

# PROCEEDINGS OF SPIE

[SPIDigitalLibrary.org/conference-proceedings-of-spie](https://spiedigitallibrary.org/conference-proceedings-of-spie)

## HETDEX tracker control system design and implementation

Beno, Joseph, Hayes, Richard, Leck, Ron, Penney, Charles, Soukup, Ian

Joseph H. Beno, Richard Hayes, Ron Leck, Charles Penney, Ian Soukup, "HETDEX tracker control system design and implementation," Proc. SPIE 8444, Ground-based and Airborne Telescopes IV, 84445J (17 September 2012); doi: 10.1117/12.925760

**SPIE.**

Event: SPIE Astronomical Telescopes + Instrumentation, 2012, Amsterdam, Netherlands

# HETDEX Tracker Control System Design and Implementation

Joseph Beno<sup>a</sup>, Richard Hayes<sup>a</sup>, Ron Leck<sup>b</sup>, Charles Penney<sup>a</sup>, Ian Soukup<sup>a</sup>,

<sup>a</sup>Center for Electromechanics, The University of Texas, 1 University Station R7000,  
Austin, TX, USA 78712

<sup>b</sup>McDonald Observatory, University of Texas at Austin, 1 University Station C1402  
Austin, TX, USA 78712-0259

## ABSTRACT

To enable the Hobby-Eberly Telescope Dark Energy Experiment, The University of Texas at Austin Center for Electromechanics and McDonald Observatory developed a precision tracker and control system – an 18,000 kg robot to position a 3,100 kg payload within 10 microns of a desired dynamic track. Performance requirements to meet science needs and safety requirements that emerged from detailed Failure Modes and Effects Analysis resulted in a system of 13 precision controlled actuators and 100 additional analog and digital devices (primarily sensors and safety limit switches). Due to this complexity, demanding accuracy requirements, and stringent safety requirements, two independent control systems were developed. First, a versatile and easily configurable centralized control system that links with modeling and simulation tools during the hardware and software design process was deemed essential for normal operation including motion control. A second, parallel, control system, the Hardware Fault Controller (HFC) provides independent monitoring and fault control through a dedicated microcontroller to force a safe, controlled shutdown of the entire system in the event a fault is detected. Motion controls were developed in a Matlab-Simulink simulation environment, and coupled with dSPACE controller hardware. The dSPACE real-time operating system collects sensor information; motor commands are transmitted over a PROFIBUS network to servo amplifiers and drive motor status is received over the same network. To interface the dSPACE controller directly to absolute Heidenhain sensors with EnDat 2.2 protocol, a custom communication board was developed. This paper covers details of operational control software, the HFC, algorithms, tuning, debugging, testing, and lessons learned.

**Keywords:** Hobby-Eberly, telescope, HET, HETDEX, wide field corrector, tracker, constant force drive, hardware fault controller, control system, Center for Electromechanics, McDonald Observatory, CEM

## 1. INTRODUCTION

The HET (Figure 1) is the largest telescope at The University of Texas' McDonald Observatory located on Mt. Fowlkes in the Davis Mountain Range in West Texas. A joint venture with The Pennsylvania State University, Stanford University, Ludwig-Maximilians-Universität München, and Georg-August-Universität Göttingen, the HET has been conducting science operations since October, 1999<sup>1</sup>. The HET support structure and primary mirror sits at a fixed zenith angle of 35° and can move in azimuth to access approximately 70% of the visible sky. The primary mirror, constructed of 91 identical hexagonal segments, forms an 11 m hexagonal-shaped spherical mirror with a 26,164 mm radius of curvature. The tracker is mounted above the primary mirror on the uppermost portion of the telescope, termed the upper hexagon or “upper hex” for short. The corrector optics mount to the tracker and are positioned via two linear drive systems and a six degree of freedom (DOF) hexapod maintaining the instruments' optical axis normal to and on the focal sphere of the primary mirror.

The Hobby-Eberly Telescope (HET) is currently undergoing a major redesign effort in preparation for the Dark Energy Experiment<sup>2,3</sup>. The upgrade, referred to as HETDEX, involves replacing the current star tracker<sup>4</sup> along with its drive systems. The catalyst for the replacement of this hardware is the wide field upgrade to the corrector optics. Replacing the current spherical aberration corrector (SAC) with the wide field corrector<sup>5</sup> (WFC) will increase the HET field of view from 4 ft to 22 ft. In addition, the current science instruments<sup>6</sup> are not in general being replaced. High resolution spectroscopy (HRS) and medium resolution spectroscopy (MRS) will be re-installed as is. Low resolution spectroscopy

(LRS) will be replaced with a new instrument, LRS2. VIRUS (Visible Integral-field Replicable Unit Spectrograph<sup>7,8</sup>) is being added to the current suite of instruments. These changes will allow the telescope to conduct the largest survey of distant galaxies ever attempted.

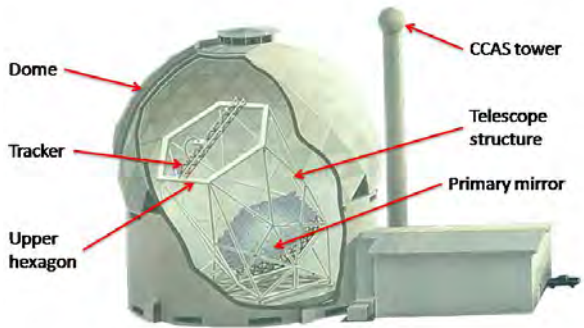


Figure 1. The figure on the left shows an aerial view of the HET with the shutter open. The computer rendering on the right reveals the major components of the telescope.

The primary impact to the HETDEX tracker caused by the replacement of the corrector optics is the approximately seven fold increase in mass of the instruments and supporting hardware. As a result, the HETDEX tracker mass increased on the order of about five times that of the HET tracker. The tracker system (Figure 2) is currently undergoing testing and commissioning at The University of Texas at Austin Center for Electromechanics (UT-CEM).

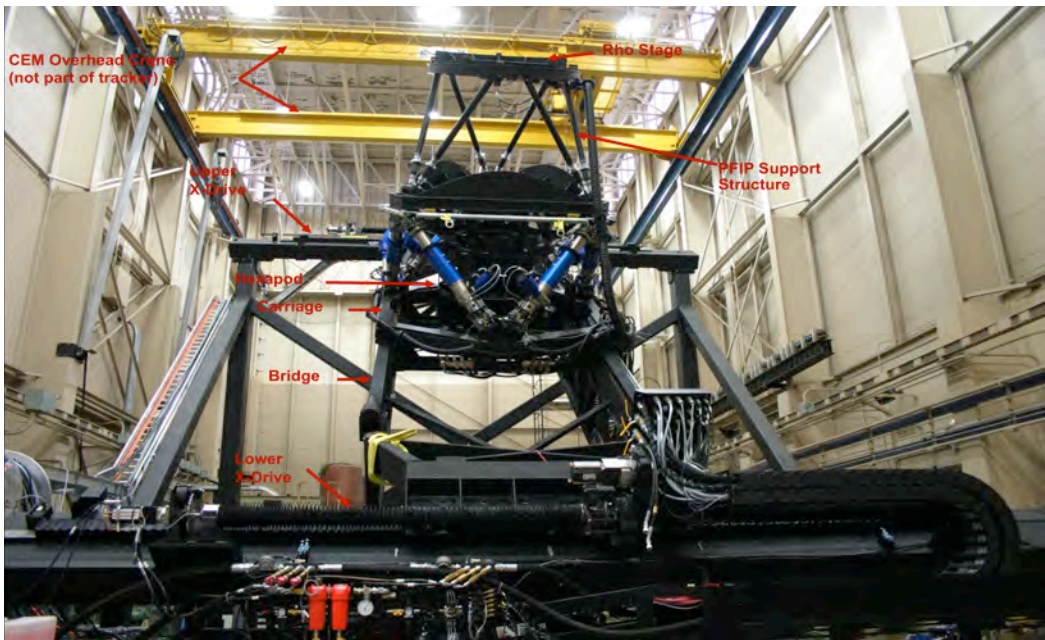


Figure 2. HETDEX tracker in CEM’s lab showing the key elements of the system

## 2. HETDEX CONTROL SYSTEM OVERVIEW

The new tracker includes a completely new control system developed by UT-CEM. The tracker control system actually includes two independent controllers, one that primarily controls motion of the tracker and a second parallel system that is only used to monitor the system and force a shutdown in the event a fault is detected. A top level overview of the system is illustrated in Figure 3 with a more complete layout in Figure 4. The major components are briefly described below with more detailed information in later sections. The focus of this paper will be on the Tracker Controller and the Hardware Fault Controller development, implementation, and testing. The primary components illustrated in Figure 3 are the following

### 2.1 Telescope Control System (TCS)

Controls the entire telescope, including telling the tracker controller what it needs the tracker to do in terms of making moves and following trajectories.

### 2.2 Tracker (dSpace) Controller

Includes two distinct software blocks operating in dSpace controller hardware. The Tracker Motion Control System – Supervisor (TMCS-S) block receives input from TCS. Based on those inputs TMCS-S will enable the servo amplifiers and brakes as required to perform specific moves and pass commands for trajectories to the Tracker Motion Control System – Algorithms (TMCS-A) block. TMCS-S also monitors sensors to assure safe operation of the system and can shut the tracker down in the event certain parameters are exceeded. TMCS-A block controls precision tracker motion by commanding the servo amplifiers for the thirteen drive motors based on input from the position sensors to meet science and engineering requirements from TCS.

### 2.3 Hardware Fault Controller (HFC)

The HFC is an independent safety/control system that operates in parallel with the tracker control system. Its sole purpose is to react to inputs from the critical sensors and load cells on the tracker and to shut the system down if preset limits are exceeded. It serves as a critical safety system for all of the primary drive systems and subsystems. When the normal sensors and tracker controls inside TMCS-S are set and working properly the HFC does nothing. It is only when TMCS-S has failed to react to an error that the HFC forces a shutdown.

### 2.4 Tracker Hardware

This group encompasses all the hardware that provides inputs to the control system such as position sensors, load cells, thermocouples, current sensors etc. and all the drives that are controlled by the tracker control system. This group makes up the majority of the components illustrated in figure 4.

Figure 4 shows a schematic of the control system, roughly organized into hardware components as they exist for the tracker system (e.g., the Tracker Control Rack is an actual electronics cabinet that contains the items depicted in Figure 4).

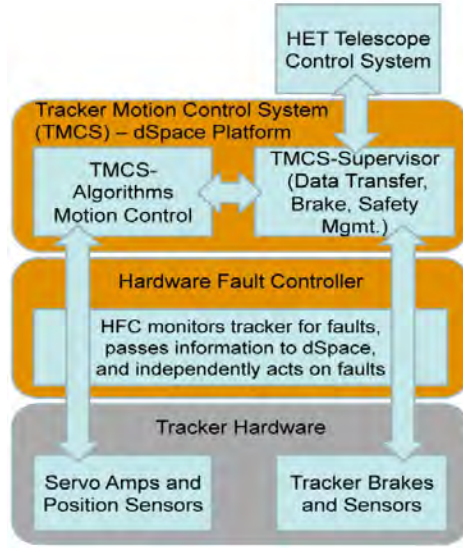


Figure 3. Schematic of top level tracker control system. The paper primarily focuses on the two orange subsystems.

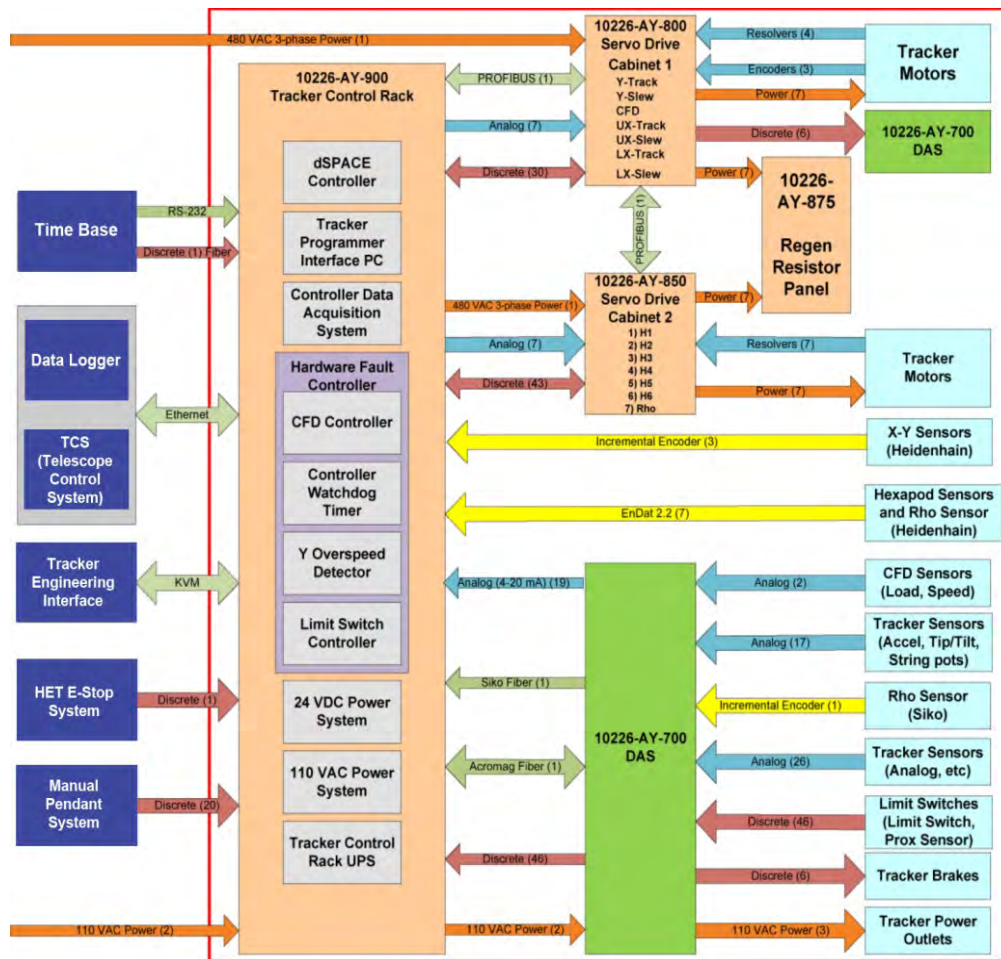


Figure 4. Schematic of control system for the HET Tracker Upgrade



### 3. TRACKER CONTROLLER DESIGN AND IMPLEMENTATION

The tracker controller is divided into two major software block: TCMS-S and TMCS-A which both reside in a single dSpace control system. All inputs and outputs that control the tracker are managed in this device including standard safety functions. The dSpace device is heavily used in the automotive industry and is particularly conducive to motion control software development in the Matlab/Simulink environment. It has proven to be an effective tool for moving control systems developed in the lab into the field at UT-CEM .

#### 3.1 dSpace Controller and Simulation-Based Design

As a tool for control system development, dSpace offers the option of developing controllers in a variety of design environments, as opposed to utilizing a dedicated processor which may require use of a specialized programming language. UT-CEM chose to use Matlab/Simulink for the tracker controls development for several reasons. First it allows the developer to test motion control software developed on a PC well in advance of hardware development. This can aid in both hardware design and debugging. Second, it is widely used in academia and in industry, resulting in a wide range of engineers and scientists who are familiar with it. Finally, it is by its nature very easy to document inside the code. This offers a significant benefit to future users who will undoubtedly be required to modify the control system software over the 20 year life of the hardware. While the software was developed primarily in Matlab/Simulink certain functions were developed in C and embedded as part of the Matlab/Simulink program.

Beyond the control function that dSpace serves, it also manages all of the inputs, outputs, and communication functions that must be accomplished to control the tracker. Many of these functions are performed by hardware that is provided by dSpace as a standardized component, although some are custom boards as shown in Figure 5.

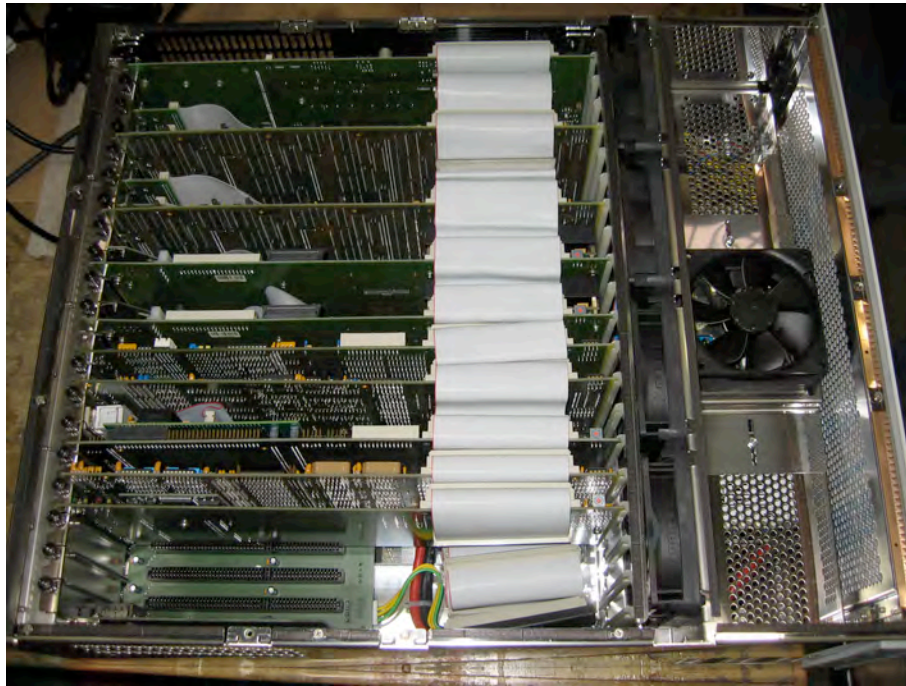


Figure 5. dSpace controller with interface and control cards installed for HETDEX tracker control.

The following items are part of the dSpace controller<sup>9</sup>:

**dSpace modular controller:**

- DS1005 – (1) PowerPC based 1 GHz processor
- DS4504 ETX – (2) 100 Mbit/s Ethernet interface board
- DS3002 – (1) 6 channel incremental encoder interface board
- DS2002 – (2) 32 channel A/D board
- DS4003 – (2) 96 channel digital I/O board
- PROFIBUS – (1) PROFIBUS master board
- DS5202 EnDat – (1) 8 channel EnDat 2.2 interface board

**dSpace I/O communication boards:**

DS4504 ETX – Two Ethernet boards: One communicates with tracker DAS (data acquisition system) mounted on top of the telescope. The DAS system on the tracker collects analog information and transmits it serially to tracker controller. The second communicates with TCS (Telescope Control System). Tracker commands, feedback and tracker logging data are sent over this Ethernet line.

DS3002 – Four incremental encoders used for position feedback (Upper X, Lower X, Y, and Rho). All four sensors are read by the DS3002 and interpolated 4096 times.

DS2002 – Direct analog measurements of various voltages and currents are measured and read into the two 16-bit A/D boards.

DS4003 – Two digital I/O boards: One is dedicated to the tracker discrete sensors (mainly limit switches and proximity sensors) feed through the DAS system to the controller. The second is dedicated to discrete sensors in the electrical room.

PROFIBUS – Fourteen motor drives are required to run the tracker that contain PROFIBUS communication cards. Each drive is uniquely addressed and setup as a slave node. The dSPACE PROFIBUS card acts as master. dSPACE sequentially cycles through each drive to send new motor commands and receive drive status information.

DS5202 EnDat – Seven EnDat 2.2 sensors are used on the tracker for position feedback, one for each hexapod actuator and one rho stage sensor. This custom board directly interfaces with Heidenhain sensors to continually read position.

### 3.2 TMCS-S Development and Implementation

TMCS-S, the tracker controller “supervisory” layer manages all of the inputs coming from the Telescope Control System and reformats commands that must be forwarded to TMCS-A for motion control. Another primary function of the TMCS-S is the frontline management of the safety system. The tracker controller runs as a multi-rate model on the dSPACE processor. The supervisor functions and motion algorithm blocks run at 1kHz while the Data Access Subsystem (DAS) runs at 200 Hz. Specifically, the TMCS-S functions are:

- Receives commands - Receives commands from the TCS computer via a 10/100 Mbit Ethernet network using the TCP/IP protocol. The commands are ASCII coded messages with a maximum length of 128 bytes that specify trajectory coordinates, engineering moves and requested data or parameters. Responses, with a maximum length of 140 bytes, return command acknowledgements and requested data or parameters. In addition, log messages are sent to a separate computer using UDP/IP protocol. The log messages can contain event or state change information useful in evaluating performance and providing diagnostic error messages in the event of component or system failures.
- Processes commands - Processes commands in 1-ms intervals and coordinates the hardware configurations and state changes to achieve the five established states (Figure 6) including moves along trajectories which are specified in real time. The real time clock is derived from a GPS disciplined signal read in from the RS232 serial port.

- Enables servo amplifiers - TMCS-S controls the servo amplifiers for the fourteen drive axis as required to move from one state to another or to force a shutdown in the event a significant fault is detected.
- Manages brake subsystem - The upper X and lower X axis have pneumatic brakes with pressure sensors to determine if the brakes are engaged or released. In addition to the pneumatic brakes, the Y axis has internal electromechanical brakes. The Y axis brakes have pressure sensors and load cells to indicate the engaged/released state. In the event of a fault, TMCS-S commands actuation of all brakes. The brake subsystem synchronizes brake operation between tracking and slewing brakes and reports state and status results to the TCS and logger computers.
- Manages States - Maintains proper state transitions and controls/monitors the servo-amp and brake subsystems. State variables indicate the current hardware/software setup and only permit allowable hardware transitions that promote safety and optimize performance.
- Manages Safety - The Fault Handling/Fault Reporting subsystems monitors over 100 analog and digital signals at three rates: 1 kHz, 100 Hz and 1 Hz to verify the system is operating within a set of safe parameters. Most of these sensors are also monitored in parallel by the HFC (discussed below) but the TMCS-S is the first line of defense. If it detects an “out of bounds” condition it will issue a warning to the TCS computer or command a shutdown depending on the condition.
- Manages communications - Manages communications between the TMCS-S and TMCS-A subsystem with the TMCS-interface block. The interface block is responsible for verifying completion of all move, hold and abort commands. An ‘EndPath’ flag reports that the requested command has completed. The interface block provides hooks via dSPACE Control Desk to run in an ‘engineering’ mode. The engineering mode bypasses the Telescope Control System interface and lets the user manually control the hardware via Control Desk’s graphical user interface. The GUI allows the user to monitor software variables and change output values without recompiling the source code. .

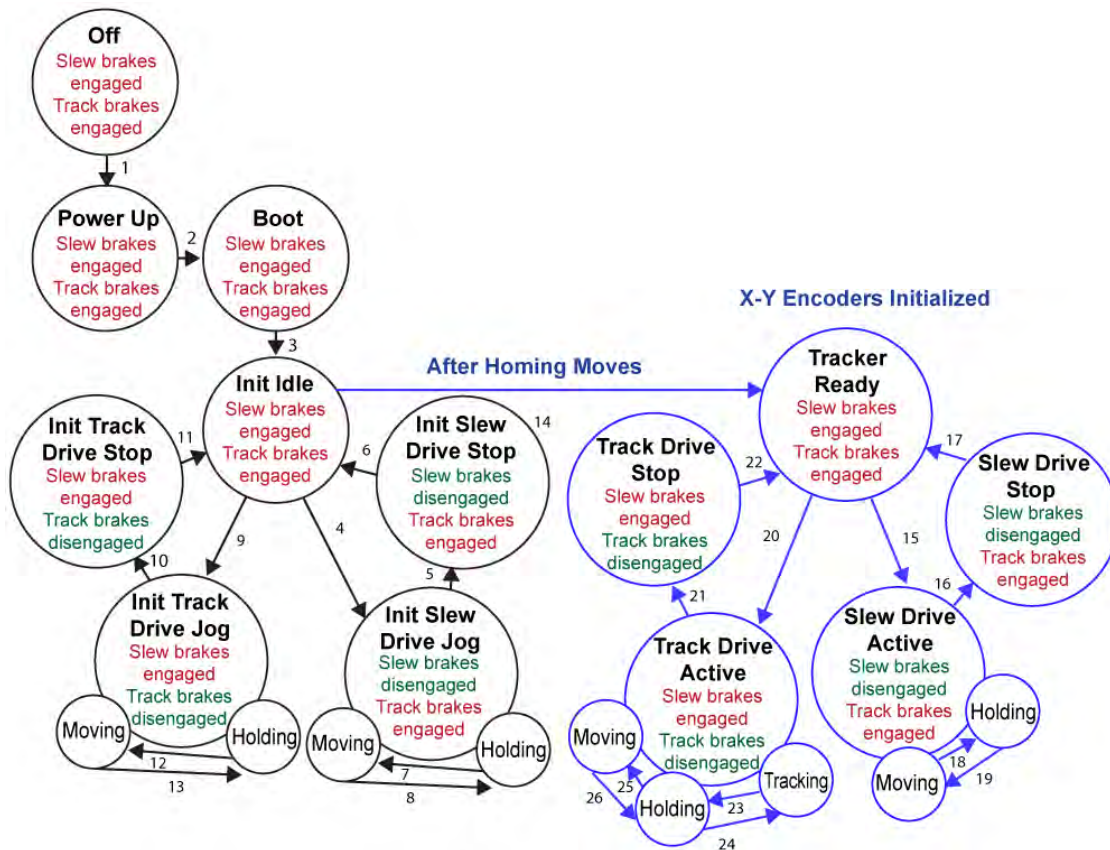


Figure 6. State diagram for tracker operation



### 3.3 TMCS-A Development and Implementation

A primary advantage of the dSPACE controller is its close linkage with the MATLAB and Simulink computational and simulation environment. Exploiting this environment, a detailed integrated simulation of the complete tracker motion system, including the X drives/actuators<sup>10</sup>, Y drives/actuators<sup>11</sup>, hexapod actuators<sup>12</sup>, rho stage actuators, and axis control algorithms was developed. Motion was modeled as rigid bodies within Simulink using the SimMechanics toolbox, with all appropriate degrees of freedom, constraints, and mass properties (from component CAD solid models). The top level model in this simulation environment is shown in Figure 7.

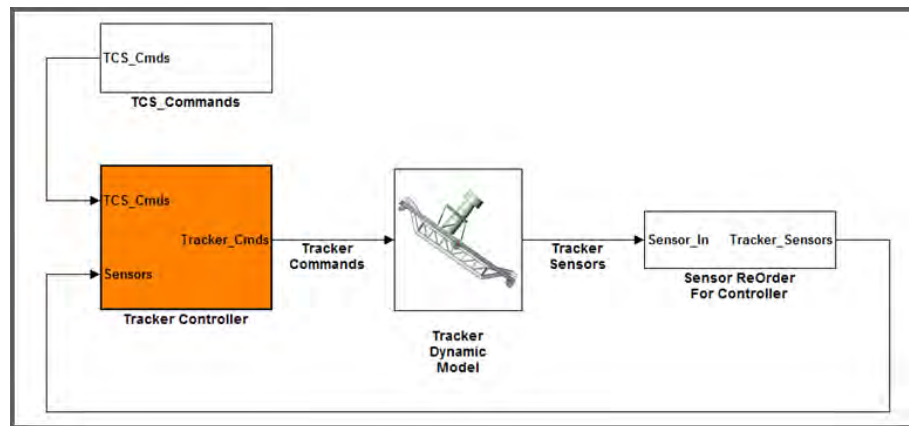


Figure 7. Simulink model of the tracker system. The “Tracker Dynamic Model” was completed using the SimMechanics toolbox. The rest of the simulation was based on normal Simulink blocks and S-functions.

Within the simulation model, modules were developed to represent sensor resolution, current resolution of the motor drive servo-amplifiers, and actuator friction characteristics. In the simulation model, all tracker control algorithms, including sensor filters, friction compensation, PID loops, and feed-forward algorithms are contained in the “Tracker Controller” block. These algorithms are built using the MATLAB GUI or built as S-functions using the MATLAB language (based on the MAPLE kernel). This Tracker Controller block then represents a stand-alone block that can be extracted (cut and paste) from the simulation environment and directly integrated into the dSPACE controller using MATLAB’s realtime workshop. The realtime workshop is an auto-code generation process to generate ANSI C code so control algorithms developed in the simulation environment transfer directly to actual tracker controller hardware without re-programming. This process allows full use of MATLAB’s powerful control development environment with very quick transition to control hardware. Some specific features of the Simulink model are the following:

- **Tracker Controller Block:** Actual control code that migrates via auto-code generation to tracker controller hardware – simplified modification process with proven industry standard Matlab-Simulink GUI driven code development process.
- **Control Algorithms:** Proven in realistic simulation environment before transferring to hardware.
- **Controller Documentation:** MS Word documents embedded in Simulink blocks for easy reference.
- **Tracker Dynamic Model:** Sim-Mechanics dynamic modeling environment linked with Matlab Virtual Reality toolbox for visualization.
- **Component Specifications:** Performance “proven” in realistic coupled simulation environment.

For the model shown in Figure 7, the block labeled “TCS\_Commands” represents the Tracker Controller link to the HET Telescope Control System. Simplified generation of tracker trajectories are developed in this block to exercise the model, debug and test the Tracker Controller, and tune the Tracker Controller. Outputs from the Tracker Controller are Tracker Commands to various tracker actuators within the Tracker Dynamic Model. As shown in Figure 7, simulated sensor output from the Tracker Dynamic Model are re-ordered and fed back to the Tracker Controller.

Prior to transferring the MATLAB based control system to the dSPACE hardware, extensive simulated tracking and slewing commands were exercised. Output from these simulations directly guided specifications for all tracker actuation subsystems. Model validation was completed as much as possible by correlating results within UT-CEM engineering models in SolidWorks using SolidWorks specialized motion and constraint tools. Further model and control algorithm validation was completed on dedicated test rigs to characterize individual actuators or subsystems (e.g., a standalone single hexapod actuator test rig and a complete hexapod system test rig)<sup>12</sup>. UT-CEM has extensive experience with developing high performance controlled motion systems and has repeatedly benefitted from the simulation based design process enabled by the MATLAB-dSPACE integrated system described in this section. Figure 8 shows a screen capture of animation tools that were used early in the project to assist in validating the general behavior and imbedded transformations of the model.

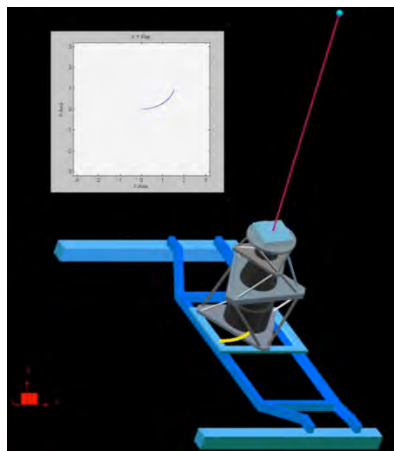


Figure 8. Screen capture of video and X-Y plot of the tracker simulation

The Tracker Controller subsystem (Figure 7), evolved into the Tracker Motion Control System – Algorithms. The TCS\_Commands subsystem shown in Figure 7 was replaced by the Tracker Motion Control – Supervisor, which interfaces with the Telescope Control System (TCS). The resulting Simulink model that evolved from early simulations to later simulations is shown in Figure 9, showing TMCS – A, TMCS – S; an interface block between TMCS – A and TMCS – S; and interface blocks to tracker hardware.

## Top Level Simulink Model of Tracker Motion Control System

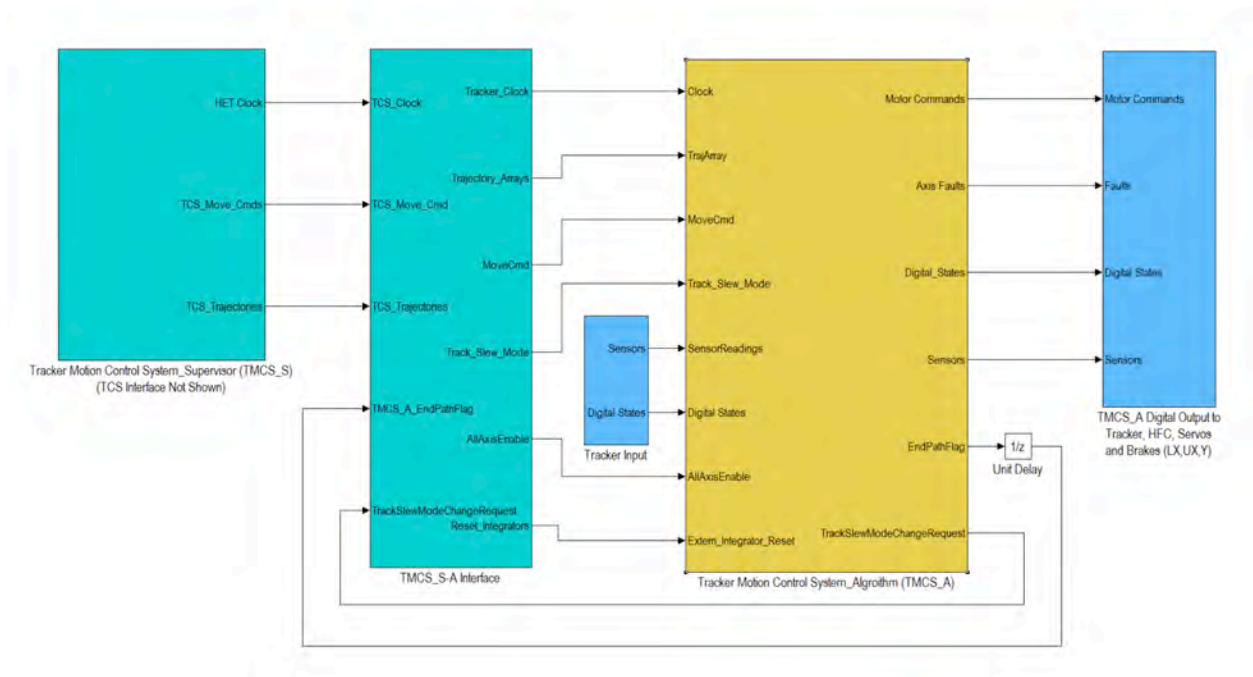


Figure 9. Top level Simulink model for the tracker motion control system

Within TMCS\_A the first sub-level has subsystems to

- (1) populate tracker move command arrays;
- (2) control abort and rapid stops;
- (3) control trajectories and path management;
- (4) control the tracker axis (e.g., servo loops);
- (5) manage faults (other fault management subsystems exist with TMCS\_S);
- (6) reset integrators; and
- (7) provide indications to TMCS\_S that the end of a path (e.g., commanded move or trajectory) has been reached.

The TMCS\_A block, exactly as shown in Figure 10 is used in BOTH the dSpace hardware to control the tracker hardware and in PC based simulations in the Matlab-Simulink environment. For the simulation task, there is also a very useful subsystem that is part of TMCS\_A that plots movements and other data (shown in red in Figure 10) Many times, while debugging and tuning the tracker system, it has been very useful to run a simulation on a laptop right next to the dSpace GUI.

# Simulink Subsystem Model of Tracker Motion Control System-Algorithm (TMCS\_A)

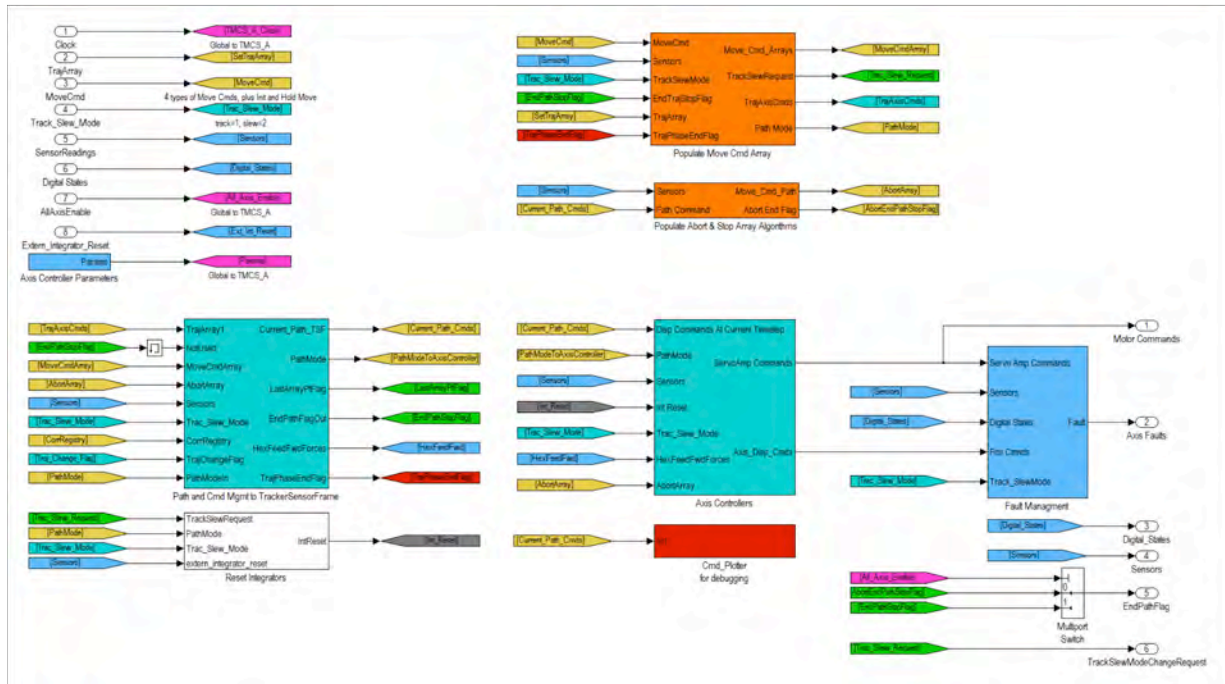


Figure 10. Top Simulink subsystem model of TMCS\_A

The dSpace environment includes a configurable hardware interface. One page of 15 Control Desk layouts is shown in Figure 11. While TMCS-S automates control of tracker functions, the Control Desk interface allows the operator to directly control each tracker function. In what is termed the Engineering Mode (non-automated mode). The particular page shown in figure 10 is used during system startup while in the non-automated Engineering Mode. *The result was that tracker hardware operation, testing, and debugging was started more than a year before completion of the first working version of TMCS-S.* Lessons learned during this period were directly integrated into the TMCS-S programming.



Figure 11. Screen capture of 1 page of dSpace “Control Desk” Tracker controller Graphical User Interface

A major advantage of the dSpace system for TMCS\_A is that experienced control engineers can easily implement a wide array of tools to overcome external disturbances, cancel adverse friction effects, and adapt to specific hardware characteristics. A companion paper<sup>9</sup> describes this process for the 13 tracker control axis. This was crucial in our ability to meet/exceed all tracker performance requirements. Figure 12 shows one plot with following errors for X motion, Y motion and one of the hexapod struts. The following comments apply to these plots:

- Y and Lower X plots show steady-state offset that is removed with sensor calibration/zeroing and with telescope closed loop control on stars.
- Lower X plot indicates reaction to cyclical screw variations. Data was taken with feed-forward cancellation terms off (e.g., compensation tables/mount models).
- Y data has cyclical variations similar to the X axis, but of smaller magnitude and partially obscured by other non-periodic control fluctuations, likely resulting from interaction with Constant Force Drive.
- Hex Strut plot also shows cyclical force fluctuations that could be improved with feedforward control terms (compensation tables/mount models).
- Despite obvious areas for improvement and the fact that various control terms were turned off to highlight mechanical characteristics of the tracker motion systems (e.g., feedforward terms to cancel cyclical fluctuations), all plots meet error budget specifications.

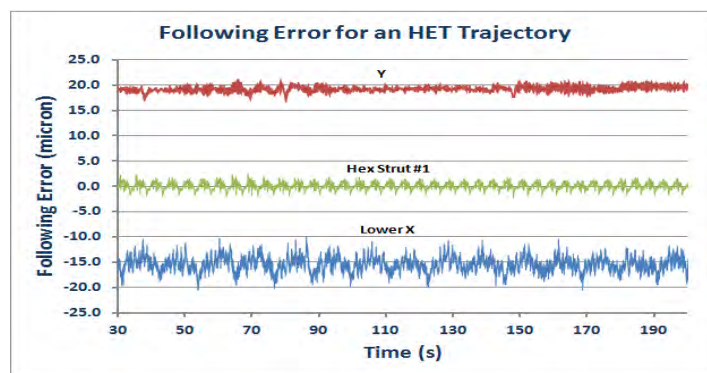


Figure 12. Following error for a HET trajectory test on the Tracker system in UT-CEM’s laboratory

#### 4. HARDWARE FAULT CONTROLLER DESIGN AND IMPLEMENTATION

The hardware fault controller (HFC) is a critical component to the long term safety of the tracker and the personnel who operate it. During normal operation, if the parameters are set correctly and sensors are working properly, the HFC should never come into play. It is only when the tracker is operating outside of the safe parameters that the HFC is called into action. This device addresses the majority of potential hardware and software faults identified during the Failure Modes and Effects Analysis (FMEA)<sup>13</sup> by forcing a safe shutdown of the system when a fault conditions is detected.

##### 4.1 HFC Design and Implementation

The main object of the HFC design is to be able to sense and command a shutdown of the tracker motion without the normal controller hardware or software. To achieve this design goal requires the HFC to be completely independent from the normal tracker controller. Another design goal was to have the HFC operate with minimal software operation, or to be as much of a hardware solution as possible. Minimizing the required software in the HFC was seen as a risk reduction tactic for the tracker. Having the HFC as a hardware based design and the dSpace tracker controller as a



software controlled system yields two controllers with two different design philosophies. With the HFC design greatly limiting the software in the HFC, the HFC has only two states, normal operation and fault state.

All of the digital control signals to and from the dSpace controller pass through the HFC. This configuration allows the HFC to have a supervisory role over all the signals. The HFC performs the normal operations by allowing all the digital control signals to pass through. In the event that the HFC detects a fault in the operation of the tracker, the HFC will interrupt several controller signals forcing them into a known state to command a safe shutdown on the tracker system.

The HFC shutdown sequence was designed to limit the forces on all axis of travel to be as low as possible at all times. The HFC safe shutdown sequence involves stopping some axes immediately and other axes in a more gradual shutdown to reduce the applied forces on the moving components. An analysis of all the axes of motion with the corresponding highest travel speeds and the respective masses yielded two groups of axes, timed or immediate shutdown. This implementation is fixed in the hardware of the system.

The types of digital signals that the HFC can force off include all brake commands, servo amplifier enable signals, and the servo high voltage contactor commands. Without the enable signals and without any high voltage, the dSpace control system is unable to send motion commands.

To determine what state the HFC should be in requires monitoring several sensor signals and comparing those signals to the predetermined limits. These sensors were determined to be the most critical during the FMEA. The sensors that are monitored by the HFC, and can initiate a shutdown, include the following:

- Y position sensor used to monitor possible over-speed condition. This is determined by the HFC using an analog position signal on the Y carriage.
- Bridge skew condition. Digital inputs from each of the four bridge corners determine if the bridge is beyond the allowable skew limits.
- Limit switches on all axis of movement in both directions. These are hardware digital switches.
- dSpace controller heartbeat signal. If the HFC does not receive a heartbeat signal every 10 Hz, the condition of the main control loop software may not be operating correctly.
- CFD load cell and speed sensors to determine if the CFD is operating outside normal operating conditions.



Figure 13. Hardware fault controller prior to installation in HETDEX control rack.

## 4.2 Constant Force Drive

One failure that generated a large amount of concern among the team was the potential for the carriage, which carries the hexapod, WFC, and PFIP to experience an uncontrolled descent down the trackers 35° incline. Many solutions were considered to reduce the likelihood of an uncontrolled descent or to mitigate the damage from the worst case scenario event. The final conclusion was a redundant carriage control system working with a redundant movement system.

There are two mechanical systems that are controlled completely independently but work together to move the tracker in the Y direction. The main Y movement system is a ball screw drive arrangement that performs the precision positioning function. The second Y movement mechanism is a wire rope and drum on a separate controller called the Constant

Force Drive or CFD. The main design goal of the CFD is to carry a set amount of the total carriage weight, at any position and at any rate of speed without causing any degradation of the desired carriage motion. The CFD functionality is implemented on the HFC<sup>11</sup>.

Although the CFD mechanical system is designed to move a portion of the carriage weight, the CFD is able to stop and hold the entire weight of the carriage. With the CFD and the Y drive systems; there are two complete systems capable of supporting the carriage. In this respect, the CFD is a redundant support system capable of holding the carriage in the event the Y drive has a failure, or there is a failure in the control system hardware or software. This was a design concern from the FMEA to minimize risk to the carriage hardware in the direction where gravity could work with a system failure to cause significant damage.

The CFD control system was incorporated into the HFC to combine all the safety systems into one piece of hardware. This minimized the extra hardware that would be required and its functionality was easy to add to the planned HFC. The CFD is controlled by the same microcontroller as the HFC functions. To control the CFD, the system uses several sensors:

- Load cell to determine the amount of carriage force the CFD is compensating for.
- Position sensor on the Y axis to determine location and speed.
- CFD drum speed also utilized for Y axis speed monitoring.

The main control item for the CFD is the set point for carriage force compensation. The force set point is the desired amount of weight that the CFD mechanical system is to support with the remaining weight supported by the Y positioning system. The HFC microcontroller uses a PID control loop to maintain the desired load on the load cell. In normal operation the main load cell target is about 20% of the total carriage weight. Testing of the actual hardware shows the best positioning performance when the CFD set point is between 5 and 20% of the total carriage weight.

The CFD is monitoring two different sensors to insure the Y carriage is not moving faster than the allowed settings. One sensor is the Y position of the carriage. The second sensor is the drum speed of the CFD. Both of these sensors must be within the allowed travel speed limits of the Y carriage or the HFC will go into the shutdown state.

## 5. CONCLUSIONS

UT-CEM's experience implementing the tracker control system, and subsequently tuning and debugging led to the following conclusions.

- The Matlab-Simulink-dSpace environment greatly facilitated successful tracker control system, especially with TMCS-Algorithm.
- Early dynamic simulation with autocode generation to transition simulation controller to actual hardware controller played major role in controller success.
- Programming for TMCS-Supervisor data handling, communication control and ASCII based interface with Telescope Control System was completed with a mix of Matlab programs and C programs. Completing TMCS-S entirely in C may have been easier to implement and debug.
- Full integration of Failure Modes and Effects Analysis (FMEA) results into the tracker design resulted in outstanding safety (personnel and equipment) for the HET tracker upgrade program.
- HFC implementation did not result in any degradation in tracker performance and the goal of stopping all control system faults from causing any hardware damage has been achieved.
- Limited testing to date indicates that performance goals are being met and performance will continue to improve with controller tuning and refinement.

## ACKNOWLEDGEMENTS

HETDEX is led by The University of Texas at Austin McDonald Observatory and Department of Astronomy with participation from the Universitäts-Sternwarte of the Ludwig-Maximilians-Universität München, the Max-Planck-Institut für Extraterrestrische-Physik (MPE), Astrophysikalisches Institut Potsdam (AIP), Texas A&M University, Pennsylvania State University, and the HET consortium. In addition to Institutional support, HETDEX is funded in part by gifts from Harold C. Simmons, Robert and Annie Graham, The Cynthia and George Mitchell Foundation, Louis and Julia Beecherl, Jim and Charlotte Finley, Bill and Bettye Nowlin, Robert and Fallon Vaughn, Eric Stumberg, and many others, by AFRL under agreement number FA9451-04-2-0355, and by the Texas Norman Hackerman Advanced Research Program under grants 003658-0005-2006 and 003658-0295-2007.

## REFERENCES

- [1] Ramsey, L.W., et al., 1998, "The early performance and present status of the Hobby-Eberly Telescope," Proc. SPIE 3352, p. 34.
- [2] Hill, G. J., Gebhardt, K., Komatsu, E., Drory, N., MacQueen, P. J., Adams, J. J., Blanc, G. A., Koehler, R., Rafal, M., Roth, M. M., Kelz, A., Gronwall, C., Ciardullo, R., Schneider, D. P., "The Hobby-Eberly Telescope Dark Energy Experiment (HETDEX): Description and Early Pilot Survey Results," ASP Conf. Series, 115-118 (2008).
- [3] G.J. Hill, et al, "Current status of the Hobby-Eberly Telescope wide field upgrade," Proc. SPIE, **8444**-19 (2012).
- [4] Booth, J. A., Ray, F. B., and Porter, D. S., "Development of a star tracker for the Hobby Eberly Telescope", Proc. SPIE 3351-20 (1998).
- [5] Burge, J. H. et al, "Development of a wide-field spherical aberration corrector for the Hobby-Eberly Telescope", Proc. SPIE 7733-51 (2010).
- [6] Hill, G. J., MacQueen, P. J., Palunas, P., Shetrone, M. D., Booth, J. A., "Present and Future Instrumentation for the Hobby-Eberly," Proc. SPIE 6269-626907 (2006).
- [7] Kelz, A., Roth, M. M., Bauer, S. M., Padilla, Y., Popow, E., Hill, G. J., Vattiat, B. L., Good, J. M., Murphy, J. D., Savage, R. D., Soukup, I. M., Mollison, N. T., "Production and performance of replicable integral field units for VIRUS", Proc. SPIE 7735-178 (2010).
- [8] Hill, G. J., Adams, J. J., Blanc, G., Booth, J. A., Chonis, T. S., Cornell, M. E., DePoy, D. L., Drory, N., Gebhardt, K., Good, J. M., Grupp, F. U., Kelz, A., Lee, H., Marshall, J. L., MacQueen, P. J., Mollison, N. T., Murphy, J. D., Rafal, M. D., "VIRUS: a massively replicated 33k fiber integral field spectrograph for the upgraded Hobby-Eberly Telescope", Proc. SPIE 7735-21 (2010).
- [9] J.R. Mock, et al., "Tracker controls development and control architecture for the Hobby-Eberly Telescope dark energy experiment," Proc. SPIE, **7733**-152 (2010).
- [10] I.M. Soukup, J.H. Beno, G.H. Hill, J.M. Good, C.E. Penney, T.A. Beets, J.D. Esquerro, R.J. Hayes, J. T. Heisler, J.J. Zierer, Jr., G.A. Wedeking, M.S. Worthington, D.R. Wardell, J. A. Booth, M.E. Cornell, J. D. Rafal "Testing, characterization, and control of a multi-axis high precision drive system for the Hobby-Eberly Telescope Dark Energy Experiment (HETDEX)," Proc. SPIE, **8444**-147 (2012).
- [11] Mollison, N. T., Mock, J. R. Soukup, I. M., Beets, T. A., Good, J. M., Beno, J. H., Kriel, H. J., Hinze, S. E., Wardell, D. R., "Design and development of a long-travel positioning actuator and tandem constant force actuator safety system for the Hobby-Eberly Telescope wide-field upgrade", Proc. SPIE 7733, 150 (2010).
- [12] Zierer, J. Z., Mock, J. R., Beno, J. H., Lazzarini, P. G., Fumi, P., Anaclerio, V., Good, J. M., "The development of high-precision hexapod actuators for the Hobby-Eberly Telescope Dark Energy Experiment (HETDEX)", Proc. SPIE 7733, 49 (2010).
- [13] R.J. Hayes, T.A. Beets, J.H. Beno, J.A. Booth, M.E. Cornell, J.M. Good, J.T. Heisler, G.J. Hill, H.J. Kriel, C.E. Penney, M.D. Rafal, R.D. Savage, I.M. Soukup, M.S. Worthington, J.J. Zierer, "Use of failure modes and effects analysis in design of the tracker system for the HET wide-field upgrade," Proc. SPIE, **8449**-56 (2012).