Design and development of telescope control system and software for the 50/80 cm Schmidt telescope

T. S. Kumar[a,b] and R. N. Banavar[a]

[a]Indian Institute of Technology Bombay, Mumbai 400076 India;
[b]Aryabhatta Research Institute of observational sciencES, Nainital 263129 India

ABSTRACT

In this paper, we describe the details of telescope controller design for the 50/80 *cm* Schmidt telescope at the Aryabhatta Research Institute of observational sciencES. The GUI based software for commanding the telescope is developed in Visual C++. The hardware architecture features a distributed network of microcontrollers over CAN. The basic functionality can also be implemented using the dedicated RS232 port per board. The controller is able to perform with negligible rms velocity errors. At fine speeds limit cycles are exhibited due to nonlinear friction. At speeds over 3.90×10^{-02} $radians/sec$, the PI controller performs with peak errors less than 1%.

Keywords: Telescope controller, CAN, GUI, PI controller

1. INTRODUCTION

The 50/80 cm Schmidt telescope at ARIES is a f/1 equatorial system with a usable aperture of 50 *cm* diameter.[1] Each of its axes is driven by two torque DC motors and the motion is sensed by an on-shaft absolute encoder and an incremental encoder coupled through a friction roller. This telescope being relatively small a simple distributed control architecture based on low cost microcontrollers is being developed. For small systems embedded controller reduces the design complexities and offers reasonably good performance. The distributed control system based on microcontrollers has been used in many telescope and mirror control systems.[2–5] Most of the embedded controllers offer dedicated hardware modules for interfacing different sensors, generating waveform for the actuators and performing serial communication. A set of prototype boards based on 8 and 16 bit PIC microcontrollers comprising the telescope control hardware has been designed. Both RS232 and CAN functionalities for communication have been implemented on each board. Currently, the basic telescope controller using five boards interfaced to PC directly through five RS232 serial ports are enabled for characterizing tracking and pointing performances. This basic system provides PI control for slew, set, guide, fine and tracking motions, interfaces with the encoders and controls the stepper motor of the focussing system. A remote PC hosting the GUI based telescope control software is used only for exchanging the user commands otherwise the telescope controller operates stand alone. This software developed under Microsoft Visual C++ runs multiple threads for hosting GUI, hardware communications and data logging service. At speeds over 3.90×10^{-02} $radians/sec$, the PI controller performs accurately within an error band of 1%. At finer speeds the telescope exhibits very prominent limit cycles due to nonlinear friction[6] which must be removed with an additional nonlinear compensator.[7] Efforts are being made for reducing the friction mechanically by fine tuning the gear alignment and bearing preload. The optical performance of the system is yet to be determined which would help in evaluating the overall performance.

The paper is organized as follows: we present the overview of the telescope structure and mechanism in Section 2, the electronics and hardware aspects are discussed in Section 3, an overview of the software aspects is given in Section 4 and finally concluding remarks are presented in Section 5.

Further author information: (Send correspondence to T.S.K.)
T.S.K.: E-mail: kumar@sc.iitb.ac.in
R.N.B.: E-mail: banavar@sc.iitb.ac.in

*Author T. S. Kumar is on leave from Aryabhatta Research Institute of observational sciencES.

Ground-based and Airborne Telescopes IV, edited by Larry M. Stepp, Roberto Gilmozzi, Helen J. Hall,
Proc. of SPIE Vol. 8444, 84445O · © 2012 SPIE · CCC code: 0277-786/12/$18 · doi: 10.1117/12.925243

2. OVERVIEW OF THE 50/80 CM ARIES SCHMIDT TELESCOPE

2.1 Telescope Structure

The telescope as shown in Figure 1 is located in the Northern hemisphere at a latitude of 29deg 22min and a longitude of 79deg 27min and supported on an off-axis equatorial mount. For smaller telescopes equatorial mount is preferred as it simplifies the drive and servo controller design. If the polar axis is aligned accurately the tracking can be achieved by controlling the R.A. (right accession) axis alone. The off-axis tube is counter balanced by dead weights and gear box mounted on the other side. Inside the tube, at the prime focus a 9 μm pixel, 4K $\times$ 4K Kodak IMGX 16801E CCD providing a pixel-scale of 3.6 $arcsec/pixel$ will be used. The main objective of the controller is to provide a desired pointing accuracy of 10 $arcsec$ for both the axes and to maintain tracking accuracy within 1 $arcsec/sec$ for the R.A. shaft.

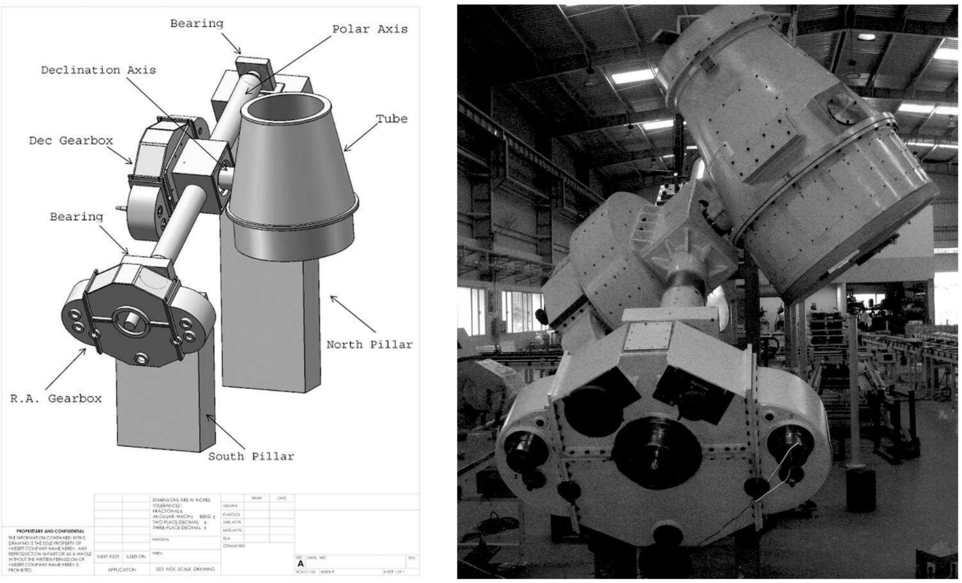


Figure 1. Left: The schematic of the 50/80 cm ARIES Schmidt Telescope; Right: Telescope being tested on shop floor on steel piers

2.2 Drive Mechanism

The gearbox and hence drive mechanism for both R.A. and declination are identical. The R.A. gear (Figures 1 and 2) box housing is anchored to the south pillar with the help of stiff torque arms.[6] A three stage gear train (27:113, 21:123 and 17:250) is used to provide an overall reduction of 360.489:1 between each motor and the output shaft. To avoid backlash in the gears each axis is driven by a set of two identical motors coupled to the bull gear of polar axis through two identical gear trains. The two motors provide unequal torque to the bull gear such that if one motor leads the motion in forward direction the other leads the motion in the reverse direction. This technique ensures that the teeth contact is never lost between any of the gears during motion reversal and thus a dead zone is avoided. A friction roller assembly is available on the top of the gear box for mounting an incremental encoder. This arrangement allows backlash free accurate velocity measurements through a reduction of 735:30 (or 1:24.5). For position measurements, an absolute encoder is mounted directly on the extended shaft of the main axis at the center of the gearbox. Provisions have been made for mounting incremental encoders or tachometers directly on the four motors and auxiliary encoders at the bottom of the two gear boxes through a reduction of 1:10.8.

Table 1 shows the mechanical specifications of the telescope. These specifications are used for designing the PI controller.

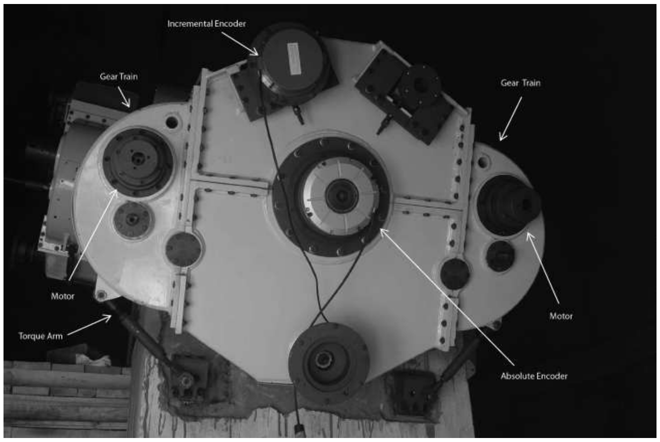


Figure 2. R.A. gear box with the control components

Table 1. Parameters for the R.A. and declination drives

Parameter	Polar axis	Declination Axis	Units
Inertia	2.2×10^6	5.79×10^5	N-mm sec^2
Stiffness	3.08×10^9	2.46×10^9	N-mm/radians
Lock Rotor Freq.	5.96	10.37	Hz
Slewing speed	12.583	12.583	radians/sec
Angular acceleration	12.583	12.583	radians/sec^2
Load torque at start	933	316	N-mm
Counter torque (25%)	233	79	N-mm
Starting Toque	1166	395	N-mm
Running torque	791	186	N-mm

3. ELECTRONICS AND HARDWARE DESIGN

3.1 Telescope Control System

The overall telescope control system is broadly divided into stand alone telescope control hardware and PC based software for user command. The controller hardware consists of multiple low cost PIC microcontrollers performing tasks like interfacing the encoders, providing a linear PI control action to the axes, controlling the focus etc. Currently for testing the basic functionality dedicated RS232 ports on the boards are used. The final telescope control architecture would be implemented as a distributed network on CAN bus. Although the number of controllers appear to be more, most of the hardware and embedded software design for the boards have been kept common. This facilitated simplicity in design and development whereas efforts were required only for implementing specific functionalities.

3.2 The Control Components

Each axis is driven by two hollow shaft QT-3124 Kollmorgen brushed permanent magnet torque DC motors. These motors are available in frameless pancake configuration, i.e., large diameter and a narrow width, and assembled directly on the gear shaft. The motors operate at 24 V DC and are capable of continuously providing stalling current of 6 A. For velocity feedback 4.608 million ppr Gurley series 8×60 incremental encoder is used. Through a reduction it provides an overall resolution of 0.011 *arcsec* accurate upto 0.05 *arcsec*. A RCN829 Heidenhein 29-bit hollow shaft absolute encoder provides direct measurement of the output shaft position at a high resolution within an accuracy of 0.5 *arcsec*. Each motor is coupled to a CP-850 incremental encoder which functions as a digital tachometer. Also a A25s Gurley 17-bit absolute encoder with an accuracy of +/- 15 *arcsec* is mounted which acts as an auxiliary encoder for cross checking the position values generated by the on-shaft main encoders.

The focus mechanism slides on a precise THK Ball Screw arrangement which travels 5 $mm/revolution$. The mechanism is coupled to a Superior Electric KM062 bipolar stepper motor using a timing belt and pulleys offering a reduction of 1:2. The motor has a incremental shaft encoder for position measurement and a PCA-116 series LVDT is used for referencing. The motor and on-shaft encoder undergoes four full rotations generating four index pulses in the range -/+5mm. At 0.9 degree per half step the housing can be positioned with a resolution of 6.25 μm.

3.3 Schematic of the Controller

The main components of the controller hardware include PWM (Pulse Width Modulation) based PI (Proportional plus Integral) controllers for the twin-motor drives, interfacing electronics for the encoders, bipolar stepper motor controller for focusing, power amplifiers for the motors and communication system. Owing to similar mechanical drive system the controller hardware design remains the same for both R.A. and declination axes. Currently, pointing is implemented with the help of slew, set, guide, fine motions requiring user intervention. These set of velocities for both axes and tracking velocity for R.A. are maintained constant with the help of PI controller. The schematic of the axis controller and focus controller are shown in Figures 3 and 4 respectively.

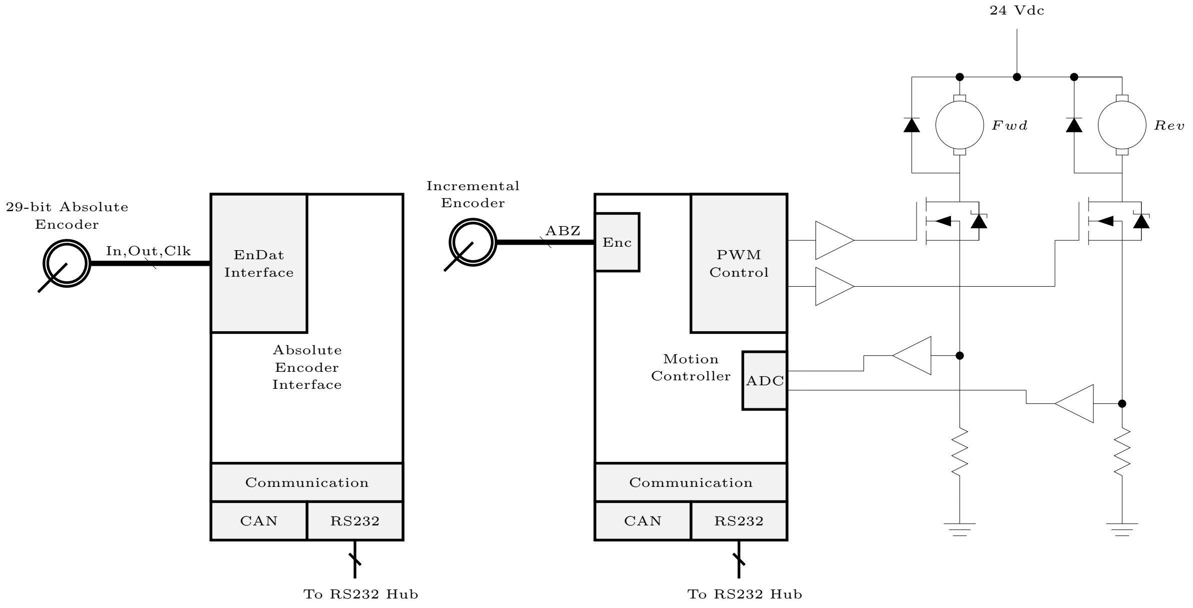


Figure 3. The schematic of the axis controller

Each motion controller is implemented on a separate dsPIC30f3011 microcontroller running an incremental PI control algorithm in PWM mode.[6] It provides control signal for both the motors on two of its PWM output channels. For driving the two motors two IRF250 power MOSFETs along with the driver stage are used to amplify the PWM signals. For reducing the mechanical backlash the PWM duty cycles on the two channels are adjusted such that 100% of the control effort is applied on the forward direction motor and 10% on the reverse direction motor. Each motor is connected between 24 V DC supply and the drain terminal of MOSFET with a 0.1 *ohm* resistance between the source terminal of MOSFET and ground. The voltage across the resistance is captured using ADC (Analog to Digital Converter) channels of the dsPIC for determining the motor armature current. The ADC module is synchronized with the PWM module using an interrupt to capture the samples during the PWM falling edge where the motor current peaks. The encoder feedback module of dsPIC is used for capturing encoder pulses from the incremental encoder for estimating the velocity. The interrupt routine is used to ensure a fixed sampling time of 5 ms.

Another set of boards are developed using PIC18f4480 microcontroller to interface the on-shaft Heidenhein position encoders on synchronous serial port. The encoder provides serial data packets on the bidirectional

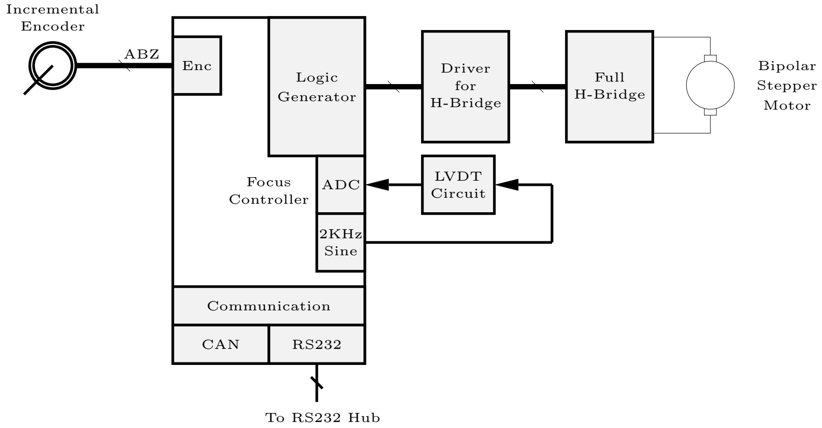


Figure 4. The schematic of the focus controller

EnDat2.2 interface which is decoded by the microcontroller for determining the absolute shaft position. The position measurements are displayed on GUI and used for setting soft limits on the telescope extremes.

For controlling the focussing mechanism a single microcontroller board was developed around PIC18F4431. Its encoder feedback module is used for interfacing the incremental encoder on the motor shaft. The controller also generates a square wave at 2 KHz frequency on its PWM channel. This is filtered using an active bandpass filter and amplified to obtain a fundamental 2KHz sine wave for the primary coil of the LVDT. The output of the LVDT is sampled by 10 bit ADC channel of the controller and combined with the index pulse of the encoder for absolute referencing. The bipolar motor is driven by a dual H-bridge in half stepping mode and consumes 1.2 A per coil at 4 V DC during operation. To prevent generation of heat during holding the motor is kept in regenerative mode through the free-wheeling diodes.

3.4 Message Timing and Communication

All boards consist of peripherals for Controller Area Network (CAN) interfacing, asynchronous serial communication (RS232) and serial debugging and programming. For PIC18F4480 boards the inbuilt CAN controller is used whereas for rest of the controller boards MCP2515 CAN controller is used. The final distributed telescope controller would have over sixteen microcontroller boards on CAN bus as shown in Figure 5. A CAN-to-USB board has been developed on a PIC18F4550 for putting the commanding PC on CAN bus. The communication bandwidth requirement between the commanding PC and hardware is very low, therefore the USB device is configured as a simple plug and play Human Interface Device (HID).

CAN is a standardized reliable system from real-time communication and has been successfully used in many telescope systems and subsystems.[8–12] A CAN bus supports upto 1 Mbit/sec for a total bus length within 40 mtrs. The distributed boards allow modularity, scalability, use of low cost microcontrollers, easy debugging and possibility of locating the boards closer to the sensors. Modularity and scalability allows independent development as well as addition of other boards as nodes to the existing network. Currently ten nodes providing position and velocity measurements for the two axes, velocity measurements for the four DC motors, digital display unit, along with USB-to-CAN node were tested in lab. The data word size per board is 8 bytes, first byte representing the device type. A total of 64 bytes are exchanged between the USB-to-CAN controller and the interfacing PC. The CAN network is yet to be implemented on the telescope.

For the purpose of telescope testing and characterization only the PI controller boards, position measurement boards and focus controller board implementing the basic functionalities are utilized. These boards are independently connected to a 8 port RS232 hub for PC interfacing. Time stamped data packets each containing 25 sets of 10 ms samples of PI control effort, telescope position, telescope velocity, ADC counts and focus position are continuously transferred to the interfacing PC at a fixed interval of 250 ms on five RS232 serial ports. The data is displayed on the GUI and continuously logged in the PC.

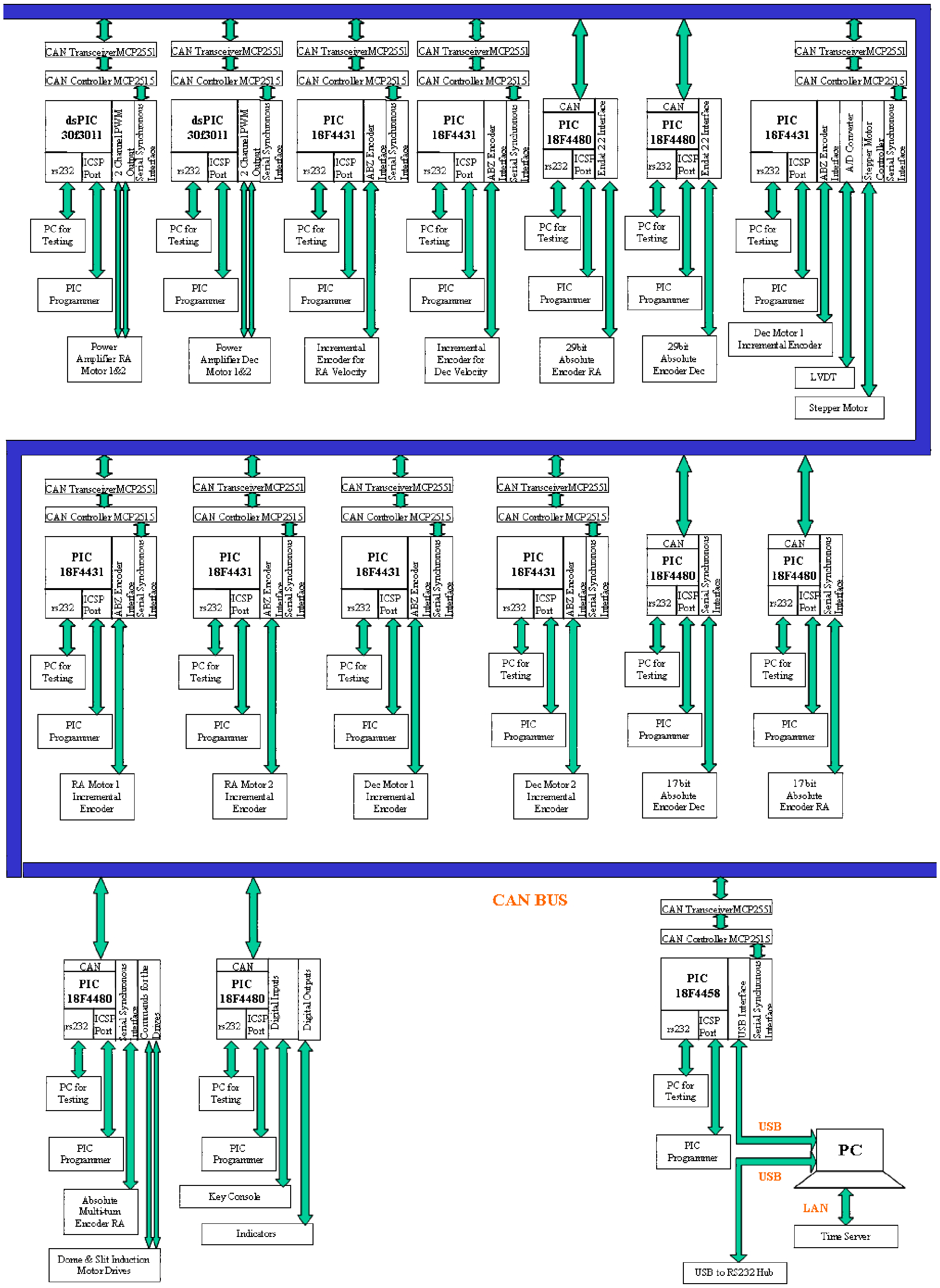


Figure 5. The schematic of the overall system under development

3.5 PI Controller Performance

A discrete time PWM based PI controller is designed for the lumped mass model based on the mechanical parameters of the telescope. The transfer function of linear time invariant second order model describing the telescope axis model reflected at the motor shaft is given by

$$\frac{\omega}{V} = \frac{\frac{K_{motor}}{J_r L}}{s^2 + \left(\frac{R}{L} + \frac{\sigma_{2r}}{J_r}\right) s + \left(\frac{R}{L}\frac{\sigma_{2r}}{J_r} + \frac{K_{motor}^2}{J_r L}\right)}, \tag{1}$$

where ω represents the angular velocity of the motor, V is the applied voltage, K_{motor} is the motor constant, R and L are the armature resistance and inductance respectively, J_r and σ_{2_r} are the reflected inertia and linear viscous friction respectively at the motor shaft. Incremental linear PI control algorithm[13] implemented in the dsPIC30f3011 microcontroller ensures bumpless transfer between different speed during pointing operation which is of the following form

$$\boxed{\begin{array}{rll} E_k = & (\omega_{sp} - \omega_k) & \text{Error = setpoint - output} \\ dU_k = & K_p(E_k - E_{k-1}) + K_i(E_k) & \text{Incremental PI control} \\ U_k = & U_k + dU_k & \text{Control effort to plant} \end{array}},$$

where E is the error representing difference between the desired velocity ω_{sp} and the actual velocity ω_k, U_k represents control command at k^{th} instant, δU represents incremental control command, K_p and K_i represent the P and I gains respectively. The duty cycle of the pulse width modulator is modulated using the control effort U_k. To prevent integral wind-up upper and lower limits are imposed on the U_k before loading the PWM duty cycle register. During the design torsional effects and all inertias which are small compared to the load are neglected. The telescope is assembled with dummy mirrors to prevent any damage to the optics and properly balanced by monitoring the motor currents in open loop. The response of the controller with current loop open and velocity loop closed for six different reference velocities are shown in Figures 6 and 7. Near the tracking speeds the limit cycles exhibited by nonlinear friction dynamics results in jittery motion. For medium speeds the errors are within 5% and for large speeds the errors are within 1% of the tolerance band.

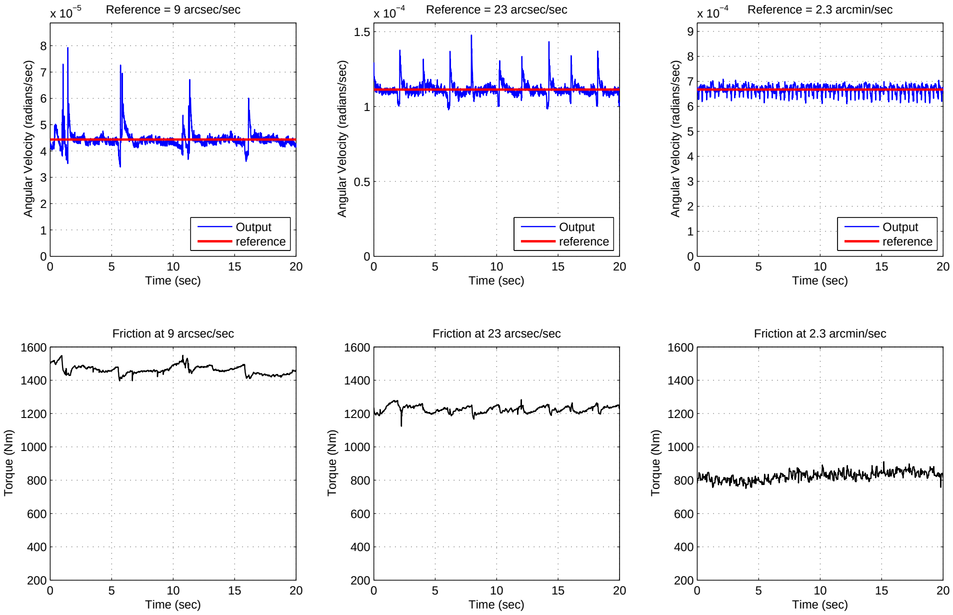


Figure 6. Top: Controller response at low speeds; Bottom: Corresponding Friction Torque

4. SOFTWARE DESIGN

4.1 Overview of the Software

A telescope control software is developed in Visual C++ which basically provides a GUI and communicates with the control hardware. It is capable of communicating with each board directly using multiple serial COM ports as well as with the USB-to-CAN board on USB port. The commands sent to and controller data continuously received from the telescope are logged at a desired interval in ASCII format. Since multitasking software especially

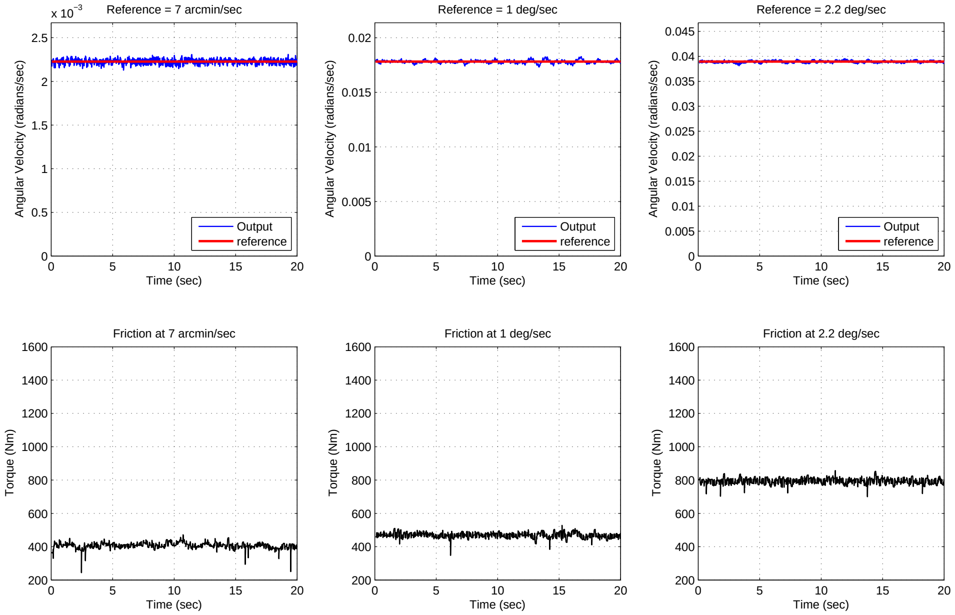


Figure 7. Top: The controller response at high speeds; Bottom: The corresponding friction torque

performing hardware communications is prone to breakdown, each function is implemented separately in a multiple software thread. This provides robustness against communication hardware errors or buffer overflows. When the GUI is launched more than ten software threads handling communication over several serial ports, USB port, logging service and GUI are launched and remain active. Multithreading also provides modularity and scalability falling in line with the advantages of the distributed telescope hardware architecture. The GUI offers several floating dialog windows which can be dragged to an extended PC monitor.

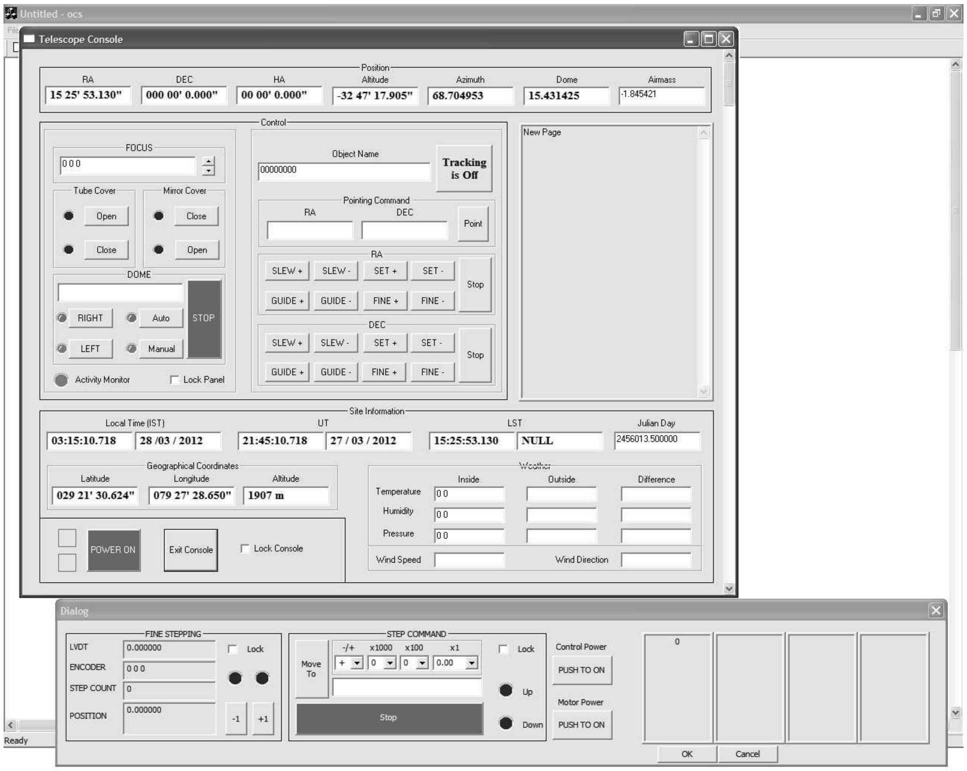
For device to device communication CAN interface is provided and preliminary tests were conducted by sending and receiving 8 bytes per board upto a transfer rate of 1 MHz. The PC picks up CAN messages using USB-to-CAN board configured as a HID device.

4.2 Software Aspects

When the software is launched the main console (Figure 8) is selected for powering up, manual pointing at preset speeds, tracking and monitoring the telescope parameters. During the power on process communication with the hardware is performed for health check. The status window displays messages related to the user command sequences, messages exchanged with the hardware, system response and errors. The console can be locked which greys out the buttons to prevent any accidental issue of command. Currently dome and weather stating control hardware have not been implemented but the features have been implemented and tested in the lab on RS232 ports.

A separate console is launched for operating the focus housing (Figure 8). The stepper motor can either be moved in single steps by using -/+ 1 buttons or moved to a desired position directly by selecting the set position and clicking on Move To button. Virtual LEDs provided on the console start blinking when the motor is running. The Motor Power OFF button provides holding operation by putting the stepper motor on regenerative mode and Control Power OFF completely frees the motor. The LVDT and encoder values are displayed in the text windows and refreshed every 250 ms.

For monitoring other parameters of the telescope useful for maintenance and troubleshooting two maintenance consoles have been provided (Figure 9). The first console is used for alignment of the absolute encoders i.e. when the telescope position is determined using a reference star, offsets can be applied on the encoders to match



RA: 15 25' 53.130"
DEC: 000 00' 0.000"
HA: 00 00' 0.000"
Altitude: -32 47' 17.905"
Azimuth: 68.704953
Dome: 15.431425
Airmass: -1.845421

FOCUS: 0 0 0
Object Name: 00000000

Local Time (IST): 03:15:10.718 28 /03 / 2012
UT: 21:45:10.718 27 / 03 / 2012
LST: 15:25:53.130 NULL
Julian Day: 2456013.500000

Latitude: 029 21' 30.624"
Longitude: 079 27' 28.650"
Altitude: 1907 m

	Inside	Outside	Difference
Temperature	0 0		
Humidity	0 0		
Pressure	0 0		

Wind Speed:
Wind Direction:

LVDT: 0.000000
ENCODER: 0 0 0
STEP COUNT: 0
POSITION: 0.000000

-/+: +
x1000: 0
x100: 0
x1: 0.00

0

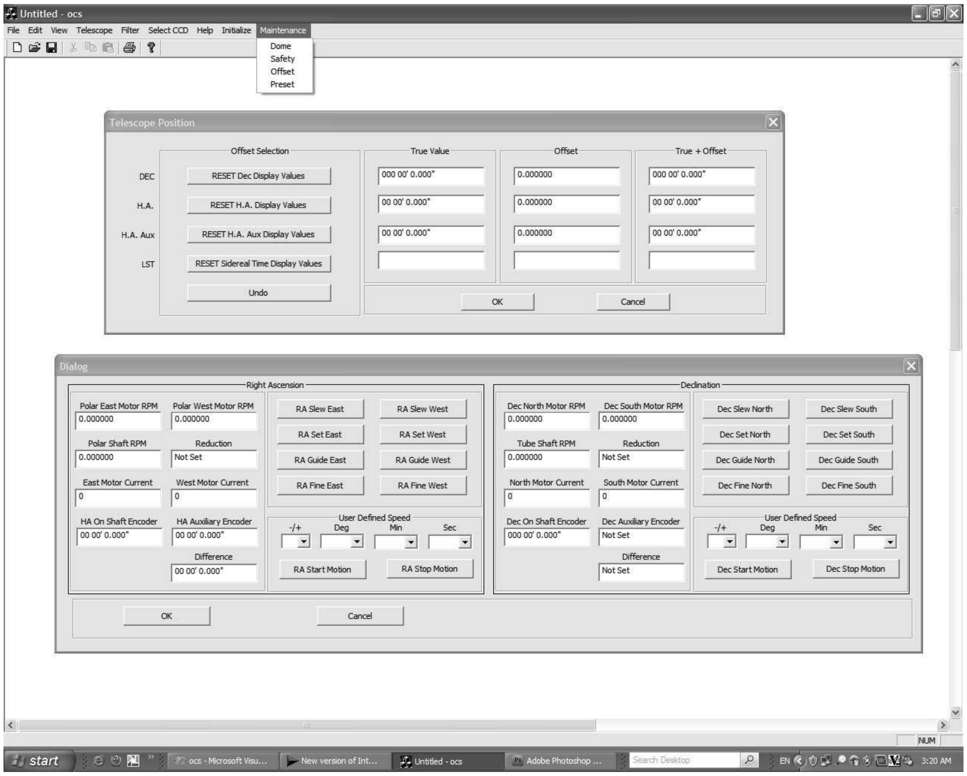
Figure 8. The consoles for telescope operation

the know telescope position. In the second console the individual motor currents and the tacho signals can be monitored. The tacho readings and the telescope velocity measurements are used for displaying the gear reduction in runtime and logged for analysing the backlash in the gear boxes. Also the auxiliary position encoder readings and its difference from the on-shaft encoder readings are displayed. The auxiliary encoder is used for the purpose of cross checking. This can also be statistically combined with the true encode readings for generating a better position estimate.

The user command sequence, device status, sensor data and errors are continuously displayed in text windows and simultaneously logged in four separate ASCII files by the data logging software thread. It continuously displays the speed, position and current data and stores it on the hard disk in ASCII format. The rate of log generation can be set as per the need which is usually kept high during maintenance procedure. One of the key advantages of developing a software in-house is the available flexibility for immediate modification, bug fixing and continuous improvement.

5. CONCLUSION

As the Schmidt telescope is a small system requiring a reasonable performance a telescope controller comprising a set of simple low cost controller boards is developed. For such telescope we favoured in-house component level design of hardware and software to enable effectiveness in maintenance and upgradation activities. Modularity and scalability is maintained in both hardware and software using separate board and communicating thread respectively for each specific task. The CAN functionality is implemented on the boards and tested in the lab, it will be implemented on the telescope at a later stage. There are some control issues at low speeds due to friction dynamics which will be further evaluated from the optical performance. The telescope is in a testing phase with the prototype boards and efforts are underway for compensating the friction. In our previous work we have already characterized friction present in the system, with the help of the addition encoders installed on the telescope we would also like to characterize backlash in the gear system and explore nonlinear compensation techniques.



	True Value	Offset	True + Offset
DEC	000 00' 0.000"	0.000000	000 00' 0.000"
H.A.	00 00' 0.000"	0.000000	00 00' 0.000"
H.A. Aux	00 00' 0.000"	0.000000	00 00' 0.000"
LST			

Polar East Motor RPM: 0.000000
Polar West Motor RPM: 0.000000
Polar Shaft RPM: 0.000000
Reduction: Not Set
East Motor Current: 0
West Motor Current: 0
HA On Shaft Encoder: 00 00' 0.000"
HA Auxiliary Encoder: 00 00' 0.000"
Difference: 00 00' 0.000"

Dec North Motor RPM: 0.000000
Dec South Motor RPM: 0.000000
Tube Shaft RPM: 0.000000
Reduction: Not Set
North Motor Current: 0
South Motor Current: 0
Dec On Shaft Encoder: 000 00' 0.000"
Dec Auxiliary Encoder: Not Set
Difference: Not Set

Figure 9. The consoles for maintenance purpose

ACKNOWLEDGMENTS

This work was carried out in joint collaboration with ARIES, Nainital. The authors would like to acknowledge the encouragement and support of Prof. Ram Sagar, Director, ARIES and engineers Shobhit Yadav, Tarun Bangia and Purushottam for technical support in this work.

REFERENCES

[1] Gupta, K. G., Bangia, T., Kumar, T. S., Sharma, S. N., and Shukla, V., "Development of 50 cm B-N Schmidt telescope at ARIES," *Proc. of The 25th Meeting of ASI, 2007* **25**, 80 (2008).

[2] Varsik, J. R. and Yang, G., "Design of a telescope pointing and tracking subsystem for the Big Bear solar observatory new solar telescope," in [*Advanced Software and Control for Astronomy*], H. Lewis; A. Bridger, ed., *Proc. of SPIE* **6274** (2006).

[3] Hirsch, P., Reif, K., and Müller, P., "A new control system for the 1m Cassegrain telescope at the Hoher List observatory," *Astronomische Gesellschaft Meeting Abstracts* **15**, 113 (1999).

[4] Ramiller, C., Taylor, T., Rafferty, T. H., Cornell, M. E., and Rafal, M., "A new control system hardware architecture for the Hobby-Eberly telescope prime focus instrument package," in [*Software and Cyberinfrastructure for Astronomy*], *Proc. of SPIE* **7740** (2010).

[5] Russiniello, G. B., Huguenin, D., and Wildi, F. P., "Robust control system for the new 1.2-m telescopes of the Geneva observatory," in [*Advanced Telescope and Instrumentation Control Software*], H. Lewis, ed., *Proc. of SPIE* **4009**, 216–226 (2000).

[6] Kumar, T. S. and Banavar, R. N., "Identification of friction in the 50/80 cm ARIES schmidt telescope using the LuGre model," *Proc. of The 18th IFAC World Congress* **18** (2011).

[7] Rivetta, C. and Hansen, S., "Friction model of the 2.5mts SDSS telescope," in [*Telescope Control Systems III of International Symposium of Astronomical Telescopes and Instrumentation*], *Proc. of SPIE* **3351**, 466–477 (1998).

[8] Cortina, J., Domingo, E., Flix, J., Lopez, J., and Martinez, M., "The control system of the MAGIC telescope," in [*International Science Symposium on The Universe Viewed in Gamma-Rays, 2002*], Enomoto, R., Mori, M., and Yanagita, S., eds., *The Universe Viewed in Gamma-Rays*, Universal Academy Press, Inc. (2003).

[9] Farris, A., Marson, R., and Kern, J., "The ALMA telescope control system," *Proc. of 10th Int. Conf. on Accelerator and Large Expt. Physics Control Systems (ICALEPCS)* (2005).

[10] Molfese, C., Schipani, P., and Marty, L., "VST telescope primary mirror active optics actuators firmware implementation," in [*Advanced Software and Control for Astronomy II*], A. Bridger and N. M. Radziwill, ed., *Proc. of SPIE* **7019** (2008).

[11] Molfese, C., Schipani, P., Capaccioli, M., Sedmak, G., and DOrsi, S., "Survey telescope control electronics," *Proc. of International Symposium on Power Electronics, Electrical Drives, Automation and Motion*, 523–527 (2008).

[12] Lefort, B. and Puig, M. P., "GTC a CAN based controlled telescope," in [*CAN in Automation*], *Proc. of the 12th International CAN Conference* (2008).

[13] Astrom, K. J. and Hagglund, T., [*Advanced PID Control*], Instrumentation Systems and Automation Society (2006).