HyRo (Working title)

Jason Klindtworth | Josh Asher | Layne Nolli

CS461

Fall 2016

Abstract

High Altitude rockets have a distinct advantage in using Hybrid propulsion systems. These systems are complex and present challenges in remote telemetry including launch initialization and controlling remote fuel filling/disconnect. High altitude rockets also contain an array of sensors that collect data which needs to be visualized in a human friendly format. The goal of HyRo is to provide mechanisms for remotely launching and controlling the fuel systems on a hybrid propulsion system through onboard embedded circuitry/software that communicates to launch team via radio waves. This circuitry/software will also communicate sensor data to the ground. Sensor data is displayed in our visualization software in an appealing, human readable way.

October 24, 2016

I. Problem Definition

Hybrid rocket motor systems have distinct advantages over traditionally fueled rocket, but tend to be more complex. A hybrid rocket fitted with proper software has improved on-board control of the systems. Humans in the ground control unit will benefit from these improvements, which will make safely handling lauch, controlling the rocket, and visualizing the rocket's data much easier than before. Desired data to be visualized includes, but is not limited to, tank pressure, chamber pressure, tank temperature, acceleration, barometric pressure, velocity, and GPS coordinates.

Our problem consists of designing software that will run on-board the rocket, on-board a ground control unit, and on a traditional computer. Communication from the rocket to the ground will be done via radio waves and to the computer via serial communication. The on-board software will also be responsible for collecting sensor data and either analyzing that data to adjust on-board system devices like valves and switches or sending the data to the ground control unit for visualization. The ground unit will be responsible for communications with the rocket, accepting commands from buttons to be sent to the rocket, and communicating to software on the traditional computer. Software on the traditional computer will be responsible for accepting user input, issuing commands to the system, and receiving data to be visualized.

II. Problem Solution

The system we are designing will need to be able to issue commands to the on-board system in the rocket, these commands include arming, disarming, system related commands, launch, fill, abort and ignition. The system will include a UI to send any vital system commands. The ground control system will be designed to have bidirectional communication with a computer that will house a graphical user interface to visualize data and issue commands to the ground system that will be sent to the rocket.

Our hybrid rocket system has 3 main independent software components working together. Each component will tackle specific problems in their own way. We will have software running on-board the rocket, the ground control unit, and on a traditional computer. This software will be able to communicate together and with it a ground crew will be able to give commands to the rocket and also collect all the required aviation data.

These three components present a solution to controlling a hybrid rocket and visualizing collected sensor data from the rocket. With these systems combined we plan to be able to control a hybrid rocket from a ground control unit in combination with a personal computer along with viewing data in a human friendly way. We will design a protocol for communication between all these system to unify the communication amongst components.

A. *On-board Rocket Software*

This piece of the system will be located on a Beagle Bone Black embedded Linux micro-controller. This embedded device will be connected to the rocket's on-board sensors, system components (like valves and switches), and to a radio frequency device for communicating data to and from the ground unit. It will be responsible for any in-flight adjustments and the sending and receiving of data and commands to the ground unit.

B. *Ground Control Unit*

The ground control unit will be a small enclosed box running software capable of monitoring button input, radio communication to the rocket and serial communication to the traditional computer. Its main responsibility will be to provide the link between the rocket and a traditional computer. Also providing physical buttons to issue the rocket commands.

C. *Traditional Computer Software*

This software that will run on a traditional computer and operating system. It will be responsible for sending and receiving data from the ground unit, providing a graphical interface to view data and providing inputs to issue commands to the rocket. This UI will present the user with a series of inputs that can be used to send commands to the rocket and a series of windows to display the sensor data from the rocket, including GPS which might be translated into a visual map to give the current location of the rocket. This vision might be expanded to include calculations and suggestions on how to adjust the current state of the rocket. This software will also be responsible for any calculation required to give recommendations on adjusting any on-board rocket system components.

III. Performance Metrics

Our proposed solution to visualizing data and controlling functions on a hybrid rocket has a growing list of metrics that will measure or success at the end. Parts of our system are yet to be designed, but some metrics we have thought of are as follows. Time is a critical metric in our system. All components need to be able to communicate and responds to each other as fast as possible. If they fail to do so launching, control, and visualization of rocket data could completely fail. We must also consider human error. If someone was to misunderstand what we have visualized, we will need to provide safety features to protect against incorrect human input. Another human based metric is that our system will be used by a wide range of engineers and needs to appeal to this interdisciplinary crowd. To do so we will have to developed user cases and test them on each individual team mate to make the platform more appealing to the entire rocket team. If everyone is happy with the system at the end we have succeeded in this metric.

To measure our success on visualizing data on the rocket the system will have to successfully communicate data from on-board the rocket, through radio transmission, through USB communication and be interpreted and visualized on a computer accurately. There are many sub components and metrics to this part. Each individual component will need to be accurate on its own, be able to send a the appropriate amount of data, and deliver it in a time frame suitable for interpretation. We will have to compare raw data to our UI and make sure we are accurately representing that sensor data. This is the same case for any commands sent through the system. Each individual component must accurately check for errors in communication and pass on the message to the onboard rocket system intact. The onboard system then will have to send signals to the onboard components to produce physical changes. This will have to be done fast and accurately as well. Failure to do so will result in lost or incorrect commands.

Students:

Jason Klindtworth

Signature

Date

Josh Asher

Signature

Date

Layne Nolli

Signature

Date

Client:

Nancy Squires

Signature

Date