Assessment Task 1:
Project Proposal and Risk Assessment

RMIT University
Semester 1 – 2024

Supervisor: Dr. Glenn Matthews

Name	Student No.	Contribution
Matthew Ricci	s3785111	50%
Jeremy Timotius	s3779178	50%

Contents **1**

Summary 2
1 Problem statement 2
1.1 Project Deliverables 2
1.2 Tools to Undertake Data Analytics and Verification 2
2 Background and literature review 2
3 Research questions 3
4 Methodology 3
4.1 Functional breakdown 3
4.2 Data storage and handling 4
4.2.1 Dataframe structure 4
4.2.2 Data typing 4
5 Synchronisation 6
5.1 Mid-flight processing 6
6 Risk assessment 6
6.1 Ethical considerations 7
References 9

Summary

1 Problem statement

The Data Analytics and Verification team is to operate as a subset within the AURC Avionics Team for the HIVE Aurora V rocket project. This subsystem's primary objective is to design, test, and validate data analytics processes for the avionics, ground communications, redundant electronic systems, and payload systems.

The project scope spans across five iterations of the Aurora rocket (A1 through to A5). As each rocket iteration progresses, data analysis processes should change to improve the accuracy of verification processes. Feedback during analysis is also to be provided to the broader Avionics team.

1.1 Project Deliverables

Data analytics and verification processes will include:

- Real-Time Processing: Research and develop algorithms for sensor fusion to integrate data from multiple sensors. Implement filtering techniques to remove noise.
- Post-Processing: Perform statistical analysis to evaluate key performance metrics of avionics systems. Implement systems to detect deviations from expected behaviour when comparing raw sensor and fused data to the data captured by the blue raven.
- Data Visualization: Develop tools for generating informative graphs and charts for data interpretation. Explore the possibility of creating simulations based on collected data.
- Data Validation: Determine the validity of the captured data. Identify the nature of errata (error with our analysis methods or hardware issue?)

1.2 Tools to Undertake Data Analytics and Verification

Due to the nature of work, most tools will be software based. The processor of the microcontroller will influence which integrated development environment (IDE) will be used. Tools such Arduino IDE, Visual Studio IDE and Microchip studio will most likely be utilized for development. Although the use of code libraries should be avoided to maintain low level control of the avionics system, some libraries may be utilized to act as a starting point. For version control and collaboration, Git will be used throughout the course of the project. A list of tools that will be required to complete the project includes:

- Git version control
- Integrated Development Environments including Arduino and STM32CubeIDE
- Arduino and Python libraries for prototyping, testing, and offline data processing
- Embedded systems analysis tools such as debug probes and logic analysers

2 Background and literature review

The Aurora V project aims to build a L3 rocket designed to fly with apogee at exactly 30,000ft. To achieve this goal, accurate data collection and analysis is a necessity both to understand the flight path of the rocket and its contributing factors to facilitate correcting modifications to the rocket itself, as well as to supply real-time state information to the airbrake system for dynamic adjustments to drag coefficients.

Inertial sensors are susceptible to noise in their measurements, which introduces error into captured data and accumulates a drift in time-varying calculations. Many model rockets such as Pioneer Rocketry's Skybreaker[1] implement some variation of a Kalman Filter to smooth out this noise for more accurate state estimation. A Kalman Filter attempts to provide an accurate estimation for noisy processes by using a model in combination with previous estimates to predict the future state of the process, correcting this prediction with measured data[2, 3]. This process accounts for the process noise - deviation between the true state of the process and the state as described by the model - as well as the noise generated by the measurement of state values. A Kalman gain is computed to weight how much of the modeled and measured data should be accounted for in the new estimate, with an error covariance matrix tracking the uncertainty in the estimate.

Rockets built by university teams such as uORocketry and NTNU generally make use of positional data such as height or displacement together with velocity measurements to control their airbrake systems[4, 5]. To accurately determine these state variables, sensor fusion techniques may be employed to reduce the accumulated error that results from the measurement process.

An example of how this may be applicable is in the altitude calculations. The most simple method of determining altitude would be to directly use the barometric pressure, however this is highly dependant on calibration at the launch site and susceptible to noise introduced from the sensor. An alternative would be to use integrated accelerometer data to obtain position, however the process of integrating itself is prone to drift due to time lag as well as flight angle offsets. A potential solution for these problems would be to combine the barometric altitude estimate together with positional calculations from the integrated accelerometer data, as described in a 2004 report by David Schulz[6]. Accelerometer data can also be combined together with gyroscopic measurements from the IMU to rotate the inertial frame, allowing for pure vertical acceleration data to be integrated down to positional information. In all cases, some application of a Kalman filter appears necessary.

3 Research questions

RQ1. What variables are necessary to track in the rocket state?

RQ2. What algorithms should be run to accurately estimate rocket state in real time?

RQ3. How will sensor fusion techniques be applied to obtain state information from available data?

RQ4. What statistical methods will be employed to assess real time validity of data?

- (a) What communication protocols will be used to inform the redundant systems of detected errata?
- (b) What metric(s) will the redundant systems use to elect which of primary and secondary sensors to sample?

RQ5. What visualisation methods will be applied to provide state representation in real time?

RQ6. What visualisation methods will be applied to provide state information offline?

- (a) What techniques will be applied to analyse trends in offline data?
- (b) What trends are important to analyse in offline data?

4 Methodology

4.1 Functional breakdown

The high-level flow chart shown in Figure 1 depicts how raw sensor data is to be captured and transmitted from a firmware perspective. The purpose of the flow chart is to guide avionics on how data is to be logged and transmitted to enable post flight analysis and verification. The system is primarily broken down into a high-resolution interval at 500Hz and low-resolution interval at 50Hz.

These resolution intervals are intended to match with Blue Raven intervals in order to sync and compare data sets post flights. The Blue Raven data will act as benchmark to determine the accuracy of data collected from various sensors of the avionics unit. Logging of data will be triggered based on a launch event recorded from the Blue Raven to enable both systems to collect data simultaneously.

During flight, captured sensor data will be written to an onboard flash memory and transmitted over LoRa to ground communications, providing a level of redundancy to mitigate the risk of data loss. This dual approach ensures that if one method encounters failure or interruption, data can still be processed and analyzed post flight.

While flash memory can only be analyzed post flight, LoRa communications provides critical live communications to the ground, enabling real-time monitoring and analysis of sensors and microcontrollers. Real-time processing through the use of algorithms is to be developed with the purpose of detecting hardware failures and or abnormalities to inform avionic systems to switch to redundant sensors or microcontrollers if required.

Only Raw data will be collected and transmitted from each sensor register during flight. The aim of this approach is to optimize data collection and transmission rather than processing which will be conducted post flight. By doing so, onboard processing will be minimized, reducing the risk of processing errors and ensuring efficient utilization of LoRa bandwidth and onboard flash memory.

During the high-resolution interval, the system reads data from an IMU (Inertial Measurement Unit) which consists of an accelerometer, gyroscope, and magnetometer sensors. These sensors are essential in determining the aircraft motion and orientation; therefore, a higher sampling rate is required to capture subtle changes accurately. Raw data from sensor registers are then combined with header to form a data frame which will be stored in a buffer array ready to be stored in flash. Sensor data will additionally be kept in RAM to be transmitted during the low-resolution interval.

At the low-resolution interval, the system reads data from a barometer, which includes a pressure and temperature sensor to determine altitude. Similar to the high-resolution interval, raw data are used to form a data frame which is added to a buffer array serving as temporary storage before the data is written to flash. Once pressure and temperature data frame is formed and saved to RAM, the buffer array is written to flash and data stored in RAM will be transmitted over LoRa. Switch I/O will also be written to RAM and transmitted to ground control to notify avionic systems if a switch has turned off.

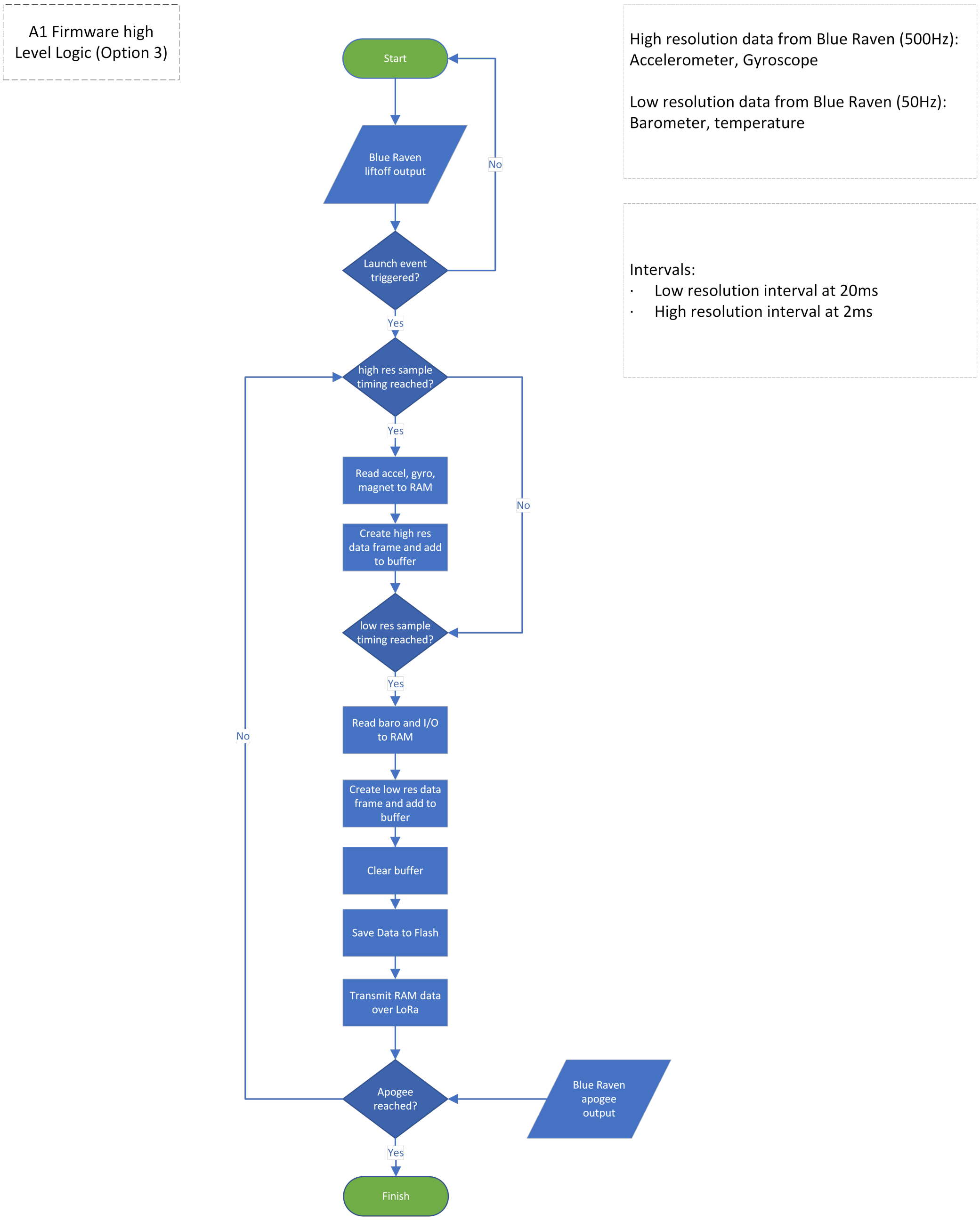
A number of alternate design methods were considered which included a separate logging interval where transmission and saving to flash occurred over a separate time frame. Although having a different interval provides greater control, flash writing and LoRa transmission occurring during on the low-resolution interval simplifies the design, consolidating data handling tasks.

4.2 Data storage and handling

4.2.1 Dataframe structure

Figure 2 provides a bytefield structure for the dataframes being constructed from the on-board avionics sensors. Each frame contains a two-byte header that identifies the type of data recorded in the payload (e.g. high resolution, low resolution or rocket payload data), describes the number of bytes contained within the payload, and provides a synchronisation byte for post processing.

These dataframes will be stored within a `uint8_t` buffer as opposed to a 16-bit equivalent. This is because some sensor data is contained within 24 bits per sample, so to make efficient use of space while maintaining simplicity of implementation 8-bit nibbles are desirable.


Start
Blue Raven liftoff output
Launch event triggered?
No
Yes
high res sample timing reached?
Yes
Read accel, gyro, magnet to RAM
Create high res data frame and add to buffer
No
low res sample timing reached?
Yes
Read baro and I/O to RAM
Create low res data frame and add to buffer
Clear buffer
Save Data to Flash
Transmit RAM data over LoRa
No
Apogee reached?
Blue Raven apogee output
Yes
Finish

A1 Firmware high Level Logic (Option 3)

High resolution data from Blue Raven (500Hz): Accelerometer, Gyroscope

Low resolution data from Blue Raven (50Hz): Barometer, temperature

Intervals:
- Low resolution interval at 20ms
- High resolution interval at 2ms

Figure 1: High-level functional flowchart of data sampling and logging

4.2.2 Data typing

Data stored by the avionics system will retain the raw structure as output by the on-board sensors, written as 8-bit unsigned blocks. This is to minimise the amount of storage used with each frame, as well as to avoid any processing overhead from computing real data.

Processing can then be done offline during analysis to obtain real data, for example by multiplying out the sensitivity of a given sensor into a floating point variable.

The payload of each dataframe will consist of consecutive groupings of sensor data, an example of which is pictured in Figure 3. No delimiter will be used to separate these groupings and instead processing will be performed by using an agreed upon structure.

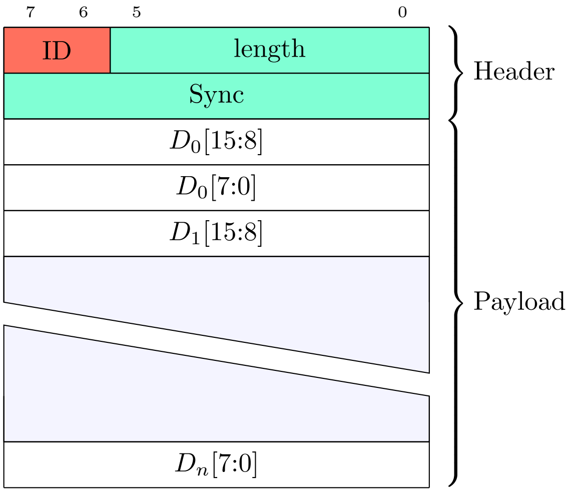


Figure 2: Dataframe structure for avionics

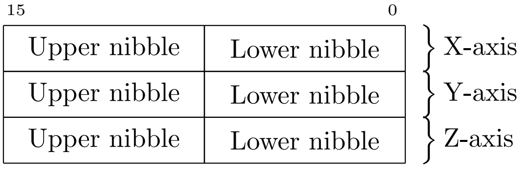


Figure 3: Example structure for a single sensor payload

Code that processes this data will need to read in each dataframe from memory, using the length field in the header to delimit where the data ends. Based on the type information from the frame header, each variable of data can then be read.

5 Synchronisation

According to the Blue Raven user's guide [7], synchronisation between high and low resolution data is achieved through a ”sync code” that is stored with the data. This code is essentially a counter that increments with each millisecond, with overflow at a count of 250.

To maintain synchronisation both with datasets collected by Aurora V avionics, as well as with the Blue Raven data (for the sake of comparison), an identical solution will be implemented within the logging system. As outlined in Section 4.2.1, this sync code is included as a full byte in the frame header, following the id and frame length. This byte will be included with all forms of saved data to maintain synchronisation parity across the board.

5.1 Mid-flight processing

6 Risk assessment

From an occupational health and safety perspective, the risks associated with this project are minimal. Due to the nature of this project, most risks will be office related, which include, visual strain, neck, and back pain from long hours in front of computers causing eye strain, headaches, and fatigue. Although these risks pose health concerns, they can be effectively managed through ergonomic interventions such as the use of ergonomic furniture and regular breaks during project work. Additionally, some aspects of testing verification will include interaction with hardware such as sensors and other electronic modules.

Risk	likelihood	Impact	Contingency Strategy
Delay arrival of avionic hardware	High	High	• Work closely with Avionics team to understand if there are any Hardware changes or substitutions. • Prioritise non hardware dependent tasks. • Focus on preliminary data analysis, algorithm development, or simulation testing while awaiting hardware arrival.
Data collected from sensors is corrupted or incomplete	Medium	High	• Develop an algorithm for data validation and error detection to identify and filter out unreliable data.
Loss of ground communication during flight	Medium	High	• Testing processes to ensure that data can be extracted from flash.
Flash data unrecoverable	Medium	High	• Ensure that enough senor data is being transmitted in both high- and low-resolution transmissions. This will still enable data analysis to take place even if flash data is lost.
Scope creep	Low	Medium	• Establish clear project scope. • Prioritise data analysis and verification task. • Regularly review project scope and assess proposed changes with Glenn Matthews.
Integration Challenges with ground communication and real-time processing	Medium	Medium	• Understand the specifications between the communication hardware and processing algorithm. • Conduct integration checks and testing.

Table 1: Risk assessment

While these interactions may introduce physical hazards such as the risk of electrical shocks or injury from handling equipment, hardware is to be handled in RMIT Laboratory rooms where proper procedures and PPE will be utilized. Hardware will also be regularly inspected and maintenance to help identify and address potential safety hazards proactively.

As data analysis and verification is heavily reliant on the avionic hardware, there are foreseeable risks that may impact how data analysis and verification is approached. Table ?? outlines possible risks and contingency strategies to ensure project deliverables are met.

6.1 Ethical considerations

As this project is primarily focused on data analysis and verification through software, ethical considerations are minimal. However, due to the nature of the aerospace and the work conducted, some considerations are as follows.

Although the test flights and the official AURC competition are held in controlled environments with safety precaution in place, the real-time data logged through LoRa may play a factor in identifying a potential safety risk.

Over the course of multiple Aurora flights, a large amount of data will be collected and stored. It should be ensured that the data and analysis should be only used for its intended purposes. As this work is conducted under RMIT and the HIVE association, these stakeholders should be consulted before data is shared externally.

The Aurora team is made of five subsystems with a total of over thirty members from different backgrounds and walks of life. Ethical collaboration is essential to ensure that all team members are treated with respect and have equal opportunities to contribute to the project. Open communication among team members should be encouraged to create a culture of inclusivity.

References

[1] J. Ellenberger, "Pioneer rocketry-2016 midwest rocket competition," in *Proceedings of the Wisconsin Space Conference*, 2016.

[2] G. Welch, G. Bishop, *et al.*, "An introduction to the kalman filter," 1995.

[3] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[4] Z. Furman, G. Beebe, F. Bjerknes, and D. Montgomery, "An autonomous air braking system for precision altitude control," in *2019 Spaceport America Cup Conference Podium Presentations*, 2019.

[5] H. Dragesund, T. Haddal, P. Holmqvist, H. Løvik, E. Rydningen, M. Sunde, B. Tapley, and T. Winter, "Apogee precision with airbrake control and srad rocket simulator," in *2019 Spaceport America Cup Conference Podium Presentations*, 2019.

[6] D. W. Schultz, "Application of the kalman filter to rocket apogee detection," *NAR Research and Development Report*, 2004.

[7] A. Adamson, "Blue raven user's guide." `https://www.apogeerockets.com/downloads/PDFs/09170-blue_raven_users_manual_september_15.pdf`, 09 2023.