MULTIPLE MIRROR TELESCOPE OBSERVATORY

Smithsonian Astrophysical Observatory *and* Steward Observatory, University of Arizona

Reply to: MMT Observatory
University of Arizona
Tucson, Arizona 85721
(602) 626-1558

MMTO Technical Memorandum 82-21

From: B.L. Ulich

Subject: Mount Control Servo Algorithm

Date: December 5, 1982

The position loops for controlling the MMT azimuth and elevation pointing are contained in FORTH software in the mount control program (MCP-12). This memo describes the algorithms used within each FORTH verb. The software is executed once for azimuth, and once for elevation, during each 50 msec tracking cycle (20 Hz). The angle units are 26th bit fractions of a circle (1 B_{26} = 0.0193 arc second).

The word which executes the mount control software is DRIVE, and this code is contained in FILE 2, LOGICAL BLOCK 49. See the attached software listing for the FORTH code. The following is a description of each verb executed in the drive loop:

WANTED

xinst = desired encoder position (B_{26})
vw = wanted target velocity (B_{26}/cycle)
= $xinst_i - xinst_{i-1}$

ENCODER $xenc_i$ = actual encoder reading (B_{26})

PERROR err_i = position error (B_{26})
= $xinst_i - xenc_i$

VELCLIP vclip = MAX [vmin, |vw| x kclip/100] (B_{26}/cycle)

vclip is the velocity command (B_{26}/cycle) at which the integral is clipped.
vmin is the minimum value allowed for vclip (B_{26}/cycle).
kclip is the percent of target velocity vw allowed for the integral. This is normally set to 10 or 20 (%) to allow for scale errors and nonlinearities between the software and the hardware velocity loop.

VELERR encvelocity = $xenc_i - xenc_{i-1}$ (B_{26}/cycle)
verr = vw - encvelocity (B_{26}/cycle)

encvelocity is the telescope velocity determined by

subtracting adjacent encoder readings.
verr is the telescope velocity error. It is not presently used in the tracking algorithm, but might be incorporated in the future.

?SLEW

If |err| > xclose (B$_{26}$)

or |vw| > vslew (B$_{26}$/cycle)

or |encvelocity| > 8 · vslew (B$_{26}$/cycle), then set slew software and set slew hardware relay (1 wslew sets relay snd sets slew-sw=1). The last condition prevents sudden changes to track speed while slewing rapidly.

Otherwise, set track software and hardware (0 wslew turns off relay and sets slew-sw=0).

The present computer system must change the hardware velocity scale with relays, in order to track accurately at sidereal rates, and also slew rapidly across the sky. In the future, this restriction may be eliminated.

?SQRT If |err| > xswitch (B$_{26}$),

then set square-root mode (sqrt-sw=1).

Otherwise set integral tracking mode (sqrt-sw=0).

?TUNE Executing TUNE will force the tracking mode to remain fixed at one of the following:

I/T → Integrate/Track
I/S → Integrate/Slew
S/T → Square-Root/Track
S/S → Square-Root/Slew

Executing NORMAL (the default condition) will allow the mode to change automatically, depending on the size of the position error and the target velocity.

VCOMMAND $vcom_i = vw + vpe$ (B$_{26}$/cycle)

vw = wanted target velocity
vpe = velocity command in square-root mode

VSQRT If sqrt-sw=1:

$$vpe = \frac{err}{|err|}\sqrt{1.8 \cdot amax \cdot |err|} \quad (B_{26}/cycle).$$

Here the velocity command is set proportional to the square root of the position error. This causes the telescope to move in a parabola, such that the position error and the velocity error go to zero at the same time.

If sqrt-sw=0 (integral mode), set vpe=0

RAMP For EL only: vmax = $avmax + $bvmax•SIN(2•EL)

$vcom_i$ = clip [$vcom_i$, vmax]

$accel_i = vcom_i - vcom_{i-1}$

$jerk_i = accel_i - accel_{i-1}$

$jerk_i$ = clip [$jerk_i$, jmax]

$accel_i = accel_{i-1} + jerk_i$

$accel_i$ = clip [$accel_i$, amax]

$accel_{i-1} = accel_i$

$vcom_i = vcom_{i-1} + accel_i$

$vcom_{i-1} = vcom_i$

This verb clips the velocity command to vmax, limits the acceleration to amax, and limits the jerk (third derivative of position) to jmax.

D/A-SET

VINTEGRAL If sqrt-sw=1 (square-root mode), set dx=0.
If sqrt-sw=0 (integral mode),
set dx=dx+err.
If wind velocity < 10 MPH
or slew-sw=1 (slew mode),
set scale=ka/100.
Otherwise set scale=ka/200.

Next let dx_i = clip [dx, vclip • 4/scale]
and let $4vdx_i = dx_i \cdot scale$.

PEFILTER $4vdx_i = 4vdx_i + [kpe/10][4vdx_i - 4vdx_{i-1}]$

$4vdx_{i-1} = 4vdx_i$

This digital filter has gain greater than unity and phase lead at frequencies near the Nyquist sampling cutoff, and must be present to stabilize the tracking loop at normal gain settings. It is the equivalent of a lag-lead active network analog filter.

#d/a = number of bits commanded to 12 bit
($\pm$5 volt) digital-to-analog converter
#d/a = vcom • kd/a + 4vdx • kd/a/4 + voff

Voff is the negative of the offset of the hardware d/a electronic voltage offset in units of d/a bits. kd/a is the scale factor (ratio of two numbers) which matches the hardware and software velocity scales. Note that 4vdx is actually four times the integral velocity command. This is done to avoid quantization errors which increase telescope tracking errors.

DITHER

For azimuth only, add dither for vw=0:

#d/a = #d/a ± amplitude

The verb dither adds the value of "amplitude" to the d/a command every +#cycles, and subtracts "amplitude" every -#cycles. For all other azimuth cycles and for all elevation cycles, zero is added. This dither reduces the amplitude of azimuth position errors at low speeds due to stick-slip oscillations.

#d/a = clip [#d/a, full-scale]
Full-scale is the maximum number of bits in the d/a (presently 2047).

D/A-SET adds the wanted target velocity, the square-root term, the integral term, and an offset together, scales the sum so that the hardware and software velocity scales match, and sends out a velocity command voltage to d/a # 0 for azimuth and to # 1 for elevation.

AZIMUTH: + velocity
= - NPL encoder reading
= + vcom
= + d/a

ELEVATION: + velocity
= + NPL encoder reading
= + vcom
= + d/a

D/A #	FUNCTION
0	AZ velocity command
1	EL velocity command
2	AZ or EL position error (set by AZERR or ELERR)
3	Instrument rotator position command (NOCHOP), or To instrument computer for IR Photometer (CHOP)

NTH (array index)	MODE
0	AZ Track
1	AZ Slew
2	EL Track
3	EL Slew

() (axis vector index)	Axis
0	AZ
1	EL

MMT SERVO PARAMETERS (December 5, 1982)

2^{-26} Circle = B_{26} = 0.01931197 arc second
1 Cycle = 0.050 second (20 Hz)

CONSTANTS	$avmax	1500		(B_{26}/cycle)
	$bvmax	12481		(B_{26}/cycle)
	full-scale	2047		(d/a bits)
VARIABLES	amplitude	50		(d/a bits)
	+#cycles	3		(cycles)
	-#cycles	- 3		(cycles)
	kclip	20		(%)
VECTORS		AZ[()=0]	EL[()=1]	
	ka	10	12	(sec^{-2})
	kpe	80	65	(-)
	vmin	75	50	(B_{26}/cycle)
	voff	- 2	- 2	(d/a bits)
	vslew	466	116	(B_{26}/cycle)
	xclose	8192	8192	(B_{26})
	xswitch	2500	1000	(B_{26})

ARRAYS nth=	0	1	2	3	
axis:	az	az	el	el	
speed:	track	slew	track	slew	
amax	25	20	20	30	($B_{26}/cycle^2$)
jmax	25	6	20	9	($B_{26}/cycle^3$)
kd/a	408/100	125/961	1500/100	140/955	(d/a bits/B_{26}/cycle)
vmax	520	13981	130	13981	(B_{26}/cycle)

PARAMETER ADJUSTMENT PROCEDURES

If hardware changes require software adjustments to optimize performance, they can be made quite simply. First, track a fixed position. Then change VOFF so that 4VDX averages zero. This will zero the hardware, so that a zero velocity software command will actually hold the telescope fixed. Next, adjust the d/a scale factor while tracking a moving target. Either SEEK a star from a catalog, or type SAT-TEST and then CHASE to track a constant-velocity dummy satellite. Next, adjust KD/A so that the average value of 4VDX is again zero. In SLEW mode, one can adjust KD/A while slewing in azimuth, so that the maximum encoder velocity is ±90 arc minutes/second. The position loop gain is set by the acceleration constant KA. This parameter should be adjusted while observing step responses, or by measuring the RMS tracking error. This can be done by first selecting the axis (AZERR or ELERR), and then typng BCOLLECT. The mount control program will then store 512 values of position error in an array (it will take 25.6 seconds to complete this task). One can print out the entire array by typing PBARRAY. To find the mean value and the RMS value in units of 24th bit fractions of a circle (B_{24} = 0.07725 arc second), type 512 RMS. When the wind is calm (< 10 MPH) or when slew speed is desired, the gain of the position loop is cut in half. This procedure has been empirically determined to minimize the RMS tracking errors under all conditions. That is, when it is windy, and the dominant disturbances are applied by wind torques to the telescope, a higher gain will result in better tracking. However, under calm conditions, and also at high speeds, the dominant disturbances are produced in the driving gears and bearings, and a lower gain will produce slightly better tracking. Similarly, the bandwidth of the software position loop can be changed by varying the parameter KPE. Larger values of KPE correspond to smaller values of the equivalent pole frequency, and therefore to reduced bandwidth for the overall loop. VMAX is set by the maximum allowable velocity, AMAX by the maximum acceleration, and JMAX is the maximum allowable jerk. Note that none of these limits affect the integral velocity command term. They only affect the sum of the target velocity and the square-root term. This will not cause any problems for normal trajectories, but tracking the Space Shuttle near the zenith in Integrate/Slew mode might cause an overspeed condition. In elevation, VMAX varies as SIN(2•EL) so that the telescope does not have significant kinetic energy near its horizon and zenith limit switches. The reduced velocity also allows the air bags to maintain proper primary mirror support at all times. The azimuth dither amplitude is adjusted empirically while observing the aximuth position error in a fixed position mode. The dither frequency is 20/(3+3) = 3.3 Hz which is nearly the locked-rotor resonant frequency in azimuth. Other frequencies are not nearly as effective in reducing the stick-slip oscillation amplitude.

When acquiring a distant target, the mount first accelerates (under JMAX and then AMAX limits) to VMAX, then runs at constant speed until it nears the target. Then it will decelerate constantly (at approximately 0.95 AMAX), while under the square-root mode of control, until the position error is less than XSWITCH. Next the square-root term is set to zero, and the integral term is enabled which quickly produces a zero mean tracking error. The overshoot is minimized by clipping the integral to a maximum value determined by VMIN or by KCLIP. Don't make either too small, or nonlinearities could cause the telescope to lose track, and drift away from the target.

Multiple Mirror Telescope Observatory

Super-FORTH II v. 25 Aug. 82 ****   Date: 22 DEC 1982

File 2 Mount-Control   Logical block 40

block # 217

```
( High Resolution D/A Test Software - 09/12/82 - BLU )

code eldac 0 s) ld @55 dob 1 s) ld @55 doa @55 nio 1p 2pop

0 variable #eldac 0 ,

: hi-res-d/a extend 8437 -1000 m*/ 131072. d+ 2dup #eldac 2!
         eldac ;



vector last-input 0 , 0 ,    ( SP last input to digital filter )

: pefilter last-input @ kpe @ -10 */ swap dup last-input !
         kpe @ 10 + 10 */ + ;
```

File 2 Mount-Control   Logical block 41

block # 102

```
( Adaptive Mount Servo Control  9/17/81)

@23 constant 'd/a
code d/a   0 s) ld  'd/a dob  1 s) ld  'd/a doa  2pop

: $newvmax  () @ if 2dup $sin.cos *. 2 *  $bvmax *. $avmax
         + 3 vmax ! then ;
(  0 variable enc-msk
: encm enc-msk @ com 4 * swap and swap ; )

: encoder  enc 2@ ( encm ) 10 d*2 p1024 2@ d- $sin.cos swap
         drop a1024 @ *. enc 2@ ( encm ) rot m+ 2dial 2@ d-
         () @ 0= if dminus 512 + <1. side @ ccw -
         if dup 256 < if 1024 + then else dup 512 >
         if 1024 - then then then 2dup xenc 2@ xenclast 2!
         xenc 2! $newvmax ;
```

Multiple Mirror Telescope Observatory

Super-FORTH II v. 25 Aug. 82 ****   Date: 22 DEC 1982

File 2 Mount-Control   Logical block 42

block # 103

```
( Adaptive Mount Servo Control  9/16/81)
2vector velocity 0. , ,  0. , ,     vector vptr 0 , 0 ,
2array vstring   40 allot       0 vstring 40 erase
code *10+  0 s) ld  0 0 add  0 3 mov  0 0 add *2  3 0 add
           1 s) ad  binary  Here #10 ,
code +10mod  0 s) ld  0 0 inc  ( 10) su # scc  0 0 sub  put
: sift    2dup 2dup vptr @ +10mod dup vptr ! () @ *10+
          vstring 2exch d- 20 10 m*/ velocity 2! ;
: nth     () @ 2* slew-sw @ + ;

: perror  xinst 2@ d- dminus 2dup derr 2!
          single dup err ! d/a# @ () @ - if drop else 4 /
          full-scale clip dup 2 d/a bstuff then ;

: azerr   0 d/a# ! ;
: elerr   1 d/a# ! ;
```

File 2 Mount-Control   Logical block 43

block # 66

```
( Servo Tuning Words - 04/29/82 - BLU )

: tune    1 'tune ! ;
: normal  0 'tune ! ;

: i/t     'sqrt 0! 'slew 0! ;
: i/s     'sqrt 0! 1 'slew ! ;
: s/t     1 'sqrt ! 'slew 0! ;
: s/s     1 'sqrt ! 1 'slew ! ;

: ?tune   'tune @ if 'sqrt @ sqrt-sw ! 'slew @ slew-sw !
          then ;
```

Multiple Mirror Telescope Observatory

Super-FORTH II v. 25 Aug. 82 ****   Date: 22 DEC 1982

File 2 Mount-Control   Logical block 44
block # 104

```
( Adaptive Mount Servo Control - 12/02/82 - BLU )

: wanted  xinst 2! x0 2@ 2dup x0last 2exch a- single poff @ +
          vwlast @ swap over - vwdotmax @ clip + dup vw !
          vwlast ! ;

: ?slew   err @ abs xclose @ > vw @ abs vslew @ > or
          encvelocity @ abs vslew @ 8 * > or
          if   1 wslew
          else 0 wslew then ;
```

File 2 Mount-Control   Logical block 45
block # 195

```
( Adaptive Mount Servo Control - 12/07/82 - BLU )

: vsqrt   sqrt-sw @ if nth amax @ 18 10 */ derr 2@ dabs
          limsqrt 4 pick 1 swap m*/ dmin rot 1m*/ limsqrt dmin
          sqrt err @ dup abs */ else 0 then dup vpe ! ;

: vintegral sqrt-sw @ if 0 else ka @ 'wvel @ 10 <
          slew-sw @ or if 200 else 100 then scale 2!
          err @ dx @ extend rot m+ single vclip @
          scale 2@ swap */ 4 * clip then dup dx !
          scale 2@ */ pefilter dup 4vdx ! ;
```

Multiple Mirror Telescope Observatory

Super-FORTH II v. 25 Aug. 82 ****   Date: 22 DEC 1982

File 2 Mount-Control   Logical block 46

block # 194

```
( Telescope Servo Loop - 12/07/82 - BLU )

: ?sqrt   err @ abs xswitch @ > if 1 sqrt-sw !
          else 0 sqrt-sw ! then ;

: velerr  verr @ verrlast ! vw @ xenc 2@ xenclast 2@ d-
          single dup encvelocity ! - verr ! ;

: velclip vw @ abs kclip @ 100 */ dup vmin @ < if drop vmin @
          then vclip ! ;

: vcommand vw @ ( vintegral + ) vsqrt + ;
```

File 2 Mount-Control   Logical block 47

block # 218

```
( AZ Dither Software - 09/15/82 - BLU )

 35 variable amplitude     ( SP # of d/a bits dither each way )
  0 variable counter          ( SP # of steps )
  3 variable +#cycles  ( SP # of +clock cycles between dithers )
 -3 variable -#cycles  ( SP # of -clock cycles between dithers )
  1 variable step             ( SP step size )
: >or= 2dup > rot rot = or ;
: <or= 2dup < rot rot = or ;

: dither  az slew-sw @ not if vw @ 0= if
          counter @ step @ + dup counter ! +#cycles @ >or= if
          amplitude @ + 0 counter ! -1 step ! then counter @
          -#cycles @ <or= if amplitude @ - 0 counter ! 1 step !
          then then then ;
```

Multiple Mirror Telescope Observatory

Super-FORTH II v. 25 Aug. 82 ****   Date: 22 DEC 1982

File 2 Mount-Control   Logical block 48

block # 179

```
( A/D Code - 12/07/82 - BLU )
octal here assembler 21 nio 0d rti @21 interrupt
here assembler 21 nio 0d rti 0 interrupt
code goo 0 s) 0 lda 21 doa 1b pop
code gett 21 dic push
200 constant 0ch 210 constant 10ch
: drag  1 0 do loop ;
: 0a/d  0ch goo drag gett ;
: 1a/d  10ch goo drag gett ;
decimal
```

File 2 Mount-Control   Logical block 49

block # 105

```
( Telescope Servo Loop - 12/07/82 - BLU )
: ramp    nth vmax @ clip extend vcom-last @ extend m- alast
          @ - nth jmax @ clip alast @ + nth amax @ clip dup
          alast ! vcom-last @ + dup dup vcom ! vcom-last ! ;
: d/a-set nth kd/a 2@ */ vintegral nth kd/a 2@ swap 4 / swap */
          + voff @ + () @ if else dither then
          full-scale clip dup #d/a ! () @ d/a ;
: drive   wanted encoder sift perror velclip velerr ?slew
          ?sqrt ?tune vcommand ramp d/a-set ;
: rmserr  az err @ el cos @ +1 */ dup m* err @ dup m* d+
          sqrt  raderr ! ;
```