Development of a Control System for a Liquid Rocket Engine Test Stand

Daniel Llewellyn[1], Barrett Twining[2], Francis Clark[3], Henry Holmberg[4],
Jack Redden[5], Rori Myers[6], Christian Dove[7]
The University of Alabama in Huntsville, Huntsville, Alabama, 35805, USA

This paper details the design of a control system for a mobile test stand (MTS) for a student-led, liquid rocket engine research and development project. The control system commands the engine to fire and ensures safe operation by minimizing risks to personnel. Since the control system is mission-critical, it must be durable, reliable, and designed for easy maintenance and future expansion. Given these requirements and its essential role in all tests, we have developed a highly detailed control system design.

The main tasks of the control system are to command open and close valves, collect temperature, pressure, and thrust data, and trigger an abort if sensor readings exceed the redline levels. The control system will be split into two main sections based on location, the MTS controller is located inside the trailer, and the base station. The base station consists of the controls computer and all physical buttons and switches necessary for safe firing. The MTS controller is further broken into two systems; the MTS computer, and the command control and instrumentation (CCI) boards. The MTS computer is a Raspberry Pi 4s that communicates with the CCI boards and the base station. The CCI boards are custom-printed circuit boards (PCB) assembled from widely available components. They connect to sensors and solenoids through XLR ports.

All of the subsystems will integrate together to create a robust and modular controls system for use on our current engine and potential future designs. The details of how these subsystems are designed will be explained in the rest of this paper.

I. Nomenclature

TC = Thermocouple
PT = Pressure Transducer
SV = Solenoid Valve
CCI = Command Control and Instrumentation
MTS = Mobile Test Stand
RPi4s = Raspberry Pi 4s
ETS = Engine Thrust Stand
GUI = Graphical User Interface
UART = Universal Asynchronous Receiver/ Transmitter
SPI = Serial Peripheral Interface (Communication Protocol)
QSPI = Quad-Serial Peripheral Interface (High-Speed Communication Protocol)

---

[1] Undergraduate, Mechanical and Aerospace Engineering Dept., dl0091@uah.edu, Student Member, 1810912.
[2] Undergraduate, Electrical and Computer Engineering Dept., barrett.twining@uah.edu, Student Member, 1811037.
[3] Undergraduate, Electrical and Computer Engineering Dept., francis.clark@uah.edu, Student Member, 1823553.
[4] Undergraduate, Mechanical and Aerospace Engineering Dept., hjh0022@uah.edu, Student Member, 1811014.
[5] Undergraduate, Mechanical and Aerospace Engineering Dept., jcr0204@uah.edu, Student Member,1745440.
[6] Undergraduate, Mechanical and Aerospace Engineering Dept., ndm0014@uah.edu, Student Member, 1811019 .
[7] Undergraduate, Electrical and Computer Engineering Dept., cd1816@uah.edu, Student Member, 1823296.

SDIO = Secure Digital Input Output (High-Speed SD Card communication Protocol)
I2C = Inter-Integrated Circuit (Communication Protocol)
P&ID = Piping and Instrumentation Diagram
RAM = Random Access Memory
PCB = Printed Circuit Board
NI = National Instruments
DAQ = Data Acquisition

II. Introduction

The Tartarus project is a student liquid rocketry team based in the University of Alabama in Huntsville; a part of the Space Hardware Club. The goal of which is to design and manufacture a 250 lbf Liquid Oxygen and Ethanol bi-propellant engine. The team must work within time, budgetary, and safety constraints, all while balancing the lives of full-time college students. The project is split into four sub-teams: fluids, responsible for designing, manufacturing, and testing the ground support equipment. Propulsion, responsible for designing, modeling, and developing the rocket engine. Systems Engineering and Operations, responsible for keeping the project on track, as well as making sure a high level of safety is maintained. And Controls, responsible for the data recording and valve actuation electronics on the test system, including designing and integrating all necessary electronic components. The team has undergone a complete overhaul, redesigning all parts of the controls system. This includes designing custom new circuit boards to gather data and control valves, a graphical user interface designed from the ground up, giving operators an unprecedented amount of control over the collection and application of data, as well as a brand new embedded system to connect said graphical interface to the aforementioned circuit boards. This system must be perfectly reliable, incredibly accurate, and low cost. These requirements have presented new challenges and necessitated radical solutions to create what we believe to be one of the best control systems in the world of student rocketry.

III. Requirements

Having a robust system of requirements for the redesign of the control system is essential to keeping the goals of the system in sight.

A. Previous Requirements For The Control System

The previous requirements for the control system, before moving away from the legacy National Instruments DAQs, were minimal and vague. The requirements were:

1) The system must monitor sensors for current temperature and pressure data and display the data in both graphical and numerical form.
2) The interface must be easy to use and understandable.
3) The system must control solenoid valves with position indication.
4) The system must be able to collect pressure, temperature, and thrust data.
5) The system must have physical and software abort abilities.

These requirements were not specific enough for a reliable system that has the capability to collect and record data at an acceptable rate.

B. Current Requirements For The Control System

With the new redesign, there was a need to have dedicated requirements for each section of the control system. These sections were therefore given a greater number of in depth requirements compared to the previous system. The sections of requirements are for the MTS computer, the CCI Boards, the MTS Controller housing, the igniter system, software, and the base station.

Requirements for the MTS Computer are as follows:

1) The MTS Computer shall be able to fit inside one of the CCI Board enclosures with said CCI Board.
2) The MTS Computer shall communicate live data to the base station via Ethernet.
3) Said live data shall be communicated with the base station at a selectable packet rate, up to 100Hz.
4) The MTS Computer shall be able to communicate with all CCI boards at the same time via a Universal Asynchronous Receiver-Transmitter (UART) connection to each CCI board.
5) Communication with each CCI board shall be at a baud rate of 115200.

The requirements for the CCI Boards are as follows:

1) The boards shall be mounted inside server chassis to allow them to be rack mounted.
2) Each board shall have 1 load cell input with an amplifier.

3) Each board shall have 8 thermocouple (TC) inputs with amplifiers, 3 of these inputs for K-type TCs and 5 of these inputs for T-type TCs.
4) Each board shall have 8 inputs for pressure transducers (PTs). These inputs shall use XLR ports.
5) The boards shall communicate with the MTS computer via UART.
6) Each CCI board shall have 8 relays, controlled by the microcontroller.
7) These relays must be able to pass a minimum 1A at 24V.
8) Each CCI board shall have 8 XLR ports connected to the relays for commanding valves.
9) The CCI boards shall have a method of measuring valve actuation status.
10) The CCI boards shall have an XLR input for MTS computer UART communication.
11) The CCI boards shall have an XT60 connector for 12V input.
12) The CCI boards must be designed with reliability and durability in mind due to them being safety critical components. Automotive grade components should be used wherever possible, and best design practices should be followed.

The requirements for the MTS Controller housing are as follows:

1) The entire MTS Controller shall be housed in an area no more than 4 square feet.
2) The MTS Controller shall be housed in a server rack system that allows for ease of assembly and upgradability.
3) The housing shall hold the network switch used for sending data from the MTS Controller and cameras to the base station.

The requirements for the base station are as follows:

1) The base station shall contain the controls computer, and all physical buttons and switches needed for a safe firing.
2) The computer must be connected to a minimum of three monitors to display the graphical user interface (GUI) with data readouts, valve actuation, and live camera feeds.
3) The physical buttons and switches shall be the abort button and the switches to activate the igniter.
4) The abort button shall be connected to the system with dedicated wiring.
5) The igniter switches shall be connected to the system with dedicated wiring.

The software requirements are as follows:

1) There shall be an intuitive and easy to use GUI.
2) The GUI must have live graphs for temperatures, pressures, and thrust.
3) The GUI shall have an interactive piping and instrumentation diagram (P&ID) that shows the status of all sensors and valves connected to the CCI boards.
4) The GUI shall be able to command valves to open and close.
5) The GUI shall be able to command system events such as engine startup.
6) The GUI shall have live camera feeds positioned around the MTS and engine test stand (ETS).
7) The GUI must be able to generate its own system event logs from the data received from the CCI board/MTS computer. Faults and other conditions such as overpressure events must be recorded in this log.
8) The system must output data into a CSV file that includes appropriately named headers for each column.
9) The software for all components must be well documented, and should be written to be understandable. Software should follow applicable best practices.
10) Version Control software like Git shall be used, hosted on a service such as GitLab or GitHub, and must be kept up-to-date for all software components.

Igniter system requirements are the following:

1) The igniter circuitry shall be able to spark the spark plug.
2) The 12V battery shall be able to power the whole spark plug system and be easily removed and changed.
3) The entire igniter system shall be mounted to the ETS.
4) The design of the enclosure shall electrically isolate the high voltage lines that go to the spark plug.
5) The enclosure shall have a high voltage warning label.
6) The solenoid valve (SVs) shall be solidly mounted to the correct panel on the ETS.

Due to these greatly elaborated requirements, the previous system using the three NI DAQs is inadequate. If the old system was kept unmodified, there would not be enough inputs on the DAQs to measure all of the pressure transducers, or a serial bus to connect the TC amplifiers to. Additionally, the old system would not have enough inputs to wire in load cells for measuring thrust.

IV. Hardware

A. Base Station Monitor Setup

There will be a minimum of three monitors connected to the controls computer. Displayed on all of the monitors will be a part of our custom GUI. One of these monitors will be exclusively dedicated to displaying live camera feed from around the MTS and ETS. The second monitor will be dedicated for displaying graphs with live data from all of the sensors and toggles for the abort procedures. The third monitor is for displaying the P&ID with all of the toggles for valve sequences. If needed there is the option of expanding to four monitors, or potentially five, if the controls computer allows for that many video outputs.

B. Hardware Abort System

The hardware abort system is the physical part of the overall abort system. This system cuts power to the SVs to shut the system down in the event of an emergency. This is achieved by switching a relay that cuts power to the SVs, controlled by the abort button located at the base station. There is also a second relay controlled by the MTS Computer. The separate relays allow for the physical switch to function regardless of the status of the MTS Computer and CCI boards.

C. Command Control and Instrumentation (CCI) Boards

The Command Control and Instrumentation (CCI) boards are at the heart of our controls system. They provide high speed data logging, valve control, and communication with the Mobile Test Stand (MTS) computer. In this section, we will review the part selection, design, and implementation of this component of the controls system.

1. Part Selection

The first part we selected was the processor, as this is the component that the rest of the board is designed around. For this part, we had quite a few requirements including the following:

1) Minimum of 50 GPIO pins
2) Minimum clock speed of 32MHz
3) At least 256 KB of program storage, and at least 128 KB of RAM
4) Minimum 250,000 sample per second, 16 bit Analog to Digital Converter (ADC)
5) I2C (For LC's)
6) Serial Peripheral Interface - SPI (for TC's)
7) Quad-Serial Peripheral Interface - QSPI (for high-speed flash memory)
8) Secure Digital Input Output - SDIO (for high speed microSD cards)
9) Minimum 1 UART (for communicating with MTS computer)
10) USB programming support
11) Automotive Grade certified component

We also wanted a chip that was widely available, and was supported by the software libraries and IDE's that we were planning on using. With all of these factors in mind, we decided to go with the Microchip SAMD51P20A[1]. Here are its relevant specifications:

1) 1MB of program memory, and 256KB of SRAM
2) Dual, 1 Million Sample-Per-Second ADC's
3) 120 MHz Main clock frequency
4) 128-pin TQFP (Thin-Quad-Flat-Pack) package, 99 GPIO pins
5) Hardware/software support for USB, I2C, SPI, QSPI, UART, SDIO
6) Existing bootloader for getting USB and other peripherals running quickly
7) AEC-Q100 automotive grade certification

With the processor chosen, we then moved on to other sections, like measuring the output of the pressure transducers. Our priorities for measurement were speed, followed by precision. We wanted to ensure we could detect minor changes in the engine during firing and accurately measure other key events occurring within. While ADCs with higher sampling rates do exist, it was determined that we would be limited by the speed of the communication protocol that they use, as we need all 8 communicating on the same bus. Also, using the internal ADCs on the SAMD51 can give us both the speed and resolution that we need. For this reason, and also for the cost savings that using the integrated ADCs provides, we decided against using discrete ADCs.

Although we are using the integrated ADCs, we will still need some circuitry to correct for the voltage output of the PTs. The SAMD51 is a 3.3V microcontroller, and the PTs we are using have an output voltage range of 1-5V. Putting 5V onto an analog input on the SAMD51 would most certainly damage it, so we scale the voltage down using a circuit called a voltage divider, as shown below:

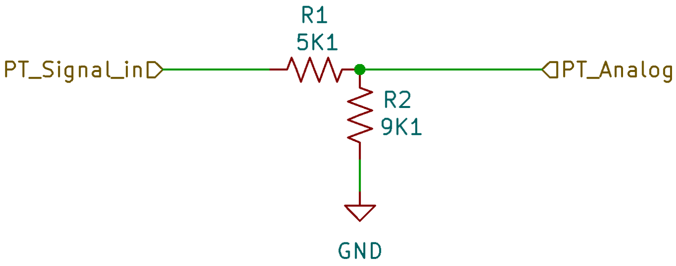

Fig. 1 Voltage divider circuit

This circuit will scale the input voltage down to an output voltage according to the equation:

$$V_{out} = V_{in}\frac{R_1}{(R_1 + R_2)}$$

The circuit above will give us a max output voltage of 3.2V, which is below the maximum input voltage of our processor (3.3V), while not being so low that we lose a significant portion of our ADC range. The resistor values above were chosen because they are fairly common values, and their resistance is high enough for the given voltage to prevent excessively loading the output of the PT. While we will purchase precision resistors, we will also measure their exact resistance when we assemble the boards and will have a "calibration value" for each channel, to account for small variations in the resistance values. After selecting the method for measuring the PTs, we moved onto the TCs.

For the TCs, we went with the Maxim Integrated (Analog Devices) MAX31856[2] precision thermocouple to digital converter IC. These were chosen for a few reasons. First, these ICs were already being used by the team (in breakout board form) for preliminary testing, and we found that they worked very well, and had a software library that was easy to use. They are compatible with K, J, N, R, S, T, E, and B type thermocouples, which allows the flexibility to change TC types if desired. They also have an impressive measurement range from -210℃ to 1800℃, and resolve temperature measurements to just 0.0078℃. The main downside of these chips is the price; they are ~$12 each, and we will need 24 to assemble the 3 CCI boards (each board has 8 TC ports), however, we decided this cost was worth it due to the reasons mentioned above. Additionally, we believe that we will be able to use some of the IC's off of the breakout boards we already possess, reducing the number that will need to be purchased.

The final "sensor" part we chose was the amplifier for the Load Cells (LC's) that measure the thrust output of the engine. These are less critical than the PTs, and mainly serve to let us know if our calculations are correct and/or if we are hitting our target values for thrust. These will be useful to measure thrust output if we decide to try throttling the engine. For the task of measuring the low voltage output of the LC, we chose the NAU7802[4], which is a high-precision 24 bit sigma-delta (Σ-Δ) analog to digital converter. The NAU7802 also features a programmable gain amplifier (PGA) with a selectable gain range from 1 to 128. Most importantly, this chip can handle a relatively wide input voltage range on its sensing terminals, which allows the use of many different load cells, including the models that we already own. There are also existing software libraries for this chip, which will make it easier for our software team to use. After the main processing and sensing circuits, we moved on to choosing simpler parts, like the relays that control the solenoid valves (SV's).

For the relays, we first considered the voltage and currents that they would need to be able to handle. The SVs we are using draw approximately 500mA when running, and operate at 12V, but the system must also be able to handle 24V up to SVs. Because the currents are so low, we decided that it would be excessive to buy 10A relays for the SVs, so we decided to use "signal" class relays that can handle up to 2A, which is well over the expected current. For the relays we went with the Panasonic AGQ200A12Z[3], as they met all of our requirements, and are made for telecom use, so they are designed to be highly reliable, with this specific model having a minimum expected life of 500,000 actuations. Shown below is the circuit we are using to control each relay. Because the GPIO pins of the microcontroller can only output about 5mA, we use a metal-oxide semiconductor field effect transistor (MOSFET) to switch the relay on and off. The resistors in series with the gate of the MOSFET limit the current drawn to turn the MOSFET on (as without them the instantaneous current could be higher than 5mA, which could damage the microcontroller). Finally, the diode between COIL- and COIL+ is called a freewheeling or flyback diode, and prevents voltage spikes caused by the inductance of the relay coil when the relay is switched off.

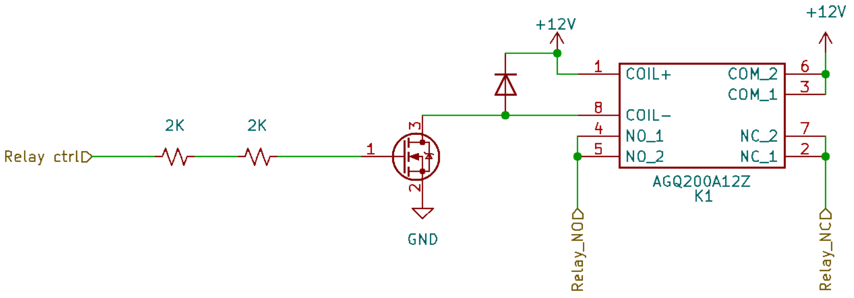

Fig. 2 Relay control circuit

With all of the parts selected, we can move on to the considerations made when designing the board.

2. CCI Board Design and Considerations

Even with the best components, a poor design can lead to unnecessary complexity, troubleshooting, costs, and limited expandability/adaptability. It is for these reasons that we have spent countless hours making sure the design of these boards is as simple, repairable, reliable, and expandable as possible.

As mentioned previously, the PTs and SVs connect to the CCI boards using standard XLR ports. These ports are on a daughterboard separate from the CCI boards that connect via a ribbon cable. This is to allow the CCI boards to be smaller, and to reduce strain on the main boards. This also means that if ports on the daughter board get damaged, we can replace them for a much lower cost than replacing the entire CCI board. For the TCs, there is a similar system, where the K and T-type TC connectors connect to the CCI boards via another ribbon cable.

In addition to being modular, we also wanted these boards to be expandable, to allow the system to adapt as the engine and overall design changes, such as allowing us to switch to throttleable valves. The main way we plan on achieving this is through the use of expansion ports. We plan to have 2 expansion ports that will expose power, ground, communication busses, GPIO, and more, to allow for other daughterboards to be designed and quickly connected via a ribbon cable. Some ideas for future expansion boards are:

1) Electric motor drivers for controlling throttleable valves
2) A camera shutter control board for starting and stopping recording on high speed cameras near the test stand.
3) A board with extra sensors or other specialty electronics that we may want to test

We will also have multiple people review the schematic and PCB designs to look for errors, possible problems, or suboptimal routing, before we order and assemble the boards.

D. Control System Housing

The control system near the test stand consists of 3 CCI boards, 1 Raspberry Pi, Ethernet-connected cameras, and a network switch that sends this data over to the base station approximately 500 ft away.

The CCI boards will each be mounted in a 2U (2 rack unit) server chassis, that will then be mounted on a server rack, along with the network switch. The Raspberry Pi will be mounted in one of the server chassis along with a CCI board. Using a server rack allows us to make the system more modular, and keeps all of the cables and wires organized. We will use a server chassis that has a blank front panel and cut out holes for each port.

On the back of the chassis, there will be 2 power inputs and the UART connections from the CCI boards to the Raspberry Pi. One of the power inputs will be for CCI board power, and one for the relay power supply. This is so the main board can still run at 12V[7],but the relays may supply a different voltage[7].

E. Data Collection

There are 4 types of data collection and one type of control node within the system: pressure transducers (PT), T-type thermocouples (T-type TC), K-type thermocouples (K-type TC), load cells (LC), and electrically actuated

---

[7] Note that the CCI board has voltage regulators that regulate the 12V down to the appropriate voltage for respective components.

solenoid valves (SV). The PTs give out an analog signal with a high enough fidelity to be able to interface directly with the CCI boards without a loss of precision. The TCs provide a much lower level of fidelity and must be passed through an amplifier which provides a significant gain in precision as well as converts the analogue signal to a digital signal. The 2 LCs interface with 2 of the CCI boards, which have onboard amplifiers for the LCs. As the PTs are directly interfacing with the CCI boards, they will have a data logging rate estimated to be 1kHz or greater. For the TCs, the maximum data rate is 60Hz, due to the amplifier boards used. This was deemed acceptable, as the TC data is not as safety critical as the PT data. Additionally, even if logged faster, the TCs are not as fast to respond to changes as the PTs are, due to their thermal mass. The PTs have a ±1.0% margin of error. Both the K and T-type TCs have margins of error of ±0.75% and the amplifier modules for the TCs have a margin of error of ±0.15%, resulting in a total margin of error of ±0.9%. The LCs have a margin of error of ±0.02%. All SVs in this system are simple binary valves with no intermediate states. These valves each first run through a simple relay before interfacing directly with the CCI boards.

F. Igniter

The igniter is a gaseous oxygen and propane torch, ignited by a spark plug. The system relies on a 12V battery which passes through a relay controlled by the base station which allows us to remotely turn the igniter on and off. The 12V, when switched on, powers a car ignition coil circuit as shown below:

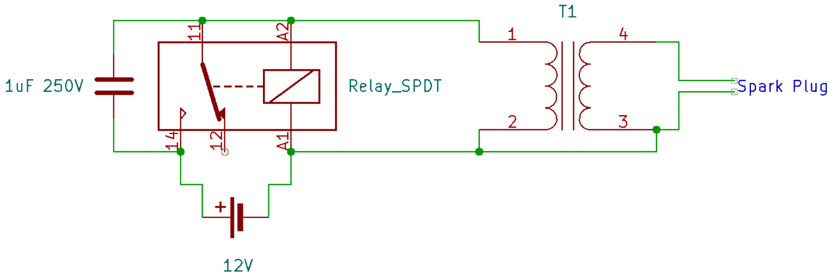

Fig. 3 Ignition coil driver circuit

While the figure represents the current circuit, alternative designs are still in consideration to further optimize the system.

On the base station side, there will be a required key switch to arm the circuit, then a covered switch that must be actuated to start the igniter. These two measures should prevent the accidental activation of the igniter system.

V. Software

A. User Interface

The base station will run a GUI application for operators to remotely control and monitor the MTS. The application is responsible for recording any data it receives as a backup, although the CCI boards will record the largest share of data. The code will be written in Python using PyQt6, a Python binding for Qt6, a C++ framework. PyQt6 was chosen for its cross-platform abilities, including its compatibility with PyQtGraph. PyQtGraph is a powerful plotting library for Python built on PyQt and NumPy, intended for math, science, and engineering applications.

The GUI application will be used for overseeing and controlling PTs, TCs, LCs, and SVs. Methods of monitoring sensors include three graphs made using PyQtGraph, one of which will display PT data, one LC (thrust) data, and the other TC data each sensor is plotted with a different colored line for readability Since operators may not need to monitor all sensors simultaneously , there will be an option to choose which sensors are graphed. there will be a block for sensor data where operators may select a sensor from a dropdown list, displaying the following information about the sensor:

1) Most recent reading
2) Current redline limit
3) Connection status
4) Whether or not the redline value has been exceeded

Finally, there will be an additional block for setting and removing the redline limits on individual sensors as appropriate. A popup dialog will ask the user to confirm any changes made to redline limits.

For control and monitoring of SVs, the GUI will also contain a manual valve control block. This will include a button to toggle manual control of the system to reduce the chance of a valve being toggled by accident. The decision to actually allow manual control of the system will be handled by the MTS computer. This block will also contain a toggle button, current signal, and current detected state. There will also be a block to activate valve sequences which will be predefined in the MTS computer, in particular the engine ignition sequence.

In addition to the aforementioned, the GUI will have a separate window containing a simplified version of the P&ID for the fluids system. Overlaid on the P&ID near each sensor and SV will be displays containing, for sensors, communication and redline trip status, and for SVs, commanded state and detected state. This will provide an intuitive way for operators to conceptualize the state of the system.

B. Backend

The backend software will be written in C++. We have chosen this language due to its speed, reliability, and usage across the industry. The backend software will be running on two different chips (the CCI boards and MTS computer).however, the ARM architecture they share allows for a simple compilation and build process, speeding up development time and allowing for quicker prototyping.

The MTS computer will connect to the base station over a Cat5 Ethernet cable, with server grade network switches on either side. Commands will be sent from the base station to the MTS computer in TCP/IP packets, containing plain text commands in a format similar to SCPI commands used in many test instruments today. This format was chosen due to its extensibility as well as it being an industry standard. The data sent from the MTS computer to the base station will first be packetized, adding much in the way of error correction. This data will then be sent to the base station between 30 and 100 times per second. Any more would be excessive.

The MTS computer and CCI boards will connect to one another via a UART connection. This protocol allows for easy integration and implementation, while still providing the bandwidth for the 30-100 hz polling rate required. The MTS computer will first send a command to the CCI board to receive data, and the CCI board will respond, sending whatever information it currently is reading from its sensors. Likewise, valve control will be issued from the MTS computer over the UART connection to the CCI board.

C. Software Abort

Backend software will have redline limits set for certain sensors, which may be updated from the base station. These limits will be checked against each sensor reading. If the sensor reading ever exceeds the redline limit, the hardware abort system will be triggered, cutting power to all SVs.

VI. Testing

A. Long Distance Communication

To ensure the safety of our team members, the control stand will be a minimum of 500 ft from the engine during firing. This presents a challenge, as there are few communication options that can handle the high data rates needed for the live data and camera feeds at that range. We first considered wireless communication using high-gain directional antennas, however we decided against this due to reliability concerns and cost. Due to the possession of fiber optic cable, this was considered to be an alternative, and the network switches we use do have SFP ports. We then decided against fiber due to our concerns about the durability of the fiber optic cable that we had, along with the additional complexity it adds to the system.

Finally, we decided on using Ethernet. This was because it is very durable, so we wouldn't have to worry about it getting damaged easily, as opposed to something like fiber. It is also a very common technology, so replacing parts or getting a longer cable would be fairly easy, in the event that something becomes damaged.

There is one problem with Ethernet though, and that is the range. A single Ethernet run is typically limited to 328ft, however we have found a way to get around this limitation. Approximately every 300 ft, we will have an Ethernet repeater with its own power supply, allowing us to reach the 500 ft target, and go even further if desired.

B. Data Collection

The data collection test will verify the capability of data logging. The test ensures that the CCI boards can collect samples quickly and trigger aborts promptly. This will be performed during a cold flow test of the engine on a test bench with a limited number of sensors. The CCI boards will take readings from all connected sensors during the cold flow test, and the data will be evaluated for accuracy and rate. The Pass Fail criteria are as follows:

Pass:

1) All PTs sampled at a rate of at least 1,000 samples per second
2) All TCs sampled at a rate of at least 25 samples per second
3) No data errors when saving data from onboard flash memory to SD cards

Fail:

1) 1,000 samples per second not achieved for all PTs
2) 25 samples per second not achieved for all TCs
3) One or more data errors when saving data from onboard flash memory to SD cards

If the system fails the test, the cause of the failure will be investigated. Once a problem and potential solution has been implemented, the test will be attempted again at the next opportunity. There are multiple cold flow tests to be run, and therefore multiple opportunities for data collection tests.

C. Full System Click Check

The full system click check test is one of the final tests required to be done before major full integrated testing, such as a full wet dress rehearsal. This test entails having the MTS and ETS fully assembled and fully wired properly, with the base station set up 500+ feet away from the ETS. The test will check continuity of all SVs in addition to checking functionality of PTs, TCs, and camera systems. Nitrogen gas will be used to simulate LOx and ethanol. During one of these click check tests, there will be a full mock firing sequence and two manufactured aborts, one to test the software abort and one to test the physical abort system. Testing the abort systems while the full system is together is vital due to the need to validate that all valves return to their safe state. The Pass Fail criteria for this test are the following:

Pass:

1) Confirmation of all 16 SVs are connected correctly and have successfully actuated their designated valve
2) Confirmation that all TCs are connected correctly and functional
3) Confirmation that all PTs are connected correctly and reading correct pressures
4) Communications between the MTS Controller and the base station remains constant and responsive
5) Software and physical abort systems work without issue

Fail:

1) One or more SVs are not connected correctly or have not successfully actuated their designated valve
2) One or more TCs are not functioning, possibly due to connection problems or part failure.
3) One or more PTs are not functioning, possibly due to connection problems or part failure.
4) Communication between the MTS Controller and the base station are not constant and there is a break in response
5) If one of the abort systems does not successfully return valves to safe state

If the test results in a failure, there will be an inspection into what caused the failure. Once the cause has been ascertained, corrective actions can be taken. The most likely solution will be to replace the sensor or SV that failed. If a replacement part is not available, the test will be scrubbed until a new part can be acquired.

VII. Conclusion

Our custom-designed control system meets all of the requirements for the application of being used to control a liquid rocket engine test stand. The system we have designed is fast to recognize and respond to a failure, minimising the risk of an incident while firing. This system is also able to gather significantly more accurate data at a higher rate than the previous system, all while costing significantly less. The system has been designed to be modular and easily serviceable. On the interface side, our GUI allows the operators to easily visualize live data from an array of sensors and live video from cameras mounted around the test stand and trailer. In order to ensure the reliability of the system, there is a series of tests that must be performed, especially before any full system tests. With all of this combined, we are sure that once complete, this will be one of the most advanced control systems in the student rocketry world.

Acknowledgements

We would like to thank the Space Hardware Club and its faculty advisor, Dr. Gang Wang, for giving us the opportunity to partake in this project. We would also like to thank the Alabama Space Grant Consortium and the University of Alabama in Huntsville for their financial support of the Space Hardware Club. We would like to thank Yellow Jacket Space Program for hosting their Liquid Rocketry Conference in November of 2024 and giving us the

inspiration to overhaul and redesign the control system. Finally, we would like to thank all members of the controls subteam of Tartarus for putting their effort into designing the new control system.

References

[1] Microchip Technology Inc., “SAM D5x/E5x Family Data Sheet”, *Microchip Technology Inc.,* 2021, URL: https://ww1.microchip.com/downloads/en/DeviceDoc/SAM_D5x_E5x_Family_Data_Sheet_DS60001507G.pdf [retrieved 12 January 2025].

[2] Maxim Integrated, “Precision Thermocouple to Digital Converter with Linearization”, *Maxim Integrated Products, Inc.*, 2015, URL: https://www.analog.com/media/en/technical-documentation/data-sheets/MAX31856.pdf [retrieved 10 December 2024].

[3] Panasonic Corporation, “High sensitivity, 100 mW Nominal operating power, 2 Form C and 2 A Compact flat body type relays”, *Panasonic Corporation Automation Controls Catalog*, 2019, URL: https://mm.digikey.com/Volume0/opasdata/d220001/medias/docus/905/AGQ_(GQ)_Relays.pdf [retrieved 12 February 2025].

[4] Nuvoton Confidential, “NAU7802 24-Bit Dual-Channel ADC For Bridge Sensors”, *Nuvoton Confidential*, 2012, URL: https://www.nuvoton.com/resource-files/NAU7802 Data Sheet V1.7.pdf [retrieved 4 February 2025].