

Faith et al-

I've been very tied up with shutdown chores but had some time to look over

the slew file data from the very disturbing elevation collision on 7/31. A

quick look at Tom's webpage with the individual plots showed rapid inflections in the feedback signal, in particular, that should not be present. I wondered, all discussion of integrator windup and observer problems aside, what behavioral data might be gleaned from the slew capture of the incident. The file is sv_20080731_193721, and has the most

inconvenient "feature" of strange entries every 100 lines that make it infelicitous to import into Matlab, but I wrote a script to import the data without the offending ones, so the data discussed here is missing points, though I do not believe this affects the result overmuch.

Attached are two plots with the absolute encoder, tape feedback, command signal, run status, position error, and dac signal all overlaid in time and normalized to the position of 90 degrees, on for the first 4 seconds of the (16.5s) complete file, and another that zooms in on the first second or so.

First, consider the unzoomed plot. Here we see that the event began with a command signal (red) that went from 90 to 89 degrees. At that time, the run status (green) was true, and the position error (magenta) was positive non-zero. The absolute encoder (blue) shows the telescope moved up, opposite to the commanded direction. Indeed, the DAC signal (yellow) remained railed positive. At approximately time 0.5s, the position error check must have tripped, freezing the position error when it exceeded 0.2 degrees. This appears to have also frozen the tape position feedback signal (black), because it then fails to track the real absolute encoder position (blue). The absolute encoder faithfully records the telescope moving up, and the run status goes to 0 a bit later at time ~0.62s, resetting integrators and what have you in the controller (was this the final limit interlock switch tripping?). However, a side effect appears of rapidly switching the command signal from 89.0 to 90.3. The command and feedback are likewise switched to the same value, zeroing the position error. A bit later at about 0.72s, the DAC finally switches from positive rail to railed negative, and then enjoys several cycles of rail to rail switching at about a 0.12s period.

Next, let's zoom in on just the first second or so of this event. Here we can see the command was 90.0, then set to 89.0 at about 9ms or so. Note that the DAC was NOT railed at the beginning, though it rapidly goes to the positive rail at about 10ms. The position error at this time was 0.02

degrees, and began going negative starting at about 0.17s. The absolute encoder tracks the position error curve with the opposite sign. The run status was true for the entire first portion of this event.

Several questions/concerns/comments about what I see here. First, we started out without the DAC at the rail, which quickly became railed. Why?

Next, the feedback shows the telescope moving opposite to the command, as if the feedback sign had been flipped; obviously this would guarantee a runaway. Once the position error check trips, it turns off the feedback signal to the controller. I can think of no good reason to do it this way;

you've opened the control loop up at that point, and the position error is set to some non-zero constant. The status signal delays 150ms or so before

it changes, and we appear to then get a spurious command signal to 90.3 degrees. The DAC signal is not zeroed, but allowed to swing rail to rail

when the final limit (I think) is exceeded. If this is the case, the DC bus on the amplifiers takes time to fall down, so even if the interlock system tripped, there could still be torque available to the drives for a

short time if the DACs are in fact not zeroed (e.g. enough to drive into

hard stops, or people). If the observer and telescope are not in agreement

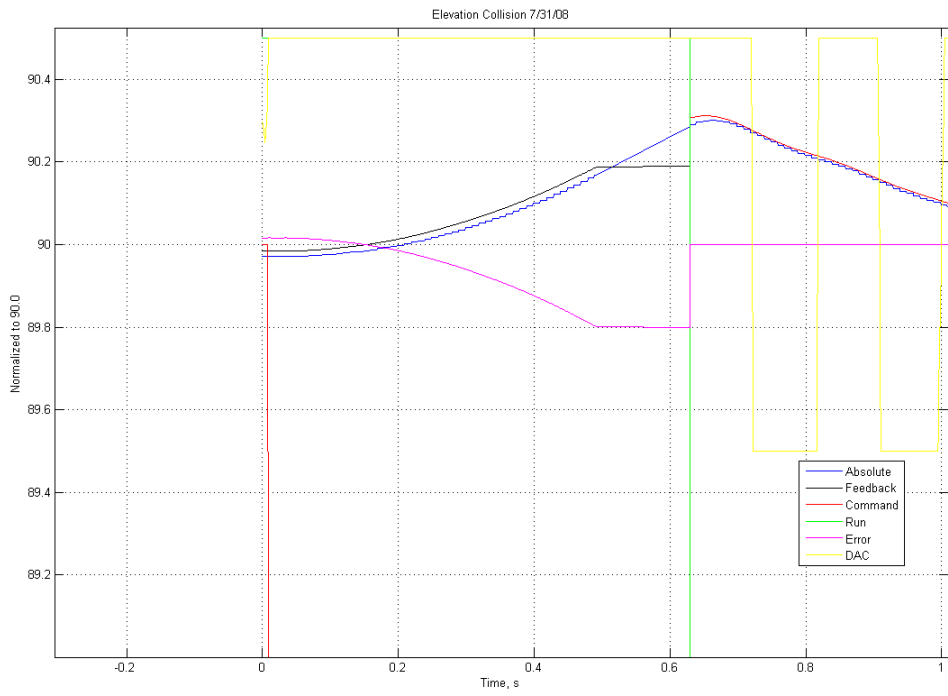
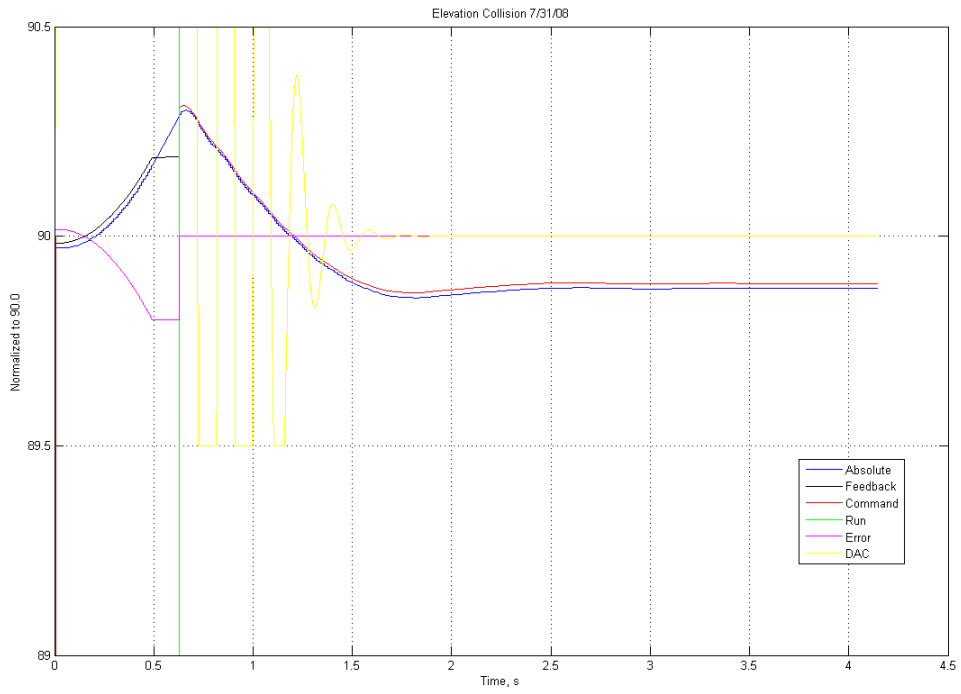
here, I wonder if the position feedback to the observer compensator is present and this is contributing to the issue. All that aside, I'm of the

opinion that we have a few easy behavioral/structural modifications to make:

1. Telemetry files without corrupted ascii lines, please. I spent considerable time working around this issue.
2. Don't turn off the controller feedback. Do zero the DAC.
3. Spurious command signals that exceed the allowable position range? Not good.
4. Detect railed DACs that don't have motion associated with them...is a brake stuck?
5. Detect motion opposite to commanded motions -- loss of encoder/sign flip? Should not happen, if it does, kill the drive QUICKLY.
6. 0.1 second delay in run status trip? Interlock computer delay?

More later,

Dusty



Faith,

One more thing I've added to the plot of the collision is the command pre-processor output (cyan) to the controllers position loop. Clearly, it is getting switched at the same time as the other signals from the preprocessed value it was outputting to command 89.0 to 90.3 degrees. This is wrong. The CPP is there to protect from rapid accelerations of the telescope that could damage things, and we had a) motion in the wrong direction to begin with, and then b) an unprocessed command output that was beyond 90...so it's to fix.

thx,

D

