

TECHNICAL REPORT

**FINE IMAGE MOTION CONTROLLER FOR KODAIKANAL TUNNEL
TELESCOPE**

by

**K. C. Thulasidharen., K. Prabhu., K. E. Rangarajan., N. Vasantharaju.,
B. Prabhu Ramkumar**

Abstract:

Automation of the spectropolarimetric observations at Kodaikanal tunnel telescope needs placement of the Sun's image on the spectrograph slit and scanning it across the region of interest. This requirement can be achieved by controlling the tilt of the secondary mirror. Hence a fine image motion controller for secondary mirror in Kodaikanal Tunnel Telescope was developed and tested with telescope. A slo-syn stepper motor drive model SS2000D6 was used for driving the Steper Motor HS50. A multi function PCI board from measurement computing model PCI 2517 was used in giving signals to the drive. Control software and user Interface were developed. The system was calibrated to facilitate regular observations.

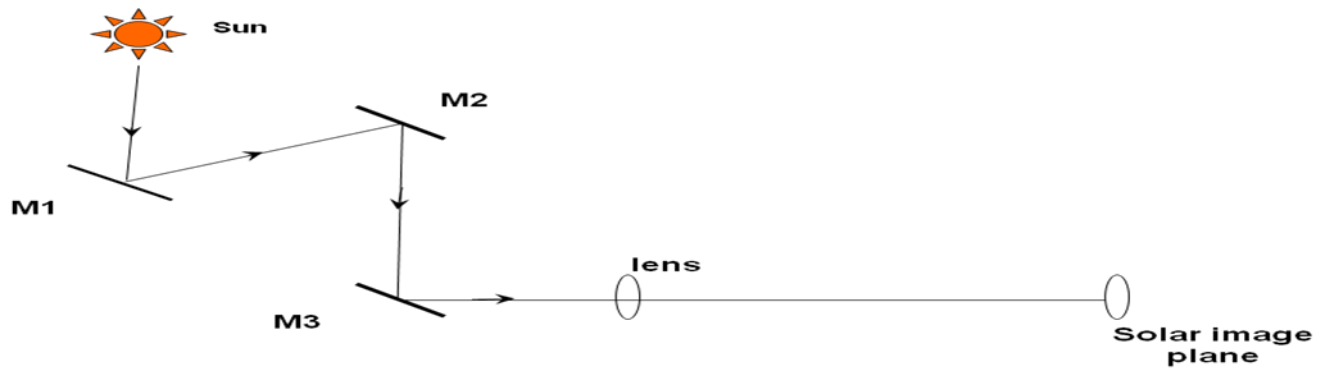
Key words: Coelostat, Stepper motor, Stepper motor drive, AC Servomotor, PCI add on board, magnetic clutch.

Introduction :

In the recent years, a dual beam polarimeter was developed to be used along with the spectrograph at Kodaikanal Tunnel telescope (Nagaraju, K. et. al 2007, Nagaraju et. al. 2008 a) to measure the magnetic fields (Nagaraju, K. et al 2008b) present in the Sun. Usually the active regions on Sun occupy several arcseconds requiring the spectrograph slit to scan across the region of interest and measurements to be taken at each position of the slit. So far, at Kodaikanal, this operation was carried out manually. That is, the secondary mirror was moved using a coarse controller and positioning the image on the slit. It was decided to automate this operation to improve temporal cadence and accuracy.

Hence, a fine image motion controller for secondary mirror in Kodaikanal Tunnel Telescope was developed and tested with telescope.

Kodaikanal Coelostat:



Coelostat mirror arrangement

1. Tracking one axis
2. Guiding two axis —RA and DEC

Figure 1.

The first mirror M1 tracks the sun. The second mirror guides the image in the image plane in RA and DEC directions.

A Siemens servo motor drives the track mirror. An AC motor-stepper motor-magnetic clutch unit drives the secondary mirror.



Fig: RA guide axis

Figure 2.

System components:

Stepper motor

- Stepper motor model
- HS 50
- Superior electric
- 3 V 4 Amp
- 85 Oz-inch
- 200 Steps/revolution



Figure 3. Stepper motor and its specifications.

Stepper motor drive

- Compatible stepper drive
- Model SS2000D6
- SLO-SYN

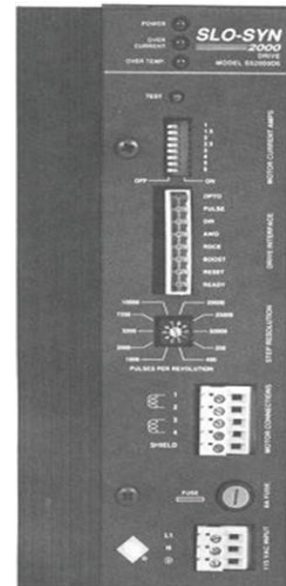


Figure 4. Stepper motor drive unit.

Slo-Syn 2000 drives are bipolar speed adjustable, two phase drives which use power MOSFET and IGBT devices. They can be set to operate

a stepper motor in full or half steps or microsteps. The maximum running speed is 10000 full steps per second.

1. Switch selectable current level of 0.5 through 6 amps depending on the unit selected.
2. Latched short circuit protection.
3. Inputs are optically isolated.
4. Self test function
5. Boost current reduced current

PCI 2517 board:



Figure 5. PCI 2517 card which has 16 bit digital I/O, Counters ADC, DAC and Software drivers.

SYSTEM BLOCK DIAGRAM:

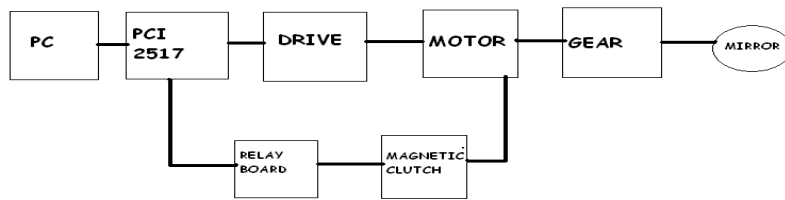


Figure 6. System block diagram.

An AC motor is used for coarse motion of the image. A handset with switches moves the image in N-S and E-W directions; normally AC motor is engaged to a secondary mirror.

When 24 volt DC is applied to the clutch stepper motor is engaged to the gear.

Figure 7. Power supply and digital I/O.



Controller working:

The controller and PC is set up near the solar image plane of the telescope. The RA stepper motor connection is given to the controller and 24 volt DC is applied to the clutch.

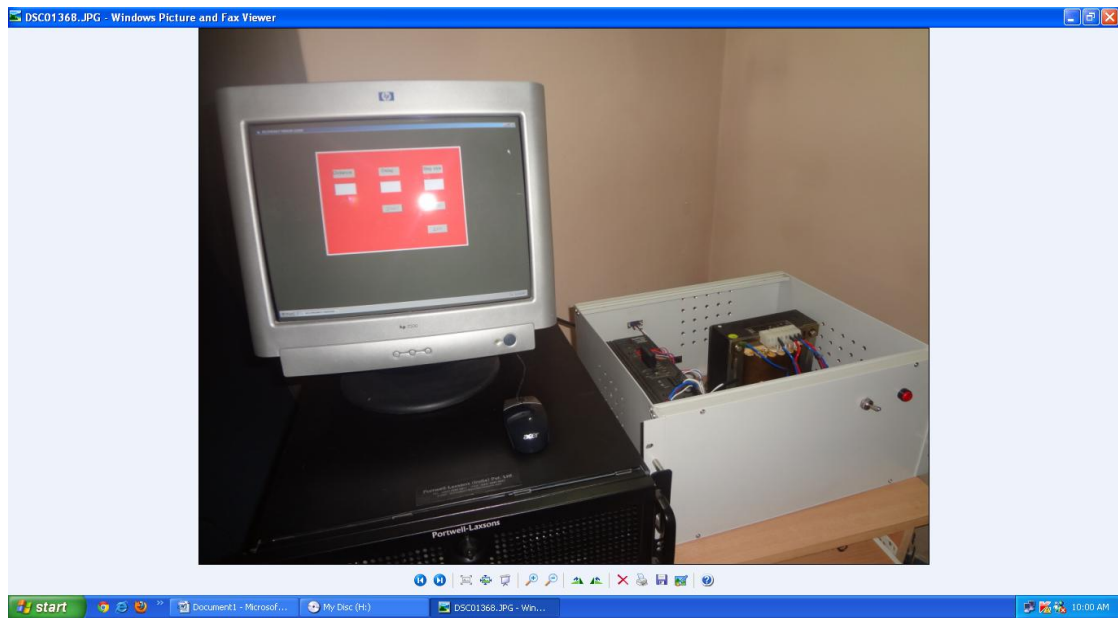


Figure 8. Telescope testing

A known number of pulses is applied to the stepper motor drive and the image moves. The fine image movement distance is entered in the user interface screen and click start. The user interface screen is shown in figure 9.

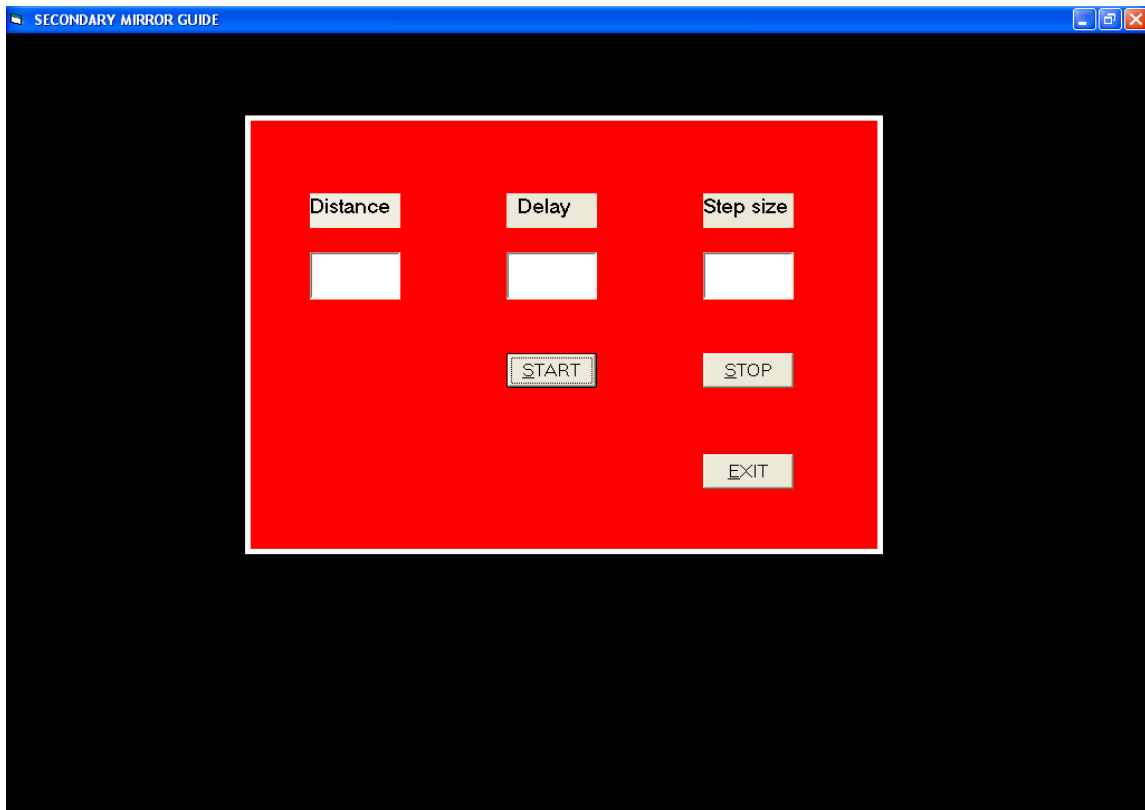


Figure 9. Screen shot of GUI. Delay is chosen between successive movements. Step size is the number of steps. Distance is the distance to be moved in the image plane.

Observational setup:

To test the fine image motion controller, a large graph sheet was firmly fixed in the image plane of the telescope. The east and west limbs were marked on the graph paper. The plate scale of the telescope is 5.5 arcseconds per mm. The image was moved using the fine motion controller by sending known number of pulses to the motor. The change in position of the limbs were measured. The number of pulses versus the movement of the image is plotted in figure 10.

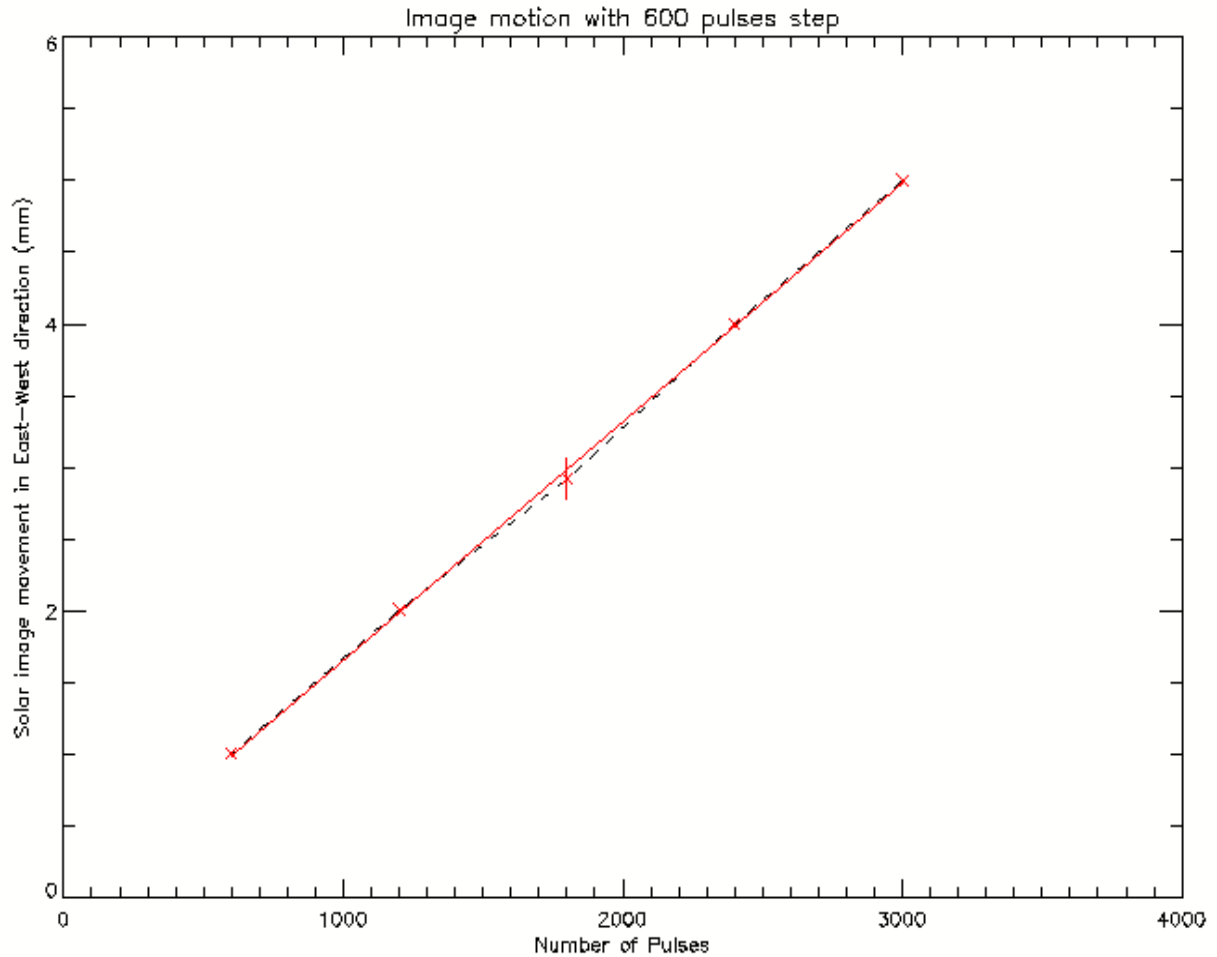


Figure 10. Calibration curve. This shows 600 pulses per mm with an error of 0.14 mm.

We obtain 600 pulses per mm with an accuracy of ± 0.14 mm. This corresponds to nearly 0.5arcsec which is of the same order of the diffraction limit of the telescope. Seeing and other distortions are usually more than the above error.

Acknowledgements:

We thank Mr. Sri Harsha of the electronics lab for wiring the unit. And Devendran for helping us to take observations.

References :

On the performance of a dual-beam polarimeter at Kodaikanal Tower Telescope by - Nagaraju, K.; Sankarasubramanian, K.; Rangarajan, K.E.; Ramesh, K.B.; Singh, Jagdev; Devendran, P.; Hariharan 2008, BASI, 36, 99

An efficient modulation scheme for dual beam polarimetry by – Nagaraju, K.; Ramesh, K. B.; Sankarasubramanian, K.; Rangarajan, K.E., 2007 BASI, 35, 307

On the Weakening of the Chromospheric Magnetic Field in Active Regions - Nagaraju, K.; Sankarasubramanian, K.; Rangarajan, K.E., 2008, ApJ, 678, 531

Quick Start Guide – Measurement Computing

Installation instructions for Slo-Syn SS2000D3 and SS2000D6 packaged drives

- Superior Electric

APPENDIX

'Option Explicit

'Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Dim DataValue As Integer

Dim Distance As Single

Dim Npulse As Single

Dim Delay As Single

Dim I, J As Integer

Dim Size As Integer

Const BoardNum = 0 ' Board number

Const PortNum% = FIRSTPORTB ' use first digital port

Const Direction% = DIGITALOUT ' program digital port B for output

'Option Explicit

Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Sub CmdEnd_Click()

 ULStat% = cbDeclareRevision(CURRENTREVNUM)

 ULStat% = cbErrHandling(PRINTALL, DONTSTOP)

 If ULStat% <> 0 Then Stop

 DataValue = &H0

 ULStat% = cbDConfigPort(BoardNum, PortNum%, Direction%)

 If ULStat% <> 0 Then Stop

 ULStat% = cbDOut(BoardNum, PortNum%, DataValue%)

 End Sub

Private Sub CmdExit_Click()

End

End Sub

Private Sub CmdStart_Click()

' declare revision level of Universal Library

For I = 1 To Npulse

ULStat% = cbDeclareRevision(CURRENTREVNUM)

' Initiate error handling

' activating error handling will trap errors like

' bad channel numbers and non-configured conditions.

' Parameters:

' PRINTALL :all warnings and errors encountered will be printed

' DONTSTOP :if an error is encountered, the program will not stop,

' errors must be handled locally

ULStat% = cbErrHandling(PRINTALL, DONTSTOP)

If ULStat% <> 0 Then Stop

' If cbErrHandling% is set for STOPALL or STOPFATAL during the program

' design stage, Visual Basic will be unloaded when an error is encountered.

' We suggest trapping errors locally until the program is ready for compiling

' to avoid losing unsaved data during program design. This can be done by

' setting cbErrHandling options as above and checking the value of ULStat%

' after a call to the library. If it is not equal to 0, an error has occurred.

' configure FIRSTPORTA for digital output

' Parameters:

' BoardNum :the number used by CB.CFG to describe this board

' PortNum% :the output port

```
' Direction% :sets the port for input or output
ULStat% = cbDConfigPort(BoardNum, PortNum%, Direction%)
If ULStat% <> 0 Then Stop
DataValue = &HA
    ULStat% = cbDOut(BoardNum, PortNum%, DataValue%)
        Sleep 5 'sleep
ULStat% = cbErrHandling(PRINTALL, DONTSTOP)
If ULStat% <> 0 Then Stop
DataValue = &HB
ULStat% = cbDOut(BoardNum, PortNum%, DataValue%)
Sleep 5 'sleep
Next I
Sleep Delay 'Observation

End Sub

Private Sub txtNPulse_Change()
Npulse = Val(txtNPulse.Text)
End Sub
```