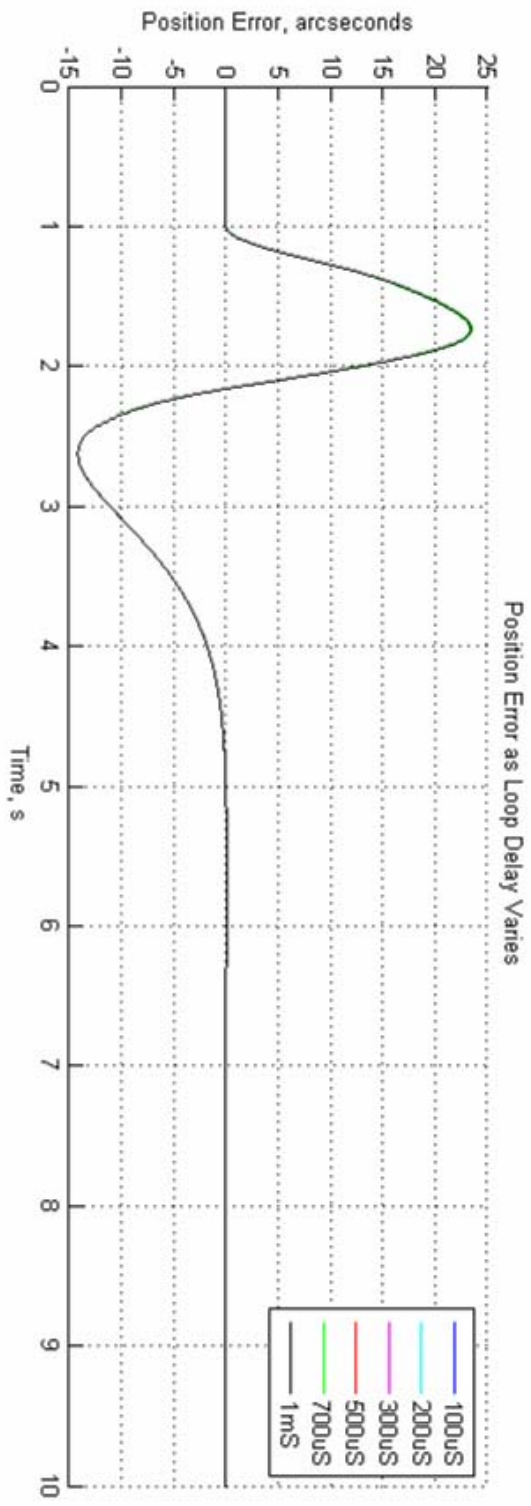
```
%Simulation of the effect of loop delays on the Elevation Axis
Controller's
%output.
%D. Clark
%5/18/07

%Item declarations
StepTime = 1;
StepSize = 100/3600; %100 arcsecond steps, arbitrarily
SimTime = 10;
DelayTime = [1 2 3 5 7 10] * 100e-6; %delay time array, in microseconds
Colors = ['b','c','m','r','g','k'];

%sim loop
figure(); hold on;
for k = 1:length(DelayTime);
    DelaySim = DelayTime(k); %varies from 100uS to 1mS
    disp('Iteration. ');
    sim('El_closedLoopDelayed.mdl');
    subplot(2,1,1);
    hold on;
    plot(timesim(:,4),timesim(:,1),Colors(k));
    grid on;
    if (k == 1)
        lowest = timesim(:,1);
    end
    if (k > 1)
        subplot(2,1,2);
        hold on;
        next = lowest - timesim(:,1);
        plot(timesim(:,4),next,Colors(k));
        grid on;
    end
end

%finish up the plotting
grid on;
title 'Difference between 1^s^t and Next Iterations';
xlabel 'Time, s'; ylabel 'Position Error, arcseconds';
subplot(2,1,1);
title 'Position Error as Loop Delay Varies';
xlabel 'Time, s'; ylabel 'Position Error, arcseconds';
legend('100uS','200uS','300uS','500uS','700uS','1mS');
```

This code plots the position error from a simulated step position command of 100 arcseconds processed with our standard command pre-processor. I also logged the velocity error and DAC outputs, but have not plotted them for the sake of brevity. The plot is on the next page:



The loop obviously becomes oscillatory in one of the plots, but as far as the others, all the simulation outputs are on top of each other. I think that as far as simulation can show us, delays in the loop are not among the Usual Suspects.