

# 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 20, 2016

## I. PROBLEM DEFINITION

Hybrid rocket motor systems have distinct advantages over traditionally fueled rocket, but tend to be more complex. There is a need for software to allow a hybrid rocket to have on-board control of systems that can monitor and adjust any components as required by the rocket's sensors. More so there is a need for a system to communicate to a ground control unit that has the ability to collect sensor data and process that data in a way that is in a human readable format. Desired data to be visualized includes, but is not limited to, tank pressure, chamber pressure, tank temperature, acceleration, barometric pressure, velocity, and GPS coordinates.

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 system that will be sent to the rocket.

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

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 this rocket has very defined metrics of success. 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. This will be successful if our UI can show sensor data accurately and in an easy way for humans to understand. If all data is accurate according to the raw data, then we have succeeded in visualizing the data from the rocket.

To measure of the success of our command protocol throughout the system is fairly easy. If a user inputs a command into the control box or the computer and the rocket does not respond appropriately we have failed. If the rocket responds to all of our commands i.e. launch and ignition, then we have successfully communicated all commands to the rocket. If our software includes any automatic adjustments these adjustments also must successfully produce the outcome they were designed to produce. For example, adjusting throttle on the rocket. This will succeed if the throttle is adjusted according to the appropriate calculations.

**Students:**

Jason Klindtworth

.....  
*Signature*

.....  
*Date*

Layne Nolli

.....  
*Signature*

.....  
*Date*

**Client:**

Nancy Squires

.....  
*Signature*

.....  
*Date*

Josh Asher

.....  
*Signature*

.....  
*Date*