Aftershock II Apogee Analysis

Will Johns, Hayden Brophy, Stracy Hoang, and Madeline S.*

USC Rocket Propulsion Laboratory
(Dated: November 2024)

Following the successful launch and recovery of Aftershock II, the second of the USC Rocket Propulsion Laboratory's Aftershock-series solid-motor vehicles, data from the on-board avionics unit was collected and used to reconstruct its flight path. After an internal review of the raw data, advanced filtering methods and reconstruction simulations were used to determine and validate an apogee of 470 400 ft ± 27 300 ft (3σ), breaking the amateur altitude record of 380 000 ft, previously set by CSXT's GoFast rocket. USCRPL also determined a maximum speed of 5283 ft/s, breaking the record of 5019 ft/s also set by the GoFast rocket.[a] This result establishes Aftershock II as the fastest and highest amateur rocket of all-time.

CONTENTS

[a] CXST

* analysis@uscrpl.com

I. INTRODUCTION

On October 20, 2024, the University of Southern California Rocket Propulsion Laboratory (USCRPL), launched Aftershock II from Black Rock Desert, NV. The vehicle was designed with the goal to become the highest and fastest amateur rocket of all time, a record previously set by CSXT's GoFast in 2004, which reached an altitude of 380,000 ft (116km) AMSL[1] and a maximum velocity of 5,019 ft/s (1,530 m/s). This flight marked USCRPL's second successful spaceshot launch and recovery, following the successful flight of Traveler IV in 2019. This paper details the post-flight analysis of the data collected by Aftershock II's avionics, with the goal of reconstructing the vehicle's flight path and determining its apogee.[2] [3]

FIG. 1. Aftershock II.

[1] Above Mean Sea Level
[2] Access to Aftershock II's flight data and other resources is available to the general public on a case-by-case basis. Access requests are to be directed to analysis@uscrpl.com.
[3] The USCRPL team would like to thank the faculty of USC's Aerospace and Mechanical Engineering and Astronautical Engineering Departments for their initial review of this document, as well as our corporate sponsors for their generous donations and the USC Viterbi School of Engineering for their continued funding and support.

I.1. Terminology

This analysis will use the following naming conventions to refer to various important physical frames of reference and concepts:

- **Apogee:** The highest point in the trajectory of the rocket, where its vertical velocity momentarily becomes zero before descending. This point marks the peak altitude during ascent.
- **Burnout:** The point during the rocket's flight when the motor ceases to produce thrust, and the vehicle transitions from powered flight to coast phase. At burnout, the motor's thrust drops to zero.
- **Quaternion:** A mathematical representation used to describe rotations in three-dimensional space. Quaternions are used to represent the orientation of a sensor or body frame relative to a reference frame (e.g., the world frame) without the gimbal lock issues associated with Euler angles.
- **Inertial frame** or **world frame:** The inertial frame of the Earth, through which the vehicle travels. In this frame, z is normal to the Earth's surface, while x corresponds to magnetic north, and y is perpendicular to x and z as described by the right-hand rule. In this frame, a free-falling object has $a_z = -g$, and a grounded object has $a_z = 0$.
- **Body frame:** The reference frame centered on the vehicle. The body frame origin is at the vehicle's center of gravity, while x points through the vehicle nose tip, and y and z are defined to point in the same direction as the y and z axes of the HAMSTER avionics unit.
- **Sensor frame:** The reference frame centered on a particular sensor. The x, y, and z axes of the sensor frame are assumed to be orthogonal and aligned with the axes of the sensor itself, which may differ slightly from the body frame depending on the sensor's orientation.
- **Transformed frame:** A transformed frame is the coordinate system of a sensor frame transformed using a quaternion such that the x, y, and z axes are oriented the same as those of the world frame, but the coordinate system origin remains anchored at the sensor origin. In this frame, a free-falling object has $a_z = 0$, and a grounded object has $a_z = +g$.
- **Earth-Centered Earth-Fixed (ECEF) Coordinate System:** A non-inertial frame that uses Cartesian coordinates with the origin fixed at Earth's center. Positive Z extends out from the North Pole, Positive X points in the direction where Earth's equator intersects the prime meridian (0°longitude, 0°latitude), and Y extends eastward along the equator.
- **North-East-Down (NED) Coordinate System:** A non-inertial frame with the origin fixed at the center of gravity. The Z axis points downward normal to Earth's surface, the X axis points North tangent to Earth's surface, and the Y axis points east tangent to Earth's surface. The specific location is defined by longitude, latitude, and altitude.
- **IMU (Inertial Measurement Unit):** A device that combines multiple sensors, typically accelerometers, gyroscopes, and sometimes magnetometers, to provide comprehensive motion tracking. The IMU measures the rocket's acceleration, angular velocity, and sometimes orientation, helping to estimate position and velocity over time.
- **GNSS (Global Navigation Satellite System):** A system of satellites that provide location and time information to receivers on Earth. GNSS is used in the avionics system to track the rocket's position and altitude during flight.

II. DATA COLLECTION SYSTEMS

The avionics system consists of multiple data acquisition systems, enumerated below.

1. The High Altitude Module for Sensing, Telemetry, and Electronic Recovery (abbrev. HAMSTER) is USCRPL's custom avionics system. HAMSTER is designed and fabricated in-house[4] and hosts a variety of sensors as detailed in Section II.1.

   (a) The *Sensor Board* is the primary controller of HAMSTER as it executes flight software, gathers data on sensors (acceleration, temperature, barometric, and GPS), and transmits it to the ground. Additionally, it handles the logic for deploying the drogue parachute recovery system.

   (b) The *Integrator Board* performs sensor fusion and live double-integration of magnetometer, gyroscope, and accelerometer data in order to determine burnout velocity and detect apogee.

   (c) The *Lightspeed Rangefinder Transponder* performs radio-based distance measurements between multiple points on the ground with the rocket as it ascends. In post-processing, the flight path of the vehicle can be determined via trilateration.

   (d) The *Guidestar GPS Board* uses the existing GNSS systems to locate the rocket's location and altitude. Since it is built in-house, it does not have the typical CoCom limitations that commercial units possess.

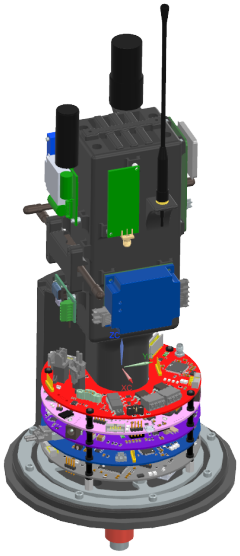

FIG. 2. HAMSTER avionics unit.

2. The *BigRedBee BeeLine GPS*[5] (abbrev. BRB) is a self-contained unit that records GPS packets to non-volatile memory and transmits those GPS packets over the 70cm RF band using the APRS protocol. For this flight, the BigRedBee was configured to send a data packet every 5 seconds.

3. The *Featherweight Blue Raven Altimeter*[6] (abbrev. Raven) is a self-contained data acquisition system often used in amateur rocketry. The Raven features a three-axis accelerometer, a three-axis gyroscope, and a barometric altimeter. It also offers high-current outputs to fire the recovery system after apogee is reached.[7]

[4] HAMSTER is fully designed and built in-house. Everything from the printed circuit board to the code and structures have been custom-created to meet our unique requirements.
[5] `http://www.bigredbee.com/BeeLineGPS.htm`
[6] `https://www.featherweightaltimeters.com/blue-raven-altimeter.html`
[7] This function serves only as a backup on our avionics system, and was not put into use during the flight of AFTERSHOCK II due to HAMSTER's nominal firing of the chutes.

II.1. Hamster Operation

The HAMSTER flight software operates as a state-machine where each state represents a specific phase of the flight, with transitions that occur in response to user commands or sensor-based criteria. For instance, the system starts in the STANDBY state, where it is idle and ready to arm. Once armed, the unit is actively monitoring acceleration to detect liftoff. The operator may send the TERM command which will put the unit into the BURNING state in ten seconds, or if the unit detects a spike in acceleration - indicating liftoff - it can transition itself into BURNING state. Shortly after liftoff, there is a 10 second window where the unit will check to see if it is in free fall or the end of the window has passed and switch to the BURNOUT state if so. After T+164.5s, the unit will begin checking the integrator board for apogee ($P_{ecef} \cdot V_{ned} \leq 0$) and transition the unit into the DROGUES_FIRING state as soon as apogee is detected or 16 seconds have passed, where it will deploy the parachute for recovery and switch to the DROGUES_FIRED state for descent.

State	Transition Time
STANDBY	-
ARM	T-120s
TERM	T-10s
BURNING	T-0s
BURNOUT	T+11.8s ... T+21.8s
DROGUES_FIRING	T+164.5s ... T+180.5s
DROGUES_FIRED	T+190s
LANDED	T+1024s

TABLE I. HAMSTER States and Transition Times.

HAMSTER also allows for various diagnostic and recovery remote procedure calls like PING, SET_STATE, HALT/STEP, SET_CACHE, and MEMCPY which can be used to extract data or troubleshoot the system. This ensures that HAMSTER operates reliably throughout the mission.

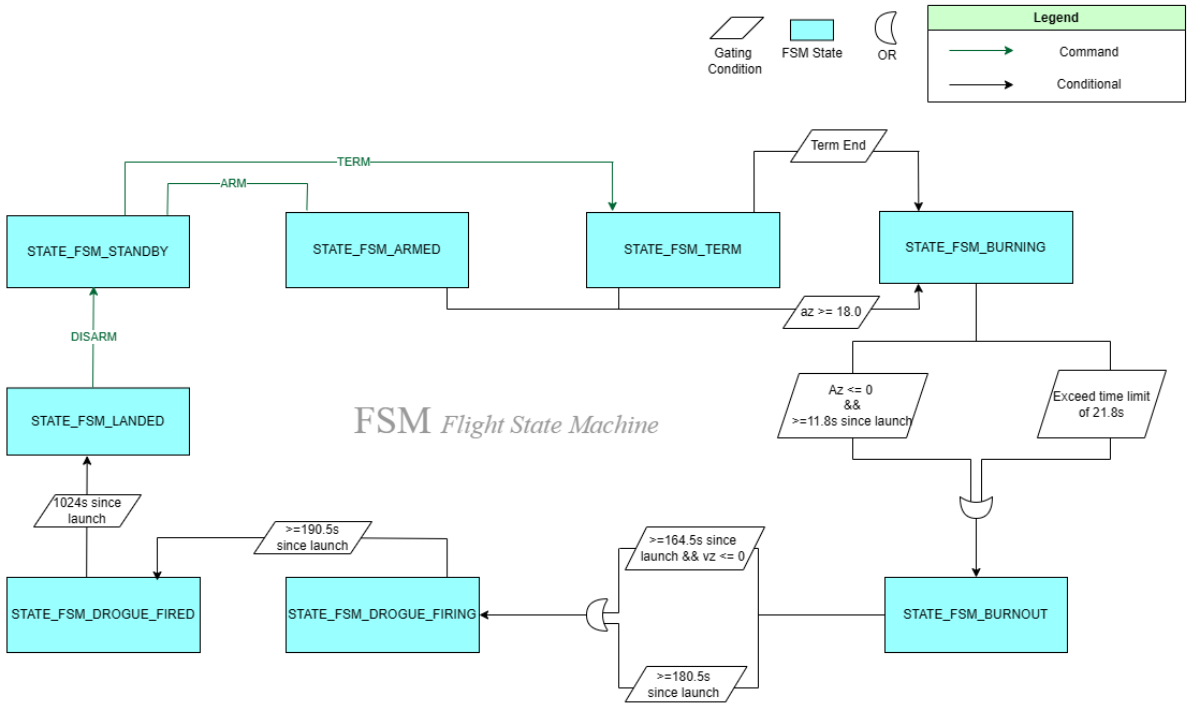

FIG. 3. HAMSTER State Machine.

The HAMSTER coordinate system is defined with the x-axis passing through the rocket's nose tip, and the y- and z-axes being orthogonal to each other. Figure 4 illustrates this principal coordinate system, which serves as the reference for all sensor data recorded during the flight.

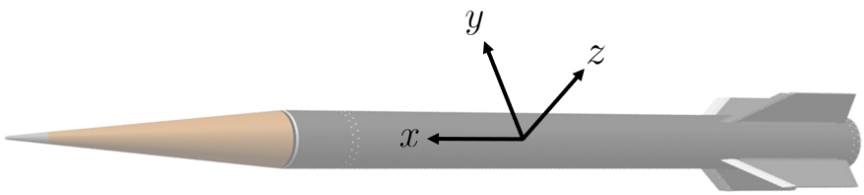

FIG. 4. Rocket's Coordinate System.

HAMSTER includes seven environmental sensors listed in Table II, which are sampled and logged at different rates before, during (for a period of 10s), and after deployment of the drogue parachute.

Model	Function	Standby	In-Flight	Landed
ADIS16467	IMU	5Hz	200Hz	10Hz
BMI323	Gyroscope	5Hz	200Hz	10Hz
KX134	Accelerometer	5Hz	200Hz	10Hz
MMC5983MA	Magnetometer	5Hz	200Hz	10Hz
BNO080	IMU	5Hz	20Hz	10Hz
MS5607	Barometer	5Hz	10Hz	10Hz
LM75B	Temperature Sensor	2.5Hz	3Hz	3Hz
ZED-F9P	GPS	1Hz	1Hz	1Hz
INA226	Power Sensor	10Hz	10Hz	10Hz

TABLE II. HAMSTER's logging rates.

HAMSTER's sensor data is recorded in an on-board 128MB NOR flash IC using RPL's TINY-TRANSFER binary format. The FlashLog system two checksums with every data packet, which allows us to confirm data integrity post-flight. Any malformed packets, or other packets which did not pass the integrity checksums (this turned out to be 0 of the 92,079 packets) were not used in our analysis. The FlashLog records the entire persistent state of the rocket (called the cache) for every packet, so the system can be fully restored to any state of flight from one packet.

Some of the MEMS sensors on the HAMSTER unit have measurable errors within their readouts, which can result from manufacturing defects, their placement on the PCB, and other process variables. These errors are fixed via calibration before flight, with exceptions as noted below. The BNO080's magnetometer was calibrated with the entire nose cone integrated in order to avoid hard iron error from the stainless steel CO2 canister.

Additionally, as a sensor designed for very high accelerations, the readout from the KX sensor is unreliable when its value is near 0 g, making it unusable during freefall.

Table III lists the specific data elements logged, along with their essential characteristics, for each of the sensors in Table II.

Sensor	Data Element	Units Stored	Min	Max	Error	
ADIS16467-1[a]	Acceleration	m/s^2	$-40g$	$+40g$	$0.697g$ RMS	±0.4%FS
	Body Rate	rad/s	-125 °/s	+125 °/s	±0.05 °/s	
KX134-1211[b]	Acceleration	m/s^2	$-64g$	$+64g$ °C	$0.814g$ RMS	±0.6%FS
BMI323[c]	Acceleration	m/s^2	$-2g$	$+2g$		±0.1%FS
	Body Rate	rad/s	-125 °/s	+125 °/s	±3 °/s	
BNO080[d]	Acceleration	m/s^2	$-2g$	$+2g$	±0.02g	
	Orientation	Quaternion			3.5°	
	Magnetic Field Vector [e]	µT	−1300 µT	1300 µT	8%	
LM75B	Temperature	°C	-55 °C	+125 °C	±3 °C	
ZED-F9P-04B [f]	Acceleration	m/s^2	$-4g$	$4g$		
	Velocity	m/s	0	1640 ft/s	±0.05%	
	Altitude	m	0	262000 ft	±6 ft	
	Acquisition Time		2 sec	30 sec		
MS5607	Altitude	m	0 ft	100000 ft	± 350 ft	
	Pressure	Pa	30000 Pa	110000 Pa	±50 Pa	
	Temperature	°C	−40 °C	85 °C	±4.0 °C	
MMC5983MA	Magnetic Field Vector	µT	-800µT	800µT	± 0.8µT	
INA226	Battery Voltage	V	0V	36V	0.036V	

[a] https://www.analog.com/media/en/technical-documentation/data-sheets/adis16467.pdf
[b] https://fscdn.rohm.com/kionix/en/datasheet/kx134-1211-e.pdf
[c] https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bmi323-ds000.pdf
[d] https://cdn.sparkfun.com/assets/1/3/4/5/9/BNO080_Datasheet_v1.3.pdf
[e] Corrected magnetic field, excluding magnetic errors from metal in the vehicle.
[f] https://content.u-blox.com/sites/default/files/ZED-F9P-04B_DataSheet_UBX-21044850.pdf

TABLE III. Types of data logged by HAMSTER.

II.2. Featherweight Blue Raven Operation

The Raven logs accelerometer and altimeter data, whose characteristics are listed below. Ravens have two modes of operation, which begin upon detection of launch. They can either fire the drogue parachutes at a fixed time offset from launch, or they can attempt to detect apogee by accelerometer and fire immediately after. In order to prevent the Ravens from prematurely firing the parachutes, they were flown in timer mode.

Data Element	Units Stored	Logging Rate	Range		Max Error	
			Min	Max		
Altitude (barometric)	ft	20 Hz	0 ft	100000 ft	±2.5 mbar	
Acceleration (x,y,z)	g	400 Hz	$-100g$	$+100g$	$0.379g$ RMS	2% FS

TABLE IV. Blue Raven data types.

III. FLIGHTLINE: DATA ANALYSIS METHODOLOGY

The Hamster, BigRedBee, and Blue Raven data was analyzed using FlightLine, USCRPL's post-flight analysis software.[8] FlightLine is written with MATLAB R2024a, relying on the Aerospace and Robotics System toolboxes for coordinate transforms. This section will review the software's methodology. The stages of processing are explained below:

III.1. Reading Data

FlightLine begins by importing a JSON file specifying flight-specific parameters (conversion factors, sensor error bars, integration options, etc.). It then imports raw data in CSV format from each Hamster sensor, as well as from the BRB and Raven.

As data is imported, it is converted to the foot-pound-second system; refer to Tables III through IV for specific data types and unit conversions.

III.1.1. Data Interpolation

To integrate data from disparate modules (e.g. Hamster acceleration with Raven IMU) each sensor's data is interpolated to match the frequency of the source with the highest sampling rate.

The following methods were used for each interpolation:

- **Linear** interpolation is used for all non-discrete data sources for which the method is mathematically valid (i.e. for all non-discrete data other than the quaternions).
- **Previous** value interpolation is used for all discrete-valued data (number of satellites, Hamster state, etc.).

III.1.2. Normalizing Timescales

Next, a time offset is applied to each module such that liftoff occurs at T-0. Liftoff is determined on the Hamster by manually finding the timestamp at which the x axis registers an initial spike; a time offset is applied to make that time correspond to T-0. The Raven has its own internal liftoff detection, so no offset is necessary. Following these operations, the dataset is usable for further analysis.

III.2. Data Fusion

To combine the multitude of recorded datasets, an Unscented Kalman Filter (UKF) was utilized to dynamically filter and integrate data in order to achieve the most accurate state estimation at all time points. This process is critical for accurate apogee estimation during rocket flight.

III.2.1. Unscented Kalman Filter (UKF)

The Kalman filter algorithm was employed for its ability to provide accurate state estimations for both linear and non-linear systems, even in the presence of significant measurement noise and

[8] For inquiries regarding the FlightLine source code, please email us at analysis@uscrpl.com

process uncertainties. The UKF was specifically chosen due to the inherent uncertainties and variations in the onboard inertial measurement unit (IMU) sensors. These variations, coupled with the compounding errors introduced by double integration, can result in highly inconsistent apogee estimations when relying on individual sensor data. By employing a UKF, this issue was mitigated, as the algorithm leverages robust state estimation techniques to optimally fuse multiple data sources.

The UKF extends the standard Kalman filter by propagating a set of deterministically chosen sample points, known as sigma points, through the non-linear system model. Unlike the Extended Kalman Filter (EKF), which relies on first-order linearization and the computation of Jacobians, the UKF uses these sigma points to capture the non-linearity of the state and measurement models without requiring explicit derivatives. This makes the UKF particularly advantageous in systems with complex dynamics or when the models contain higher-order nonlinearities.

III.2.1.1. Overview of Kalman Filter Operation The Kalman filter operates through two primary steps:

- **Prediction Step:** The filter propagates the current state estimate forward in time using the system's dynamics. This step provides an a priori estimate of the state.

$$\hat{x}_{k|k-1} = \mathbf{f}(\hat{x}_{k-1}, u_{k-1})$$

$$P_{k|k-1} = F_k P_{k-1} F_k^\top + Q$$

  where $\hat{x}_{k|k-1}$ is the predicted state vector, $P_{k|k-1}$ is the predicted covariance matrix, F_k is the state transition Jacobian, and Q is the process noise covariance matrix.

- **Update Step:** The filter incorporates new measurements to correct the predicted state, minimizing the estimation error. The measurement model is used to compare the predicted measurements with actual observations.

$$K_k = P_{k|k-1} H_k^\top (H_k P_{k|k-1} H_k^\top + R)^{-1}$$

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k(z_k - h(\hat{x}_{k|k-1}))$$

$$P_k = (I - K_k H_k) P_{k|k-1}$$

  where K_k is the Kalman gain, H_k is the measurement Jacobian, R is the measurement noise covariance matrix, and z_k is the measurement vector.

The UKF generalizes these steps by replacing linearized models with a sigma point-based approach. Figure 5 illustrates the high-level operation of the UKF. In this implementation, the system state is corrected based on the fusion of inertial measurements and any available external references, ensuring robust performance across the entire flight envelope.

III.2.1.2. Filter Dynamics The UKF model is constructed around the following state vector:

$$x = \begin{bmatrix} \mathbf{r}_N & \mathbf{v}_N & \mathbf{a}_N & \sigma_{BN} & \omega_{BN_B} \end{bmatrix}$$

where:

- $\mathbf{r}_N$: Position in the ENU (East-North-Up) frame,
- $\mathbf{v}_N$: Velocity in the ENU frame,
- $\mathbf{a}_N$: Acceleration in the ENU frame,
- σ_{BN}: Attitude represented using Modified Rodrigues Parameters (MRP),
- ω_{BN_B}: Angular velocity in the body frame.

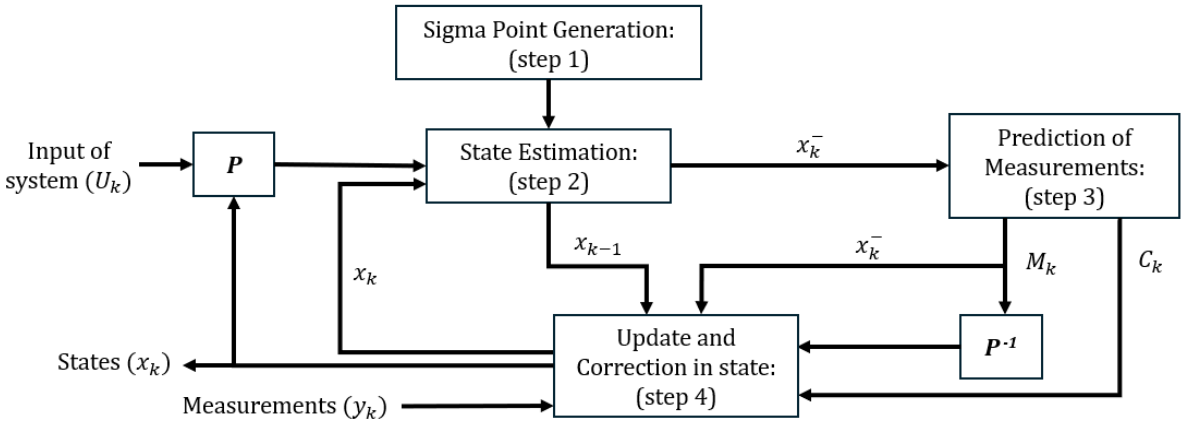

FIG. 5. Diagram of Generalized Unscented Kalman Filter.

III.2.1.3. State Transition Model The state transition function, $\mathbf{f}(x, dt)$, predicts the next state using the system dynamics:

$$\begin{aligned}
\mathbf{r}_N(t+dt) &= \mathbf{r}_N(t) + \mathbf{v}_N(t) \cdot dt, \\
\mathbf{v}_N(t+dt) &= \mathbf{v}_N(t) + \mathbf{a}_N(t) \cdot dt, \\
\sigma_{BN}(t+dt) &= \mathrm{MRP}\big(\sigma_{BN}(t), \omega_{BN_B}(t) \cdot dt\big), \\
\omega_{BN_B}(t+dt) &= \omega_{BN_B}(t).
\end{aligned}$$

The attitude update utilizes MRP kinematics, providing a compact and computationally efficient representation of orientation. Additionally, the angular velocity, ω_{BN_B}, is propagated under the assumption of quasi-static rotational dynamics for short time steps.

III.2.1.4. Process and Measurement Models The process noise covariance matrix, Q, and the measurement noise covariance matrix, R, represent the stochastic uncertainties in the system dynamics and sensor readings, respectively:

$$Q = \mathrm{diag}(Q_{\mathrm{pos}}, Q_{\mathrm{vel}}, Q_{\mathrm{acc}}, Q_{\mathrm{att}}, Q_{\mathrm{ang\_vel}})$$

$$R = \mathrm{diag}(\sigma^2_{\mathrm{ADIS}}, \sigma^2_{\mathrm{KX}}, \dots)$$

where σ terms denote the variances associated with individual sensor types. By tuning Q and R, the filter can prioritize certain measurements or dynamics based on their reliability and expected noise characteristics.

The measurement model, $z = h(x)$, relates the system state to observable quantities, such as accelerometer, gyroscope, and GPS data. These observations are fused with the predicted state to continuously refine the estimation.

III.3. Inertial Frame Transform

To make the accelerometer data usable for kinematic analysis, the acceleration vectors must be transformed into the transformed frame (and ultimately into the ECEF frame). This is done with the Aerospace Toolbox's `quatrotate` function, which given sensor x, y, and z acceleration components and a quaternion, converts from the sensor frame to the transformed frame. The IMU is the only source of complete orientation data, so the IMU quaternion is used whenever

acceleration data is converted into world frame. This is acceptable because the IMU's orientation relative to the all other sensors is constant.

Transforming the accelerometer data is straightforward, as there are x, y and z components from the sensor; thus, `quatrotate` with the IMU quaternions can be directly applied.

III.4. Integrating Acceleration

Before integrating, the data is converted from the transformed frame into the ECEF frame by subtracting g from the z axis. For increased accuracy, the value of g is recalculated at each altitude using the WGS84 gravity model.[9]

Integration is straightforward. The trapezoidal sum method of integrating the data is used, first to find velocity, and once more to find position along each axis. The initial velocity is assumed to be 0, and the initial position is found using the BRB GPS.

IV. FLIGHTON: SIMULATION METHODOLOGY

At this point, FlightLine possesses 4 separate sources of sensor-based absolute position data (each with different merits and drawbacks, as discussed in Section V.2): BigRedBee GPS, Hamster GPS, Raven barometric altimeter, and the processed and integrated combined accelerometer+IMU data described above. For the remainder of this analysis, during which data sources are selected from the above for final use in reconstructing an apogee and flight path, various simulation techniques are used to validate and augment the sensor data.

Data collected by the onboard sensors was validated against results from FlightOn, USCRPL's internally developed proprietary flight and solid rocket motor simulation toolchain. The flight simulation code was qualified following the successful flight of USCRPL's Traveler IV in 2019 and Fathom II in 2016, and validated against the benchmarks in NASA/TM-2015-218675. FlightOn's solid rocket motor simulation code is a time-unsteady lumped parameter model that accounts for erosive burning effects, and has been anchored to multiple static firings including a successful full-scale static firing of the motor design used on Aftershock II. Six degrees of freedom (6-DOF) and 3 degrees of freedom (3-DOF) flight simulations were also run for the portions of the flight before and after parachute deployment, respectively.

IV.1. 6DOF

6-DOF simulations were run in FlightOn as a companion to the sensor data for the portion of the flight up to parachute deployment. FlightOn solves the kinematic and Euler equations in an ECEF coordinate system reference frame, using quaternions to represent vehicle orientation and a Dormand-Prince 5(4) Runge-Kutta numerical integration scheme to propagate the state vector over time. Thrust was determined using FlightOn's solid rocket motor simulation code. When performing 6-DOF simulations, FlightOn uses the WGS 84 gravity model (for consistency with the accelerometer integration technique discussed in SectionIII.4) and MSISE-00 atmosphere model, and wind data for the time and location of the launch from NOAA's READY system using the HRRR model.[10] The surface wind speed and direction are set to values recorded at launch using an anemometer.

[9] Specifically, the `'Exact'` option of the MATLAB Aerospace Toolbox's `gravitywgs84` implementation of the model.
[10] For more information on these models, see Appendix A.

IV.2. 3DOF

3-DOF simulations were used to verify that the ascent accelerometer data, which did not accurately capture the subtle world-space deceleration of parachute descent, could feasibly agree with GPS and barometer data during the descent phase. The same gravity, atmosphere, and wind models were used as in the 6DOF simulation, but the kinematic equations were solved in NED (North-East-Down) coordinates, with the origin set at the launch site. The complex dynamics of the vehicle body, nose cone, and parachute were simplified into a single non-rotating body. Drag was computed using a C_d derived from flight data for the vertical axis. Specifically, the Raven's barometric data was used to the compute the C_d during the portion of the descent recorded by the Raven. This empirical estimate was then curve fitted for the simulation; in the initial portion of the descent, in which no barometric data was available, a constant value was used. The value was chosen to match the C_d when the vehicle regained GPS lock and valid barometric readings.

V. REVIEW OF FLIGHT DATA

Now that the data sources and general methods have been established, each dataset will be reviewed and its usage justified based on quality and relevance.

V.1. Flight Timeline

The following timeline describes the most significant events of the flight:

T-120 s	**On Tower:** The vehicle is stationary in the tower as the countdown proceeds. Hamster enters ARM state.
T-8 s	**Terminal Count:** Hamster enters TERM state.
T+0 s	**Liftoff:** First motion is recorded by the accelerometers.
	Ignition: The motor burn begins.
T+2 s	**Supersonic:** The vehicle exceeds the speed of sound (Mach 1.0).
T+10 s	**Max Q:** The vehicle encounters maximum dynamic pressure.
T+13 s	**Hypersonic:** The vehicle exceeds 5X the speed of sound (Mach 5.0).
T+19 s	**Burnout:** The motor burn ends.
	Coast: As the vehicle ascends into thinner atmospheric layers, aerodynamic forces decreases rapidly. Transitions to a ballistic coast phase, gradually decelerating under the influence of gravity.
T+85 s	**Atmosphere Exit:** The vehicle crosses the Kármán Line, denoting its exit from Earth's atmosphere and entry into Space.
T+177 s	**Apogee:** The vehicle hits zero vertical velocity, achieving its maximum altitude of 470 400 ft. Subsequently beginning its descent.
T+181 s	**Parachute Deploy:** The parachute is deployed by Hamster as its integrated velocity estimation hits zero within the allowable firing window.
T+270 s	**Atmosphere Entry:** The vehicle once again passes through the Kármán Line, denoting its reentry into the Earth's atmosphere.
T+309 s	**Max Reeetry Heating:** The vehicle is hitting the lower layers of the atmosphere at supersonic velocity, heating the vehicle via intense air friction. After this point the heating subsides into a more benign descent.
T+311 s	**GPS lock regained:** The BRB regains GPS lock, giving precise data for the remainder of the flight. A few seconds later, the Hamster GPS also regains lock.
T+754 s	**Touchdown:** The vehicle hits the ground.

Many of these events and the methodology for describing them will be explained in greater detail throughout the remainder of this section.

V.2. Review of Data Sources

V.2.1. Atmospheric and Altitude Data

V.2.1.1. Altimeters There were three functional sources of altimeter data aboard the vehicle - ZED GPS, BRB GPS, and Raven Barometer as seen in Figure 6.

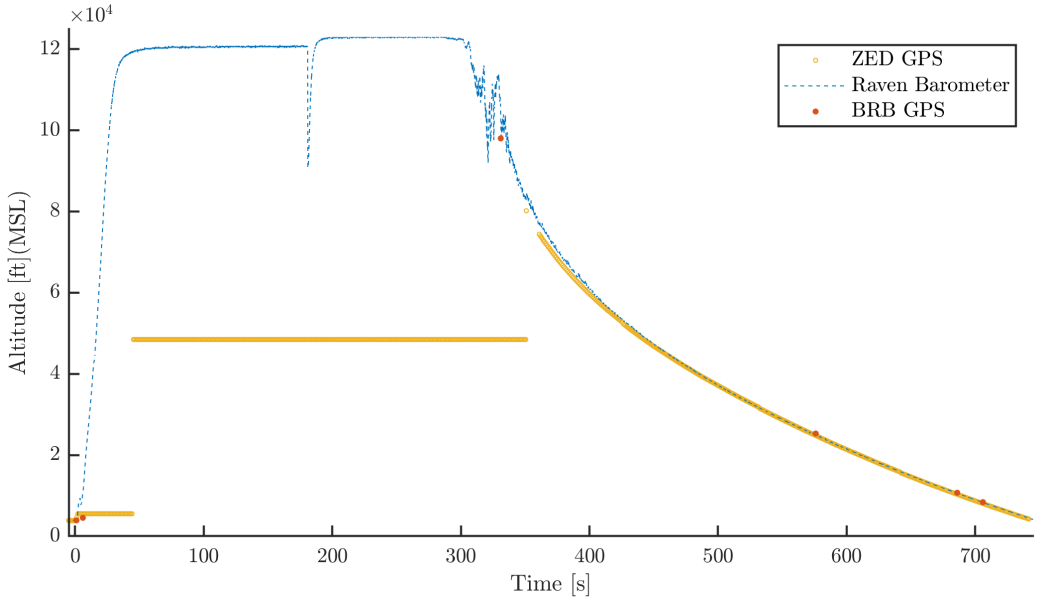

FIG. 6. Altitude readings from the altimeters and GPS units.

During the flight, the Raven's MS5607 barometer, HAMSTER's ZED-F9P GPS, and BRB's GPS all show good agreement. On ascent the barometer reports abnormally high values, however it's assumed these anomalous readings are due to the vehicle passing through the transonic regime of flight. During the descent, there is a very good agreement between all three sensors, as generally the deviation never exceeds greater than 500 ft.

V.2.1.2. Atmospheric Model In addition to the direct sensor measurements, we utilized the NRLMSISE00 atmospheric model to estimate atmospheric properties when sensor data was either unavailable or unreliable. The NRLMSISE00 model, developed by the U.S. Naval Research Laboratory, is a well-established empirical model used for determining the atmospheric density, temperature, and pressure at various altitudes. It incorporates data from both ground-based observations and satellite measurements, providing an accurate representation of atmospheric conditions from the Earth's surface up to altitudes exceeding 100 km. This model was particularly useful during the flight's reentry phase, where sensor readings were anomalous, allowing us to fill in the gaps in sensor data with reliable atmospheric predictions.

V.2.2. Orientation Data

V.2.2.1. ADIS16467 The IMU recorded body rate data in sensor frame, as seen in Figure 7.

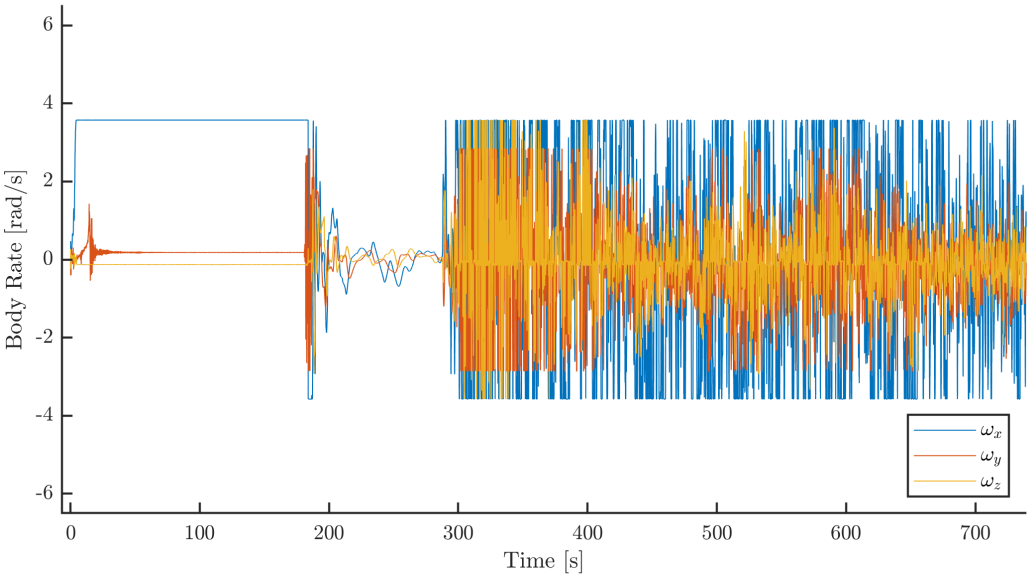

FIG. 7. Body rate data recorded by the ADIS16467 IMU in the sensor frame.

Due to a configuration error, the ADIS operational range was set too low and as a result the vehicle's roll rate (ω_x) exceeded the measurable range. This resulted in the sensor's ω_x readings becoming 'railed', or flat-lined.

V.2.2.2. BMI323 The IMU recorded body rate data in sensor frame, as seen in Figure 8.

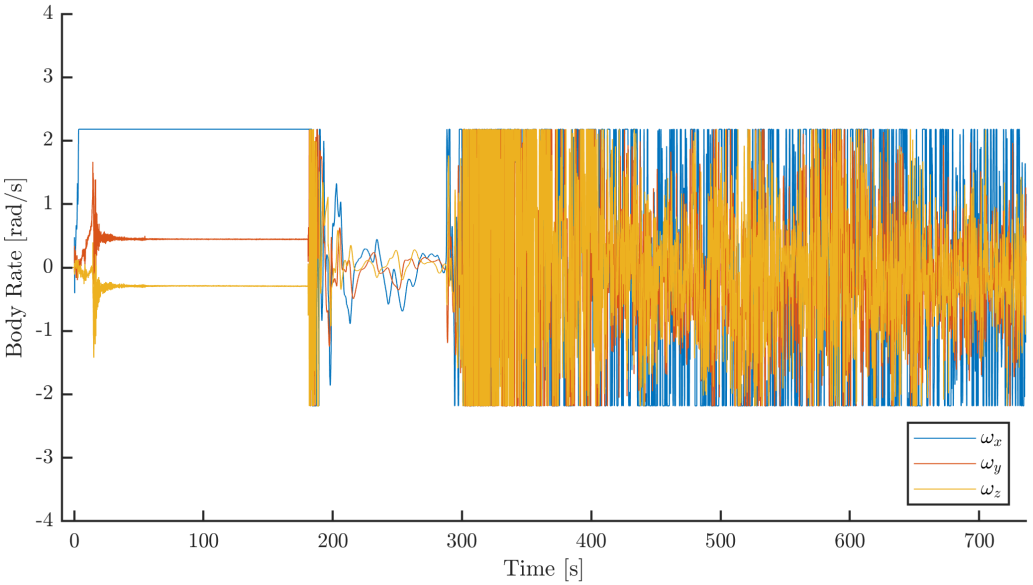

FIG. 8. Body rate data recorded by the BMI323 IMU in the sensor frame.

The BMI operational range was also set too low and as a result the vehicle's roll rate (ω_x) exceeded the measurable range.

V.2.2.3. RAVEN The IMU recorded body rate data in the sensor frame.

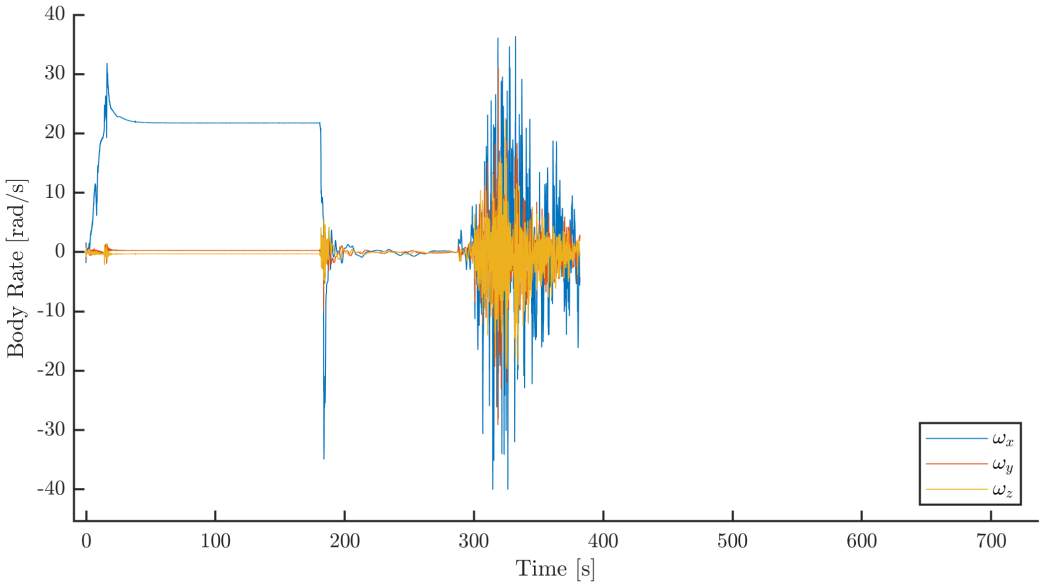

FIG. 9. Body rate data recorded by the RAVEN IMU in the sensor frame.

Raven IMU was the only IMU to record data through the entire ascent and partially through descent.

V.2.3. Accelerometer Data

V.2.3.1. ADIS16467-1 The ADIS Triaxial accelerometer recorded acceleration data in its sensor frame, as seen in Figure 10.

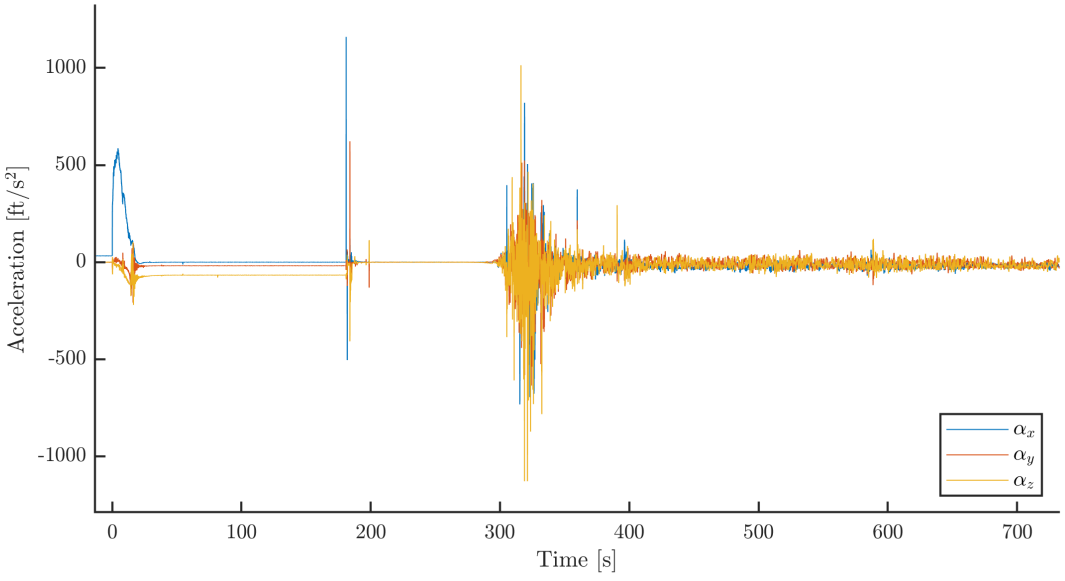

FIG. 10. Raw ADIS triaxial acceleration data in sensor frame.

V.2.3.2. KX134 The KX triaxial accelerometer recorded acceleration data in its sensor frame, as seen in Figure 11.

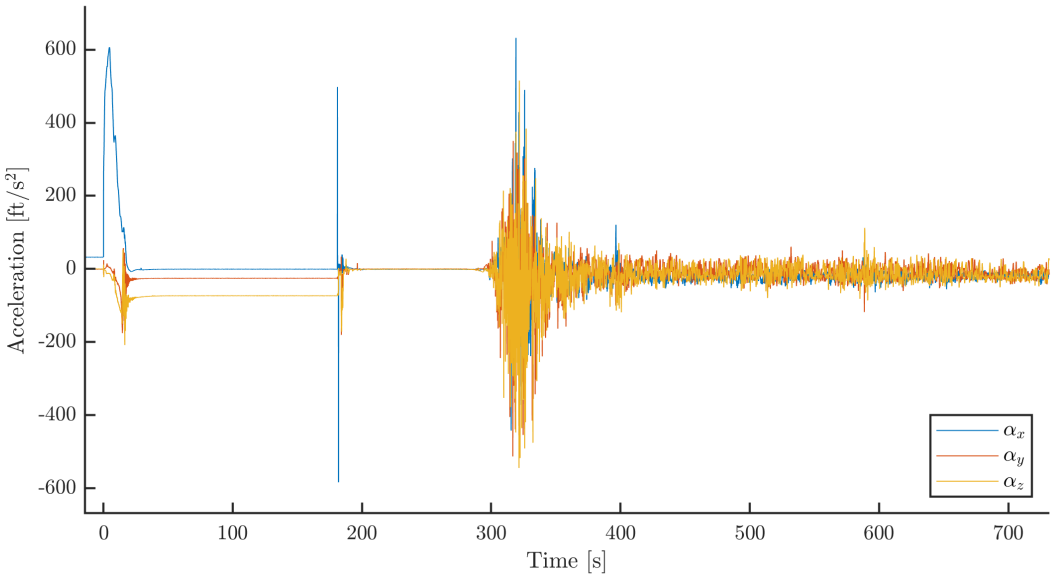

FIG. 11. Raw KX triaxial acceleration data in sensor frame.

V.2.3.3. Blue Raven Accelerometer The Raven triaxial accelerometer recorded acceleration data in its sensor frame, as seen in Figure, as seen in Figure 12.

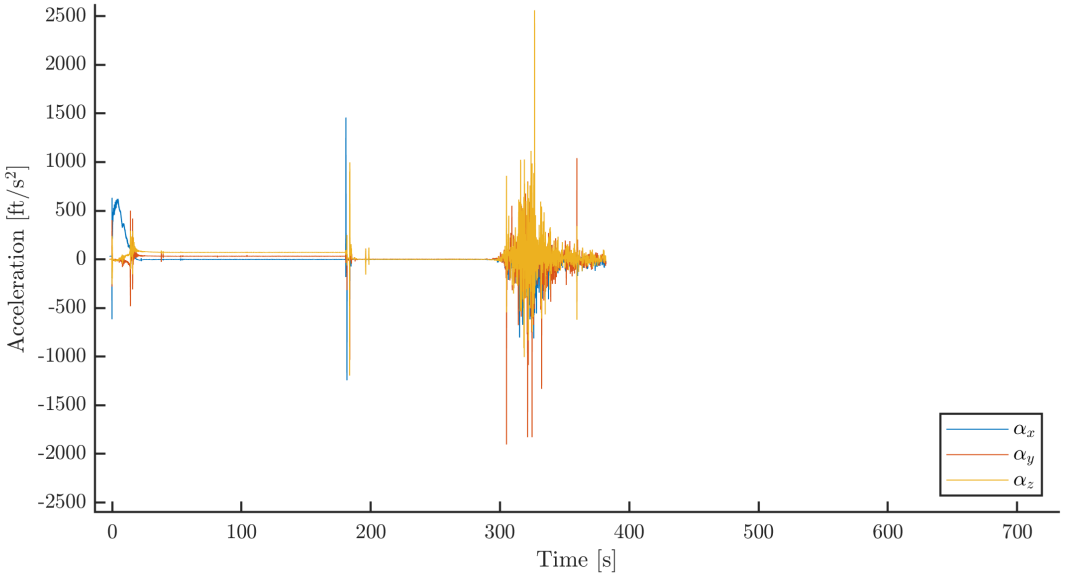

FIG. 12. Raw Raven triaxial acceleration data in the sensor frame.

The datasets for the ADIS, KX, and Raven accelerometers were all observed to have performed nominally as they recorded continuously throughout the entirety flight (with the exception of Raven which ran out of storage at approx. 375s), and none experienced any major errors.

V.3. Data Overview

Based on the quality of each redundant data source, a set of best data was identified to be used in the final apogee calculations. The choices were as follows:

- **Orientation (world):** Raven IMU
- **Acceleration x, z, and y (sensor frame):** ADIS, KX, and Raven
- **Altitude (world):** Raven Barometer, BRB GPS, and ZED GPS.

V.3.1. Choice of Data

To integrate the acceleration in the world frame, the gyroscopic data from the Raven IMU was used to transform all three accelerometers into world frame. The quaternions necessary for this transformation sourced directly from an integration of the IMU data. This was a consequence of the HAMSTER onboard IMU's railing during the high roll-rate portion of the flight. The acceleration data from all three accelerometers showed good agreement and thus were all used. Due to hardware malfunctions, the *Lightspeed Rangefinder* and *Guidestar GPS* were not operational for this flight and their data was discarded. The Raven Barometer, BRB GPS, and ZED GPS were all selected as the altimeter sources for descent.

The burn phase acceleration data was compared from all three accelerometers in order to evaluate the relative agreement between independent accelerometers. In Figure 13, the Raven burn recorded with the highest sampling rate and thus contains a relatively large amount of noise and variation. The KX is more stable, but has near-complete agreement with the Raven at all stages of the burn. Whilst the ADIS agrees with the other two sensors at low-G accels, there is a difference which develops at the highest-G portion of the burn.

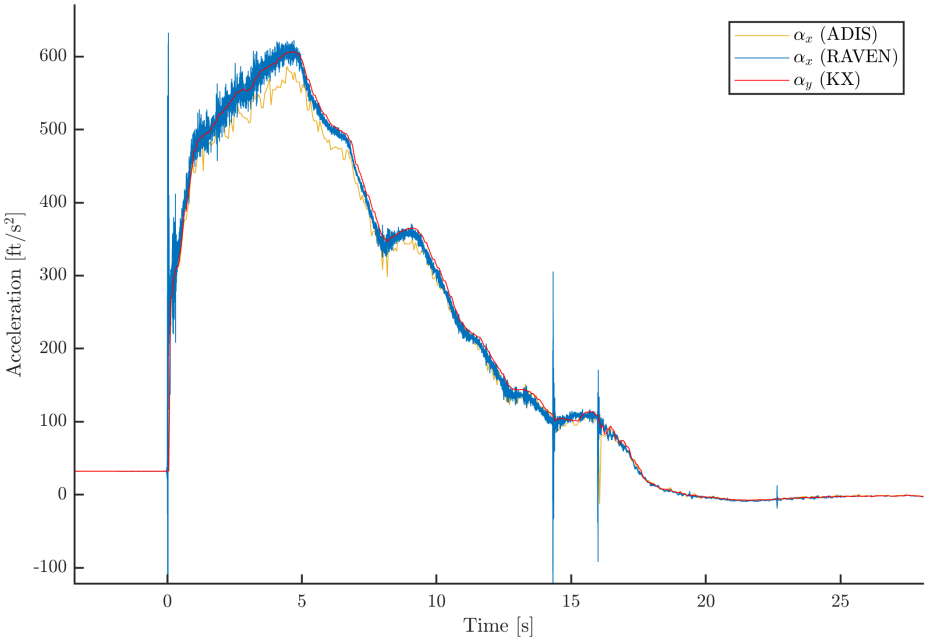

FIG. 13. Comparison of burn x-axis accelerations between the accelerometers: KX, RAVEN, and ADIS.

VI. RESULTS

Two methods were used to reach a final apogee result: the direct result based on integration of the accelerometer data all the way to apogee, and a flight reconstruction using reentry data. Final uncertainty is defined as a 99.7% confidence interval on the set of apogees calculated from the Monte Carlo trials with the Kalman filter defining the mean line. Apogee estimates from flight reconstruction fall within these uncertainties.

VI.1. Flight Reconstruction Using Reentry Data

The flight was reconstructed using FLIGHTON 3DOF simulations in order to match the reentry data — prioritizing the barometric because of the larger overall dataset and superior agreement with launch and landing site elevations. Due to tangling and higher than expected choking on the parachute, the original C_d was no longer representative. A new C_d for the 3DOF parachute model was thus back calculated from the available barometer descent data and approximated with a Gaussian curve-fit model, with C_d being a function of Mach. Note that barometer data is only reliable at altitudes below 100 000 ft. The C_d corresponding to Mach 1.1 was used for Mach 1.1+ portions of the flight.

Two nominal trajectories were simulated using the derived C_d values. The apogee was tuned to match the lower and upper bound of the descent profile, accounting for an error of ±2.5 mbar. These simulations produced two simulated 3DOF trajectories, representing the lower and upper apogee estimates. Based on this flight reconstruction method, the apogee of AFTERSHOCK II was determined to be in the interval of [453 000 ft, 473 000 ft].

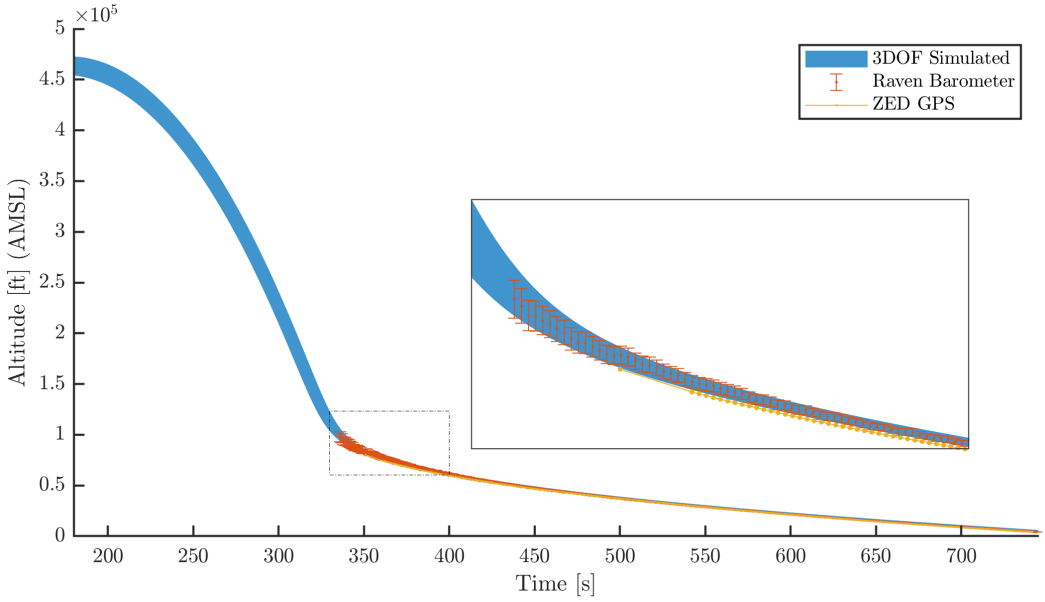

FIG. 14. Flight trajectory reconstruction using 3DOF simulations matched to barometric and GPS data on reentry.

VI.2. Integrated Accelerometer Data

A 1024 trial noise Monte Carlo was run in FLIGHTLINE on the sensor data, and the resulting acceleration data distribution was recorded. Using this data, the Unscented Kalman filter algorithm was executed on the suite of sensors in order to determine a combined acceleration dataset, including time-varying uncertainty according to the standard deviation spread of the noise Monte Carlos. Once the combined dataset and uncertainty were determined, the acceleration was run through FLIGHTLINE to be double-integrated in the world frame and the overall flight trajectory calculated.

Based on this method of integration, the apogee of AFTERSHOCK II had a mean value of 470 400 ft, and was in the interval [443 100 ft, 497 700 ft] with 3 sigma confidence.

σ Level	Apogee Uncertainty	Velocity Uncertainty	Confidence
$\pm 1\sigma$	$\pm$ 9100 ft	$\pm$ 50 ft/s	68.3%
$\pm 2\sigma$	$\pm$ 18 200 ft	$\pm$ 101 ft/s	95.4%
$\pm 3\sigma$	$\pm$ 27 300 ft	$\pm$ 151 ft/s	99.7%

TABLE VI. Apogee and velocity confidence levels at $\pm 1\sigma$, $\pm 2\sigma$, and $\pm 3\sigma$.

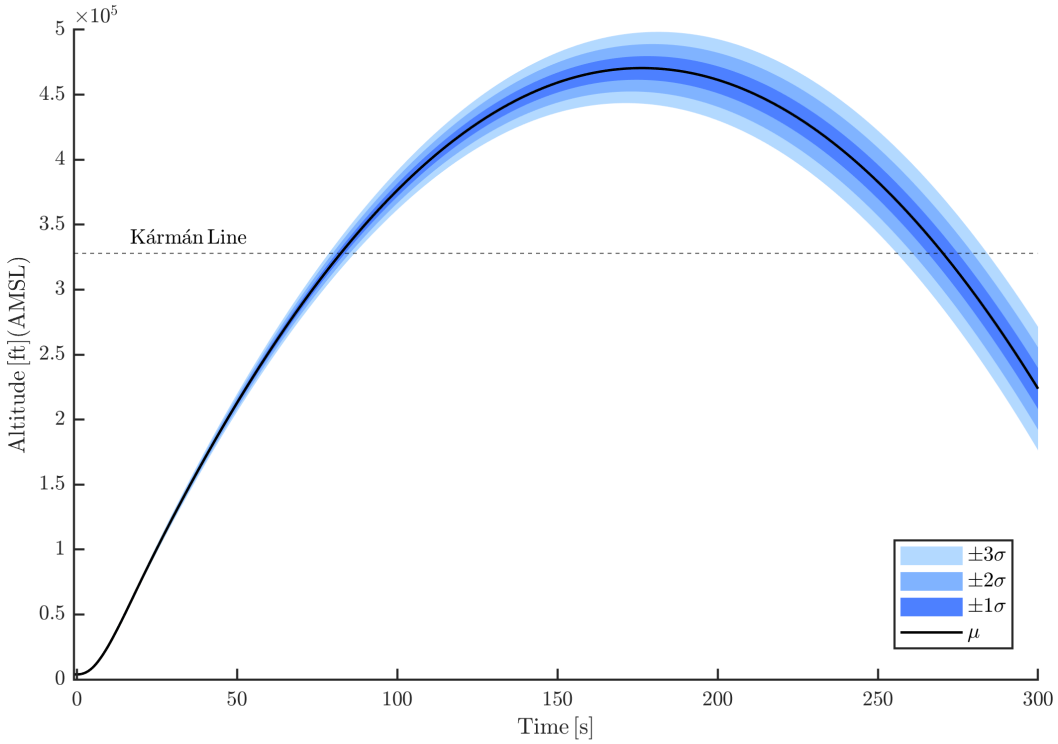

FIG. 15. Altitude from integrated accelerometer data with mean trajectory (μ) and uncertainty bounds ($\pm 1\sigma$, $\pm 2\sigma$, $\pm 3\sigma$). Maximum Altitude: 470 400 ft $\pm$ 27 300 ft.

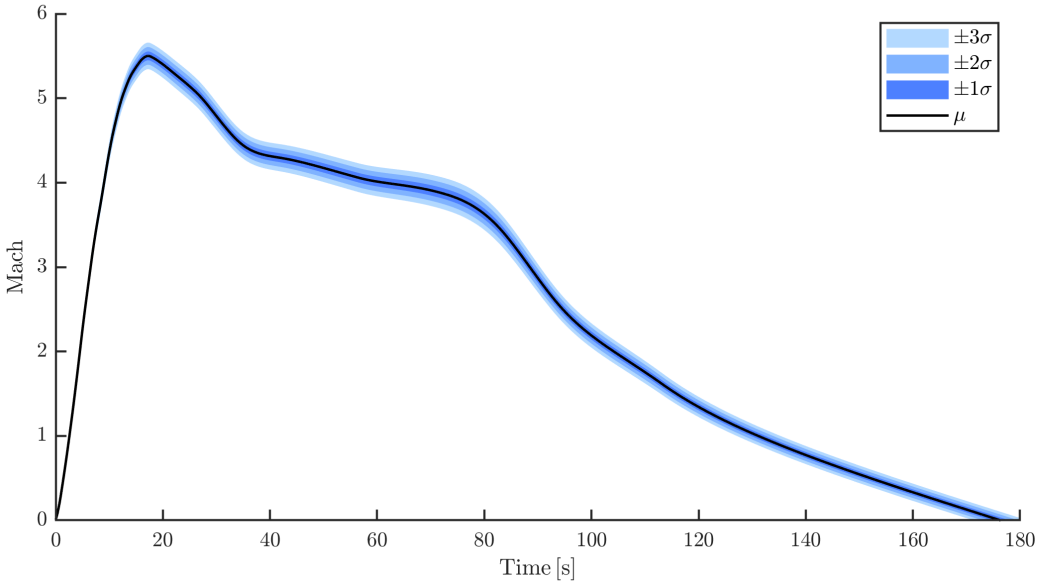

FIG. 16. Mach number from integrated accelerometer data with mean profile (μ) and uncertainty bounds ($\pm1\sigma$, $\pm2\sigma$, $\pm3\sigma$). Maximum Mach: 5.50 ± 0.15.

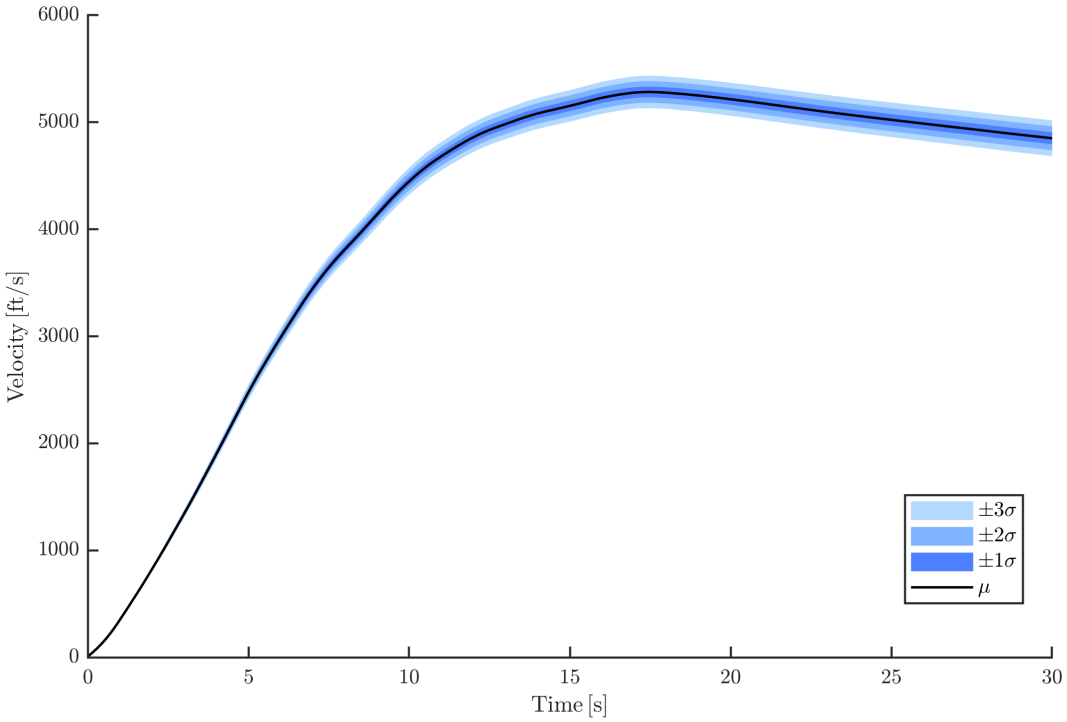

FIG. 17. Velocity from integrated accelerometer data with mean profile (μ) and uncertainty bounds ($\pm1\sigma$, $\pm2\sigma$, $\pm3\sigma$). Maximum Velocity: 5283 ft/s ± 151 ft/s.

VII. CONCLUSION

After careful analysis of data recorded during the Aftershock II launch, vehicle apogee was determined to be 470 400 ft ± 27 300 ft AMSL breaking the amateur altitude record of 380 000 ft previously set by CSXT's GoFast rocket. Traveling significantly higher than the Kármán line at 328 064 ft AMSL, Aftershock II is the second rocket entirely designed by students to reach space, surpassing the altitude reached by the Traveler IV rocket. Our analysis represents a conservative approach to determining the vehicle's apogee, and as such has the full support of the USCRPL team.

Appendix A: Environmental Models

1. Atmosphere: NRL MSISE-00

To model atmospheric density, pressure, temperature, and sonic velocity as a function of altitude for 6-DOF simulations, FLIGHTON uses a C implementation of the Naval Research Lab's MSISE-00 model[11]. For AFTERSHOCK II flight simulations, the parameters listed in Table VII were used as inputs to MSISE-00, along with launch time, date, and vehicle geodetic coordinates.

2. Gravity and Ellipsoid Model: WGS 84

For FLIGHTON 6-DOF simulations, an approximation of WGS 84 where only the J_2 term is considered is utilized to calculate gravitational acceleration[12]. This approximation is accurate to within $\sim 3 \times 10^{-4}\,\mathrm{m/s^2}$ RMS, which is sufficient for USCRPL purposes. For the Earth reference ellipsoid, WGS 84 is used. The J_2 term and WGS 84 constants used in FLIGHTON are listed in Table VII.

Model	Parameter	Value
MSISE-00	F10.7 cm	153.0
	AP	5
WGS 84 Approx.	Gravitational parameter	$3\,986\,004.418 \times 10^8$ m/s^2
	Semi-major axis	6 378 137 m
	Semi-minor axis	6 356 752.3142 m
	First eccentricity squared	$6.694\,379\,990\,14 \times 10^{-3}$
	Second eccentricity squared	$6.739\,496\,742\,28 \times 10^{-3}$
	Inverse flattening	298.257 223 563
	J_2	$1.082\,626\,684 \times 10^{-3}$

TABLE VII. Atmosphere and Gravity Constants

[11] `https://www.brodo.de/space/nrlmsise/`

[12] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls, Design, and Autonomous Systems* (Wiley, 2016), 3rd ed.