Traveler IV Apogee Analysis

Adam Aitoumeziane, Peter Eusebio, Conor Hayes, Vivek Ramachandran, Jamie Smith, Jayasurya Sridharan, Luke St. Regis, Mark Stephenson, Neil Tewksbury, Madeleine Tran, and Haonan Yang*

USC Rocket Propulsion Laboratory
(Dated: May 2019)

Following the successful launch and recovery of Traveler IV, the fourth of the USC Rocket Propulsion Laboratory's Traveler-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, analytic methods and Monte Carlo flight simulations were used to determine and validate an apogee of $339\,800\,\text{ft} \pm 16{,}500$ ft, passing the Kármán line with a confidence of 90%. This result establishes Traveler IV as the first rocket designed and built entirely by students to reach space.

CONTENTS

* analysis@uscrpl.com

I.   INTRODUCTION

On April 21, 2019, the University of Southern California Rocket Propulsion Laboratory (USCRPL), launched TRAVELER IV from Spaceport America in New Mexico. The vehicle was designed with the goal of crossing the Kármán Line, the boundary between Earth's atmosphere and space defined as 328,084 ft (100km) AMSL [1]. This flight marked the first successful operation and recovery of USCRPL's on-board avionics system on a high performance vehicle.[2] This paper details the post-flight analysis of the data collected by TRAVELER IV's avionics, with the goal of reconstructing the vehicle's flight path and determining its apogee.[3] [4]

I.1.   Terminology

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

- **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 z points through the vehicle nose tip, and x and y are defined to point in the same direction as the x and y axes of the avionics unit's ADXL accelerometer.

---

[1] Above Mean Sea Level

[2] The avionics unit on Fathom II–USCRPL's previous record-breaking rocket, suffered a brownout immediately following drogue deployment due to an electrical design issue. This led to a software fault that wiped all recorded data and prevented continued recording.

[3] Access to TRAVELER IV'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.

[4] 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.

- **Sensor frame:** The reference frame centered on a particular sensor. A distinction must be made between the sensor frame and the body frame, since the accelerometers are mounted at different positions relative to the vehicle's center of mass, and therefore measure different centripetal forces as the vehicle rotates. While this makes direct data comparison between sensors difficult, the data has been utilized in such a way as to minimize this effect, as justified in Section III.3.

- **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 $a_z = +g$.

II.   DATA COLLECTION SYSTEMS

The avionics system consists of multiple independent 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, and hosts a variety of sensors as detailed in Section II.1.

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 Raven 4 Altimeter*[6] (abbrev. Raven) is a self-contained data acquisition system often used in amateur rocketry. The Raven features a two-axis accelerometer and a barometric altimeter. It also offers high-current outputs to fire the recovery system after apogee.[7]

II.1.   Hamster Operation

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

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

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 steel ballast at the nose cone's tip. The ADXL375 sensitivity factors for each axis have been applied in post rather than in flight. The conversion from ADXL375 LSB to g is different for each accelerometer axis. The conversion factors were measured using a two-point linear fit at 1g and -1g, and are summarized in Table III.[8]

---

[5] http://www.bigredbee.com/BeeLineGPS.htm

[6] https://www.featherweightaltimeters.com/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 TRAVELER IV due to HAMSTER's nominal firing of the chutes.

[8] For future flights, the ADXL375 will be calibrated at multiple higher g values using a centrifuge for better accuracy.

Model	Function	Logging Rate (Hz)		
		Pre-Deployment	Deployment	Post-Deployment [a]
MS5607	Altimeter	4	10	0.5
ADXL375	Accelerometer	20	5	0.5
BNO080	IMU	10	2.5	0.5
LM75B	Temperature Sensor	1	1	0.5
FGPMMOPA6H	GPS	1	1	0.5
INA226	Power Sensor	0.2	10	0.5

[a] The slow post-deployment logging rate rendered the HAMSTER's collected motion data mostly useless during re-entry, as will be addressed later in this analysis.

TABLE I. HAMSTER logging rates.

Sensor	Data Element	Units Stored	Min	Max	Error	
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	
ADXL375	Acceleration (Sensor frame)	LSB	$-200g$	$+200g$	$0.179g$[a]	2% FS[b]
BNO080	Acceleration (World frame)	m/s²	$-2g$	$+2g$	$±0.02g$	
	Orientation (World frame)	Quaternion			3.5°	
	Magnetic Field Vector [c]	µT	$-1300\,µT$	$1300\,µT$	8%	
	Calibration [d]	N/A				
LM75B	Temperature	°C	-55 °C	+125 °C	±3 °C	
FGPMMOPA6H [e]	Latitude & Longitude	°				
	Altitude	km				
	Number of Linked Satellites	Satellites				
	Fix Quality [f]	N/A				
INA226	Battery Voltage	V	0V	36V	0.036V	
	Battery Current	A				
	Regulated 5V Current	A				

[a] Mean RMS noise. Unfortunately, despite only polling the ADXL at up to 20Hz, the sensor itself was configured for 800Hz mode, giving a much larger RMS noise than would normally apply at this polling rate.
[b] Sensor non-linearity error, defined as a percentage of full-scale measurement.
[c] Corrected magnetic field, excluding magnetic errors from metal in the vehicle.
[d] Integer from 0-3 denoting calibration level, and thus accuracy, of rotation data.
[e] This module is also known as the Adafruit Ultimate GPS
[f] 0 → no fix; 1 → GPS fix; 2 → Differential GPS fix.

TABLE II. Types of data logged by HAMSTER.

Axis	Conversion
x	0.0501 g/LSB
y	0.0510 g/LSB
z	0.0484 g/LSB

TABLE III. ADXL375 LSB conversion factors.

Additionally, as a sensor designed for very high accelerations, the readout from the ADXL375 sensor is unreliable when its readout is near $0\,g$, making it unusable during freefall.

HAMSTER's sensor data is recorded in an on-board 16MB NOR flash IC using RPL's FlashLog encoded binary format. The FlashLog system includes a checksum 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 checksum (only 8 of the 24,306 recorded packets) were not used in our analysis. The FlashLog also records system state and other information useful for debugging, but which is not relevant to this analysis.

II.2. BigRedBee BeeLine GPS Operation

The BRB logs only GPS data, consisting of latitude, longitude, altitude, and time. It logs data at 0.2Hz.

II.3. Featherweight Raven 4 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. However, standard Ravens have a maximum timer setting of 51.2 seconds, which is well below TRAVELER IV's predicted time at apogee. For this flight, *Featherweight* generously provided the team with Ravens flashed with custom high-altitude firmware, which increments the firing timer twenty times slower (but does not affect the sensor polling rate).

Data Element	Units Stored	Logging Rate	Range		Max Error	
			Min	Max		
Altitude (barometric)	ft	20 Hz	0 ft	100000 ft	±300 ft	
Axial Acceleration (z)	g	400 Hz	$-100g$	$+100g$	$0.379g$ RMS	2% FS
Lateral Acceleration (x)	g	200 Hz	$-100g$	$+100g$	$0.268g$ RMS	2% FS

TABLE IV. Raven 4 data types.

The Featherweight Raven uses the same MS5607 altimeter that the HAMSTER unit does. To sense acceleration, it uses the H3LIS331DL MEMS accelerometer, which is factory-calibrated.

A second Raven was also launched on Traveler IV, but this Raven did not have appear to have any data on it after the unit was recovered. It is possible that it failed to detect the vehicle's launch, or that the vehicle's high acceleration damaged one of its components.

III.   FLIGHTLINE: DATA ANALYSIS METHODOLOGY

The Hamster, BigRedBee, and Raven 4 data was analyzed using FlightLine, USCRPL's post-flight analysis software.[9] FlightLine is written with MATLAB R2019a, 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 II through IV for specific data types and unit conversions. All data is kept in the frame in which it is recorded. At this point, packet quality checks are done, and packets without a valid checksum are discarded.

III.1.1.   Data Interpolation

To reach a balance between exactness of data and ease of use, each sensor's data is interpolated to the frequency of the most frequent source within its module (the Hamster, BRB, or Raven).

In the case of the Hamster, this is achieved using *tails.csv*, a file listing the time since power-on (in μs, but converted to s for easier use) and packet type (ADXL, Barometer, etc.) for each packet recorded by the unit. Each Hamster data source is interpolated to this timescale.

The Raven interpolates its readings to the timescale of its axial acceleration sensor, which is both the most frequently recorded source and the only data from the Raven used in the final results.

The BRB does not need to be interpolated, as it has a single data source.

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).

- **Slerp** (Spherical Linear intERPolation) is used for the quaternions. The implementation is an adaptation of the `quatinterp` function from the Aerospace Toolbox such that it functions like `interp1`.

- **Previous** value interpolation is used for all discrete-valued data (number of satellites, Hamster state, etc.).

The same strategy is used at later points when data from two different modules are interpolated (e.g. when using IMU quaternions with Raven acceleration).

---

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

<div align="center">

III.1.2.   Normalizing Timescales

</div>

Next, a time offset is applied to each module such that liftoff occurs at T0. Liftoff is determined on the HAMSTER by manually finding the power counter time at which the ADXL z axis registers an initial spike; a time offset is applied to make that time correspond to T0. The BRB offset is found by aligning its altitude to that of the GPS sensor on the HAMSTER during the vehicle's descent. The Raven begins collecting data when liftoff is detected, so no offset is necessary. Following these operations, the dataset is usable for further analysis.

<div align="center">

III.2.   Noise Correction

</div>

Each accelerometer has a specified root mean square (RMS) value for noise, as well as a specified non-linearity error bounded by a percentage of the "full scale" of the sensor (e.g. 2% FS for the $\pm 100g$ Raven accelerometers, which means their true non-linear sensor transfer function's output is bounded to within $\pm 4g$ of the given linear approximation). Although the actual noise and non-linearity present may be quite different for each axis, the RMS value and the non-linearity bound are the same across all axes. It follows that:

$$a_{\text{measured}} = (a_{\text{actual}} + X_{\text{RMS}}) * (1 + \text{NL}_{\text{max}} * X_{\text{NL}}) \tag{1}$$

where:

- a_{measured} is the acceleration reported by the sensor.

- a_{actual} is the true acceleration of the sensor in its sensor frame, with all measurement and sensing error removed.

- X_{RMS} is a random variable selected from a normal distribution with a standard deviation of the given accelerometer RMS noise value. A separate X_{RMS} is selected for each a_{measured} reading.

- NL_{max} is the non-linearity bound of the accelerometer in decimal form, typically given by the manufacturer in percent of full-scale (% FS).

- X_{NL} is a random variable selected from a uniform distribution ranged from -1 to 1. A single X_{NL} is selected to be applied across the entire flight to every a_{measured} reading in the current trial.[10]

Solving for a_{actual} gives:

$$a_{\text{actual}} = \frac{a_{\text{measured}}}{1 + \text{NL}_{\text{max}} * X_{\text{NL}}} + X_{\text{RMS}} \tag{2}$$

To model these random variables, a Monte Carlo simulation is run as follows: Random noise value X_{NL} is selected from its distribution, to be applied globally across the run. Then an individual X_{RMS} value is selected from its distribution for every raw data point a_{measured}. These values are then used as in Eq. 2 to arrive at a hypothetical a_{actual} for each data point.

---

[10] In the case of the Raven accelerometer z-axis, for which this method is particularly important and whose error bars dominate those of the final apogee result, this method essentially approximates the true non-linear transfer function of the sensor as a 10-section piece-wise function, in which only 2 pieces are actually used (-20g to 0g, and 0g to 20g). These two linear pieces are anchored such that they pass through the 0g and 1g calibration points, and their slopes are varied within the specified non-linearity bounds during the Monte Carlo analysis.

Next, the remaining steps of the analysis are completed using these a_{actual} values. This full analysis process is then repeated many times for each axis of each accelerometer, using new X_{RMS} and X_{NL} values. Finally, the results of each repeated trial are compared, and are used to estimate an error distribution for the final products of the analysis.[11] The true final result, with measurement error removed, should fall within the distribution created by this simulation method.

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 inertial 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 ADXL and Raven is constant.

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

The Raven data poses a problem: as the sensor has only x (radial) and z (axial) accelerometer axes, a transform cannot be easily performed. Furthermore, the Raven is at a different radius from the center of mass than the ADXL and thus senses different centripetal forces, each in its own sensor frame. These forces are not problematic when considering a single accelerometer, as integrating the world frame path of a single sensor rotating about the vehicle's z axis will produce the vehicle's actual path with a small "corkscrew" effect added in. However, the presence of these forces *does* mean that the ADXL cannot be used to provide raw y data for the Raven, which only has x and z axes, because the centripetal accelerations on the two sensors about the z axis are too different.

To overcome this problem, the assumption is made that centripetal acceleration a_c due to rotation about the x and y axes in body frame are the same for all accelerometers for the duration of motor burn and free fall. This assumption is justifiable due to the following:

1. The only rotation about the body frame x and y axes during the ascent is a slight, low-frequency wobble.

2. All accelerometers in the vehicle are tightly concentrated in the nose cone, far from the vehicle's center of gravity about which they rotate. The radial distance between any two sensors is 2 orders of magnitude less than than their distance to the x and y axes.

The combination of these two lemmas means that a_c measured by the sensors about the x and y axes is both very small (1) and nearly the same (2) due to $a_c = \omega \times r$,[12] [13] allowing the assumption of equal a_c for all sensors about x and y.

----

[11] This process could be applied to other sources with a known noise distribution, but would ultimately be irrelevant to the apogee analysis as integrated-accelerometer position error is several orders of magnitude larger than the error produced by other sensors that can describe position.

[12] The assumption of equal centripetal accelerations for each sensor *cannot* be made for rotation about the z axis, as the sensors' radial distances to the z axis are not similar, and the rotation about z is extreme ($\sim$8Hz).

[13] Centripetal acceleration about x and y increases significantly when the vehicle re-enters the atmosphere, as can be seen in Figure 1. Due to this as well as the low polling rates after deployment, accelerometer data for that portion of the flight is discarded for the remainder of this analysis.
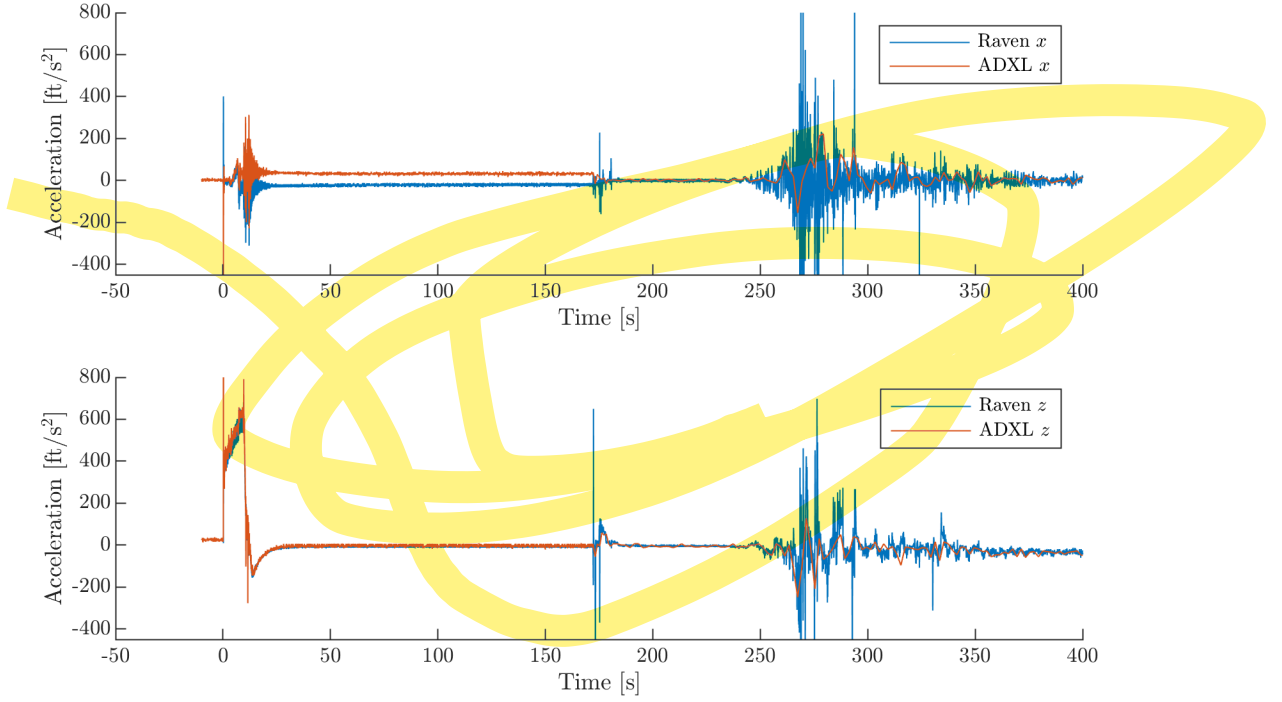
FIG. 1. Raven x and z axis compared to ADXL x and z-axis acceleration. Note that the x axes do not match, but the z axes do.

With this assumption in place, it follows that the z-axis sensor on any two accelerometers is measuring the same real acceleration (since a_c felt *along* the z-axis is due to rotation *about x* and y). This allows the ADXL x and y data to supplement the Raven z for the sake of the quaternion transform, such that the `quatrotate` function receives

$$\begin{pmatrix} a_{\text{ADXL}} \cdot \hat{x} \\ a_{\text{ADXL}} \cdot \hat{y} \\ a_{\text{Raven}} \cdot \hat{z} \end{pmatrix} \tag{3}$$

which is the accelerometer data construct used for the remainder of this analysis.

III.4. Integrating Acceleration

Before integrating, the data is converted from the transformed frame into the world frame by subtracting g from the z axis. For increased accuracy, the value of g is recalculated at each altitude using the WGS 84 gravity model.[14]

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.

---

[14] Specifically, the 'Exact' option of the MATLAB Aerospace Toolbox's `gravitywgs84` implementation of the model.

IV.   FLIGHTON: SIMULATION METHODOLOGY

At this point, FlightLine possesses 5 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, Hamster 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 Fathom II vehicle in March 2017, and validated against the benchmarks in NASA/TM-2015-218675, "Check-Cases for Verification of 6-Degree-of-Freedom Flight Vehicle Simulations", as well as the Traveler III ballistic flight[15], prior to the Traveler IV launch. 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 Traveler IV. Six degrees of freedom and 3 degrees of freedom 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.[16] The surface wind speed and direction are set to values recorded at launch using an anemometer.

The nominal simulations, anchored to measured vehicle dimensions and mass properties, outperformed any of the onboard sensor-based results. After tuning thrust and drag penalties to -0.9% and +20% respectively, there was close agreement. These penalties are consistent with damage to the nozzle and fins seen on the recovered vehicle.

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. Complete simulations were constructed by connecting the 6DOF simulation to the 3DOF simulation, and partial simulations

---

[15] Post-flight Monte Carlo trajectory reconstructions, performed using as-measured dimensions of Traveler III, launch tower angles, and local winds as inputs to FlightOn, all landed within 3 miles of the actual impact point.

[16] For more information on these models, see Appendix B.

were run by initializing the 3DOF simulation with sensor data from the ascent. The complex dynamics of the vehicle body, nose cone, and parachute were simplified into a single non-rotating body. Drag was computed using a constant C_d and area for the horizontal axes, and a C_d and area partially interpolated from flight data for the vertical axis. Specifically, the BRB GPS data was used to the compute the product $C_d \cdot A$ during the portion of the descent recorded by the BRB. This empirical $C_d \cdot A$ estimate was then smoothed and interpolated for the simulation; in the initial portion of the descent, in which no GPS data was available (and atmospheric pressure was at its lowest), a constant value was used. The value was chosen to match the quantity $C_d \cdot A$ when the vehicle regained GPS lock, as determined by analysis of BRB GPS data. [17] Because the chute's tangling, shredding, and varying degrees of openness could not be directly modeled in the simulation, Monte Carlo simulations were run where the interpolation of the product $C_d \cdot A$ experienced an associated 1-σ error deviation of 20% (determined based on the overall change in empirical $C_d \cdot A$ over the course of the flight). Due to the inexact nature of the descent simulation, these results were used exclusively to bound the possible apogee values that were produced by the sensor-measurement analysis, based on landing time.

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−10 s	**On Pad:** The vehicle is stationary in the tower as the countdown proceeds. HAMSTER enters ASCENT state.
T+0 s	**Liftoff:** First motion is recorded by the accelerometers. **Burn begin:** The motor burn begins. **GPS lock lost:** Both HAMSTER and BRB GPS modules lose their lock to the GPS constellation and stop producing position data.
T+13 s	**Burnout:** The motor burn ends. **Extreme drag:** Strong aerodynamic forces rapidly decelerate the vehicle.
T+∼40 s	**Free Fall:** The effects of aerodynamic forces become minimal as atmospheric density decreases with altitude. The main force acting on the vehicle is gravity.
T+∼151 s	**Apogee:** The vehicle reaches its maximum altitude and begins its descent.
T+173 s	**Drogue Deploy:** The drogue parachute is deployed after the HAMSTER deployment countdown times out (apogee was not successfully detected due to the loss of HAMSTER GPS lock).
T+∼230 s	**Entry Interface:** The vehicle reenters the atmosphere at ∼250 kft, at which point aerodynamic forces manifest in measured acceleration. This point can also be identified in audio from in-flight footage, as air passing over the open ends of the case and nose cone creates high-pitched noises. Shortly after this point, the vehicle (now a parachute, nose cone, and case tethered together) begins to tumble wildly, and no accelerometer on the vehicle is recording data at a high enough rate to be meaningfully integrated.

---

[17] This method likely underestimates the speed of the descent under chutes before regaining of GPS data, as the working theory for why the GPS units did not regain lock until that moment is because the chutes were tangled and the vehicle was moving too fast. This, again leads to a conservative apogee estimate

T+278 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.
	Drogue Descent: At some point prior to GPS lock until the end of the flight, the drogue parachute opens fully, making the descent (relatively) gentle.
T+658 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 of Data Sources

V.2.1.   Atmospheric and Altitude Data

V.2.1.1.   Temperature Sensors   The Hamster has two thermometers: the MS5607 barometer and the dedicated LM75B thermometer, as seen in Figure 2.
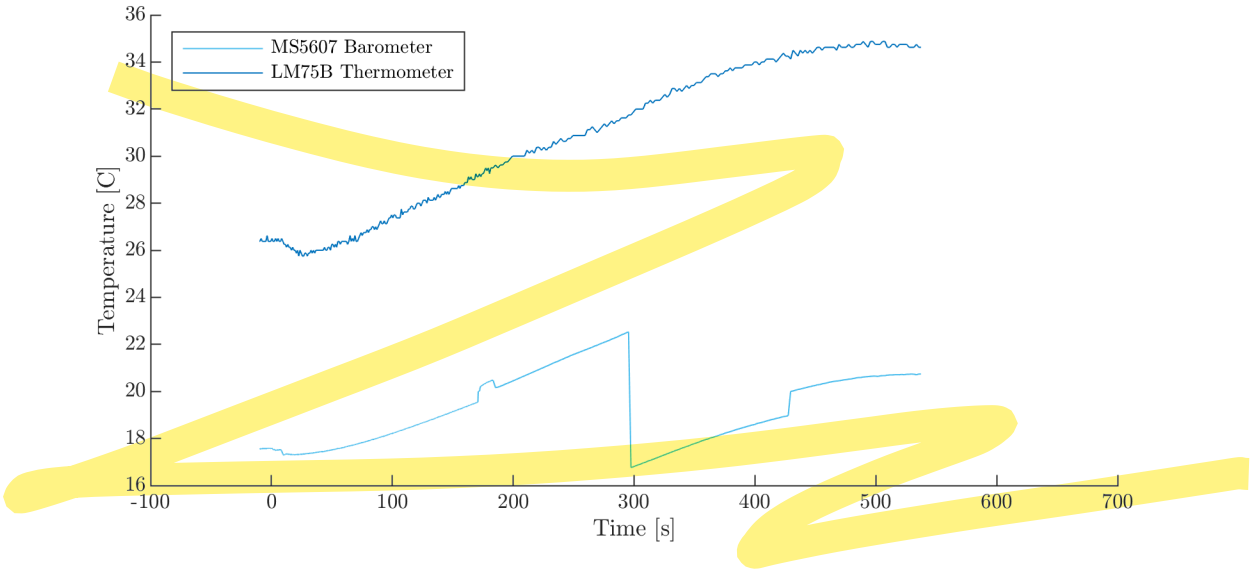

FIG. 2.   Temperature readings from the MS5607 barometer and LM75B thermometer.

The temperature data from the barometer has a number of strange discontinuities, most notably at T+295 and T+428 s. Additionally, the error bars around the two sensors do not overlap ($\pm 3°$C for the thermometer and $\pm 4°$C for the barometer). The LM75B was known to produce incorrect temperature readings, even before flight. However, as this data was not critical to flight, no action was taken to resolve this discrepancy. This data is not used at any point in the analysis, so these issues are not of great concern.

V.2.1.2.   Altimeters   There were four sources of altimeter data aboard the vehicle: two pressure-based altimeters (the Hamster's MS5607 barometer and the Raven's barometer) and two GPS units that recorded altitude (the Hamster's FGPMMOPA6H GPS and the BigRedBee), as seen in Figure 3. The Hamster and BRB GPS values from launch to T+$\sim$ 275 s, respectively, are artifacts of data interpolation and sensor handling of lost lock. The Hamster GPS continues to report and record its last known position (on the launchpad) in case of lost lock, while the BRB stops recording data points entirely.
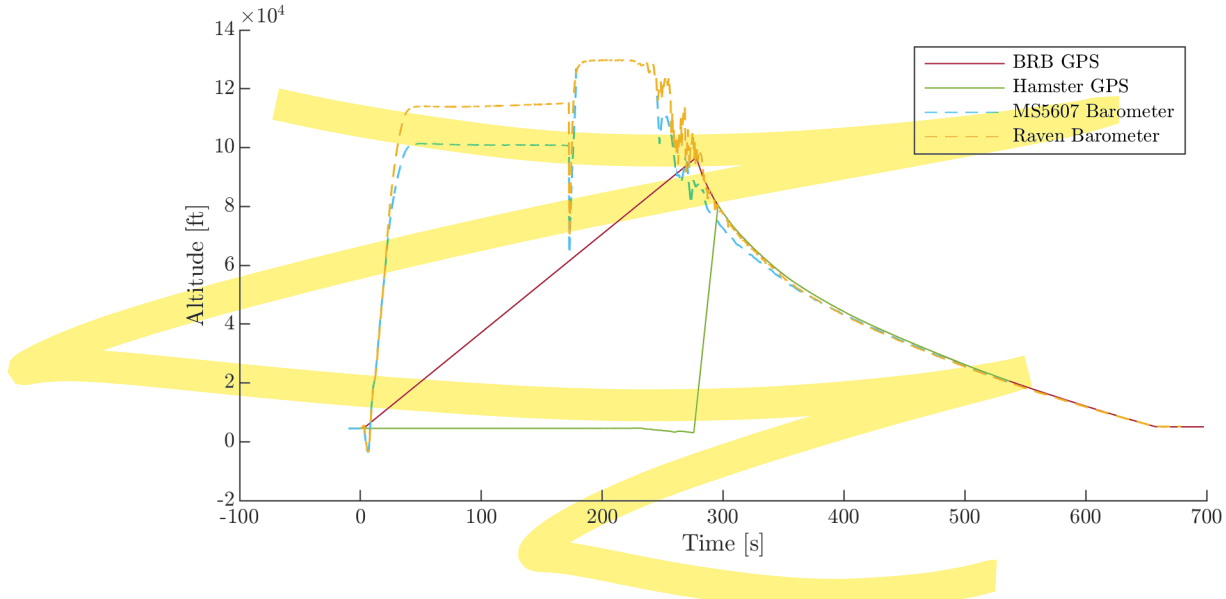
FIG. 3. Altitude readings from the altimeters and GPS units.

While the four sensors agree for the time spans for which they recorded data, there are still a number of limitations. Both pressure-based altimeters recorded a drop of -8000ft over the first 6 seconds of ascent. For the remainder of the ascent until the altimeters go out of range, they continue to read below the estimated actual altitude. Because the altimeters measure pressure within the nose cone, which is vented to ambient via two 3/16" diameter holes in the airframe, these phenomena are likely related to changing flow conditions over the vent ports as the vehicle moves through transonic, supersonic, and hypersonic flight regimes.

At drogue deployment, both altimeters read a sudden drop in altitude. This is due to the release of pressurized CO_2 used to eject the nose cone from the case.

From the time GPS lock is regained until touchdown, all four sensors record nearly the same values. The BRB data is believed to be most representative of this section, as the GPSs are more reliable than barometers, and the BRB regains lock slightly earlier than the HAMSTER's GPS. After the HAMSTER regains lock, its GPS output is nearly identical to the BRB's, confirming the BRB's results.

V.2.2.   Orientation Data

V.2.2.1. BNO080 IMU Quaternion The IMU recorded quaternion data that was used to transform acceleration data from sensor frame to transformed frame. However, despite the IMU's accurate absolute orientation data due to the combined magnetometer and gyroscope, the HAMSTER's sampling rate of its data during the ascent was low enough to inspire worries of the true rotation rate of the craft about body z being aliased in the orientation data that was recorded. Although not used for final accelerometer integration, the rotation rate of the craft is of great interest to USCRPL in explaining the performance of the craft. Therefore, GoPro footage was used to calculate the true rotation rate about the body z axis as described below.

V.2.2.2. GoPro The onboard GoPro was used to calculate the angular velocity about the body z axis. The GoPro was mounted such that it pointed out the side of the motor case just below the recovery bay. As TRAVELER IV launched in the early morning, the sun was near the horizon. This meant that the camera's degree of overexposure would vary periodically with the

same frequency as the rocket's rotation about its z axis, as shown in Figure 5. By finding the frames of peak overexposure, the angular frequency about bobdy z can be derived. The results of this methodology can be seen in Figure 4.
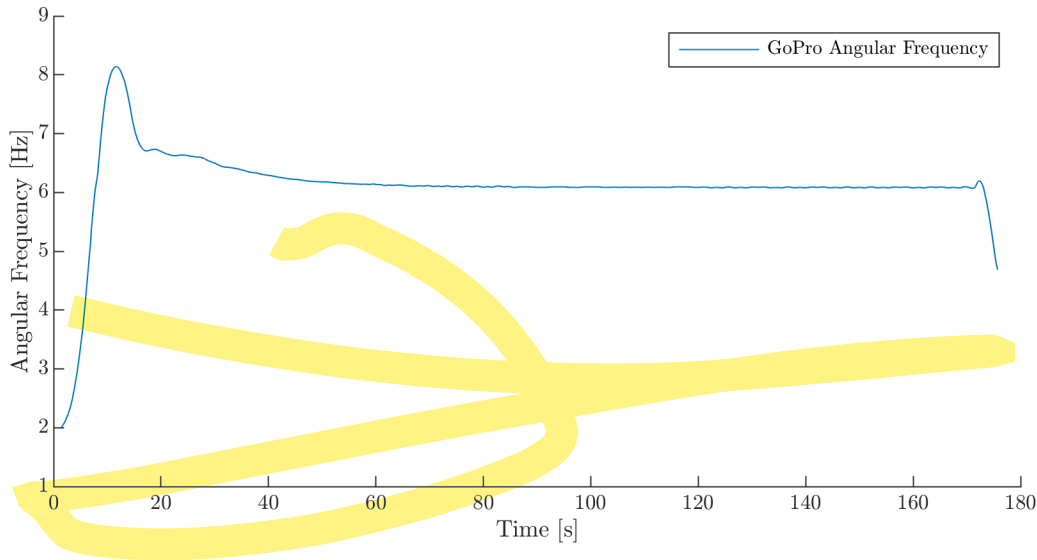

FIG. 4.   Angular frequency of rotation about the vehicle z axis as derived from onboard GoPro footage.

FIG. 5.   A typical GoPro frame vs. a highly overexposed frame.

The in-flight angular frequency as reported by the Gopro was higher than expected, as pre-flight 6-DOF simulations predicted a rotation rate of ~ 3 Hz as opposed to the sustained 6 Hz experienced.

V.2.3.   Accelerometer Data

V.2.3.1.   BNO080 IMU   The IMU recorded acceleration data and auto-transformed it into the world frame, as seen in Figure 6. It was intended to be a higher-accuracy low-load alternative to the ADXL.

The auto-transformation of this data ultimately made it unusable. Intended for virtual reality headsets, the BNO080 features auto gravity vector detection and subtraction, operating within a
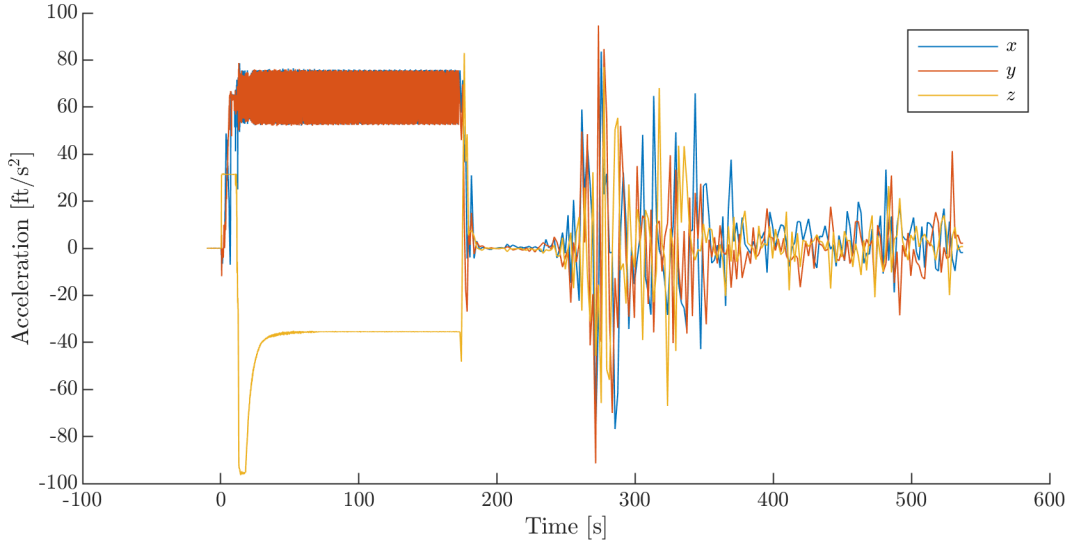
FIG. 6.   IMU acceleration data auto-transformed into the world frame.

$\pm 2g$ range. However, the manufacturer's algorithm is not optimized for free fall situations like TRAVELER IV experienced for much of the flight. Because of this, the magnitude of acceleration subtracted from the z axis is both incorrect and discontinuously time-variant, and thus cannot easily be derived to correct manually. The shifting offset is visible in a few places. During and just after the burn, the z sensor maxes out at $+1g$ and $-3g$, indicating a correct $-1g$ offset. During free fall, the offset shifts erroneously as the average z acceleration is $\sim -1.1g$, which is physically impossible and well outside the sensor's error bars. Yet another shift occurs after drogue deploy, as the IMU then reads $0g$ until entry interface, while a free fall reading of $-1g$ is still expected. This shift is also indicated by $+2.5g$ spike at drogue deploy, which would be outside the previously established $-1 \pm 2g$ range if a shift had not occurred.

The oscillation of the x and y data in a single band reflects what is expected from a sensor rotating about the vehicle's z axis, post-transform.

V.2.3.2.   ADXL375 Accelerometer   The ADXL recorded acceleration data in its sensor frame, as seen in Figure 7.

Providing raw data in the sensor frame, the ADXL encountered fewer complications than the IMU. The first notable data point is recorded at the start of burn, at which the z axis records a single point at $+119g$ (cut off on Figure 7 to preserve a reasonable scaling of the axes). It cannot be determined whether this was a real occurrence or a misreading. Conceivably, this spike could have been a brief mechanical shock from igniter firing that was stretched out by the ADXL's low polling rate. This contributed to the decision not to use the ADXL as the primary z-axis acceleration source, as no other sensor recorded such a spike and including it could erroneously increase the apogee calculated by this analysis. The other primary issue with the ADXL data is its performance during free fall. Like the IMU, the ADXL is not optimized for recording free fall data; it is intended for higher loads and thus loses accuracy for readings near $0\,g$. Because of this low-value error, once transformed into the inertial frame, the ADXL records values ranging from $-0.8g$ to $-1g$ in the world z axis. No combination of offset and sensitivity calibration values within the acceptable range could shift this value to an average of $-1g$ while preserving zero acceleration on the pad.

V.2.3.3.   Raven 4 Accelerometer   The Raven recorded acceleration data in its sensor frame for its maximum timespan of 8 minutes, as seen in Figure 8.

As mentioned in Section III.3, the primary limitation of the Raven is its lack of a y axis. Because
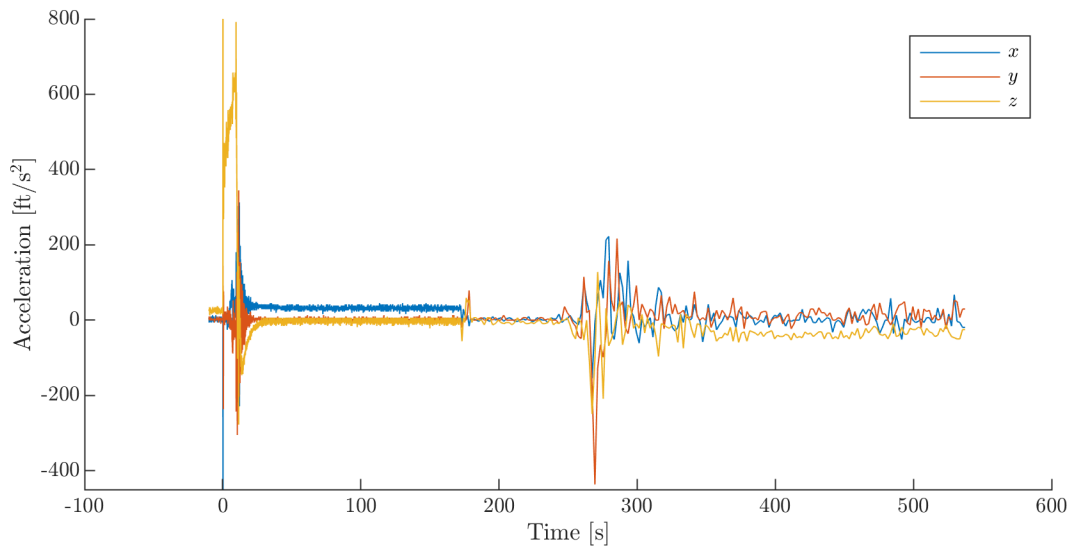
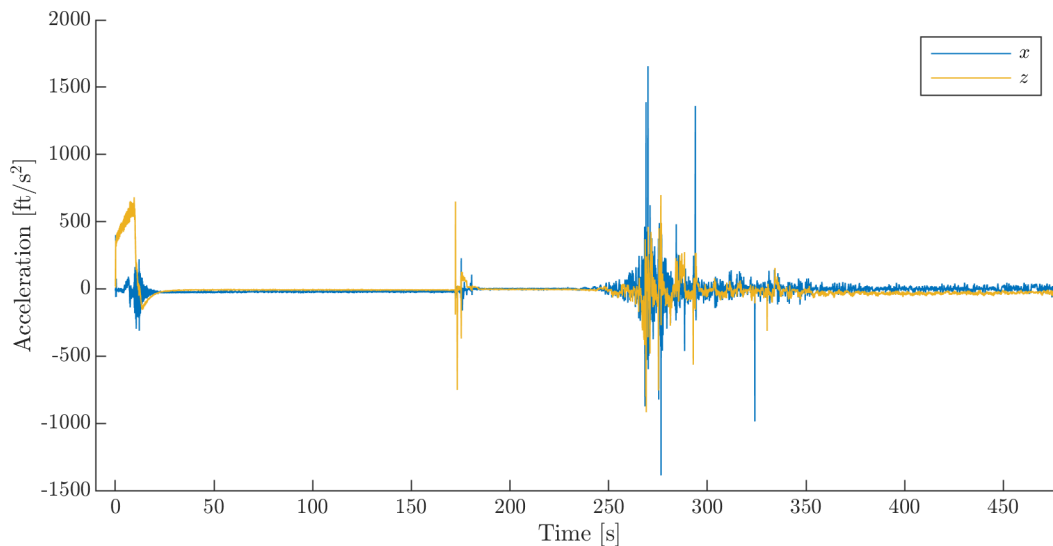FIG. 7.   ADXL acceleration data in the sensor frame.

FIG. 8.   Raven acceleration data in the sensor frame.

of this, the sensor must be supplied with x and y axis data in order to be transformed into world frame. Other than this, the Raven provides the highest quality acceleration data due to its high sensitivity and sampling rate. Even then, it is not known whether the sensor recorded with a high enough fidelity to capture the tumble at entry interface, as data of the same quality from other sensors is not available to reconstruct position and orientation. No notable anomalies appear in any other point of the data.

<div align="center">

V.3.   Data Overview

</div>

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):** BNO080 IMU quaternions

- **Acceleration x and y (Sensor frame):** ADXL375

- **Acceleration z (Sensor frame):** Raven 4

- **Altitude (world):** BIGREDBEE GPS Altimeter

<div align="center">

V.3.1.   Choice of Data

</div>

To integrate the acceleration in the world frame, a quaternion and three sensor acceleration axes are required. The quaternions were sourced directly from the IMU–despite the sampling rate's aliasing of angular velocity, the absolute orientation readings themselves were judged to be accurate. The Raven provides the highest quality of acceleration data, lacking low and high force calibration issues faced by the ADXL. Using the method discussed in Section III.3, the Raven z is combined with the ADXL x and y for transformation and integration. The ADXL was selected for this because it was the only other accelerometer with sensor frame acceleration data for the whole flight. The BRB was selected as the altimeter source for descent, as it has the largest range of reliable data.

The burn acceleration was used as a cross-sensor reference to verify data accuracy. Other than the initial spike in the ADXL described in Paragraph V.2.3.2 and visible in Figure 9, the accelerations between the Raven and ADXL are consistent. This can be checked through integration to yield velocity curves, which are identical in shape other than an initial offset caused by the ADXL spike. For a more conservative apogee calculation, the Raven z was selected.

The burn data was compared to simulated burn acceleration curves. In Figure 9, the simulated burn (with the -0.9% thrust penalty and +20% drag penalty described in Section IV.1) is consistent with the recorded data. The simulated burn was longer ($t_{b,sim} \approx 13\,\text{s}$) and produced a greater net acceleration but less acceleration per unit time than the recorded data.

<div align="center">

V.3.2.   Limitations

</div>

The main limitation faced with all of the orientation and acceleration data was the inability to capture the tumbling experienced at entry interface. While the Raven potentially captured the true x- and z-axis accelerations, the HAMSTER's automated switch to a low sampling rate following drogue deployment led to extreme aliasing and thus non-integrable acceleration and orientation data. Because of this, the accelerometer integration method can only be considered reliable until entry interface, at which point a simulation must be used to confirm that a physically plausible path exists between the end of the valid integration data and the start of the BRB altimeter data, which returns part-way to the ground.

Because of the aforementioned difficulty of transforming data into the body frame, the integrated path is that of the combined ADXL-Raven sensor.
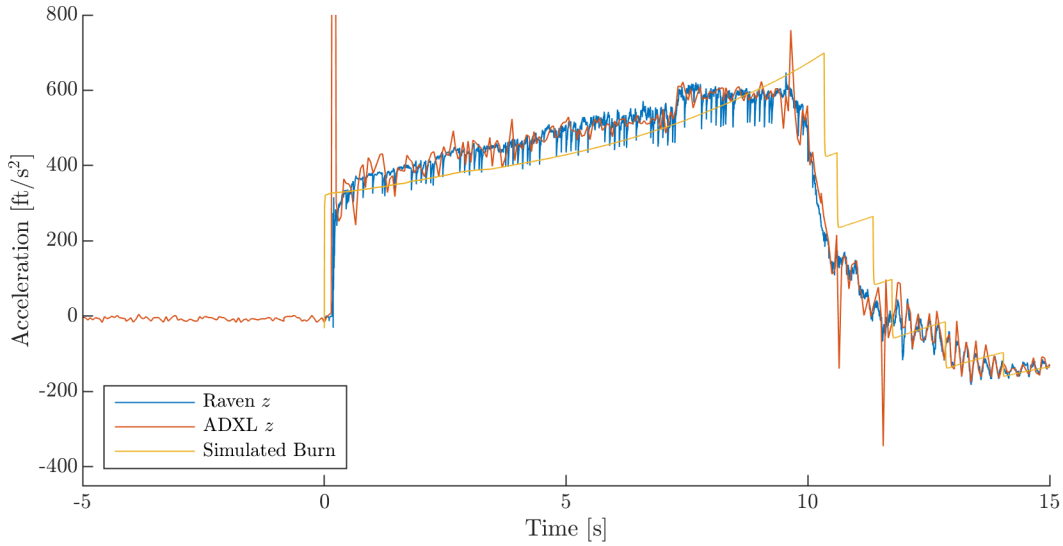
FIG. 9.   Comparison of burn z-axis accelerations between flight data and simulations. According to the Raven's z-axis data, a conservative estimate of maximum acceleration is $605\,\mathrm{ft/s^2}$ ($18.8g$)

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 combined sensor-kinematic method which kinematically estimates vehicle apogee based on sensor-determined velocity as the vehicle ascends through $200\,\mathrm{kft}$, where aerodynamic forces become negligible.[18] In both cases, uncertainty is defined as a 95% confidence interval on the set of apogees calculated from the Monte Carlo trials.

VI.1.   Integrated Accelerometer Data

A 1024 trial noise Monte Carlo was run in FLIGHTLINE on the sensor data, and the resulting integrated apogees were recorded. The resulting distribution of apogees was not normal; because of the uniform distribution of the non-linearity estimation (which dominates the overall uncertainty), the distribution has no tails.

Based on this method, the apogee of TRAVELER IV had a mean value of $340\,000\,\mathrm{ft}$, and was in the interval $[324\,200\,\mathrm{ft},\ 357\,000\,\mathrm{ft}]$ with 100% confidence. TRAVELER IV crossed the Kármán line with a confidence of 90.4% by this method.

VI.2.   Kinematic Prediction

The velocity of the vehicle at $200\,\mathrm{kft}$ based on integrated data was taken for the same trials as the direct integration method. A simple 1-D integration assuming the only force was gravity (computed using the WGS 84 model described in Section III.4) was then used to determine the apogee of each flight trial, in order to ensure that accelerometer and gyro drift did not produce

---

[18] Despite defining entry interface as $250\,\mathrm{kft}$ AMSL in Section V.1, $200\,\mathrm{kft}$ is used during ascent. This is appropriate because atmospheric density at $200\,\mathrm{kft}$ AMSL is roughly 10x that of $250\,\mathrm{kft}$ AMSL, but the projected area of the tumbling case, nosecone, and parachute during reentry is at least 10x that of the vehicle during ascent, pre-nosecone deployment.

results significantly different from physics-based predictions. The resulting distribution had the same shape as in the previous section.

Based on this method, the mean apogee of TRAVELER IV was 338 300 ft, and the apogee was in the interval [323 200 ft, 354 800 ft] with 100% confidence. TRAVELER IV crossed the Kármán line with a confidence of 85% by this method. Most importantly, the error bars and distribution for this apogee determination method overlap significantly with the entirely integration-based method, providing further validation for our primary, purely accelerometer-based apogee calculation.

In order to provide a conservative apogee estimate that falls well within physical expectations, yet makes use of the collected sensor data, the integrated accelerometer data will be used, after removing all trials that exceed the upper bound of the kinematically predicted apogee distribution. This combined method results in a final mean apogee of 339 800 ft, within the interval [324 200 ft, 354 800 ft] with 100% confidence (again due to the uniform distribution's dominating effect). This confirms that TRAVELER IV crossed the Kármán line with 90% confidence.

VI.3.   Analytic vs. Simulated Results

Both of these results were shown to be physically possible based on the 6DOF and 3DOF FLIGHTON flight simulations. In the ascent phase, Figure 10 shows that the integrated and simulated altitudes over time agree.
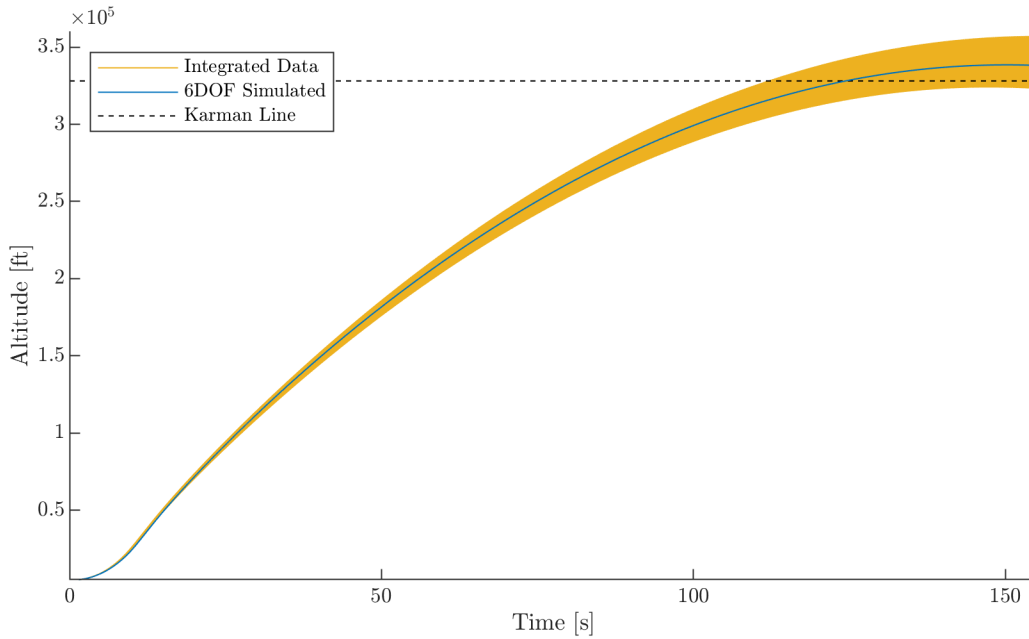

FIG. 10.   Comparison of 6DOF simulated altitude over time and the results of integrating the accelerometer data. Apogee occurs here at 151 s.

There is especially good agreement at 200 kft where the integrated data was sampled for use in the kinematic prediction. As seen in Figure 11, descent from integrated apogee to the first BRB altitude data point is possible and could match the BRB descent profile.

Limitations in the parachute simulation prevent as close of an agreement as in the ascent phase, but the results of the Monte Carlo show that the recorded trajectory from the BRB can be reasonably achieved from the reported apogee.
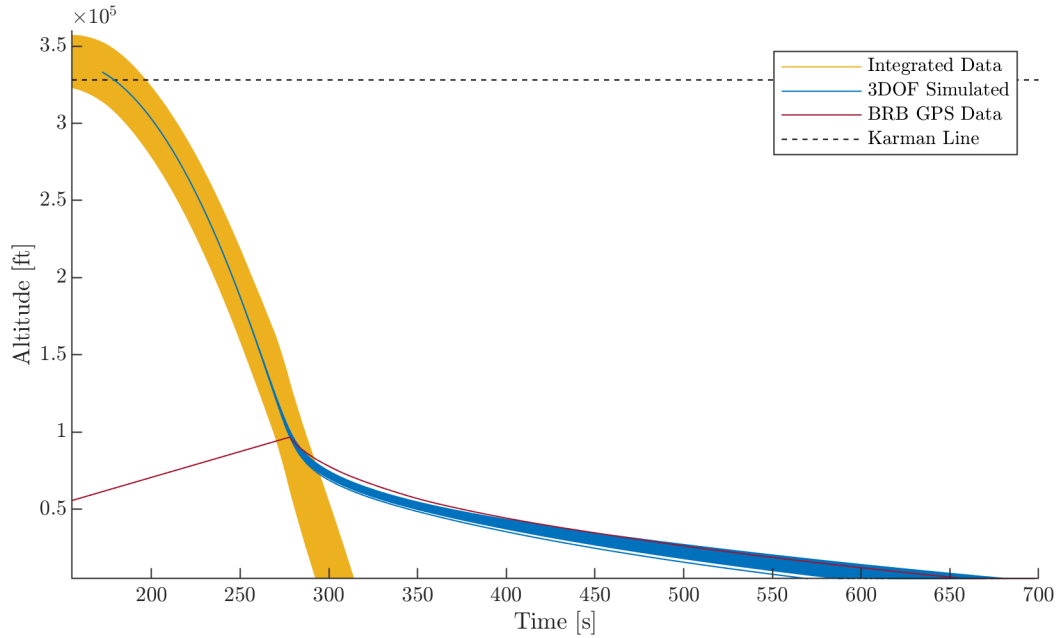
FIG. 11.   Comparison of 3DOF simulated altitude over time and the results of integrating the accelerometer data and the BRB GPS altitude.
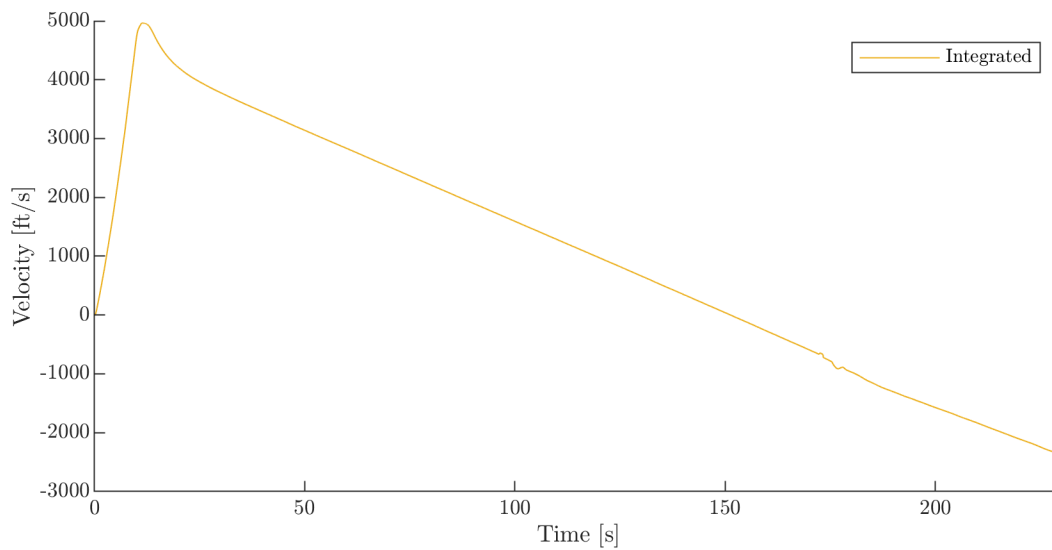

FIG. 12.   Vertical velocity versus time, integrated from nominal combined Raven-ADXL data. Maximum vertical velocity is 4966 ft/s, which occurred at T+11.5 s.
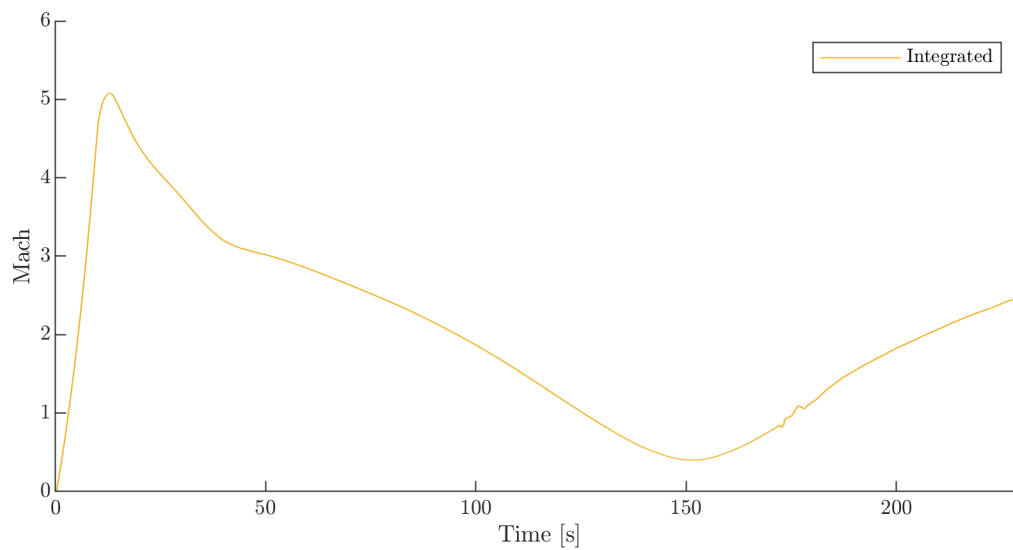
FIG. 13.   Mach number versus time, considering change in atmospheric temperature and pressure with altitude, calculated using integrated velocity from nominal combined Raven-ADXL data. Maximum Mach is 5.1, which occurred at T+12.9 s.

VII.   CONCLUSION

After careful analysis of data recorded during the Traveler IV launch, vehicle apogee was determined to be $339\,800$ ft $\pm$ 16,500 ft AMSL, higher than the Kármán line at $328\,064$ ft AMSL with a confidence of 90%. As such, Traveler IV is the first entirely student designed-and-built rocket to have flown to space, as well as the first university rocket to be recovered from a space flight.

This conclusion has been reached by taking all possible care to validate all collected data, tabulate and quantify all sources of error and uncertainty, and–when necessary–err on the side of *underestimating* the final apogee number. Data points which pushed our final estimate higher, yet were not validated by other live readings (such as the initial ADXL a_z spike) were rejected from the analysis. Overall, the aforementioned confidence interval of 90% reflects a highly conservative approach to determining the vehicle's apogee, and as such has the full support of the USCRPL team.

Appendix A: Avionics: Next Steps

Despite the overall success of Traveler IV, USCRPL recognizes that the quality of the recovered dataset requires improvement before the next flight. To that end, the team has several changes planned for the next iteration of the avionics unit, which should address many of the uncertainties present in the Traveler IV data in future flights. In particular, the following changes are scheduled for immediate development:

- **Polling frequency increases**: Sensor polling frequencies will be increased by at least an order of magnitude in order to improve the accuracy of accelerometer integration-based position calculations, especially during entry interface. Due to a number of uncertainties inherent to launching a space vehicle, the team was concerned about exhausting the unit's onboard memory, leading to the selection of the low data recording rates during descent. However, as only a few percent of the available memory space was actually filled during the flight, this worry turned out to be unfounded, allowing the team to take a less conservative approach in the future.

- **IMU settings changes**: The BNO080 IMU will be taken out of auto-subtract-gravity mode, removing the unknown element from its accelerometer data and allowing it to better serve its intended role: a high-accuracy supplement to the noisy low end of the ADXL375 accelerometer's range, particularly for the freefall and re-entry stages of the flight.

- **New high-fidelity IMU**: In addition to the changes to the treatment of current hardware, the team intends to add an extremely high-fidelity accelerometer and IMU to supplement the readings from the current sensor set. The team is currently testing the ADIS16467, which has much lower error bounds than the present sensor set and is still able to measure the large accelerations of motor burn by itself. It also includes a low-drift gyroscope that can reinforce the combined gyro-magnetometer readings from the present BNO080 IMU. This sensor will be flown and qualified on an RPL vehicle by next spring, and pending good results, it will be included in the avionics package on the next high performance vehicle.

- **New GPS hardware**: The team is also investigating new GPS hardware. Standard consumer hardware such as the GPS module flown on Traveler IV may, for a variety of reasons, be unable to handle the extreme conditions of a high-performance flight. For the next high-altitude launch, USCRPL is seeking partnerships with aerospace-grade GPS manufacturers and research teams, as well as investigating the possibility of a home-brew recording-only GPS unit, which would free the system from relying upon on-line calculation of GPS location by hardware that's simply not qualified for flight conditions.

- **Ground-based positioning systems**: The team is also considering ground-based absolute position determination methods, such as ground-based triangulation or radar tracking. If in-flight GPS doesn't work, some method of absolute position tracking is strongly preferred to supplement even the highest-fidelity IMU data, which suffers from integration drift and significant noise, as seen throughout this report.

In future flights, the changes above should result in a more easily discernible and less uncertain reconstructed flight profile.

It is the firm belief of the avionics team that the issues addressed above and in this analysis were due not to negligence in system design or testing, but rather to the unexpected conditions encountered when launching an amateur vehicle to space. For the avionics team, this launch represents the first time a full set of flight data has been recovered from a USCRPL high-altitude

vehicle. New problems were not only understandable, but to be expected. Fundamentally, the flight of TRAVELER IV served as an unprecedented proof-of-design for the USCRPL avionics system: the vehicle was recovered, a complete and valid set of flight data was obtained, and the unit remained completely intact and functional throughout the highly dynamic flight, descent, and landing. Moreover, its cargo of highly valuable system data is in the hands of the team, and will be used to inform the next iteration of the design, allowing USCRPL to reach ever greater heights.

Appendix B: 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[19]. For TRAVELER IV flight simulations, the parameters listed in Table VI 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[20]. This approximation is accurate to within $\sim 3 \times 10^{-4}$ m/s² 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 VI.

Model	Parameter	Value
MSISE-00	F10.7 cm (daily)	70
	F10.7 cm (90 day avg.)	71.2889
	AP	5
WGS 84 Approx.	Gravitational parameter	$3\,986\,004.418 \times 10^{8}$ m/s²
	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 VI. Atmosphere and Gravity Constants

---

[19] https://www.brodo.de/space/nrlmsise/

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