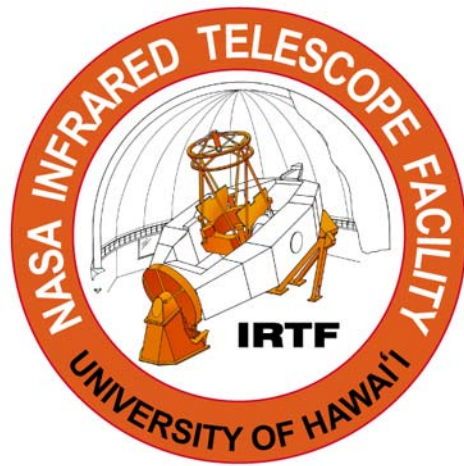


# NASA Infrared Telescope Facility

University of Hawaii • Institute for Astronomy  
2680 Woodlawn Drive, Honolulu, HI 96822



## TCS3 Servo System Design: Software

Tony Denault  
May 26, 2004  
Version 1.0

## 1. Software Mode

There are 2 linux processes related to the servo, VTCS and RTCS. And the PMAC can be considered an independent process (although it is actually running several threads).

- VTCS – The Virtual TCS task performs the astronomy related calculation. At 20Hz it will transform a RA,Dec means coordinate (ie: FK5) to an HA,DEC mount coordinates. The transformation includes refraction and pointing correction.
- RTCS – The Real TCS task is responsible for commanding the servo system based on the VTCS's mount coordinates. At 20 Hz, it check the VTCS position and commands the servo controller towards the VTCS position.
- PMAC is the servo controller. It responds to the RTCS command. The PID loop is located in the PMAC.

This table summaries the software algorithm of the servo processes.

VTCS Modes	RTCS Mode	PMAC Notes	Comments
Idle	Idle	OpenServo Brakes Locked Amps disabled	
Track	Track	To Be Determined: PVT mode, or Jog at velocity	Target is an RA,DEC means coordinates.
Slew	Slew	Jog to position	Target is an RA,DEC means coordinate. Upon completion switches to track.
MP	MP	Jog to position	Move Position will move an axis to a HA,DEC mount position. This is an engineering mode.
MV	MV	Jog at velocity	Move Velocity will move the axis to a specific velocity. This is an engineering mode.

Currently the only issue is whether to use the PMAC's PVT or Jog mode of motion for tracking.

## 2. PMAC Configurations Notes

### 2.1 Dual Output

Each telescope axis has 2 motors requiring 2 PMAC DAC per axis.

The PID algorithm will produce a single correction value based on a position error.

The general scheme is to have a PMAC PLC program run every servo cycle and write 2 DAC value based on the PID correction value. The following pseudo code illustrates the algorithm:

```
dac0 = PID output
dac1 = motor 1 DAC
dac2 = motor 2 DAC
base = backlash value for dac

if( dac0 < 0 )
    dac1 = -base + dac0
    dac2 = base
else
    dac1 = -base
    dac2 = base + dac0
```

This algorithm is current running on the LAB Development System.

### 2.2 Using DC tachometer as a velocity sensor.

The PMAC does allow for a separated position sensor to close the velocity loop.

However, only a single input is allowed per motor axis. Our PMAC is fitted with an 16-bit ADC. The 2 axis tachometers output must be combined into a single voltage for input to the PMAC.

TBD: What the scaling factor is needed for to map to +/- 10 volts to a desired velocity range.

We have successful programmed the PMAC to sample the Lab Simulated Tachometers. However, we normally run the Lab System as an axis with a single position sensor (Encoder on main shaft).

### 3. PVT vs JOG

**PVT** is Position-Velocity-Time. The user specifies the axis state at transitions between moves using the end position, end velocity, and piece time.

Advantages:

- Tighter control of profiles during the 50ms between RTCS = better BS performance.

Disadvantages:

- RTCS cannot always determine the last PVT move used by the PMAC. Duplicate PVT points or missing PVT points during transition periods may cause problems during tracking.
- Exact timing of individual PVT points difficult to schedule due to look-ahead nature.
- Must be implemented as a PMAC motion program – PVT cannot be implemented using online commands. These motions program must be started/stop on-the-fly when moving in and out of track mode.
- RTCS must provide a new PVT table before current table is completed.

**Jog** mode allow you to command the axis to move at a constant velocity.

Advantages:

- Much simpler to implement, understand, and execute = Less prone to error.
- No PMAC motion program to load, run, and monitor.
- Able to keep PMAC's dpos close in sync with RTCS position = better tracking.

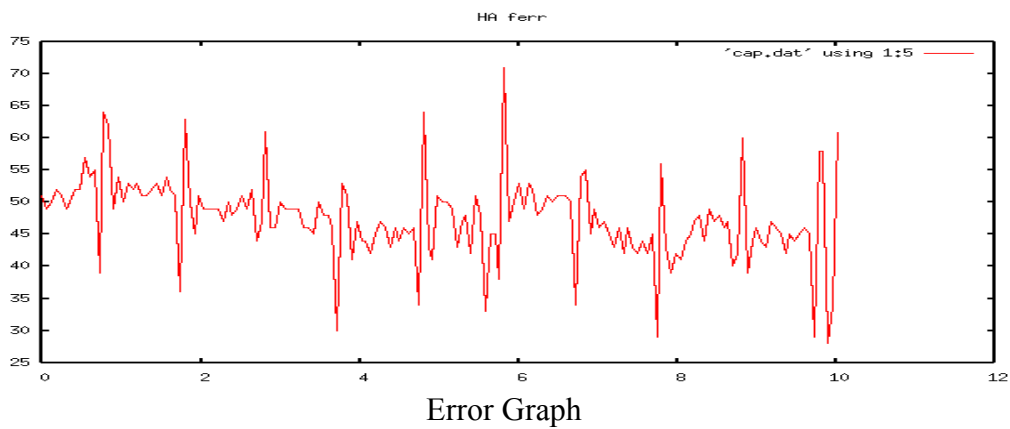
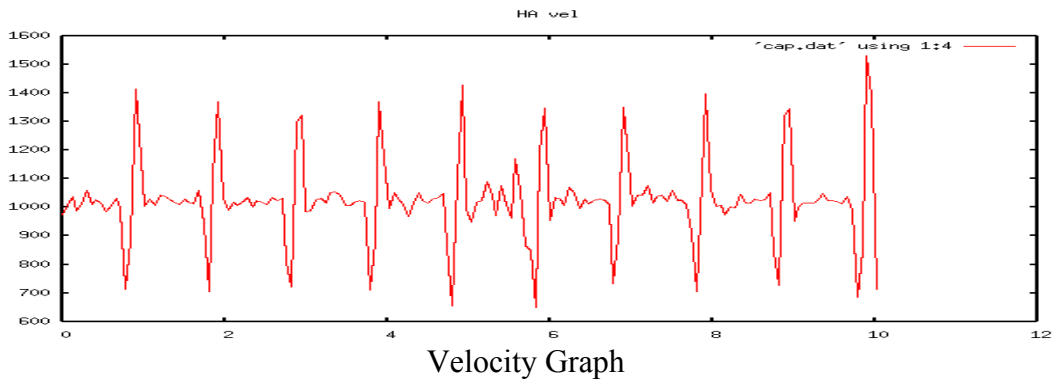
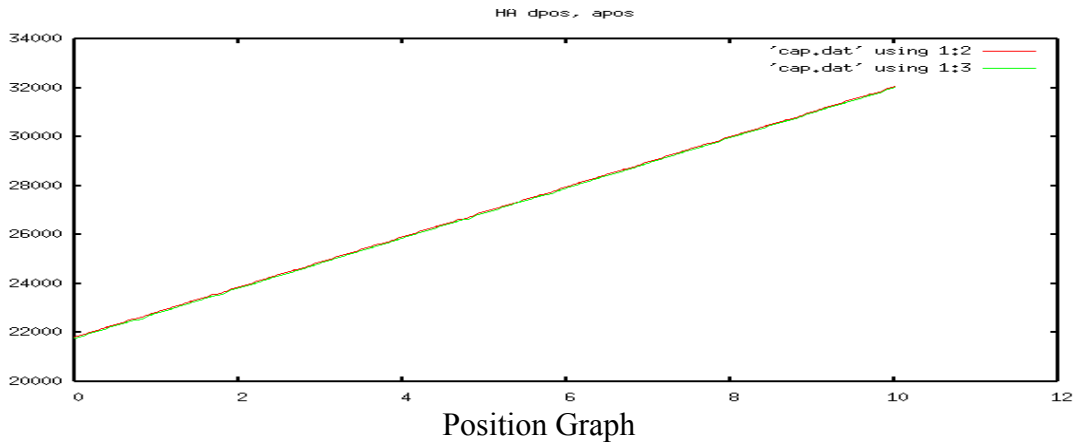
Disadvantages:

- Can only make adjustment to velocity at RTCS intervals (20Hz).
- Beam switch require more setting time.

### 3.1 PVT Test1

The following graphs illustrated the velocity and following error for a 10 AS/sec track for 10 seconds. The PVT period is 200 ms. The Linux host download a 10 element PVT table each seconds. The PMAC motion program switch to the new PVT table when downloaded.

Rate is 10 arcseconds/sec (~1000 count/second).



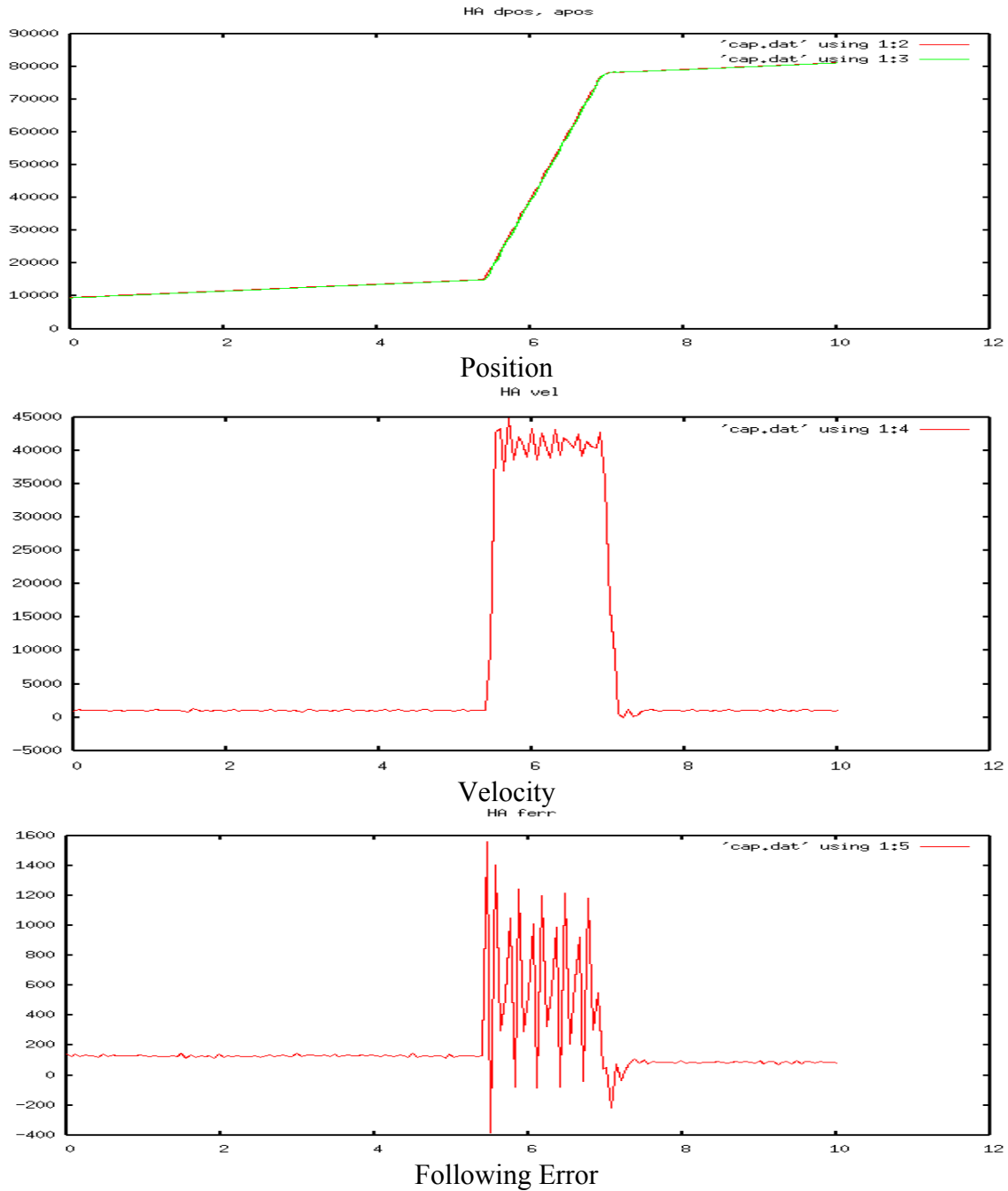
Note the spike which occurs every second, this is when the PMAC motion program tries to switch PVT tables.

### 3.2 PVT Test2

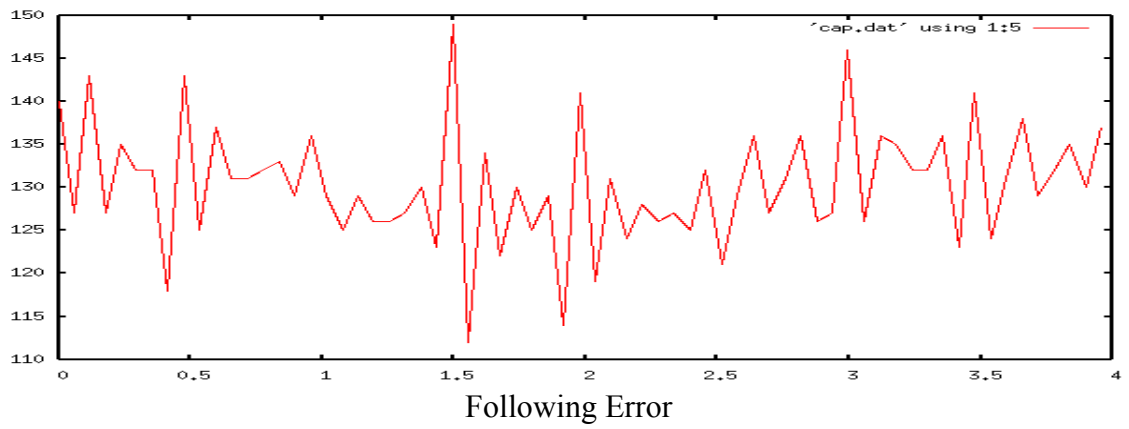
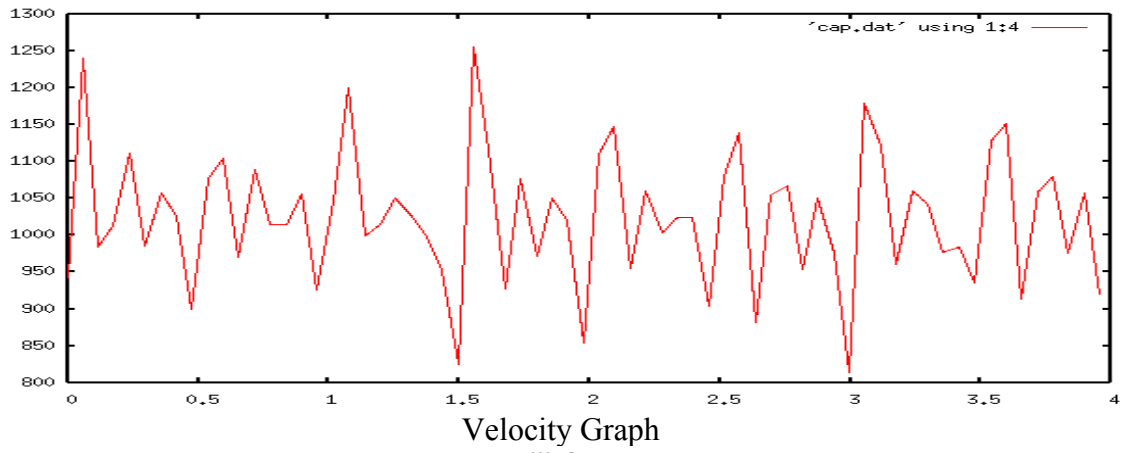
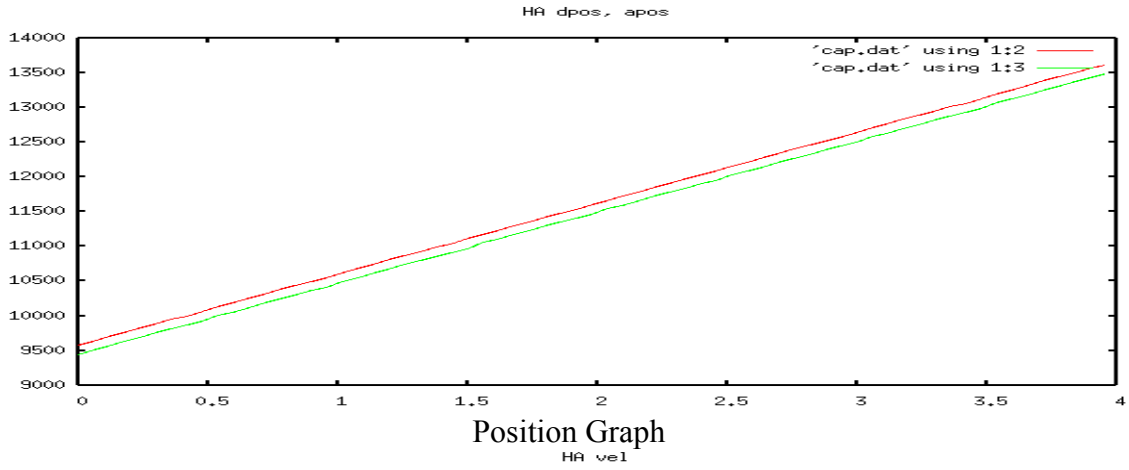
Using a PVT period of 100 ms, and switching buffer every 500 ms.

Improvement in the PVT algorithm.

The axis is tracking at 10 AS/s with a 600 AS beam switch.



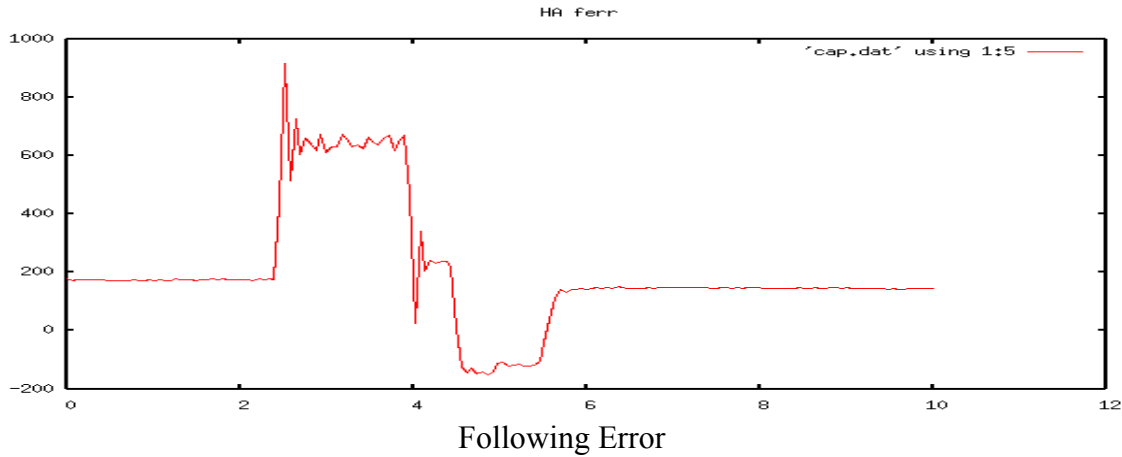
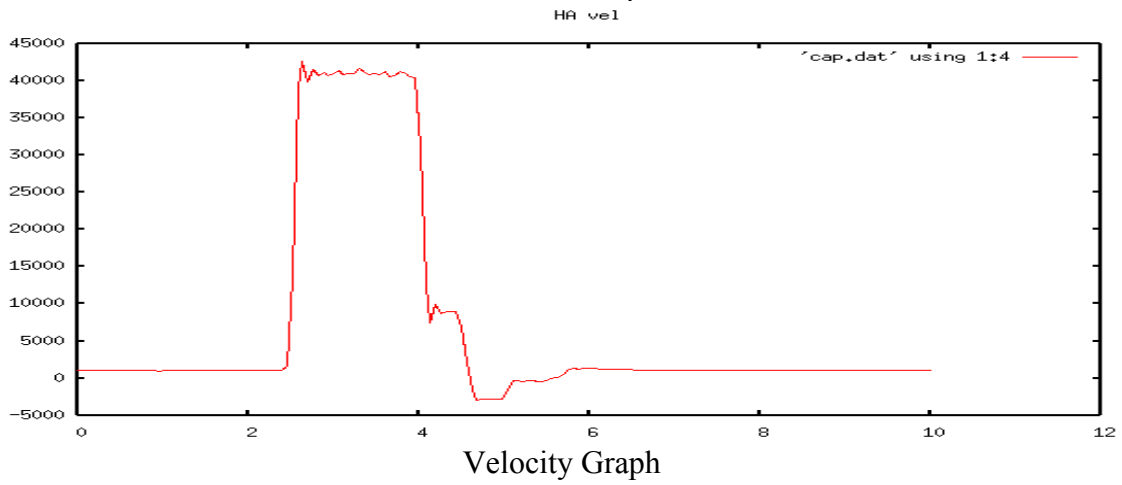
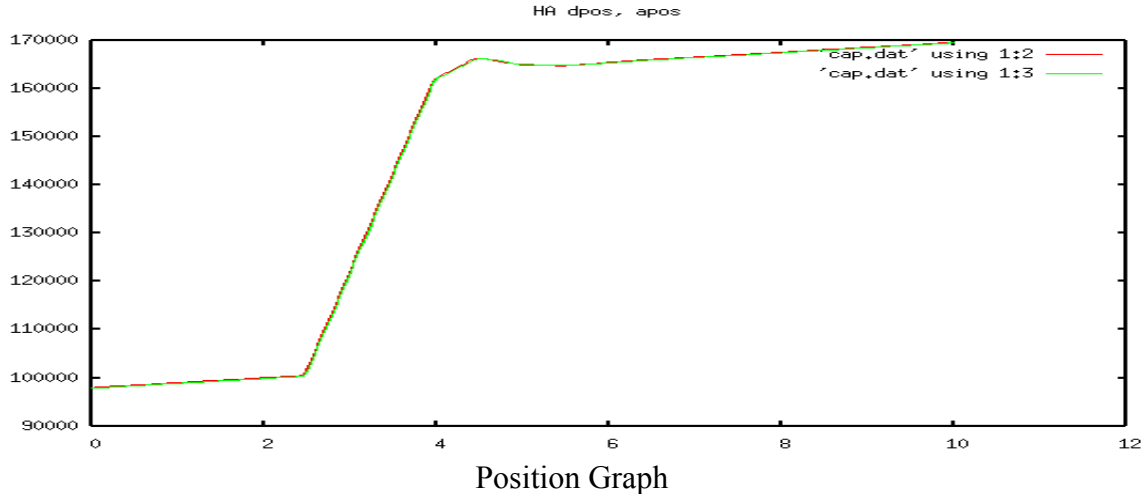
Test2 data, but without the beam switch



### 3.3 Jog Test

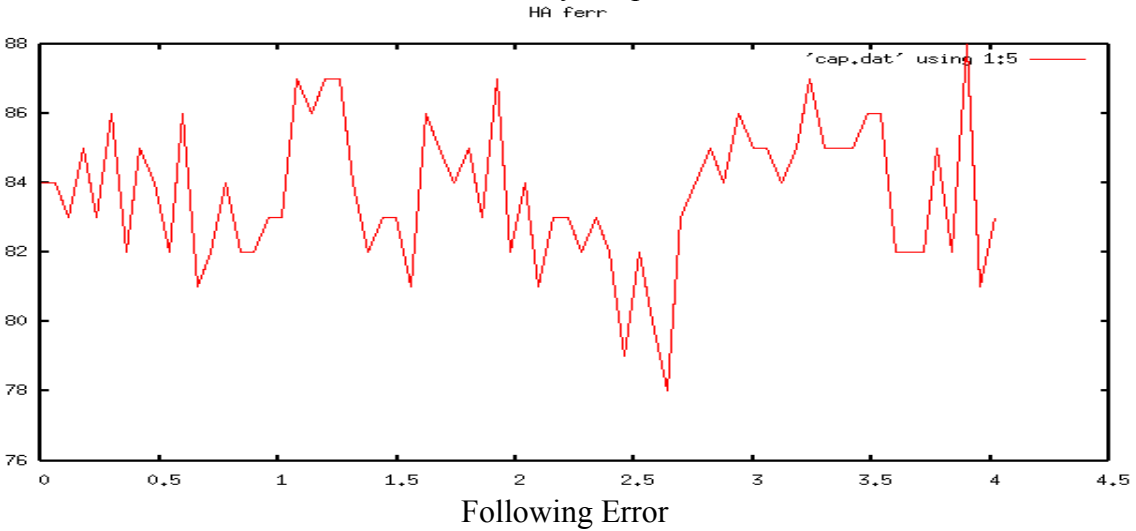
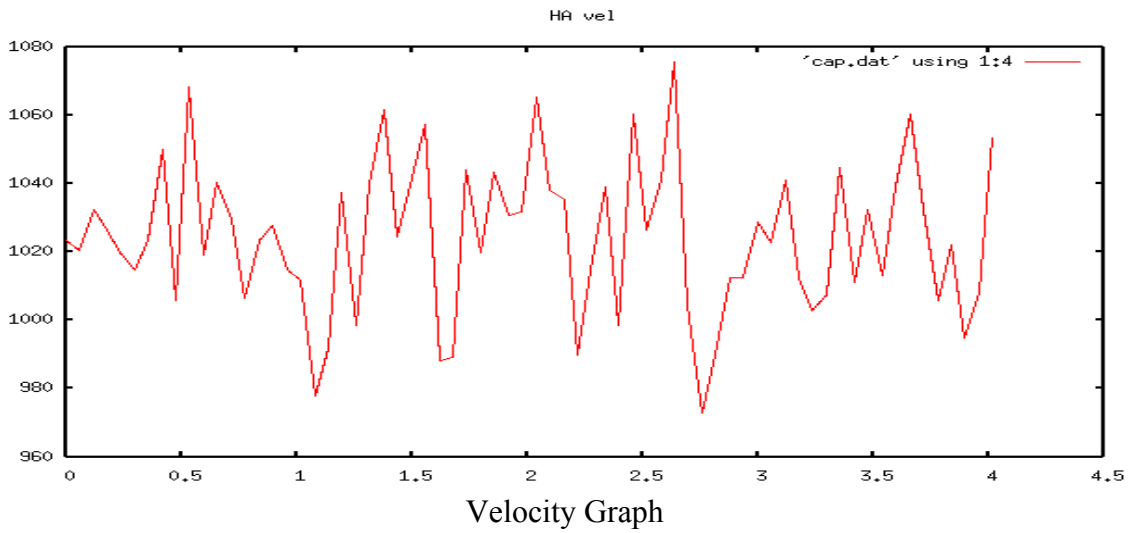
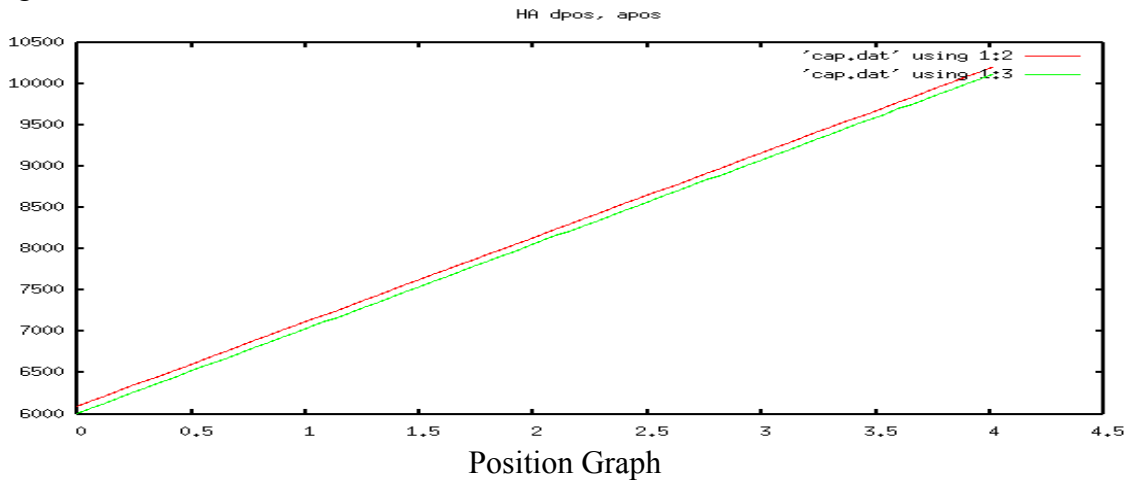
Using an update 500 ms.

The axis is tracking at 10 AS/s with a 600 AS beam switch.





Jog Test without the beam switch.



#### 4. Conclusions

- The PMAC is an acceptable solution for TCS motor control. The IRTF's servo system has some unique feature which requires customized PMAC configuration and programming.
- The requirement for dual motor output per telescope axis can be implemented using a PMAC PLC program. The algorithm is actually very simple.
- The PMAC can only accept a single velocity sensor per telescope axis. A built-in A/D input can accept the tachometer signal, but treats it as a position sensor.
- The jog mode of the PMAC is very flexible and can be used to implement all required TCS3 software mode: Track, Slew, Motor Position, Motor Velocity.
- The PMAC PVT is an alternative to using the JOG mode during tracking.
  - The advantage of PVT is it allows the RTCS to specify change in position and velocity independently of the RTCS's loop cycle of 20 Hz. PVT points can be specified, for example, at 5 ms (10 PVT trajectories per RTCS loop). PVT could improve settling time of the telescope.
  - The PVT mode does have disadvantages: It is an extremely complex mode to program. PVT requires independent motor programs for each axis, which consume CPU resources and must be started/stop on demand. The RTCS must constantly feed data to the PVT motion program, thus imposing a responds requirement on the VTCS.
  - I believe the using JOG mode for track is the more desirable solution.