

Servo Project Progress Report
Dates covered: 1/15/07 to 1/19/07
D. Clark and T. Trebisky

At last report, we had planned the following activities:

1. Develop and test an xPC Target controller for verifying controller operation and discovering the source of the 20Hz oscillation, and correct it.
2. Measure open-loop telescope response.
3. Complete documentation and code changes as necessary to ensure the proper feedback signal flow and signs in the mount PC.
4. Complete data analysis for the differential positions reported by the absolute and tape encoders. Plan a strategy for dealing with measured position errors from the drive arc encoder(s).
5. Test the new mount controller on the sky (weather and Director permitting).
6. Collect closed-loop and disturbance rejection data for documentation of the new controller performance.
7. Install telemetry in the mount controller for collection of high-speed tracking error data to verify tracking performance and measure modal responses.

To support items 1 and 2, Clark worked on building a version of the xPC Target controller that allowed for absolute positioning of the telescope in closed-loop form. This is new in that previous iterations of the xPC test controller used offsets from the starting position as the command position, making safe motion to positions that are near the elevation limits inconvenient to accomplish. Safely operating the telescope with the xPC controller needs a way to interact with the human interface, with the ability to stop motion if an unsafe event occurs. The most straightforward way to do this was to design the controller with an Ethernet interface to a user GUI hosted on another computer. The user GUI allows commands to be entered, or motion to be halted, and displays information about the telescope position, velocity, and position error.

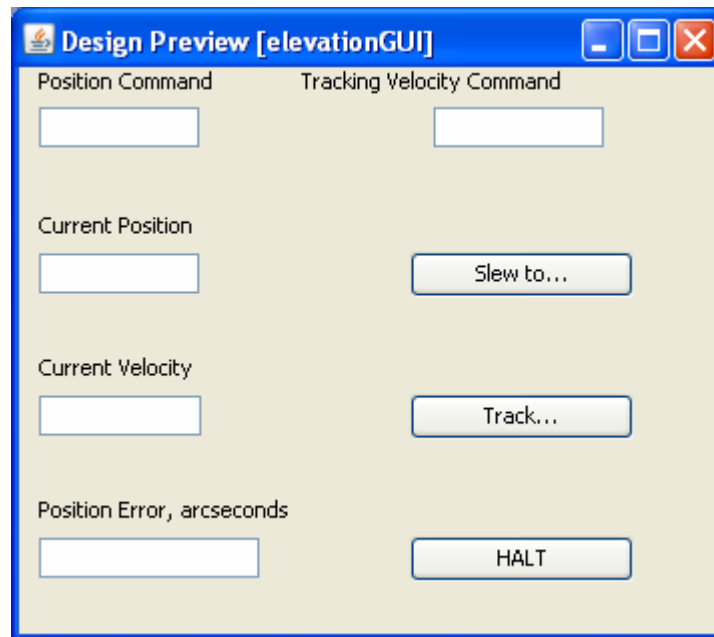
The first portion, the xPC Target controller, is viewable here at:

http://www.mmt.org/~dclark/Reports/testServoModes_slwebview.html

It is the same controller that has been in use for testing for the past ~18 months, with some additional blocks to handle switching the motor feedback signal between the hardware and a couple of different telescope models. In this way, the xPC machine can then be used in a simulation-only mode to test the controller behavior without actually needing to run the telescope. Another section of the block diagram handles the Ethernet connection using UDP packets to receive the commanded motion type, position command, and velocity command. A UDP transmit section sends the current absolute position, velocity, and position error back to the same host the commands are sent from. The last section is the “decision switch” that connects the controller command signal to the appropriate input value, which is either the starting position in the case of the controller just coming up, or the commanded slew position/track rate in the case of a UDP packet command. With this logic, slews to an absolute position with a final velocity

of zero, and tracking commands to a specific velocity are supported. Additional logic on the remote host end allows the current motion to be halted by setting the tracking velocity to zero and making the command position equal to the last transmitted actual position. This has the side effect of causing the telescope to decelerate and reverse its position in the case of a motion halt being selected when the telescope has any significant velocity due to the inherent acceleration limits of the hardware and software. This is the same as the “Cancel” button available on the existing operator GUI.

The operator GUI is simple:

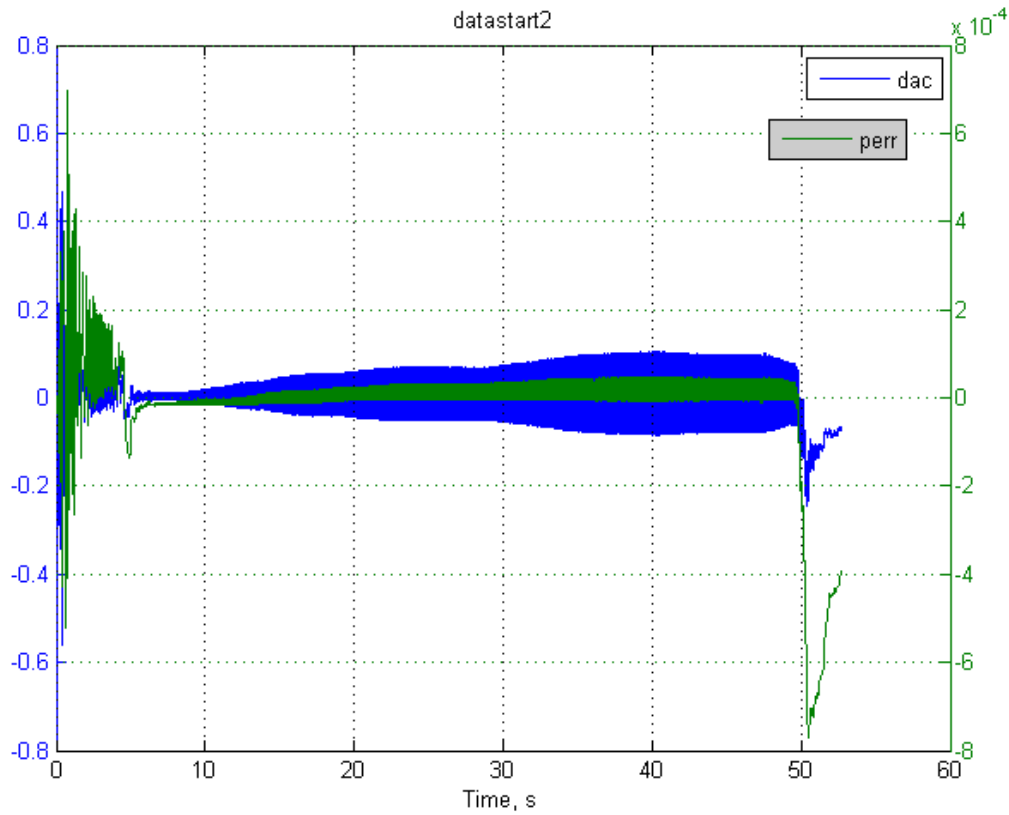


It is written in Java by Clark using some resurrected code in support of the project documented at <http://www.mmt.org/MMTpapers/pdfs/itm/itm04-3.pdf> for those interested, with new code as needed to get it to control the xPC controller as intended.

A round of initial tests was conducted with the xPC controller in simulation mode, with the controller feedback connected to the standard state-space model we have used for the controller design. With some minor tweaks, the Java GUI and the simulation controller were working well, giving confidence in our ability to actually control the telescope with this new network interface.

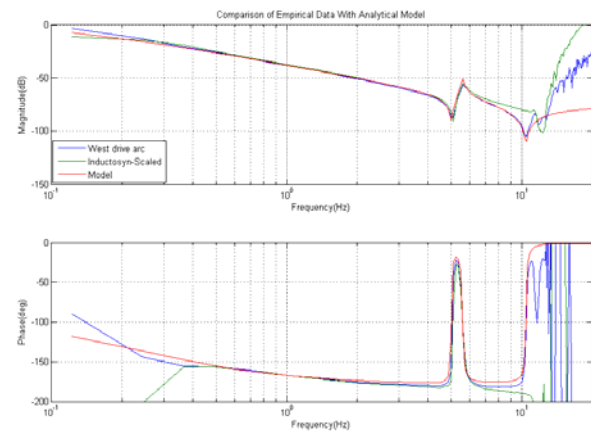
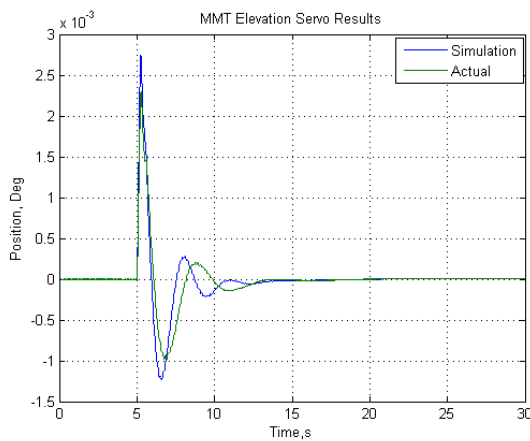
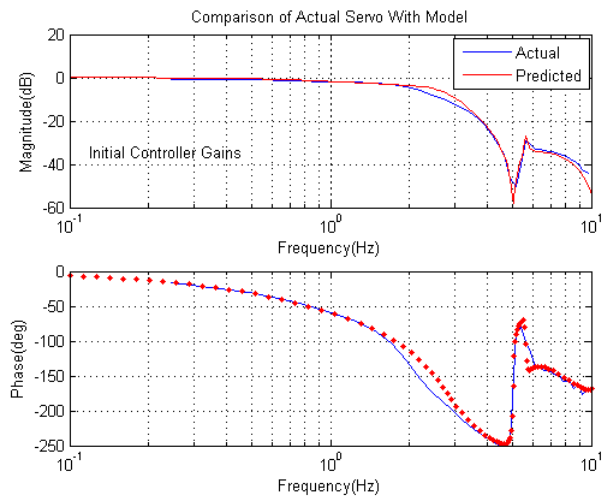
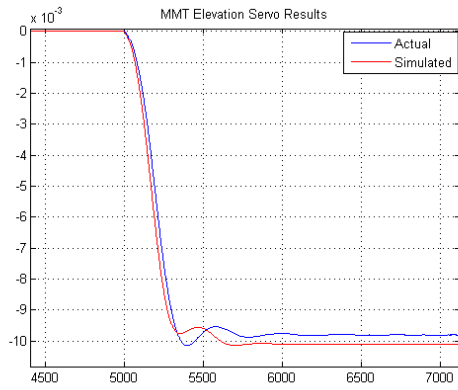
Once we had the new GUI and xPC controller code in place, we wanted to confirm if the oscillations seen during last week’s series of commissioning tests was present. If present, it means that the xPC and VxWorks controller implementations are equivalent, which serves as a check on the controller code generation and incorporation into the VxWorks mount code infrastructure. If not, means that there are problems with this code transitioning process.

Unfortunately, the telescope oscillation is still present. When starting up the controller software and drives, we experience an unstable condition with an increasing oscillation at 20Hz. The plot below shows the DAC output in volts, and the position error in degrees for the system startup.



The oscillation is generally not noticeable at startup, but grows into audibility after ~15s; the glitch at the end is the drive amplifiers being turned off for safety's sake.

The oscillation with both controllers is surprising; there have been no results with this oscillation when testing the controller, as some plots below comparing the model results with the actual telescope (with the xPC Target controller) illustrate:



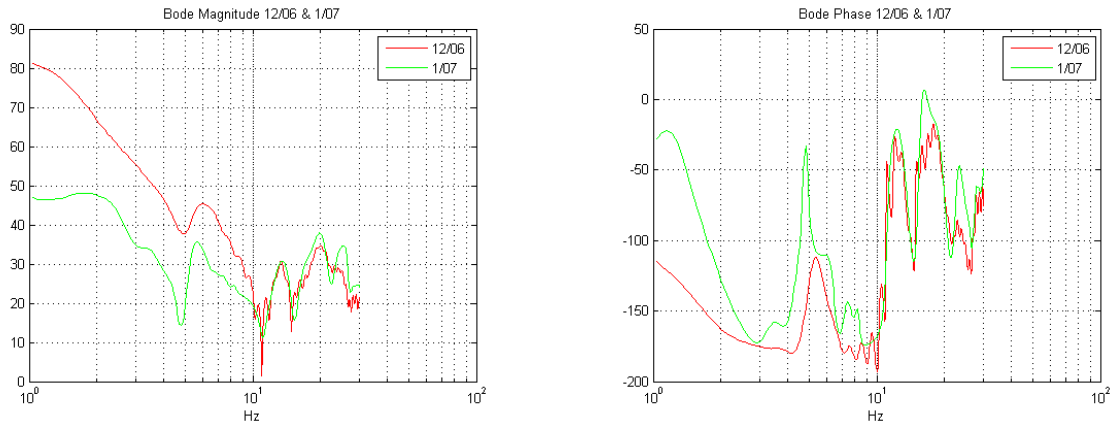
All of the above are from the early controller design and testing in 2004 and 2005. The excellent agreement in the results is why we have been confident in the controller design. There appears to have been a change in the elevation plant response since this design work was done. No good theory for this change comes to mind.

Nearly all of the night of 1/17 was consumed in an effort to understand and correct the oscillation. A major difficulty was that neither the original model, nor new ones created using the System Identification Toolbox from the measured time-response data, predicted the 20Hz oscillation. This in turn meant that it was impossible to have any certainty as to the actual response of the telescope hardware would be to changes in the controller. In hopes of having some actual progress, many changes in the controller were attempted:

- design and implementation of 20Hz notch filters (several types) in the forward path of the controller resulted in instability.
- insertion of a 20Hz notch filter in the feedback path also gave unstable results.
- low-pass filtering the discrete differentiators in the rate feed-forward and velocity estimation paths was not stable.

- gain changes in the controller loops, including turning off rate feed-forward, and adjusting the velocity loop derivative-path gain was likewise unstable.

To get better data on the elevation plant response, on 1/18 more open-loop data was collected, with emphasis on collection of frequency response data between 10 and 30Hz. For comparison purposes, here is a transfer function estimate of the west tape encoder output using the data collected in December 2006 and the data from this week:



The difference in the low-frequency results is because this week's data was from an excitation chirp signal with its lowest frequency at 10Hz. As you might expect, there is agreement between the two estimates, with smoother results in the January data because more signal dwell time over the interval of 10 to 30Hz was applied. The December data are more reliable in the region from 1 to 10Hz because that has generally been the region of interest in past open-loop tests.

The rapid phase changes at ~14 and 20Hz show that there really is a modal resonant-frequency pair in the telescope at those points. Why they have not before been a problem in closing the servo loop is unknown.

Clearly, the controller design requires iteration to a) capture and predict the plant response more faithfully, and b) properly design the controller to deal with the actual structural response of the telescope.

The bright spot in this is that the controller passed all the safety-related checks in the commissioning checklist, and we were able to do some limited dry runs on acquiring stars and non-sidereal objects, so once the controller iteration is completed and tested, we can easily move on to implementation and final commissioning.

Once we have a new controller design candidate, we also now have a handy GUI and controller environment to speed up the general testing process, so we would not necessarily need to spend a lot of time ensuring that the controller is working.

As to the activity list, Trebisky completed items 3, 4, and 7 during the week, with items 5 and 6 needing further action to complete (controller design iteration).

Upcoming activities:

1. Meet with Keith Powell to discuss results and formulate controller iteration design plan.
2. Pursue system identification and simulation as needed to predict the telescope plant response, including verification of open-loop response now that the telescope configuration has changed (f/9 to f/5).
3. Develop controller design changes (filters, gains, etc.) to bring the controller into working order.
4. Test controller iteration with xPC Target test machine for initial verification.
5. Turn over to Trebisky for inclusion in the mount computer code infrastructure.
6. Test VxWorks implementation with the remainder of commissioning checklist items.