Flight Data Analysis Using Excel - Part 1

Thomas B. Fetter
NAR 15551

Introduction

Being an engineer, I like to design and build things, and then measure the results and look for ways to improve the design. And when it comes to designing, building, and flying rockets, I like to measure everything I can about their flight trajectory. To do this, I fly my rockets with both a recording barometric altimeter and a recording GPS. The GPS data contains all the information that is needed, latitude, longitude, as well as altitude, to create a 3D flight profile. But the altimeter has a higher sample rate than the GPS so it has additional information about the flight that the GPS misses. When analyzed together, more details can be learned about the flight than can be learned using either one alone. The key to comparing the GPS and altimeter data is getting the data into an Excel spreadsheet where the flight data can then be easily manipulated and plotted together on the same graph. The results from flight simulation software can also be imported into Excel to enable comparing the simulated to measured results to see how closely they match. Using a recent flight of one of my rockets, I will show you how to import the data into Excel and analyze and compare the data.

The example rocket I will use is Speedmotion 54. The flight I will be analyzing occurred at the Dairy Aire launch, sponsored by TCC at the Maddox Dairy in the Central Valley of California, in May, 2013. This rocket is 3 inches diameter, 72 inches long, with a 2 inch motor mount. The rocket has redundant dual deploy, using two PerfectFlite altimeters, one HiAlt45 and one StratoLogger located in the ebay at the middle of the rocket, and the GPS receiver with a HAM band data transmitter, a BeeLine GPS, located in an ebay in the nosecone of the rocket. For this flight, I flew on an AeroTech J415W motor. Ready for launch, Speedmotion weighs 10.5 lbs. It has an 18 inch drogue, and 48 inch main parachute, both from Fruity Chutes. This was the first flight of Speedmotion 54, but its larger brother, a 4 inch diameter, 75mm motor mount version of the same design, has flown many times to over 10,000 feet. On its first flight, Speedmotion 54 flew to around 6,000 feet, and landed about a half mile from the pad.

Figure 1 - On the pad, ready for launch at Dairy Aire

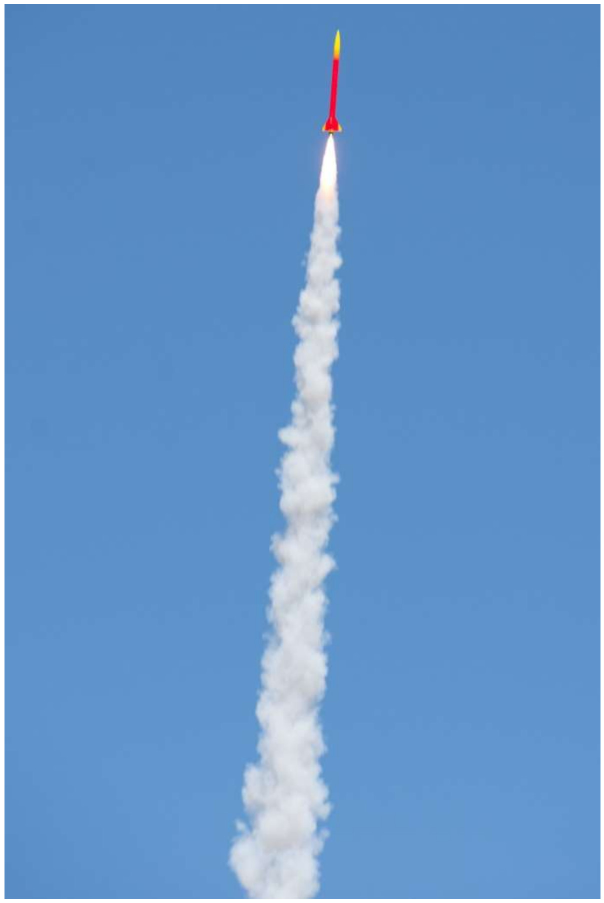

Figure 2 - Speedmotion 54 on its way to 6000 feet

The Altimeter Data

Speedmotion 54 uses two altimeters, a main and a backup, just to help ensure a successful recovery, as shown in Figure 3. Both are set up for dual deployment. On Speedmotion 54, the primary altimeter is an older HiAlt45, but this unit does not have data logging capability. For the recording altimeter, I used the newer model PerfectFlite StratoLogger which can record the altitude data to on-board memory. Because the StratoLogger can be set up with a time delay after apogee detect to ensure the two deployment charges do not occur simultaneously and over pressurize the rocket , I use the HiAlt45K as the primary deployment altimeter, and the StratoLogger as the backup for both drogue, and main deploy as well. For main deployment, I set the HiAlt45k to deploy at 1000 feet, and the StratoLogger to deploy at 800 feet, again, to ensure the deployment charges do not fire at the same time. I'll say more about this choice later. Both units are barometric altimeters, meaning they use the barometric pressure to determine altitude. I have used the HiAlt45K for many years and I have never had a deployment failure. The StratoLogger, with its data logging capability, is a welcome update to the HiAlt45. PerfectFlites are very simple and easy to use and have high current outputs for reliable deployment, all reasons why I like them, but there are other good choices for recording altimeters that would work for this purpose as well.

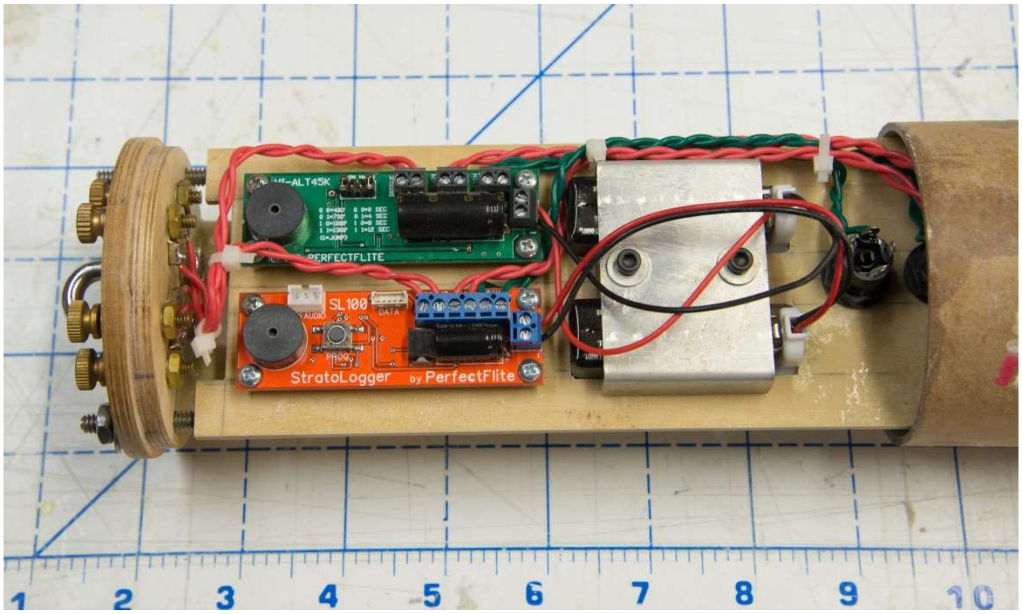

Figure 3 - Speedmotion 54 altimeter bay

The PerfectFlite StratoLogger has a software application, PerfectFlite DataCap, which can be used to set up the unit, as well as for downloading the logged flight data and plotting the altitude vs. time (Figure 4). After downloading the StratoLogger data into the PerfectFlite DataCap software, the data file can then be saved to the PC. The PerfectFlite manual does a good job of describing the process. The file is saved to a .pf2 file, which is unique to PerfectFlite, but the format is a simple txt file with comma separated values. Later, I will describe the process to load this data into Excel to do additional analysis of the data.

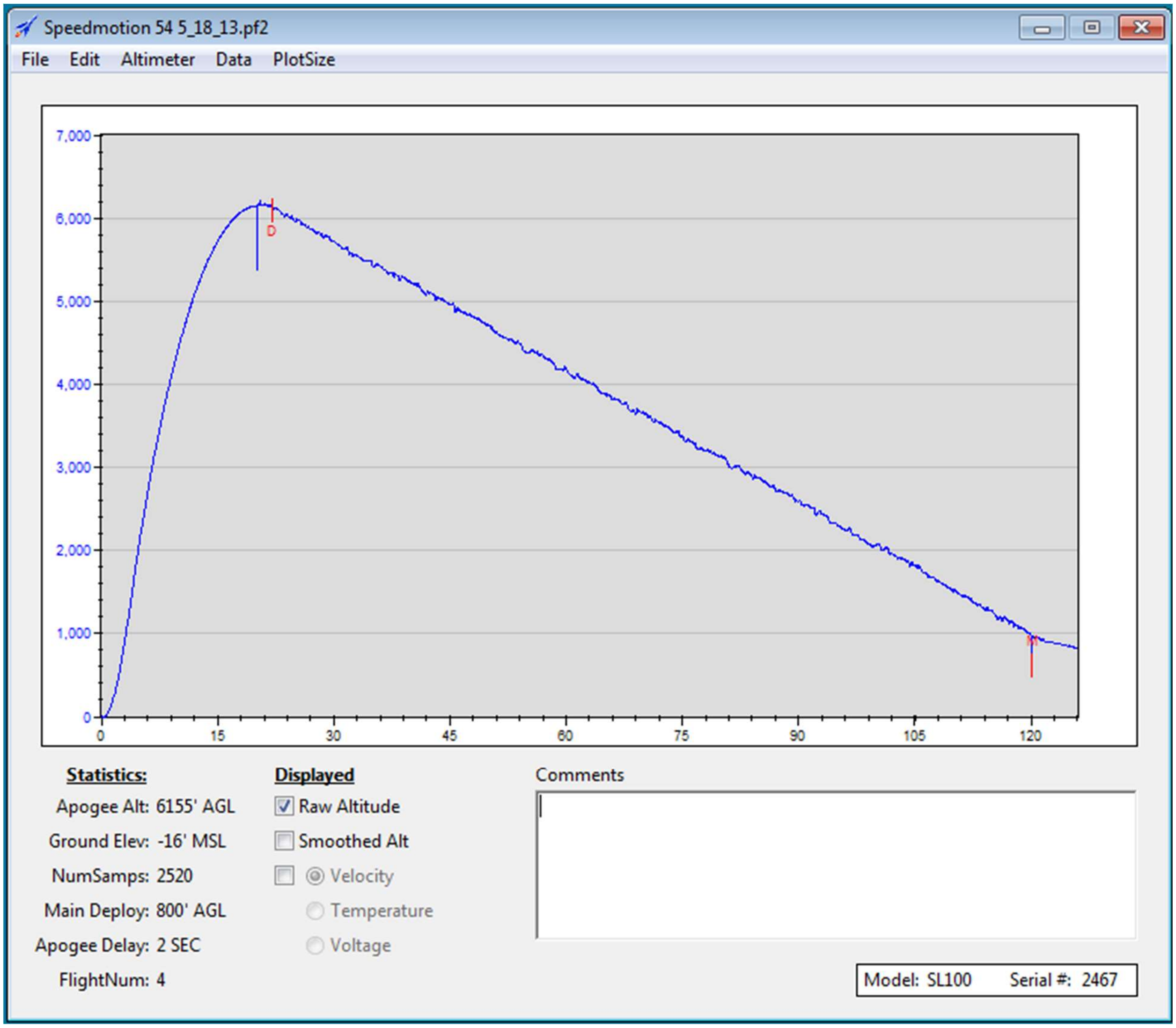

Figure 4 – PerfectFlite DataCap

The StratoLogger has a sample rate of 20 samples per second. This is enough time resolution to get a good picture of the ascent portion of the flight. The StratoLogger also has enough memory, at over 9 minutes, to capture the complete descent for most flights.

There are several interesting things to note about the data in Figure 4. The two red markers indicate the drogue and main parachute deployments. The data capture for the flight of Speedmotion ended at 126 sec, shortly after the main ejected at 1000 feet, as indicated by the second marker, and did not include the final portion of the flight below 800 feet. The StratoLogger did not run out of memory, as it can store well over the 126 seconds of data. There are two positive pressure spikes, seen as lower altitude points, one at the apogee, and one at 1000 feet. The apogee charge fired 2 seconds after apogee, as set (remember the StratoLogger is set up as the backup to the HiAlt45) and indicated by the first deployment marker, but the main backup charge went off at 1000 feet, as indicated by the second deployment marker, rather than at 800 feet as the StratoLogger was set to do. What happened? I will come back to this later.

The GPS Data

The StratoLogger provides altitude information versus time, but for complete trajectory information in 3D space, GPS data is required. For the GPS, I use a BeeLine GPS. To mount the BeeLine GPS in the rocket, I wrap it in anti-static bubble wrap and slide it into a 38mm tube mounted in the nosecone as shown in Figure 5. That way, it is easy to move the unit from one rocket to another. This is easy to do since the GPS does not need to be hard-wired into the rocket as required for the altimeter to connect to the deployment charges. It is important to wrap the BeeLine for a tight fit in the tube so that it does not slide easily. The BeeLine is easily damaged when the nosecone is ejected if it is allowed to slide in the tube either forward or backward. I once had a BeeLine slam into the compartment's rear bulkhead at ejection, due to lack of adequate padding behind the unit, and shear a component from the PC board.

In addition to broadcasting the current altitude and coordinate locations on the 440MHz HAM band, the BeeLine GPS also records the data to onboard memory.   The data recorded to onboard memory is used to determine the flight trajectory after the completing of the flight, but the real-time broadcast GPS data can be used in conjunction with a Ham receiver, an AX.25 packet decoder, and a GPS receiver to track rocket during the flight and find its location after landing. For this, I use a Kenwood TH-D7 transceiver that has a built in TNC (packet decoder) and a Garmin 60CS GPS receiver. Patching the decoder output of the Kenwood to the Garmin using the NMEA interface places a real-time waypoint at the current location of the rocket. Even though the TCC field consists of miles of flat and open farmland, GPS makes finding a rocket so much easier. I've had the big brother Speedmotion 75, a 4 inch version of Speedmotion with a 3 inch motor mount, land as far as 2 miles from the pad in a field of alfalfa. Even a field of 12 inch alfalfa can completely hide a rocket the size of Speedmotion until you are within about 20 feet. Without GPS, the rocket would have been impossible to find.

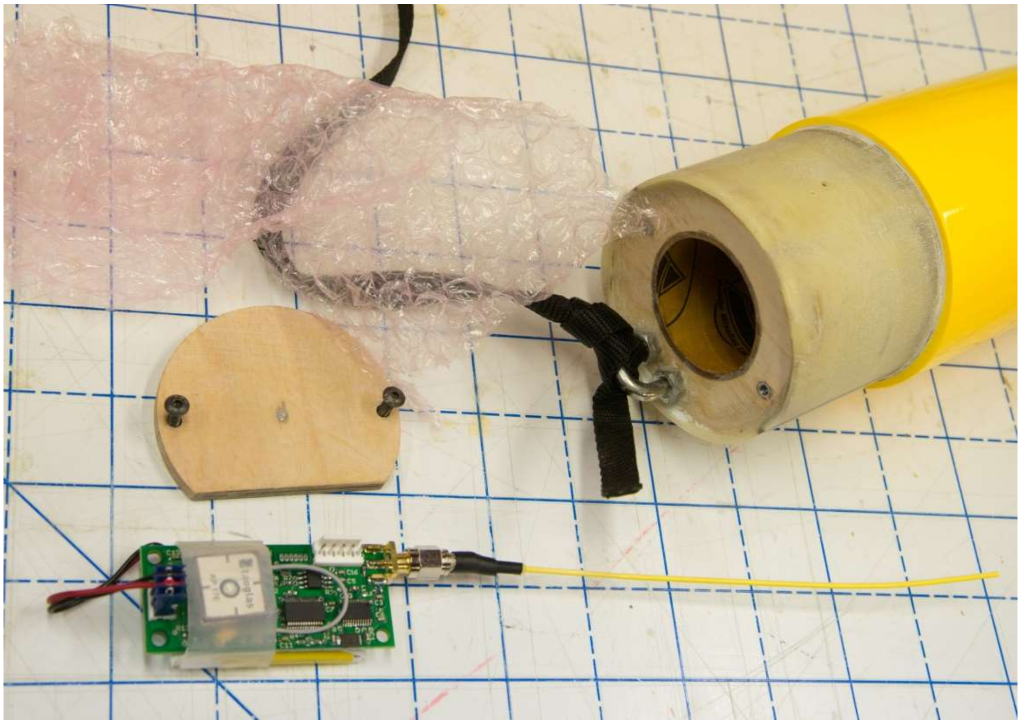

Figure 5 - Speedmotion 54 BeeLine GPS and nosecone bay

The BeeLine GPS Communicator software is used to program the BeeLine GPS. Figure 6 shows the parameters that can be set with the software. The BeeLine GPS unit used on this flight is one of the new BeeLine GPS units introduced a little over a year ago. The current version of the Communicator software talks to both the current and older BeeLine GPS units. The parameters for the older models are a little different than those shown here for the new model.

The new BeeLine units use a new Ublox GPS chipset, as opposed to a Trimble chipset used on the older units. The new BeeLine units do a much better job of maintaining lock as the rocket is first launched. The older units would typically loose lock at the beginning of the powered ascent and would miss the first 5-10 data points. The new units also have more memory, eliminating the need for the Smart Launch Detect feature or the g-switch, which sometimes caused reliability problems. Also, the data on these new units is much less noisy. I highly recommend using one of the newer units from Big Red Bee. These units may appear to be expensive, but, compared to the cost of losing a rocket, they are well worth the money, and the newer units appear to work much more reliably than the older.

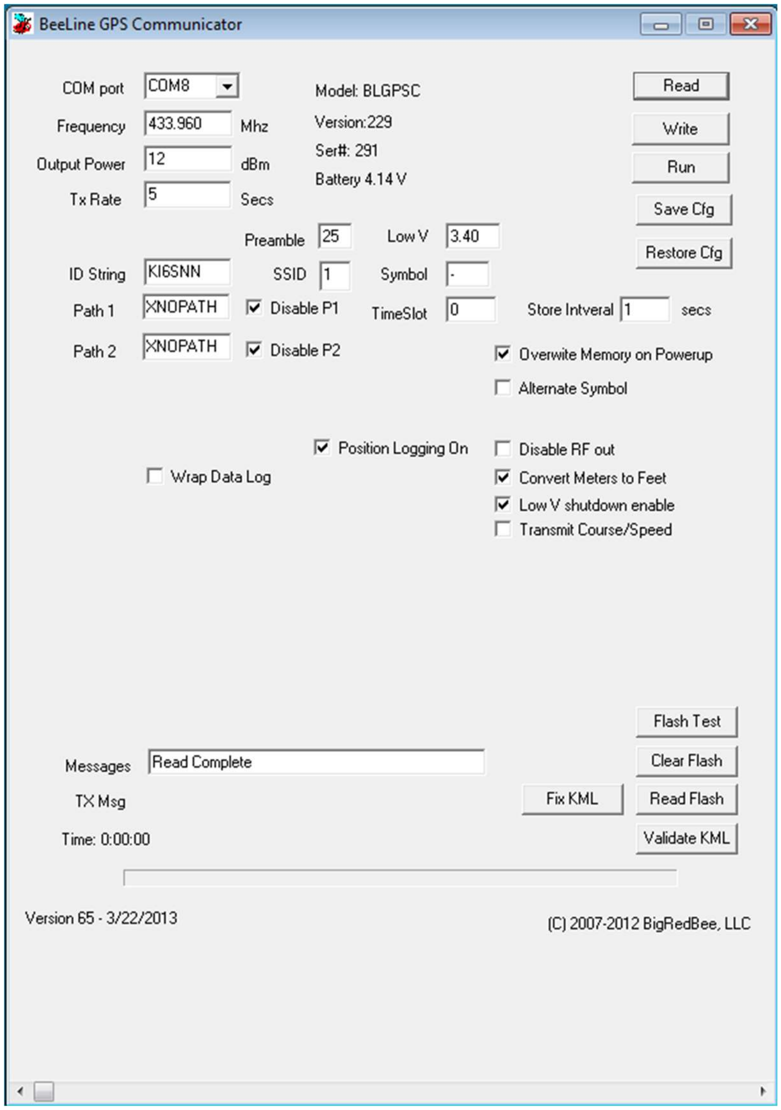

Figure 6 - BeeLine GPS Communicator software

The fastest rate that the BeeLine GPS saves the trajectory data is once per second. Although this rate is fine for capturing the general flight trajectory, some details of the flight might be missed. As we will see, if the GPS and the altimeter data are used together, a much more complete view of the flight can be constructed.

The BeeLine GPS software also allows saving the data file after download. The file is saved as a .kml file, which is the file format used by Google Earth. If you have Google Earth installed on your computer, which is available as a free download from the Google web site, then double clicking on the saved BeeLine GPS file automatically opens Google Earth and plots a 3D rendition of your rockets flight trajectory overlaid on an aerial image of the flight field. Figure 7 shows the flight path of Speedmotion 54. In this view, the rocket was prepped at the far left of the yellow tacking path line, then walked to the pad due east. From there, it was launched to a peak altitude of 6512 feet, and then drifted under parachute to the south-east, where it was recovered and returned to the starting location via the roads dividing the half mile square fields. The center of the view perspective on Google earth is a little north of the launch point, and so the flight path is seen in perspective as the arching line that rises up the west side of the arch, with deployment of the drogue at the peak, and a descend down the east side of the arch.

Google Earth requires the units for the altitude data in the .kml file be meters. The data file must be in meters even when the readout units are set as feet in Google Earth. In the BeeLine GPS Communicator interface, there is an option to select “Convert Meters to Feet”. For the new BeeLine GPS units, even if the “Convert Meters to Feet” option is selected, this applies to the transmitted data only. The data is always saved in meters, so the data is always in the correct format for Google Earth. For the older BeeLine GPS units, the data will be saved in feet when “Convert Meters to Feet” is selected. If plotted in Google Earth, then the altitude data will be 3.28 times too high. In the past, I have used a simple Excel spreadsheet that converts the altitude back to meters. I then re-save as a text file, and then re-name the file with a .kml suffix.

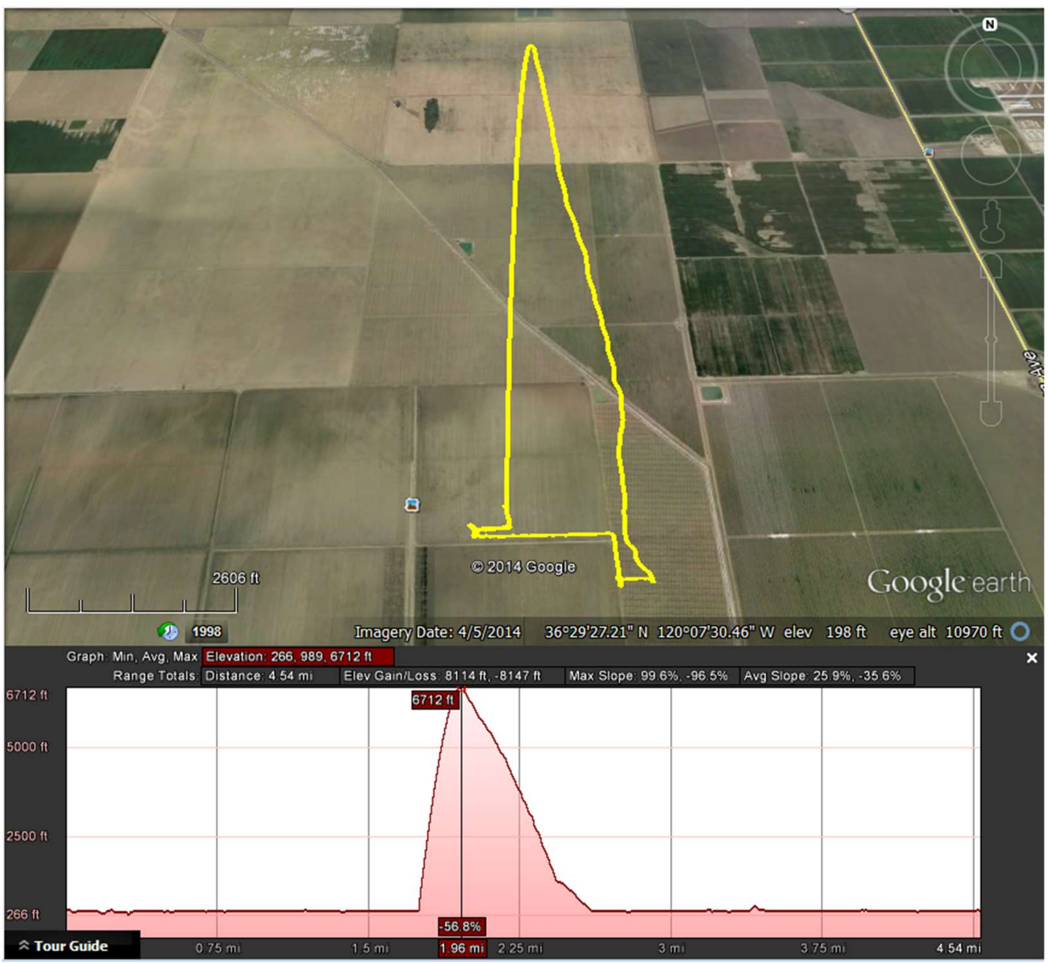

Figure 7 – BeeLine GPS data in Google Earth

It's fun viewing the rocket's trajectory in Google earth. The quality of the aerial images are amazingly good for many locations, and you can change the perspective and zoom down to the ground level to view a profile of the flight trajectory. Turning on "Show Elevation Profile" under the "Edit" menu shows the elevation of the flight profile (see Figure 7). But the plot is elevation vs. distance rather than time like the StratoLogger plot. This is good viewing the elevation profile of a bicycle trip over mountains, but not so for plotting the flight trajectory of a rocket. The Google Earth plot is missing the quantitative information that we are looking for to compare this data to the StratoLogger data. The information required to generate the desired plots is in the GPS data, but additional work is required to take full advantage of the logged GPS data.

Getting the GPS Data into Excel

The first step is to get the BeeLine GPS data into a tool to enable analyzing the data in more detail and plotting the results. For this, I use an Excel spreadsheet. My Excel spreadsheet is available for download at www.Speedmotionrockets.com under the spreadsheets tab.

The following steps are used to get the BeeLine GPS data into an Excel spreadsheet. This example is for Excel 2010; the steps for previous versions of Excel are similar.

1. Create a copy of the BeeLine . kml file
2. Change the extension from .kml to .txt
3. Run Excel
4. Select File, Open, then select Files of Type: txt, and open the BeeLine GPS .txt file
5. The Text Import wizard will open. On the first page select Delimited & press next
6. On the second screen select comma, and for the BeeLine file, also special character and enter <. Unselect any other delimiters (space and tab). Press next
7. Review proposed columns and if OK press Finish

The first few lines of the file are the header that gives general information about the flight, but scrolling below the header lines, the data is now organized into columns. The column widths may need adjusting to see all of the digits. When saving the Excel file, use SaveAs, give the spreadsheet a new name, and be sure to change the file format from the default .txt file to an .xls or .xlsx spreadsheet file. Figure 8 shows the data in Excel with a little formatting of the columns. If you have downloaded my spreadsheet, then the data can simply be copied and pasted into the data columns on the Beeline data tab.

Beeline File			
Longitude	Latitude	Altitude (m)	counter:#sat:time (s)
-120.1263	36.48157	56	!-- 63795 sats:12 UTC 19:37:27 --
-120.1263	36.48157	56	!-- 63810 sats:12 UTC 19:37:28 --
-120.1263	36.48157	55	!-- 63825 sats:12 UTC 19:37:29 --
-120.1263	36.48157	55	!-- 63840 sats:12 UTC 19:37:30 --
-120.1263	36.48157	56	!-- 63855 sats:12 UTC 19:37:31 --
-120.1263	36.48157	56	!-- 63870 sats:12 UTC 19:37:32 --
-120.1263	36.48157	56	!-- 63885 sats:12 UTC 19:37:33 --
-120.1263	36.48157	56	!-- 63900 sats:12 UTC 19:37:34 --
-120.1263	36.48157	56	!-- 63915 sats:12 UTC 19:37:35 --
-120.1263	36.48157	56	!-- 63930 sats:12 UTC 19:37:36 --
-120.1263	36.48157	56	!-- 63945 sats:12 UTC 19:37:37 --
-120.1263	36.48158	56	!-- 63960 sats:12 UTC 19:37:38 --
-120.1263	36.48157	56	!-- 63975 sats:12 UTC 19:37:39 --
-120.1262	36.48163	161	!-- 63990 sats:12 UTC 19:37:41 --
-120.1262	36.48174	311	!-- 64005 sats:12 UTC 19:37:42 --
-120.1261	36.48191	505	!-- 64020 sats:12 UTC 19:37:43 --
-120.1261	36.48208	702	!-- 64035 sats:12 UTC 19:37:44 --
-120.126	36.48224	880	!-- 64050 sats:12 UTC 19:37:45 --
-120.1259	36.4824	1039	!-- 64065 sats:12 UTC 19:37:46 --
-120.1259	36.48255	1185	!-- 64080 sats:12 UTC 19:37:47 --
-120.1258	36.48269	1316	!-- 64095 sats:12 UTC 19:37:48 --
-120.1258	36.48283	1433	!-- 64110 sats:12 UTC 19:37:49 --
-120.1257	36.48296	1537	!-- 64125 sats:12 UTC 19:37:50 --
-120.1256	36.48309	1629	!-- 64140 sats:12 UTC 19:37:51 --
-120.1256	36.48321	1711	!-- 64155 sats:12 UTC 19:37:52 --
-120.1255	36.48332	1780	!-- 64170 sats:12 UTC 19:37:53 --
-120.1255	36.48344	1840	!-- 64185 sats:12 UTC 19:37:54 --
-120.1254	36.48355	1888	!-- 64200 sats:12 UTC 19:37:55 --
-120.1254	36.48365	1928	!-- 64215 sats:12 UTC 19:37:56 --

Figure 8 – BeeLine GPS data imported into Excel

Looking at the data from the BeeLine, there are four columns of data. They are Longitude, Latitude, Altitude, and a column that contains a counter, the number of satellites being received, and seconds counter. We will start by using the altitude vs. time data to create a plot in Excel, and compare that plot to the StratoLogger altitude plot. Again, the units for altitude in the table are meters, even if the "Convert Meters to Feet" option is selected in the BeeLine GPS Communicator application, as this applies to the transmitted data only. I like to plot altitude in feet, and so a conversion factor of 3.28 feet per meter is applied to the data in the spreadsheet before plotting.

Figure 9 shows the plot of altitude vs. time for the GPS data entered in Excel. The insert plot function was used in Excel to create the graph from the data table. On this plot, I have enabled the data point markers to show the density of the actual data samples. During the ascent, the one second sampling interval results in a relatively small number of actual data points during that portion of the flight.

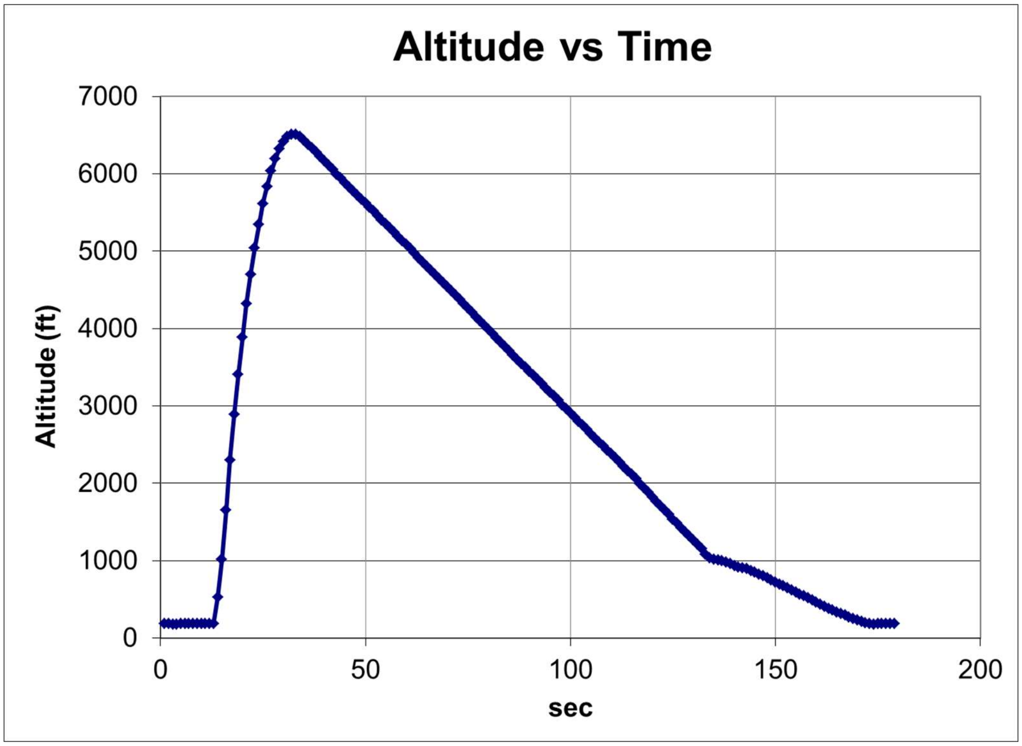

Figure 9 – BeeLine GPS altitude plot in Excel

There are several things to notice from this plot. The data is offset by the ground elevation of 184 feet since the GPS altitude is always referenced to sea level. The peak altitude is 6,512 feet, which is more than the 6,155 feet reading from the StratoLogger even after offsetting the GPS data by the ground elevation level. The next step is to pull the StratoLogger data into the same spreadsheet so that the two altitude plots can be compared more closely.

Useful References

PerfectFlite DataCap: http://www.perfectflite.com/Download.html
BeeLine GPS Data Communicator: http://www.bigredbee.com/BeeLineGPS.htm
RockSim: http://www.apogeerockets.com/Rocket_Software
My Excel spreadsheet: http://www.Speedmotionrockets.com

Flight Data Analysis Using Excel - Part 2

Thomas B. Fetter
NAR 15551

Adding the Altimeter Data into Excel and Comparing Results

Adding the StratoLogger data into the Excel spreadsheet follows a similar process to entering the BeeLine GPS data.

1. Create a copy of the StratoLogger.pf2 file
2. Change the extension from .pf2 to .txt
3. Run Excel
4. Select File, Open, then select Files of Type: txt, and open the StratoLogger.txt file
5. The Text Import wizard will open. On the first page select Delimited & press next
6. On the second screen select comma, and unselect any other delimiters (space and tab). Press next
7. Review proposed columns and if OK press Finish
8. Because the data is imported into a new spreadsheet, this data must be cut and pasted into the same spreadsheet that was used for the GPS data.
9. Save the spreadsheet with both the BeeLine GPS and StratoLogger data, the new spreadsheet is no longer needed and can be closed without saving.

Figure 10 shows the StratoLogger data once imported into Excel with the addition of the column headers.

Stratologger Parameters				
Time	Altitude	Velocity	Temp (F)	Voltage
0	0	0	81.57	9.5
0.05	1	13	81.57	9.5
0.1	1	-21	81.57	9.5
0.15	0	-8	81.57	9.5
0.2	-4	-37	81.57	9.5
0.25	-2	-40	81.57	9.5
0.3	-13	-29	81.57	9.5
0.35	-16	-5	81.57	9.5
0.4	-13	26	81.57	9.5
0.45	-6	59	81.57	9.5
0.5	1	86	81.57	9.5
0.55	6	103	81.57	9.5
0.6	11	114	81.57	9.5
0.65	17	122	81.57	9.5
0.7	22	132	81.57	9.5
0.75	28	146	81.57	9.5
0.8	38	163	81.57	9.5
0.85	47	179	81.57	9.5
0.9	57	192	81.57	9.5
0.95	67	201	81.57	9.5
1	79	204	81.57	9.5
1.05	92	204	81.57	9.5
1.1	98	205	81.57	9.5
1.15	107	209	81.57	9.5

Figure 10 – StratoLogger data imported into Excel

Figure 11 shows both the BeeLine GPS and the StratoLogger raw data plotted on the same graph.

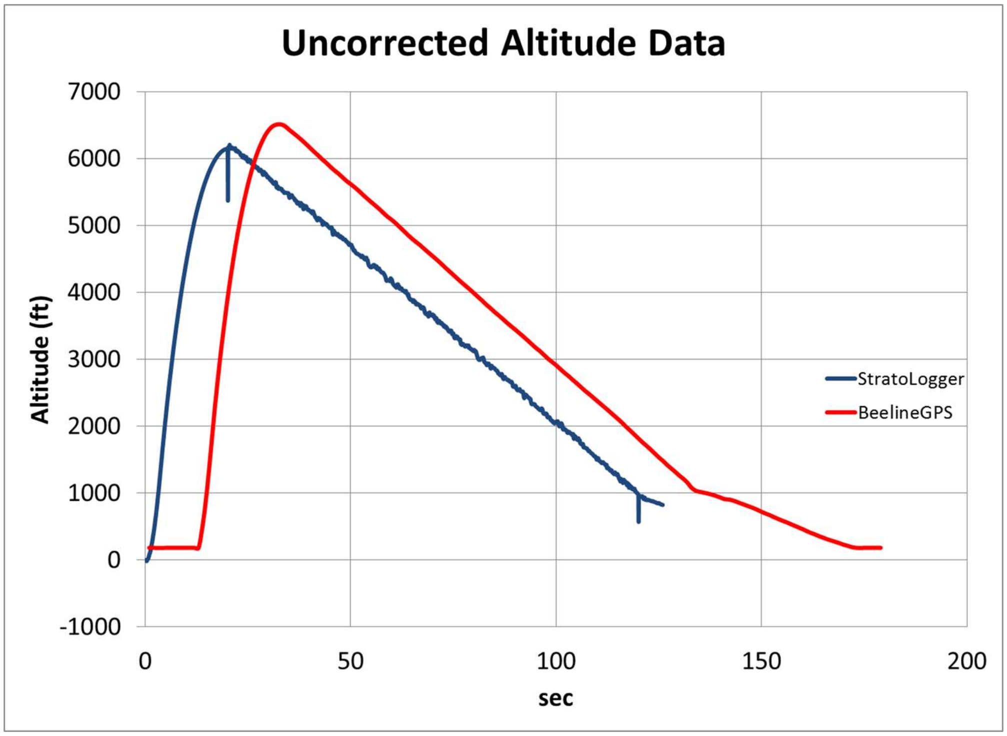

Figure 11 – Comparing raw StratoLogger and BeeLine GPS altitude plots

Several steps are required to align these two altitude plots. We will use ground level as the zero reference for the altitude data, so first, the altitude of the launch location must be subtracted from the GPS data, which is zero referenced to sea level. Looking at an average across a number of samples prior to the launch gives a number to be subtracted from all the altitude samples, 184 feet in this case. After subtracting 184 feet from all GPS altitude points, the GPS peak reading, 6328 feet, is still 172 feet greater than the peak barometric altitude reading, 6155 feet.

The next step is to align the peak altitude of the two graphs. But which unit, the BeeLine GPS, or the StratoLogger is the more accurate reference? The StratoLogger uses a barometric sensor to measure the air pressure. The altimeter automatically zeroes the sensor after turning the unit on, which corrects for the sensor offset error, but the sensor can still have a gain error. Rather than having an error that is a function of altitude, the GPS is more likely to have an offset error that is a function of the number of satellites being acquired by the GPS. Figure 12 shows a plot of the BeeLine GPS altitude data while sitting stationary on the ground. The same average ground elevation has been subtracted from all the data points so just the drift can be seen. During the flight of Speedmotion 54, 12 satellites were acquired. During this test, the number of acquired satellites ranged from 7 to 11, but never reached 12, as seen in Figure 12 on the second axis. The change in altitude is less than +-10 feet once 10 satellites have been acquired. Given the variance is much smaller than the 172 foot difference between the BeeLine and StratoLogger curves at apogee, after being aligned at ground level, I will assume the GPS is more accurate than the barometric sensor at apogee, so a correction will be applied to the StratoLogger data to align it to the GPS data.

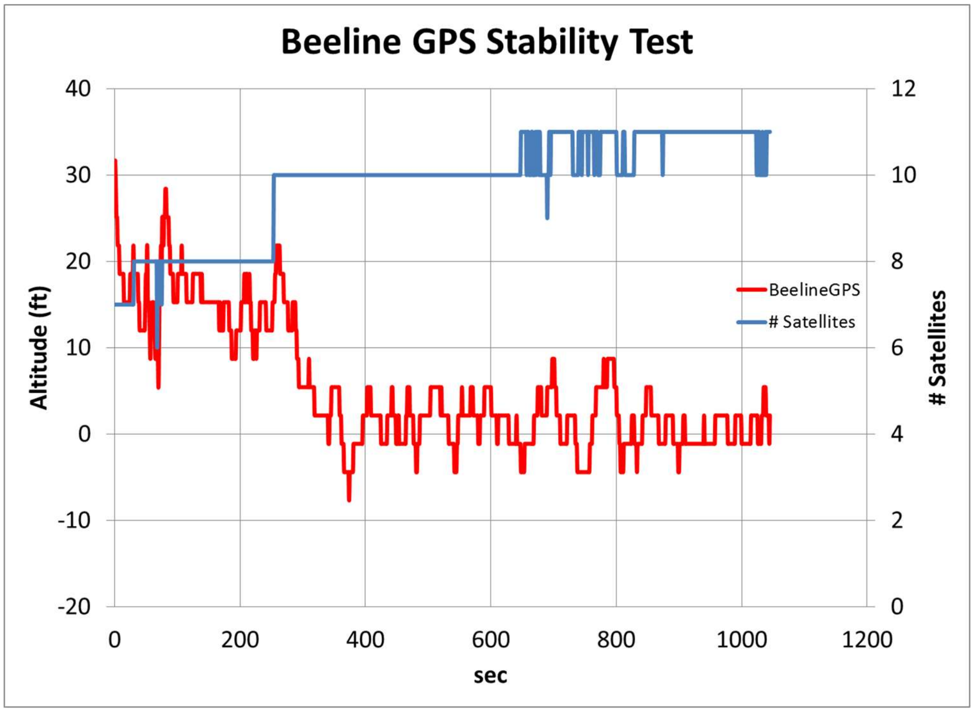

Figure 12 – BeeLine GPS stability

A gain term (a multiplier) can be added to each sample of the StratoLogger until the peak altitude of the StratoLogger matches the peak altitude reported by the GPS. A single gain number is used for all the samples. For this flight, a gain of 1.0285, which is a 2.85% error, aligns the peak altitude of the StratoLogger very closely with the peak altitude of the BeeLine.

The next step is to align the two plots in time. It is easier to do the first pass time alignment after the gain correction term has been added. The StratoLogger data starts at the launch of the rocket. Its higher sample rate means it has good resolution of the actual time of the launch, so the StratoLogger will be used as the time reference. A time offset is added to each sample of the BeeLine data to find a best fit between the two plots. When I trimmed lines of the BeeLine data file after importing it into Excel, I chose to start the table about 10-20 samples ahead of the launch, just to have enough data to establish an average reference for the launch altitude. Note, the altitude readings ahead of the launch are all very stable at 184 feet. The amount of time correction necessary depends on the number of samples included ahead of the launch. For this case, a time offset of about -12 seconds aligns the two plots very closely. With a 1 second sample rate, there can be a delay in the GPS altitude reading during the ascent when the rate of change in the altitude is large. The GPS is probably the most accurate in the region where the rocket is moving slowly, on the way down under parachute, so I align the plots on the descent side of the flight (see Figure 14).

Figure 13 shows the results of applying the ground level offset and the time shift to the GPS data, and the gain term to the StratoLogger data. The process of selecting the best GPS time shift and

StratoLogger gain is a iterative process of adjusting each number until a combination is determined that gives a very close fit between the curves. The fact that the data points now align over the entire flight path indicates that modeling the altimeter error as a simple gain term is a good assumption after applying the ground offset. For higher altitude flights, multiple gain corrections terms for the altimeter might be required. Note that at the end of the flight, the altitude GPS readings are right back to 184 feet, so the GPS was very stable over the entire flight.

Figure 14 shows a close look at the peak of the flight. The actual data points are shown, and the difference between the sample rates is very clear. Also, at first look, the StratoLogger data appears noisier after the drogue chute ejects. But looking closer, the data has a sinusoidal component that is not random data noise. Expanding the scale further, in Figure 15, the peak swings in the altitude data are around plus and minus 20-30 feet about the average with a period of about 1 second. What's causing the oscillation in the altitude data? Speedmotion 54 has 15 feet of shock chord plus the rocket body length between the drogue chute and the altimeter bay. The peak swings of up to about 20-30 feet could be explained by the rocket swinging fully around the drogue chute in a vertical plane. But the period of the sinusoidal pattern in Figure 15 is around 1 second, which is likely too fast to be caused by the rocket's swinging around its parachute over a 15 foot shock cord length. Observing rockets of this size as they descend, the period of a complete swing cycle is closer to 5-10 seconds. Something else must be causing the apparent altitude oscillation. We will revisit this after looking at the calculated velocity and acceleration. The GPS data has a much slower sample rate and is much smoother, so it does not show the oscillation, but instead, shows the average altitude.

Also, from Figure 14, a small delay in the GPS data can be seen on the ascent portion of the flight, after the descent portion has been time aligned. The time delay is mostly constant during the rapid portion of the ascent, but the difference does converge as the rocket slows as it approaches apogee. Figure 16 shows the detail about the middle of the ascent. With the descent aligned as closely as possible, the GPS delay on the ascent can be seen to be a little less than 0.2 sec, which is well less than one sample period of the BeeLine.

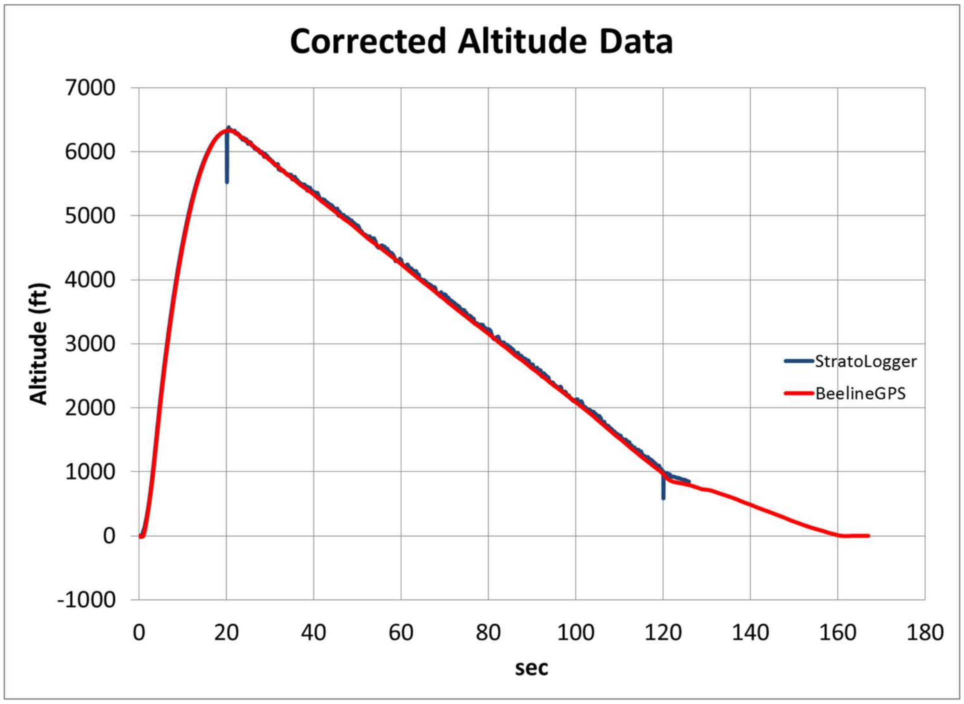

Figure 13 – Corrected altitude data

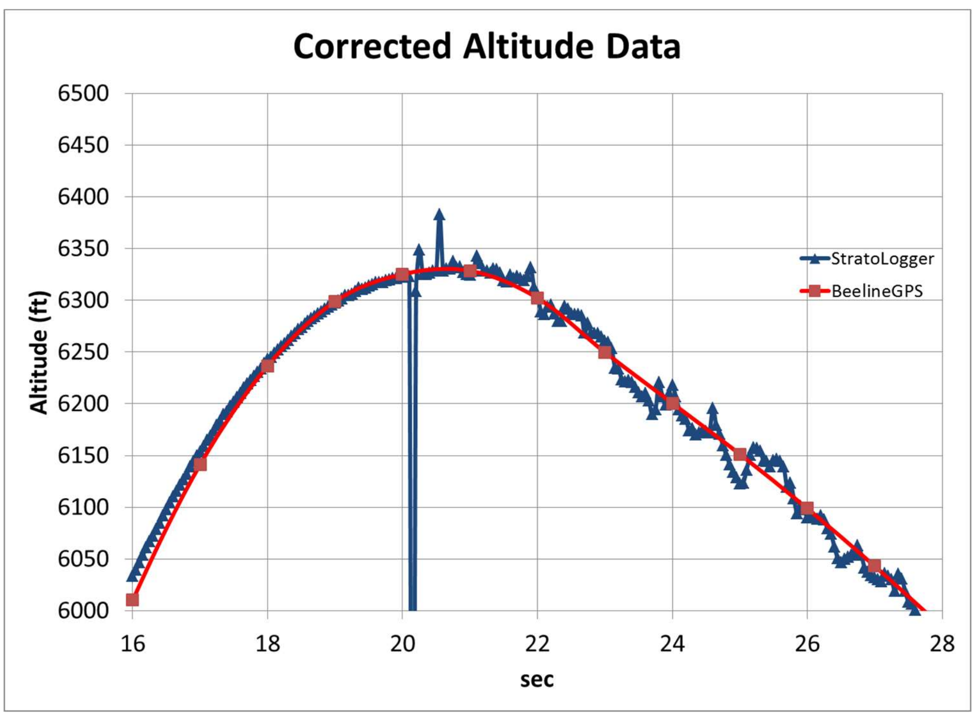

Figure 14 – Corrected altitude data at peak altitude

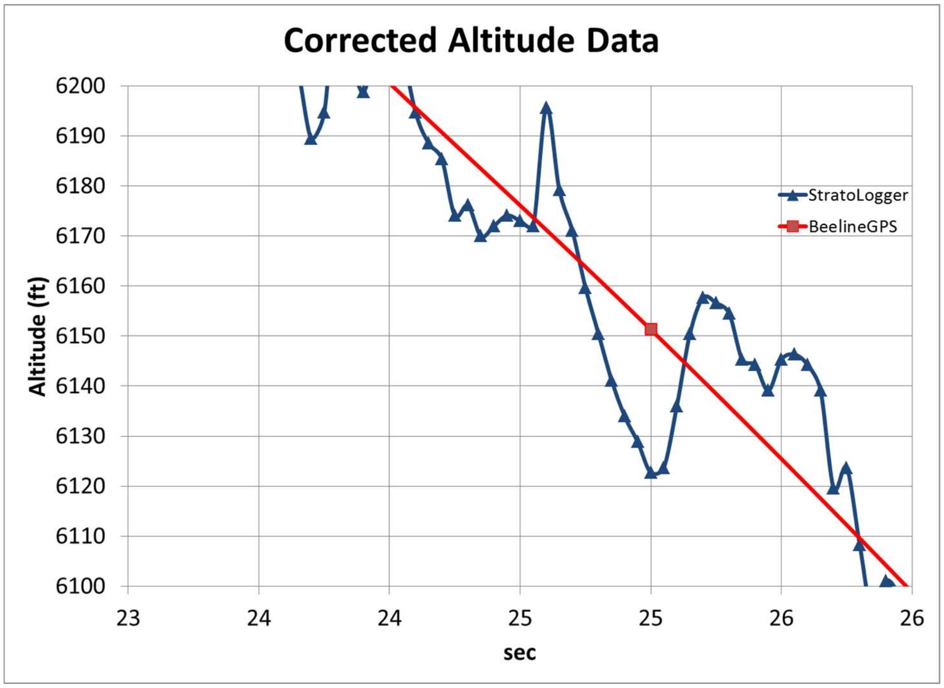

Figure 15 – Corrected altitude data on descent

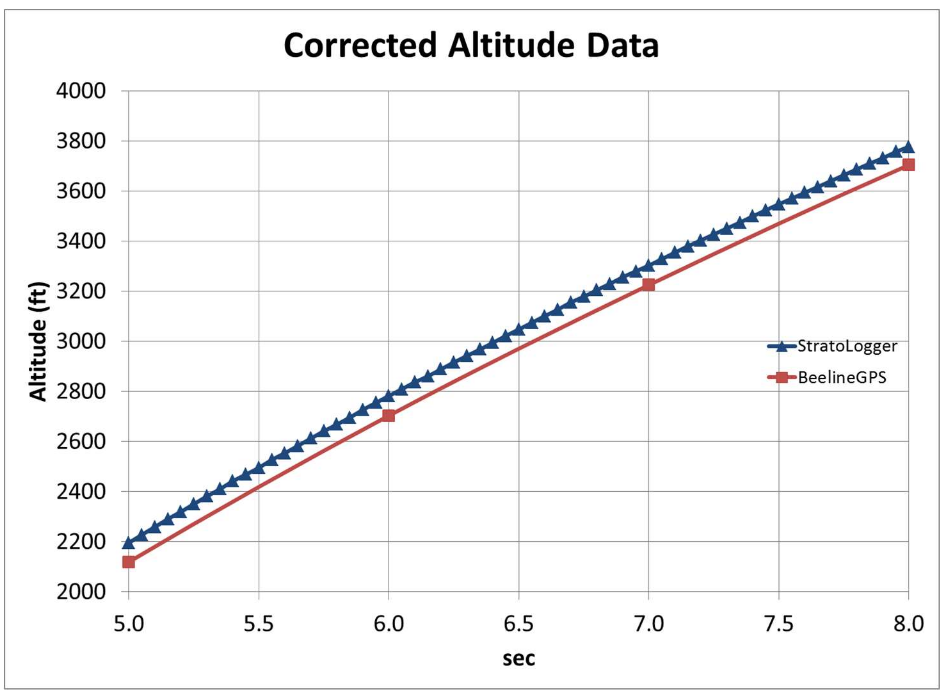

Figure 16- Corected altiude data on ascent

As mentioned earlier, this GPS data is taken from one of the new BeeLine GPS units introduced within this past year. The new BeeLine units do a much better job of maintaining lock as the rocket is first launched. The older units would typically loose lock during most of the powered ascent and miss the first 5-10 data points for a flight with an initial acceleration of around 10 g's. Figure 17 shows a data capture from an earlier BeeLine GPS unit. The data has been aligned so that t=0 is the actual start of flight. It can be seen that the GPS missed the first 10 data points, or 10 seconds of the flight before logging the first valid point at over 6000 feet. But as can be seen in Figure 9 (Part 1), not a single data point is lost on data captured using one of the new units.

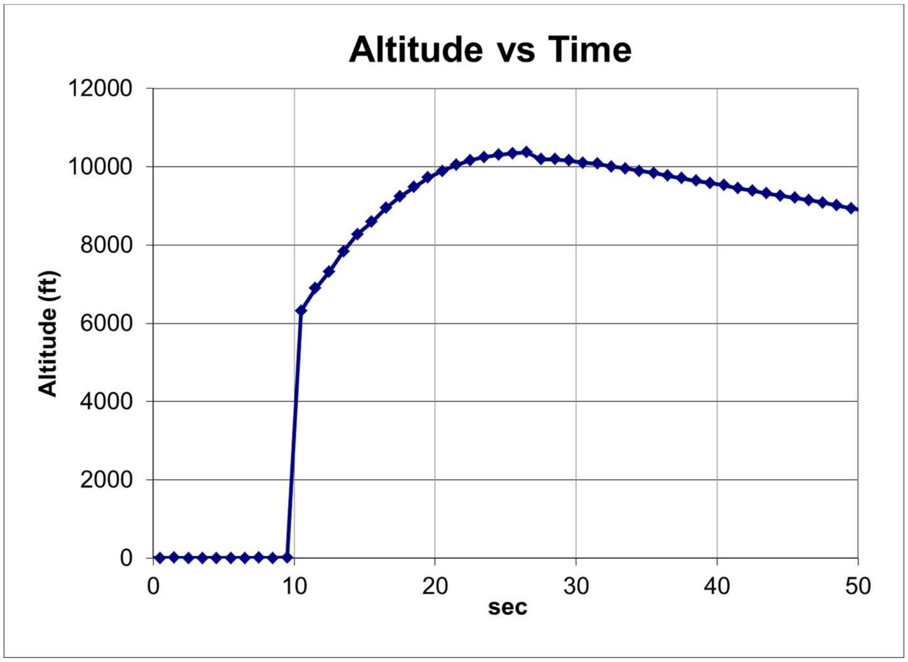

Figure 17 – Loss of lock at launch for earlier BeeLine GPS unit

Adding the Position Data

So far, I've only talked about altitude data. But the objective is to make a complete dimensioned plot of the entire flight trajectory, including the distance traveled east-west and north-south, so the latitude and longitude data from the GPS must be converted into units of distance, either feet or meters, to do this.

Latitude is the distance north-south of the equator. The distance due to latitude lies along circumferential lines that intersect at the north and south poles of the earth. Making the assumption that the earth is a perfect sphere, which is good enough for our calculations, then a degree of latitude is always the same linear distance anywhere on the earth's surface.

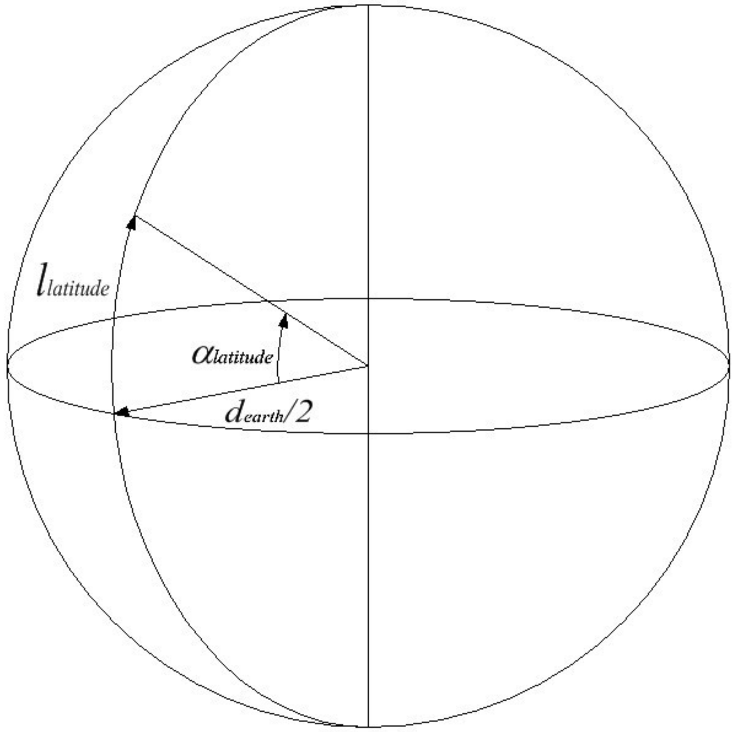

Figure 18 – Calculating latitude distance

From Figure 18, the distance along the circumference of a circle is given by:

$$l_{latitude} = \frac{d_{earth}}{2} * \alpha \tag{1}$$

where α is in radians. Given the diameter of the earth at the equator is 7926.4 miles and

$$1^\circ = (2\pi / 360)\, radians \tag{2}$$

solving Equation 1,

$$l_{latitude} = 69.1708 * \alpha_{latitude} \tag{3}$$

where $l_{latitude}$ is in miles, and $\alpha_{latitude}$ is in degrees, or, given 1 mile = 5280 feet

$$l_{latitude} = 365222.0 * \alpha_{latitude} \tag{4}$$

where $l_{latitude}$ is in feet, and $\alpha_{latitude}$ is in degrees.

Longitude is the distance east-west of the prime meridian. Latitude is measured along lines that are parallel to the equator. Since the circles become smaller as the latitude approaches the poles, a degree of longitude represents a shorter linear distance as the latitude increases.

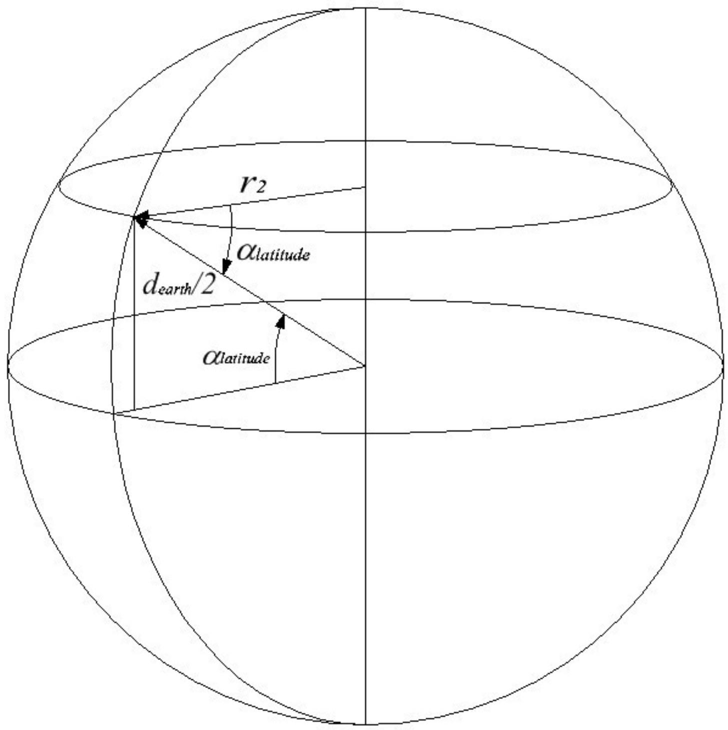

Figure 19 – Calculating earth radius at latitude

From Figure 19

$$r_2 = \frac{d_{earth}}{2} * \cos(latitude) \tag{5}$$

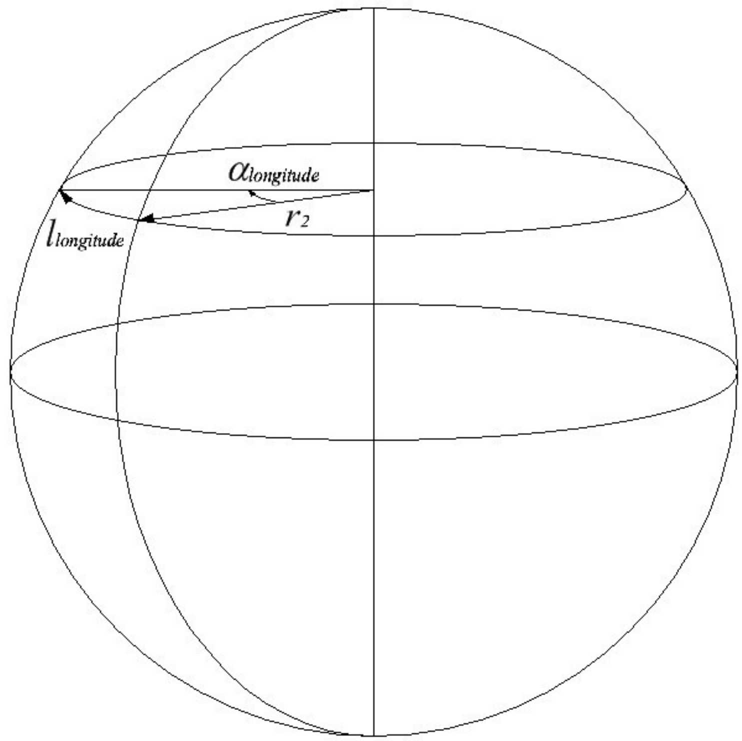

Figure 20 – Calculating longitude distance

Once the radius of the longitude circle is known, then the distance along the circumference of that circle can be calculated just at the distance along the latitude circle was calculated.

From Figure 20

$$l_{longitude} = r_2 * \alpha_{longitude} = \frac{d_{earth}}{2} * \cos(\alpha_{latitude} * \frac{2\pi}{360}) * \alpha_{longitude} \quad (6)$$

or

$$l_{longitude} = 365217.6 * \alpha_{longitude} * \cos(\alpha_{latitude} * \frac{2\pi}{360}) \quad (7)$$

where $l_{longitude}$ is in feet, and $\alpha_{latitude}$ and $\alpha_{longitude}$ are in degrees.

With equations (3) and (7), the GPS coordinates can be converted to north-south (latitude) and east-west (longitude) distance.

The following figures show different plots of the rocket's 3-dimensional trajectory. They are based the GPS data corrected for time and ground offset.

Figure 21 shows a plot of the trajectory from above, with the axes being north-south and east-west. This is the same projection as the view from Google Earth when looking straight down, but with dimensioned axes. It is much easier here to see exactly how far the rocket has traveled. The rocket moved to the north-east during the ascent and was around 1000 feet from the range at its peak altitude. It then drifted nearly 3000 feet to the south-east on its way back down.

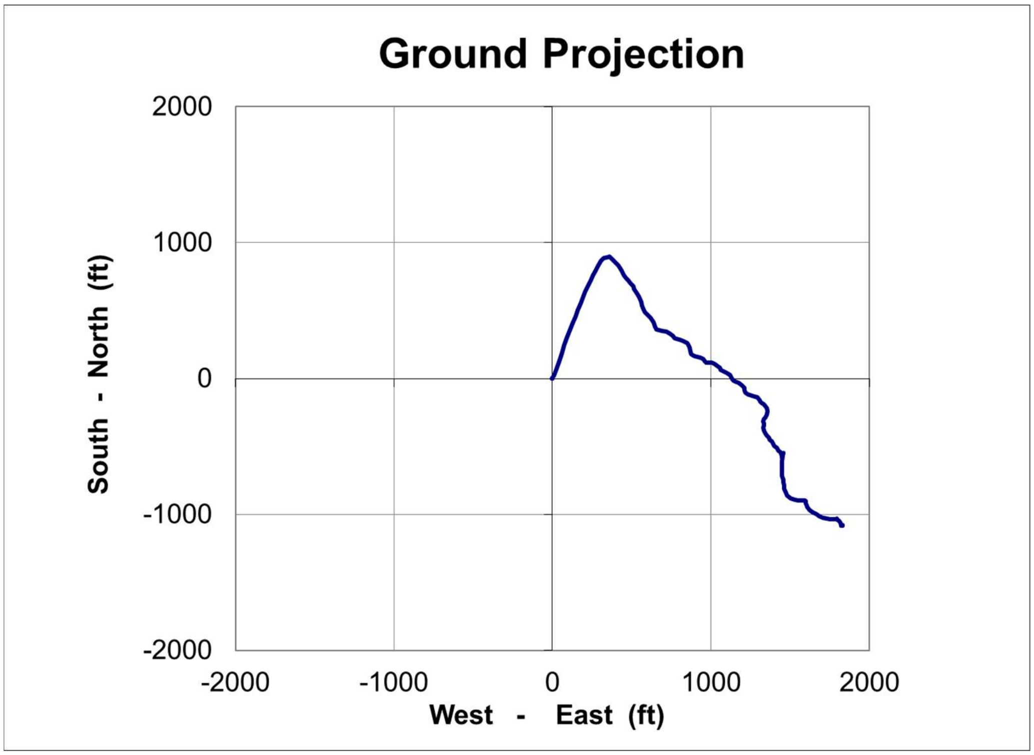

Figure 21 – Ground projection of flight path

From Figure 21, from the descent portion of the flight, starting at the northern most point, the wind direction was very consistently from the north-west, with just a short duration swing from the north north-east, as seen by the rocket moving to the south south-west, in the middle of the descent. The wind must also have been from the north north-east at launch as the rocket moves toward the north north-east after launch. Either the wind at ground level was coming from the north north-east during the first second of the flight when the rocket is moving slowly and most of its turn into the wind occurs, or the launch rail flexed at launch, causing the turn.

Figures 22 and 23 show altitude versus north-south and east-west cross sections of the flight. It is helpful to scale the x any y axes to roughly the same scale to keep the distance versus distance plots in proper perspective.

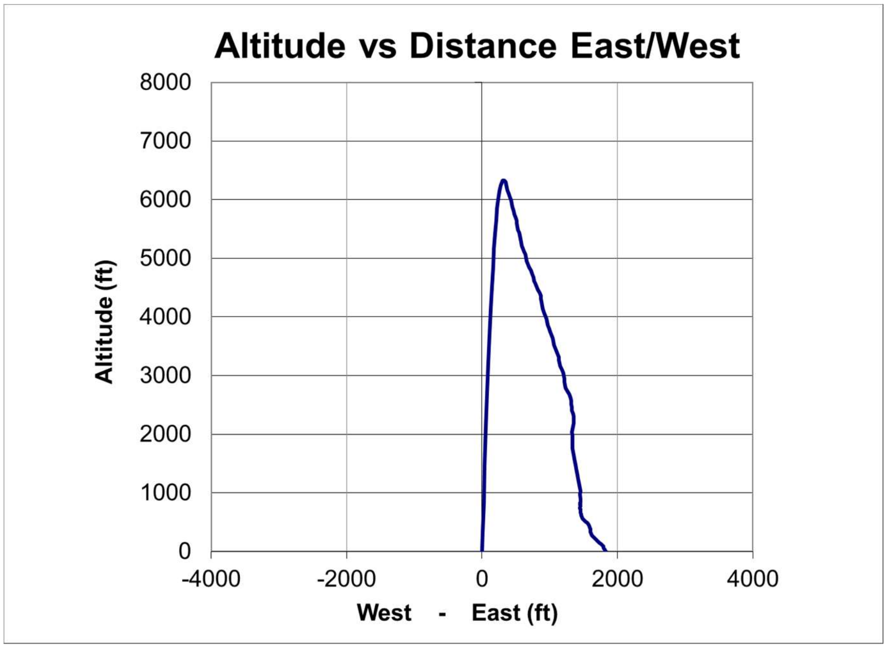

Figure 22 – East/west slice of flight path

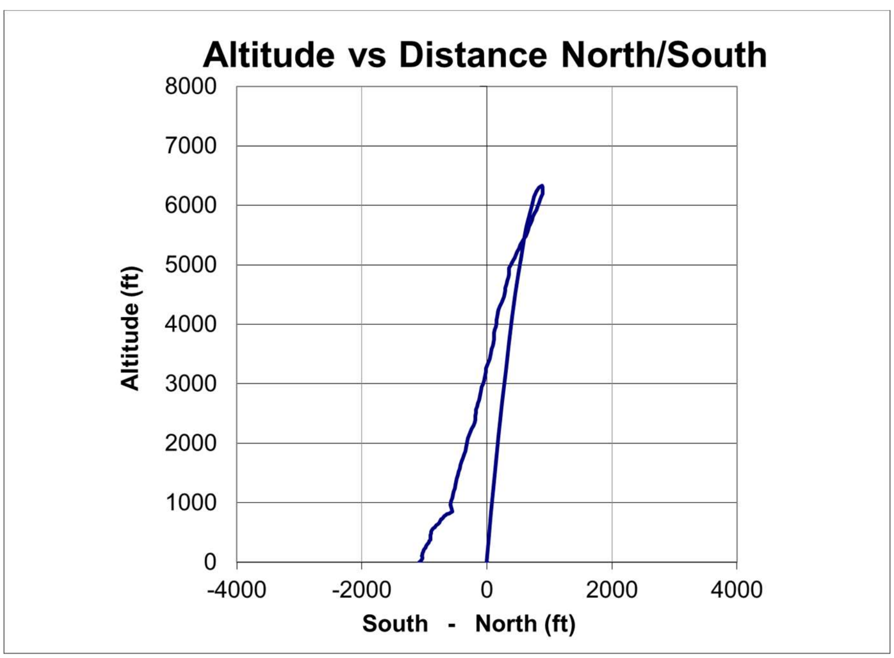

Figure 23 – North/south slice of flight path

The horizontal distances can also be plotted versus time to show the rate at which the distance was covered. Figures 24 and 25 show the north-south and east-west distance versus time, much like the altitude versus time plot. Note that a negative north distance is south of the launch point. Since the rocket drifted nearly directly south-east, the slopes of these two plots are about equal. The data points on these graphs are at equal time intervals and quite dense, so I have turned off the data point markers in these plots to allow seeing the data more clearly.

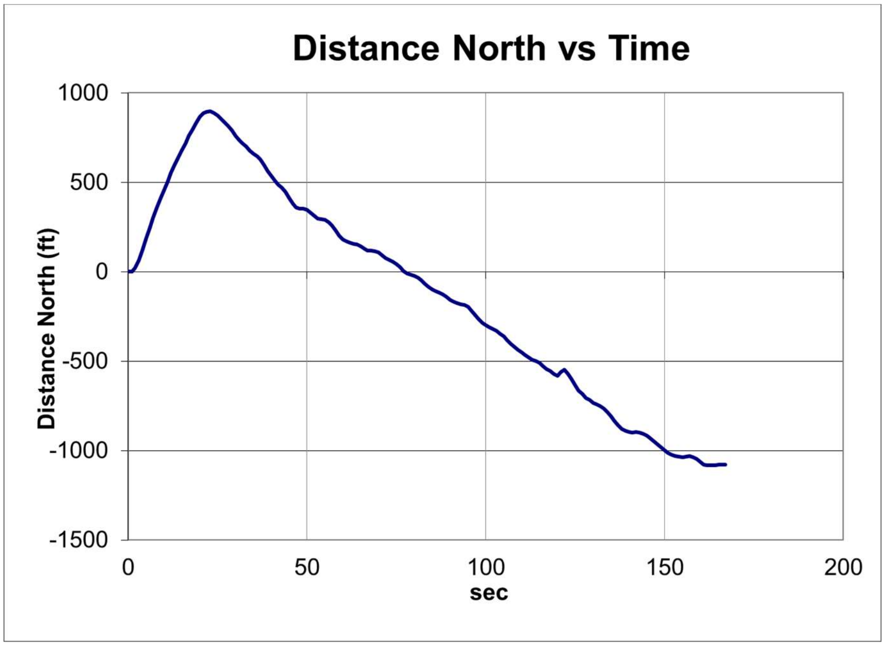

Figure 24 – North/south vs time

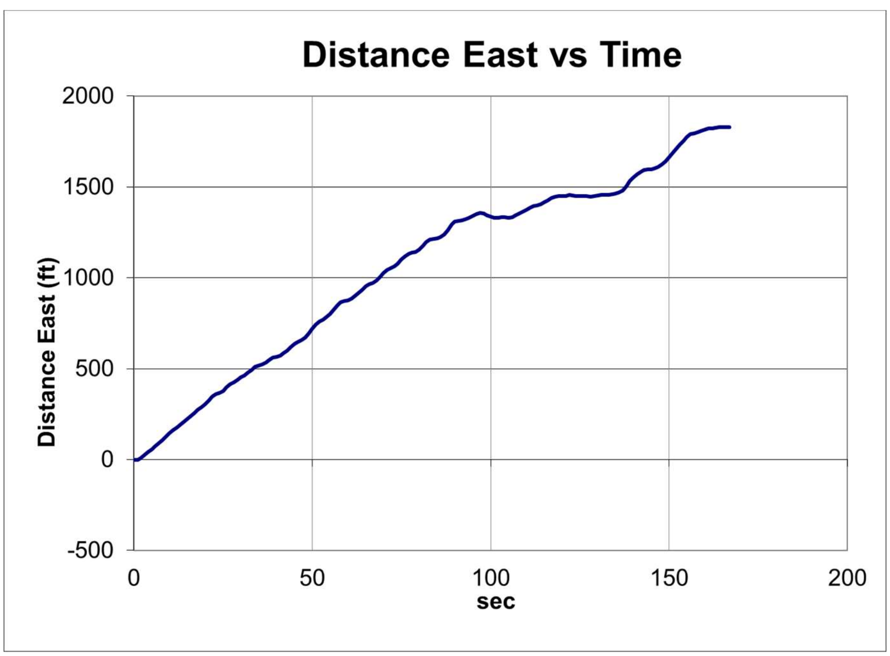

Figure 25 – East/west vs time

Useful References

PerfectFlite DataCap: http://www.perfectflite.com/Download.html
BeeLine GPS Data Communicator: http://www.bigredbee.com/BeeLineGPS.htm
RockSim: http://www.apogeerockets.com/Rocket_Software
My Excel spreadsheet: http://www.Speedmotionrockets.com

Flight Data Analysis Using Excel - Part 3

Thomas B. Fetter
NAR 15551

Calculating Velocity from Altimeter Data – Part 1

Neither the BeeLine GPS nor the StratoLogger record velocity data directly, but the velocity can be calculated from the altitude data. The velocity is simply the rate of change of the altitude versus time, or the slope of the altitude curve. Mathematically, velocity is the derivative of the distance traveled. For a discrete time system, the velocity can be calculated by taking the difference of two adjacent altitude data points and dividing by the difference in time between the two points.

$$\frac{\Delta d}{\Delta t} = v \tag{8}$$

or

$$\frac{d_n - d_{n-1}}{t_n - t_{n-1}} = v_n \tag{9}$$

Figure 26 shows the result of calculating the velocity using adjacent altitude and time data points. The ascent portion of the flight is visible, but when apogee is reached at 20 seconds and the descent begins under parachute, the data becomes extremely noisy, and the average descent velocity is very hard to see. It is unlikely that the rocket is actually moving in short bursts of velocity approaching 300 mph on descent as shown in this graph, so this is data noise on top of the actual motion of the rocket. Why is the data so noisy?

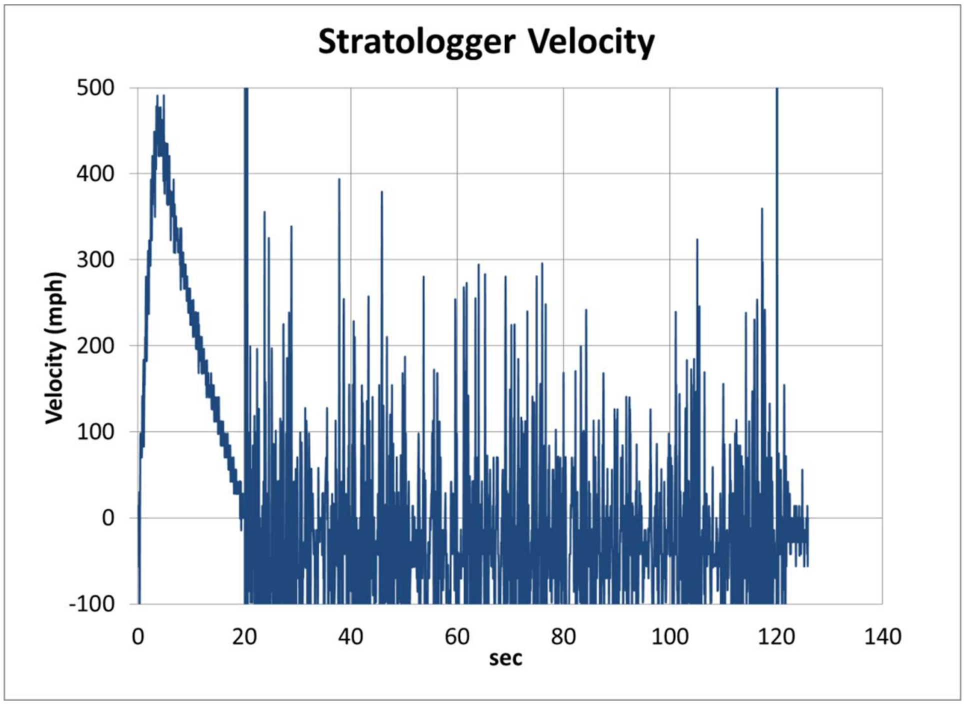

Figure 26 – Raw velocity calculated from altitude data

The PerfectFlite DataCap software calculates the velocity data. The software can be set to plot the velocity data, but the velocity data is included when the StratoLogger is saved to a file from DataCap. Figure 27 shows a plot of the velocity calculated by DataCap. It is clearly much less noisy than Figure 26. What is different about the way DataCap calculates the velocity that producers such a clean trace?

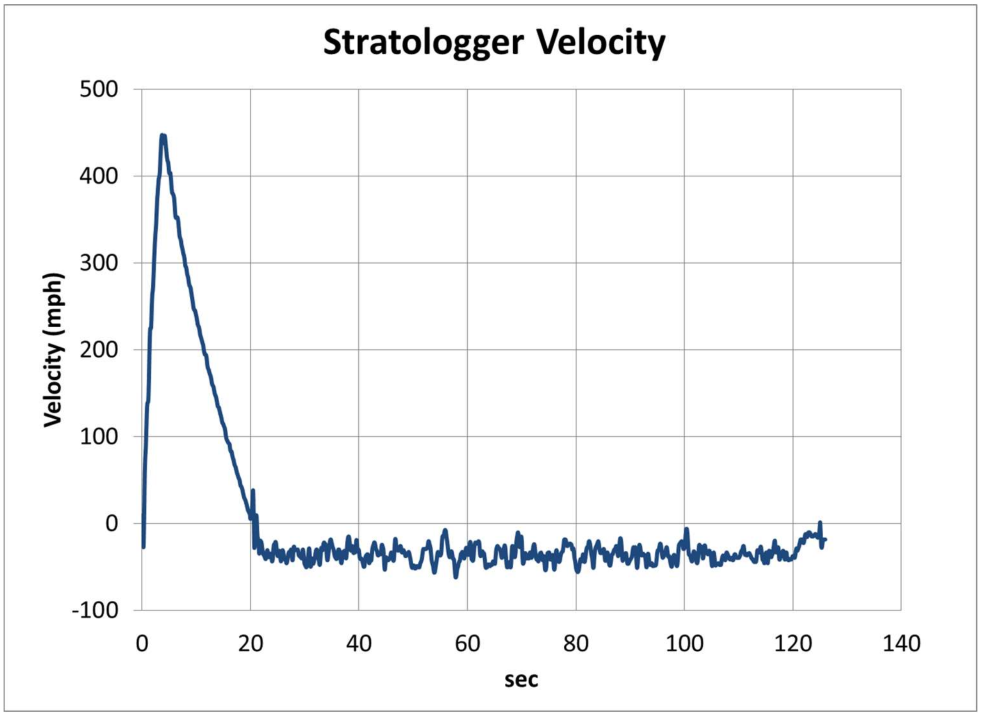

Figure 27 – Velocity data from PerfectFlite DataCap

Since the velocity is the slope of the altitude data, a rapid change in altitude, even over a very small change in altitude, will result in a very high calculated velocity. If the rapid change in altitude occurs over a very small change in altitude, then the high velocity will only last for a very short time. This is exactly what is seen in Figure 26, a series of very short bursts of very high velocity.

A small amount of noise in the altitude data can cause very large spikes in the velocity trace. For example, a change in altitude of 20 feet from one 50 ms sample to the next, as can be seen in one interval in Figure 15 (Part 2), would result in a velocity spike of 400 fps, or 273 mph. Calculating velocity from the altitude by taking the derivative accentuates the noise in the altitude data. It is unlikely the rocket is actually going from 0 to 400 fps from one 50 ms sample to the next. The altitude noise can come from several sources. It can be electrical noise in the analog portion of the barometric sensor electronics, it can be quantization noise in the A/D converter, or it could due to actual changes in pressure measured by the altimeter due to something other than actual changes in altitude.

Looking at the altitude data in Figure 15, there is very little random variation from sample to sample. Rather, there are rapid swings in altitude that continue in the same direction in a sinusoidal pattern over a period of a number of samples. Electrical or quantization noise would have a truly random pattern sample to sample, so this noise must be due to actual measured pressure variations caused by some external phenomenon. Previously, when looking at the altitude variations in Figure 15, we said it was possible that +-20 foot swings about the average altitude could be caused by the rocket swinging under chute based just on the magnitude of the swings, but the 1 second period of the oscillation was too rapid to be caused by swinging alone. Now, examining the rate of change of the altitude swings, or the velocity, which approach +- 300 fps, many times the average descent rate of around 50 fps, it is clear these variations are not due to swinging around the parachute. Instead, these variations are more likely caused a change in air speed as the altimeter bay swings and rotates as it moves through the air and the orientation of the vent hole changes. In addition to changes in pressure caused by a change in altitude, the rate of change of the air blowing over the vent hole can impact the pressure in the altimeter bay. Looking at Figure 14 (Part 2), the large altitude variations begin after the deployment of the parachute. Before deployment, the orientation of the vent hole is not changing rapidly. After deployment, the bay could be swinging and rotating, and the changes in the orientation of the vent hole could be causing changes in air velocity over the hole, which, in turn, would cause changes in pressure in the altimeter bay. The altimeter bay could easily be changing orientation at a rate of once per second. This is the most plausible explanation for the noisy velocity readings. Given this 1 second variation does not represent true altitude variations of the rocket's descent, it will have to be removed from the data to enable seeing the actual underlying velocity of the rocket.

Given the velocity data is very noisy, smoothing, or averaging the data will enable seeing the actual velocity more clearly. A simple technique for smoothing is to take a running average of the data. For example, after calculating the instantaneous velocity using Equation 9, a new velocity data point is created by averaging the current data point with several adjacent velocity data points. This has the effect of reducing the sample to sample variance and smoothing the trace. Figure 28 shows the effect of averaging 5 total velocity points, the center point plus two adjacent points on either side. The resulting trace is now a little smoother, but still not smooth enough to see the average velocity. Increasing the number of adjacent points included in the average provides additional smoothing. Figure 29 shows the result of including 31 points, the center point plus 15 points on either side. The resulting trace is much smoother, and the deviation is similar to the StratoLogger calculated velocity plot in Figure 27. But the disadvantage of including a larger number of points on either side trace is the average applies to the actual changes in rocket velocity as well as the noise. The information about the peak velocity, which occurs over a very small number of data points, is now lost as those peak velocity data points are averaged in with a large number of lower velocity adjacent data points. It is clear the amount of smoothing needs to be a balance between the amount of noise reduction required and the need to respond to actual rapid changes in the velocity.

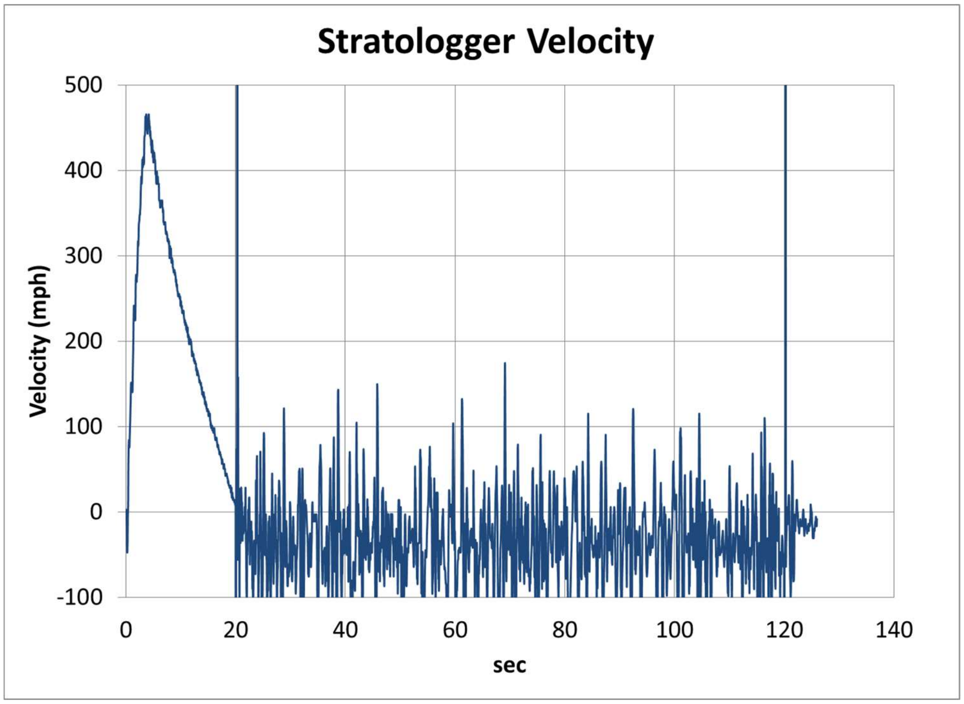

Figure 28 – Calculated velocity with 5 point running average

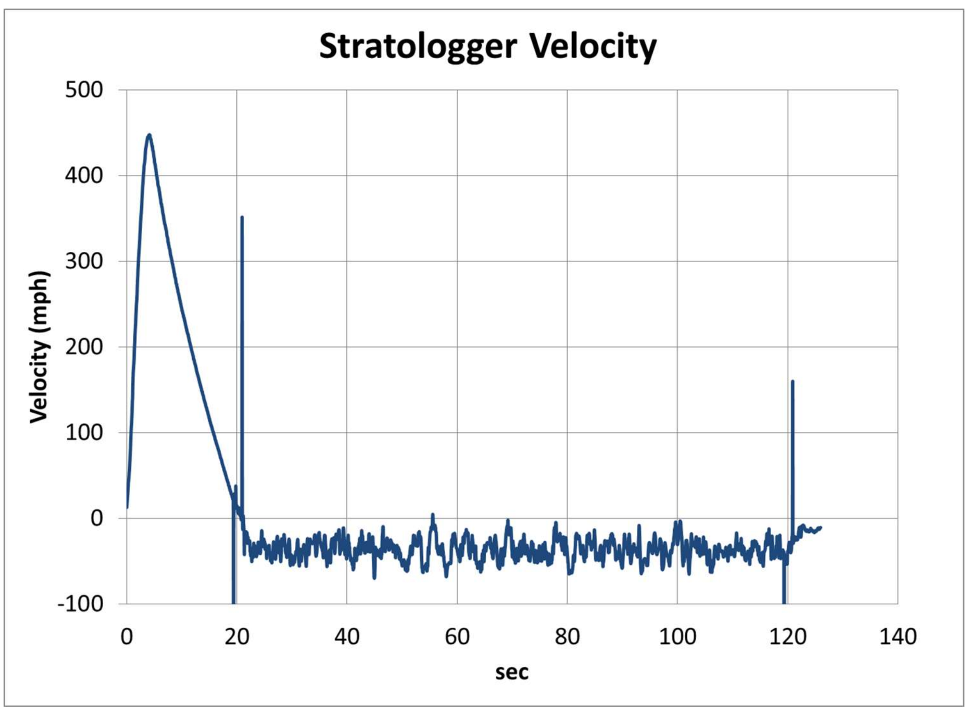

Figure 29 – Calculated velocity with 31 point moving average

Data Smoothing

A running average smoothing function is a form of a low pass filter; it passes the desired signals that are changing slowly, or low frequency signals, unchanged, and attenuates the undesired rapidly changing, high frequency signals.   There are other filter functions that could be used to smooth the data, but a moving average filter is very simple to implement in an Excel spreadsheet, and it works quite well.

Increasing the number of adjacent data points used to smooth the curve reduces the bandwidth of the filter, or ability to see rapid changes in the signal, but attenuates more of the higher frequency noise. To understand how to choose the number of samples to include in the averaging window to optimize the running average filter to reduce noise while retaining the integrity of the actual signal requires an analytical understanding the filter's behavior.

The sample rate of the signal also has an effect on the frequency response of the system. If the data sample rate is 50 ms, as is the case for the StratoLogger, then the trace responds to changes that are longer than 50 ms, but cannot respond accurately to changes that are less than 50 ms. More accurately, the sample rate must be at least twice the highest frequency of the data represented in the data trace. This is called the Nyquist rate. Above the Nyquist rate, the signal is sampled less than twice per sinusoidal cycle, and the signal will be moved, or "aliased", to a differ frequency. For the StratoLogger, the Nyquist rate is 10 Hz (1/50ms/2 = 10 Hz). We will be setting the bandwidth of the smoothing filter well below the Nyquist rate.

Figure 30 shows the frequency response of a constant amplitude sine wave that is swept in frequency while being sampled at 50ms with an 11 point moving average filter applied. This plot is made by first creating a sinusoidal signal data stream at a number of test frequencies. The running average filter is applied to the data stream at each frequency. The magnitude of the amplitude of the resulting sinusoidal signal is plotted versus the frequency of each test signal. The calculation was done using a visual mathematical programming application called Mathcad.   The plot shows the log of the amplitude response versus the log of the frequency of the applied signal. Log scales are used to show the response over a wide dynamic range of frequency and amplitude. Note that the passband is flat, and has a gain of 1, so it will pass all signals in that frequency range un-attenuated. The stopband has an attenuation of roughly 10, and starts to roll off around 0.8 Hz, with a first null at 1.8 Hz, and then another null spaced every 1.8 Hz from there. An eleven data point window is 0.55 sec long. 1/0.55 sec is 1.8 Hz, so the nulls are spaced at 1/window length.

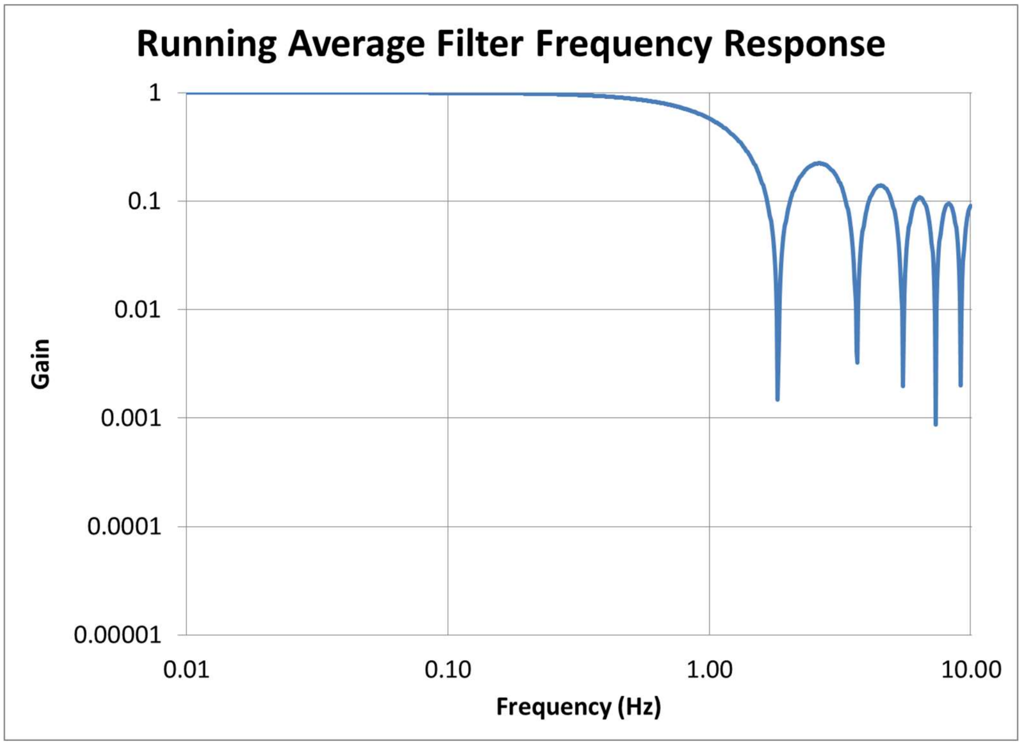

Figure 30 – Frequency response of moving average filter 11 samples in length

Figure 31 shows the frequency response of two running average filters, one, with a window of 5 data points, and a second, with a window of 51 data points, both applied to a data stream with a 50ms sample rate. A five data point window is 0.25 sec long. 1/0.25 sec is 4 Hz, and first null in the frequency response occurs at 4 Hz. Likewise, the 55 data point window is 2.5 seconds long. 1/0.5 sec is 0.4 Hz, and the first null for this filter is at 0.4 Hz. Changing the length of the averaging window changes the cutoff frequency of the filter, but the attenuation of the stopband always remains the same. If a factor of 10 is not enough of a reduction of the higher frequency signals, then another means to achieve further reduction must be employed.

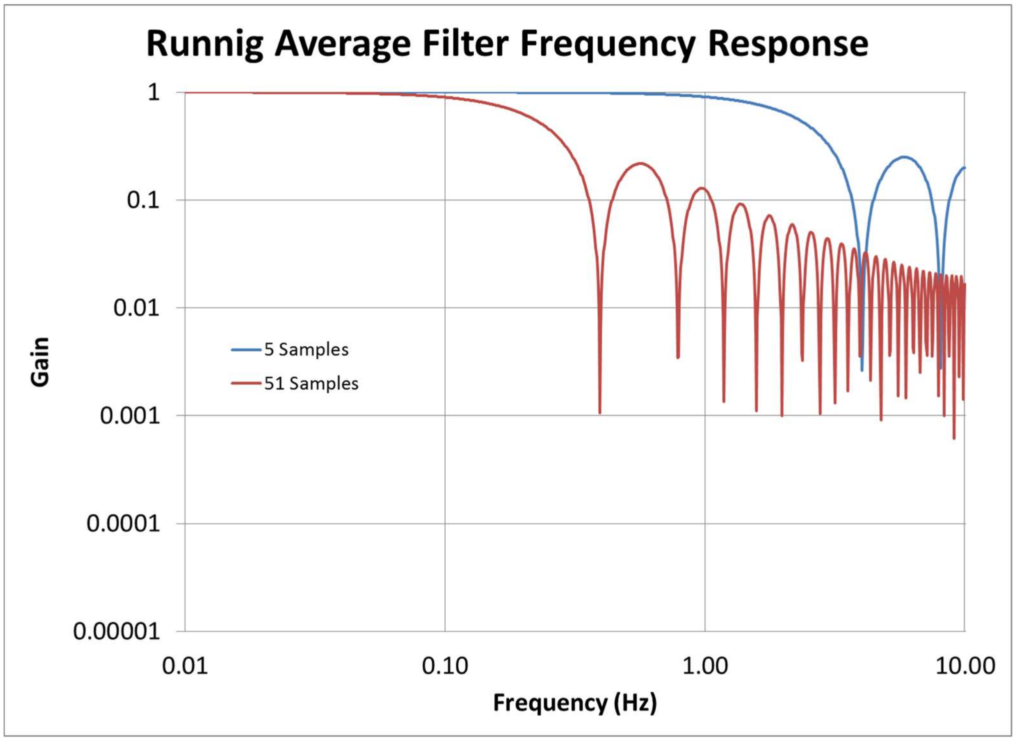

Figure 31 – Frequency response of single pass moving average filter 5 and 51 samples in length

If additional attenuation is required in the stopband, then multiple passes of the moving average filter can be applied to the data stream. Figure 32 shows the results of applying the same averaging window successively, multiple times. Each pass increases the stopband attenuation by an additional factor of 10. Each pass also causes some lowering of the roll-off frequency, so the number of samples in the average window would have to be reduced to maintain the same net bandwidth as more passes of the filter are added.

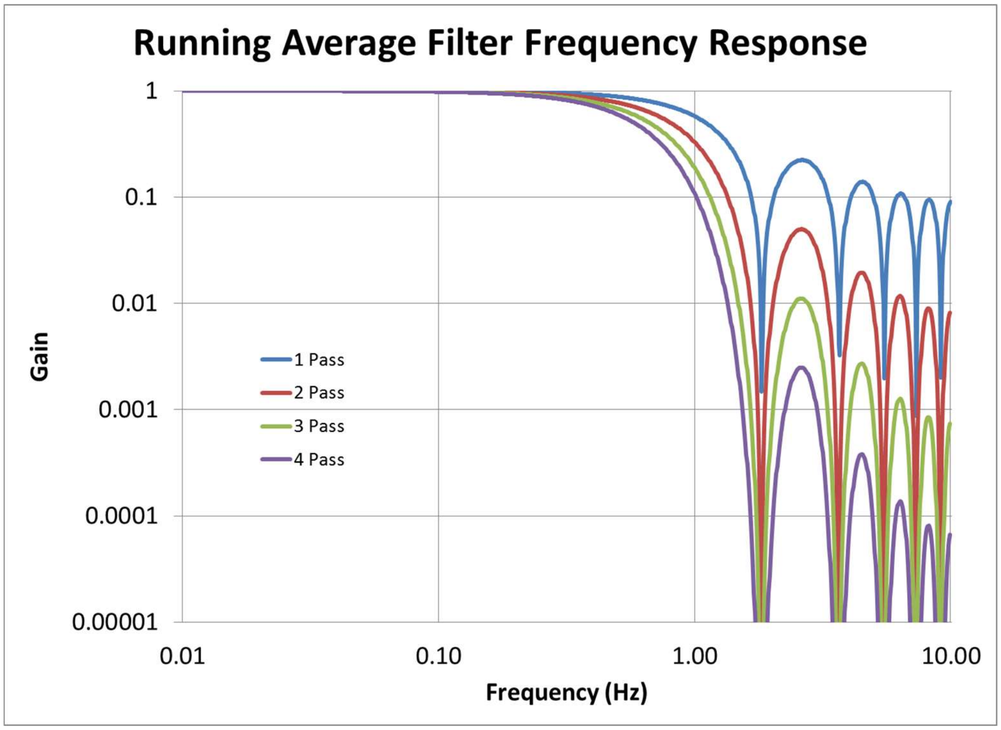

Figure 32 – Frequency response of 1, 2, 3, & 4 pass moving average filters 11 samples in length

It is easy to construct a multi-pass filter in an Excel spreadsheet. The first column of data is the raw, unfiltered data. The second column is constructed by creating a new value for each row that is the average of the same row in the previous column, and as many additional values from rows on either side of the center row as desired, for example, the prior and next two rows for a 5 point filter. The second pass of the filter is constructed in the next column, using the same method of filtering but using the values from the first pass filter column. Then the third pass filter is constructed in the next column in the same way. Figure 33 shows the columns of the spreadsheet with the original raw data and the three passes of a 5 point moving average filter.

Cal Vel (mph)	Cal Vel Avg^1	Cal Vel Avg^2	Cal Vel Avg^3
	2.81	-1.68	-3.03
14.03	0.00	-2.81	-7.18
0.00	-11.22	-10.66	-12.79
-14.03	-5.61	-20.76	-17.84
-56.10	-39.27	-28.05	-22.10
28.05	-47.69	-26.93	-21.09
-154.28	-36.47	-24.12	-13.13
-42.08	-5.61	-5.61	1.12
42.08	8.42	19.07	18.51
98.18	53.30	43.20	38.04
98.18	75.74	60.03	55.31
70.12	84.15	73.49	68.67
70.13	78.54	80.78	78.32
84.15	75.74	85.83	86.51
70.13	89.76	91.44	94.70
84.15	100.98	100.98	103.90
140.25	112.20	114.44	113.77
126.23	126.23	126.79	123.64
140.25	143.06	135.20	132.28
140.25	151.47	140.81	138.12
168.30	143.06	144.18	141.04
182.33	140.25	143.62	143.17
84.15	143.06	141.37	145.75
126.23	140.25	145.86	149.90
154.28	140.25	153.71	157.53
154.28	165.50	164.93	169.65

Figure 33 – 5 point three pass filter example

So, to design a moving average filter, first the desired stopband attenuation is selected, which sets the number of passes, or applications of the filter window. For example, for a stopband attenuation of 100, two passes would be required. Then, the roll-off frequency is selected. For example, if the data contains useful information that changes at rates less than 1 second, or up to 1Hz, a 1 Hz roll-off frequency would be selected. The number samples in the window is then chosen to match the desired cut-off frequency for the given number of passes. Note that the number of samples in the averaging window will be different for the same roll-off frequency for a different number of passes of the filter.

It is also helpful to look at the time domain response of the moving average filter. The time domain response shows how quickly the filter responds to a rapid change in the input data. The longer the average, the more slowly the filter will respond. A typical way of measuring the time domain response is to apply a step to the input, and compare the output response over time. Figure 34 shows input step, and response of the .55 sec window 1, 2 and 3 pass filters (11 sample at a 50 ms sample rate) applied to the step input. Note that the filter response actually precedes the input step in time. This is because the filter window is centered about the data point that represents the average. As the filter center moves toward the step, the average starts to include the input data beyond the step transition for several time increments (half the filter width for a single pass filter) ahead of the filter being centered on the actual step transition. For a single pass, the data reaches full amplitude at half the filter window (.275 sec) beyond the step transition. For multiple passes, the signal is spread on either side of the step transition even further.

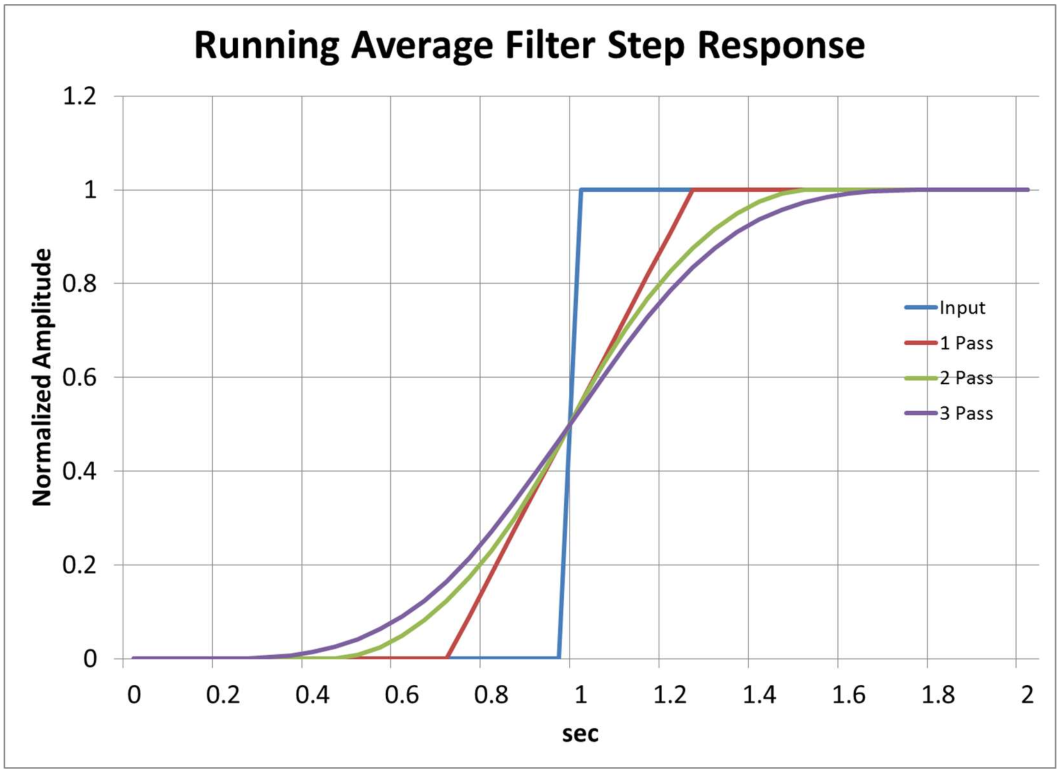

Figure 34 – Step response of 1-3 Pass 11 sample filter

Since it is difficult to plot the frequency response of the filter in Excel, the method I use to select the window length is to look at the actual velocity plot and change the length of the average by trial and error, balancing noise reduction with the response to rapid changes in actual velocity. But it has been helpful to examine the characteristics of the moving average filter to better understand how to apply it.

Useful References

PerfectFlite DataCap: http://www.perfectflite.com/Download.html
BeeLine GPS Data Communicator: http://www.bigredbee.com/BeeLineGPS.htm
RockSim: http://www.apogeerockets.com/Rocket_Software
My Excel spreadsheet: http://www.Speedmotionrockets.com

Flight Data Analysis Using Excel - Part 4

Thomas B. Fetter
NAR 15551

Calculating Velocity from Altimeter Data – Part 2

Now that the process for designing a moving average filter is understood, the method can be applied to the StratoLogger velocity data. Looking at the data in Figure 26 (Part 3), the velocity has noise spike peaks of 300-400 mph. An attenuation of approximately 100 will be needed to smooth the trace acceptably so that the noise is less than 3-4 mph. To achieve this, I will select a two pass smoothing filter. Next, the roll-off frequency is selected.

Looking at the altitude data in Figure 14 (Part 2), it is clear that the noise increases significantly after the parachute is deployed at apogee. This increase in noise will be accentuated as the derivative, or slope of the amplitude is calculated to create the velocity trace. This can be seen in Figure 26, which is the unfiltered velocity data trace. But the difference in the frequency of the noise during the descent portion of the flight and the most rapid changes during the ascent portion of the flight is not very large. If a filter is applied with a low enough roll-off frequency to adequately reduce the noise, it will also attenuate the rapid velocity changes during ascent. This would have the effect of reducing the apparent peak velocity, as the peak of the data would be smoothed as shown in Figure 29 (Part 3).

A simple way of dealing with the conflicting amount of smoothing needed during the ascent, where the noise is low, but the rate of change is high, and descent, where the noise is high, but the rate of change is low, is to apply different window lengths during each portion of the flight. This is easy to do in Excel. Complete data traces are calculated using both the short and long windows, and then a composite data trace is created that uses the shorter window for altitudes less that the peak altitude, and the longer window for altitudes greater than the peak altitude.

The actual window lengths are selected by trial and error. The longest window is selected that still preserves the most rapid changes in that portion of the velocity data. Figure 35 shows a 5 sample (.25 sec) and 11 sample (.55 sec) window three pass filter applied to the peak velocity portion of the data. The 5 sample window still has some sinusoidal noise remaining, whereas the 11 sample filter nicely soothes the data without reducing the peak value significantly, so an 11 sample window is chosen for the ascent portion of the flight data. Actually, the 5 sample window gives very similar results to StratoLogger Data Capture software for this portion of the flight data, but for this flight, I chose an 11 sample filter. For the descent portion, a 33 sample window smooths the data nicely as seen in Figure 36. The frequency response of the 11 and 33 sample three pass filters are shown in Figure 37. The roll-off frequencies of these two filters are around 0.2 Hz and 0.6 Hz.

Figure 36 shows the results of a composite 11/33 sample filter applied to the velocity data. With this dual bandwidth filter applied, the entire velocity trace now clearly shows the actual velocity of the rocket over its entire flight. The peak velocity is preserved, and the descent velocity under drogue is easy to see. The decrease in velocity at the end of the trace is due to the main deployment.

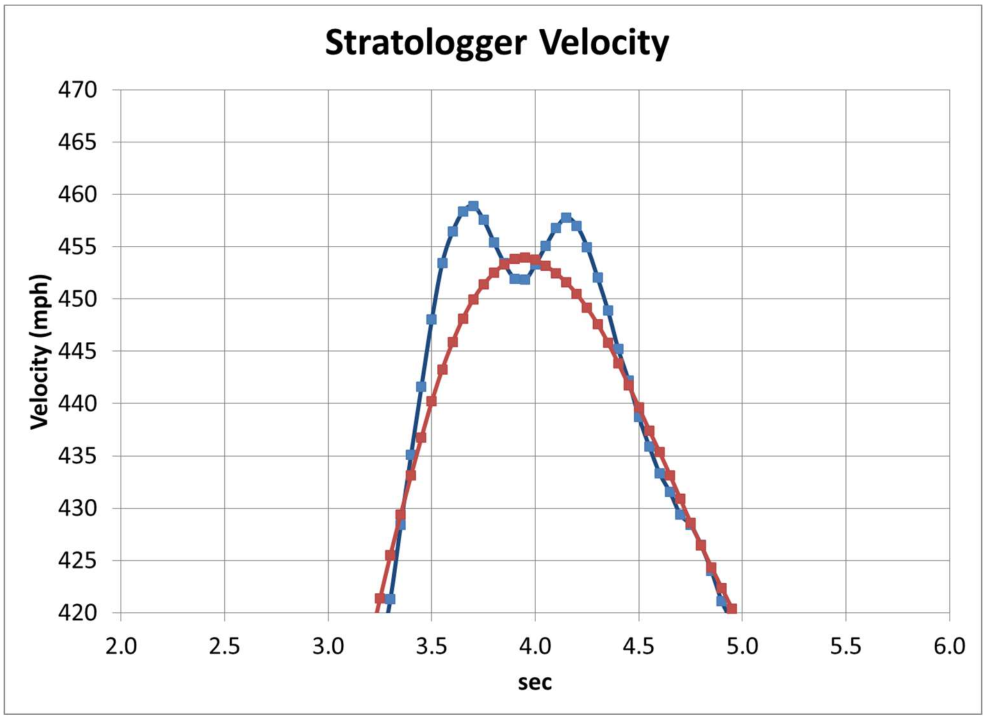

Figure 35– Calculated velocity with three pass 5 and 11 sample filters at peak velocity

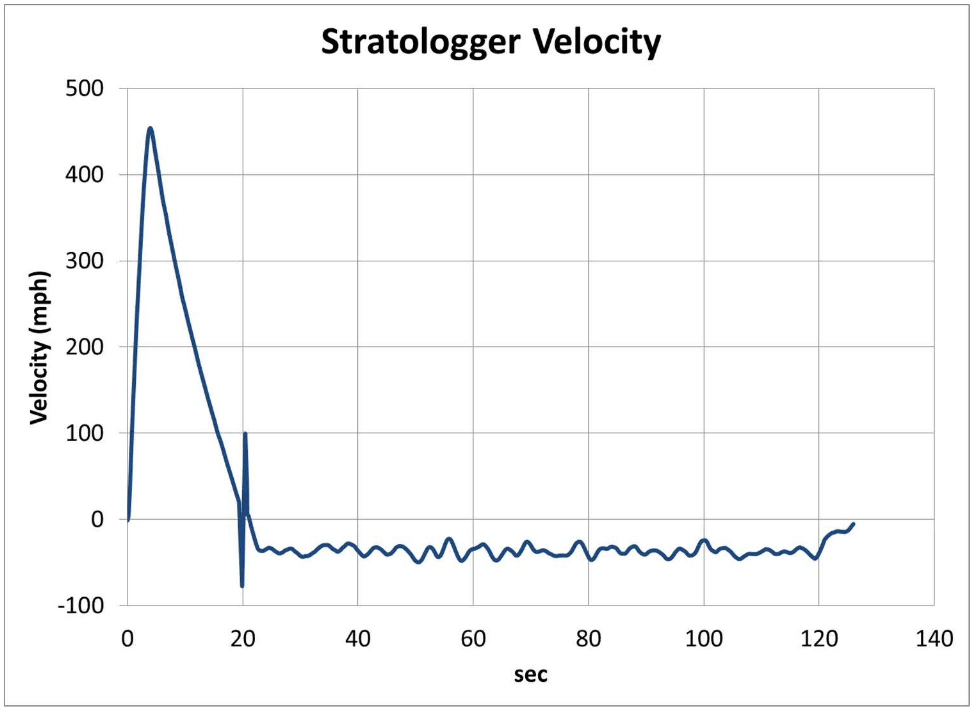

Figure 36 – Calculated velocity with three pass 11/33 dual sample filter

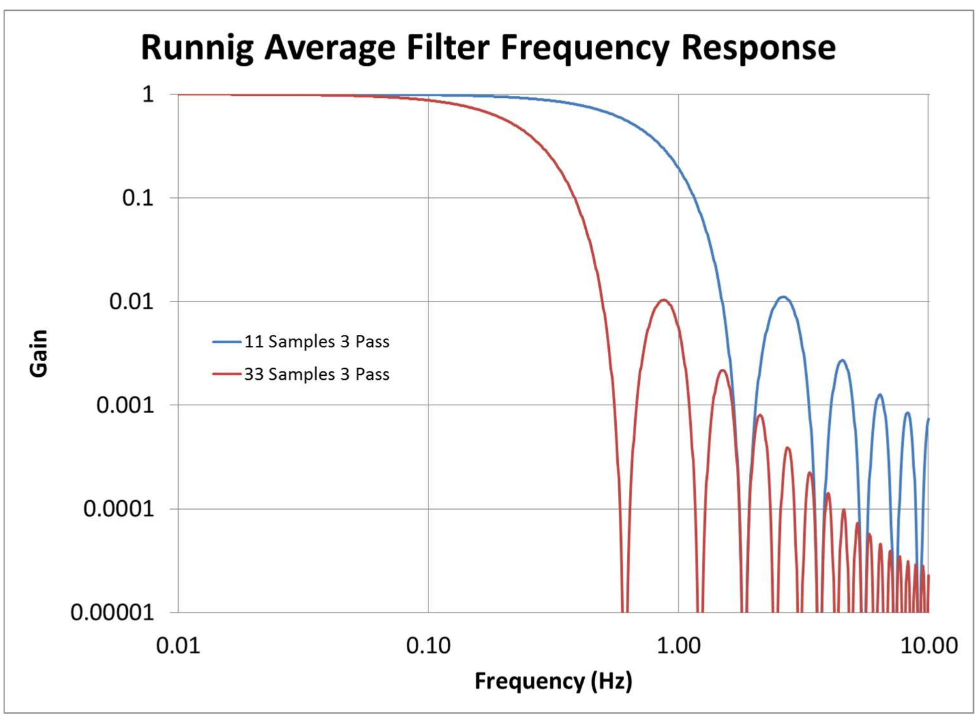

Figure 37 – Frequency response of three pass moving average filter 11 and 33 samples in length

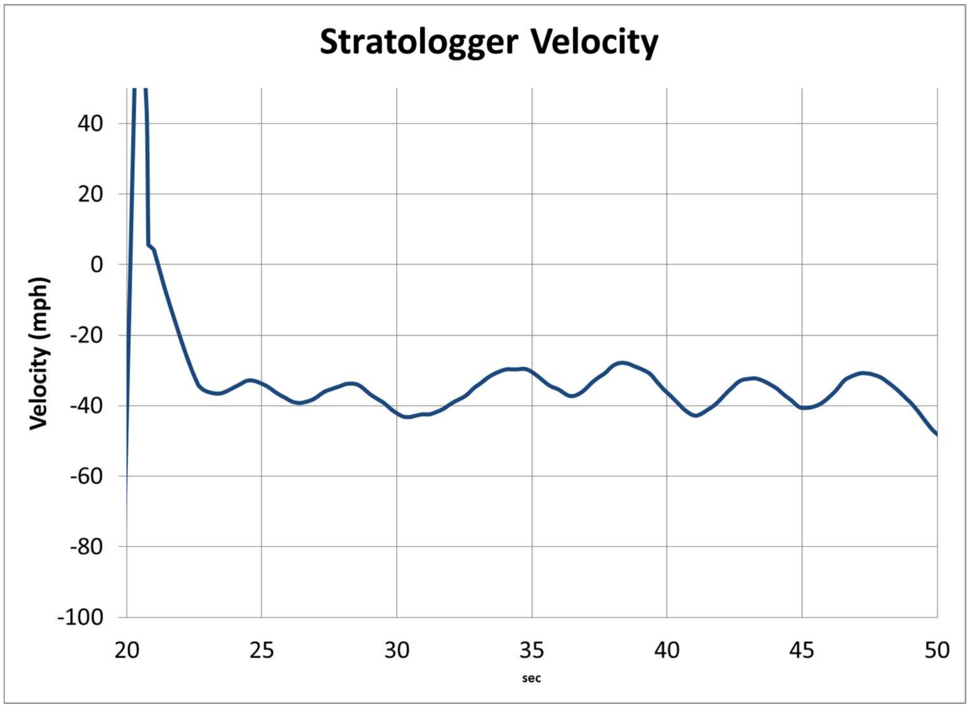

Figure 38 – Velocity during drogue descent

Figure 38 shows the velocity during drogue descent after the filtering has been applied. The 1 cycle per second oscillation is gone, but there is now an oscillation with a period of around 5 seconds and a peak-to-peak amplitude of 10 mph. It is reasonable that this is due to the actual swinging of the rocket under the drogue. To verify the corresponding altitude variation, Figure 39 shows the altitude plot for just one of these cycles during the descent. For this graph, the same 33 sample three pass filter was applied to the amplitude data to take out the higher frequency noise. Although it is harder to see because of the relatively rapid descent, the variation from a straight line descent (the dotted red line) is about 10 feet, which is a reasonable peak amplitude for swinging.

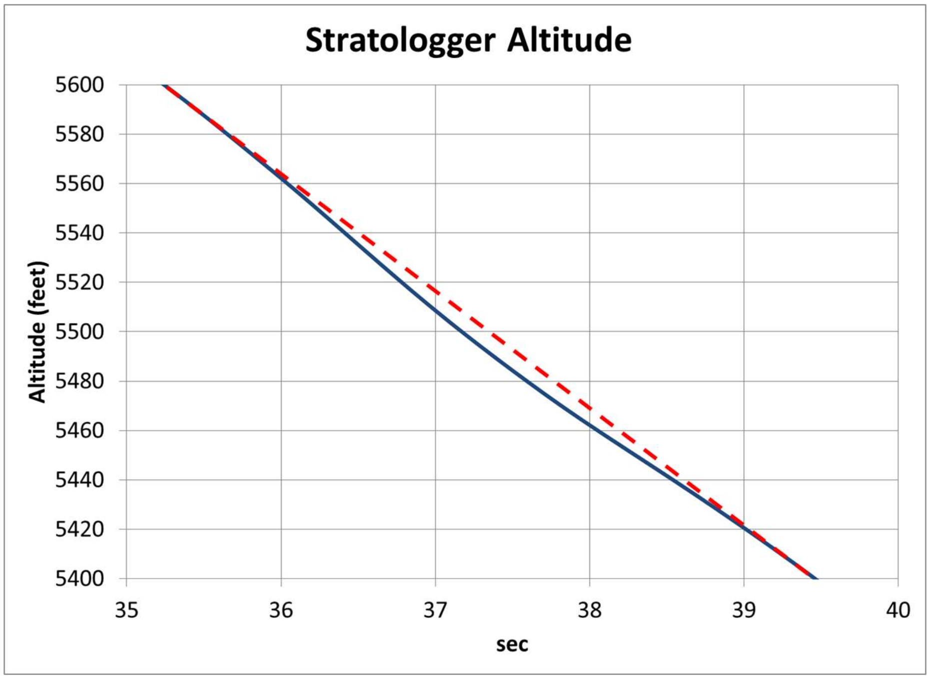

Figure 39 – Altitude variation during drogue descent

Calculating Velocity from BeeLine GPS Data

It might be expected that the velocity calculated from the BeeLine GPS data would be noisy as well, and would require filtering. But the data from the newer BeeLine GPS units has very little sample to sample noise, probably because it is filtered internally, so simply taking the difference of adjacent data points produces a nice smooth velocity plot as seen in Figure 40. If noise is a problem, then using the same techniques described for the StratoLogger data will work. The older BeeLine GPS units do require external filtering when calculating velocity.

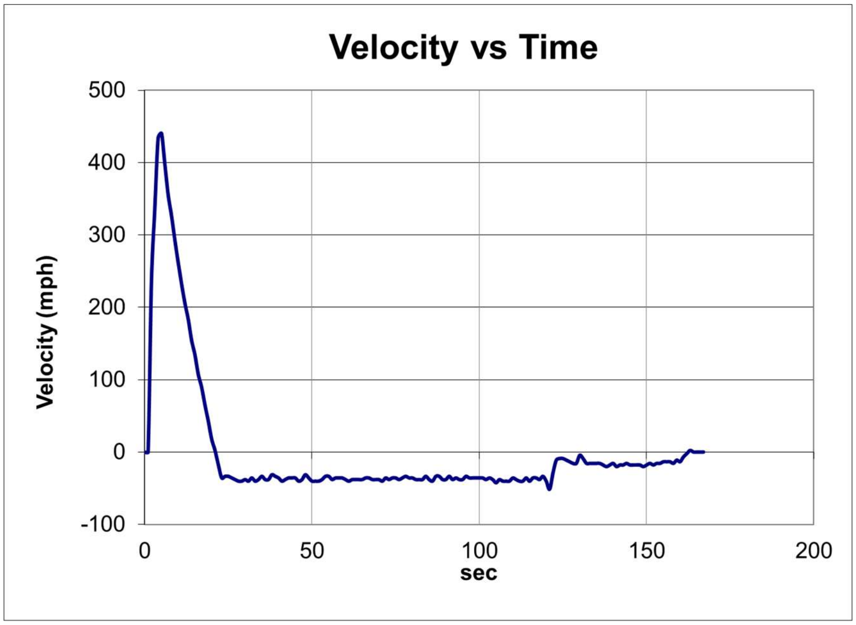

Figure 40 – Velocity calculated from BeeLine GPS altitude data

Figure 41 shows a plot of the velocity, calculated from the altitude BeeLine data, of just the descent portion of the flight. For the descent, units of feet per second, fps, are used. I like using mph for the ascent, and fps for the descent, which is typically used when describing the rate of descent under parachute. To calculate mph from fps, mph = fps * (3600 sec/hr)/ (5280 ft/mi). Under drogue, the descent rate was a little over 55 fps. When the main deployed at just under 200 sec, the rocket slowed to 20-30 fps. This graph is a good check of the size of the parachutes used for both drogue and main. 55 fps under drogue is fine, especially on a moderately windy day like this was. But 20-30 fps under main was a little fast for the final descent. Normally, I like to keep the final descent velocity under 20 fps to minimize paint chipping on landing.

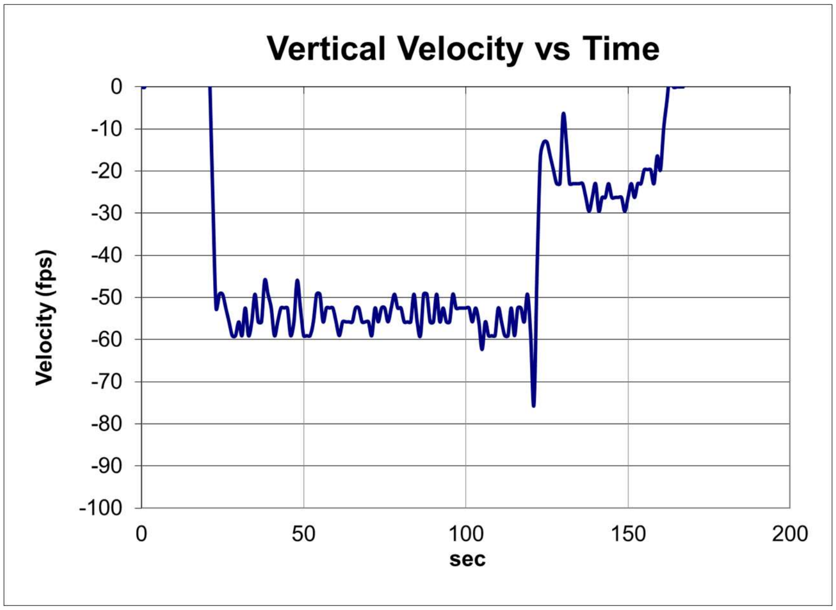

Figure 41 - Velocity calculated from BeeLine GPS altitude data

The horizontal, or ground speed can also be calculated and plotted using the GPS data. This is a good check of the wind speed as the rocket descends. The ground speed is determined by calculating the total distance traveled between each data acquisition and dividing by the time increment. The total ground velocity is:

$$v_{ground} = \sqrt{\Delta_{east}{}^2 + \Delta_{west}{}^2} \,/\, \Delta t \qquad (10)$$

Figure 42 shows that the ground speed reached a maximum of 45 mph at motor burnout, and then decreased to 25 mph by the time the drogue ejected at 20 sec. The rocket then slowed to an average of around 12 mph, which would have been the wind speed. It's interesting to see that the ground velocity has a sinusoidal oscillation with a peak-to-peak amplitude varying between 5 and 10 mph and a period of about 5-10 seconds. Figure 43 shows the detail of the oscillation. It can be seen to occur over many 1 second data samples, and is sinusoidal. This is similar to the oscillation rate seen in the altitude data, and is, again, likely caused by the rocket's swinging.

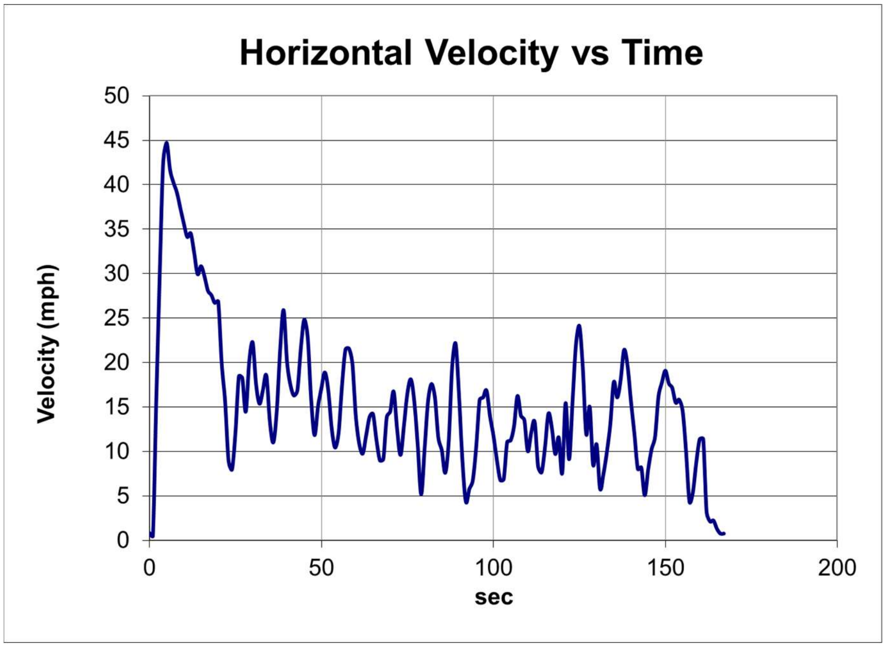

Figure 42 – Horizontal velocity

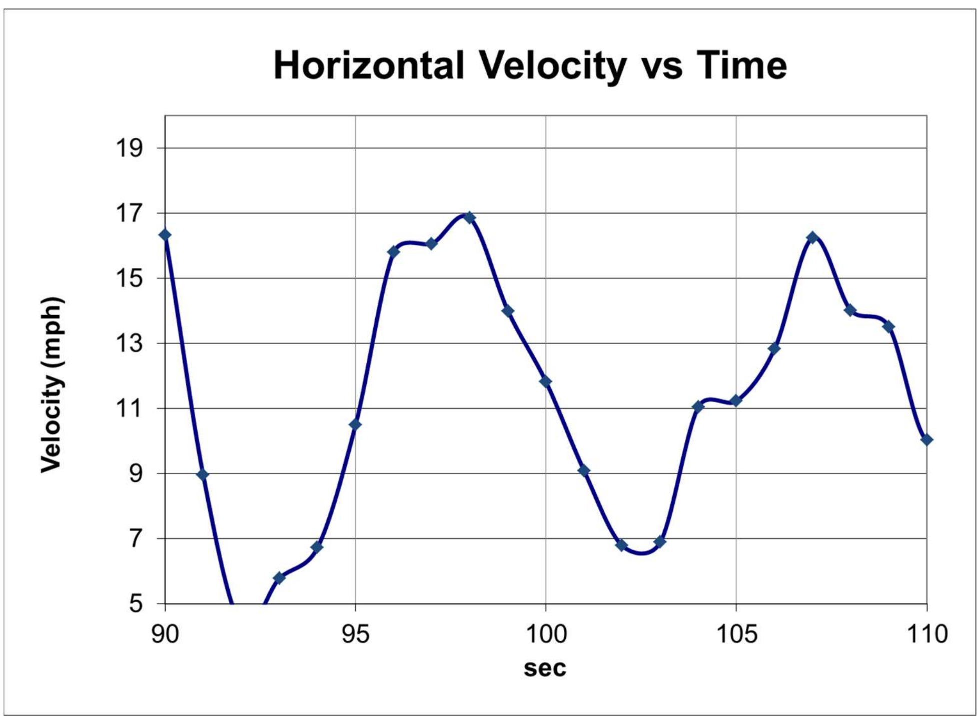

Figure 43 – Horizontal velocity during drogue descent

With a 12 mph wind, the advantage of dual deploy is clear. Even with the small drogue, Speedmotion drifted nearly 2000 feet from the launch point. Had only a main been used, given the 25 fps descent rate under main, Speedmotion would have drifted nearly a mile.

$$\frac{6500\,ft}{25\,fps} * 12mph \frac{5280\,ft/mi}{3600\sec/hour} = 4576\,ft \qquad (11)$$

It's also interesting to note that the wind speed was very constant from 6500 feet all the way to ground level.

Calculating Acceleration

The acceleration of the rocket can also be calculated. The vertical acceleration is calculated by taking the difference between two adjacent velocity points and dividing by the time interval. To convert to units of g's, the result is scaled using *1g = 32 ft / sec²*

$$a_{vertical} = \frac{1}{32} * \frac{\Delta v_{vertical}}{\Delta t} \qquad (12)$$

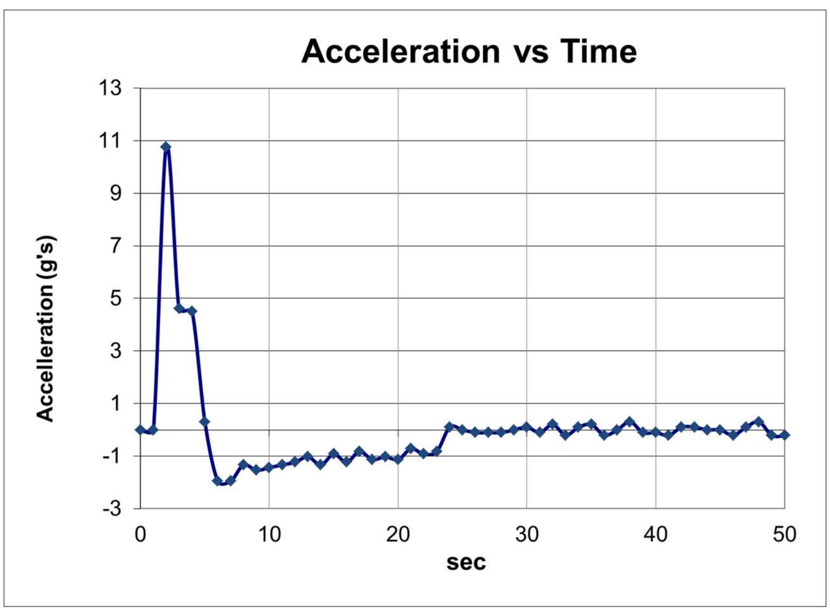

Figure 44 – Acceleration calculated from GPS data

Figure 44 shows a plot of the acceleration using the GPS data. From this graph, after reaching a peak of 11g's during the motor burn, the rocket decelerates at a little over -1g due to the combination of the force of drag plus gravity, and then reduces to just -1g as the rocket slows and the drag force, which is a function of velocity, becomes small compared to the force of gravity. Once the parachute is deployed, the average acceleration goes to zero as the velocity is constant. The sample rate is really not high enough to capture the motor burn portion of the flight in complete detail as there are only 3 data points during the motor burn.

Since the StratoLogger has a higher sample rate, perhaps it will show the acceleration of the burn portion of the flight with more detail. Figure 45 shows the results of calculating the acceleration from the StratoLogger data. The acceleration was first calculated by taking the slope of the un-filtered velocity data, and then the same 11/33 sample three pass filter used on the velocity data trace, producing clean results.

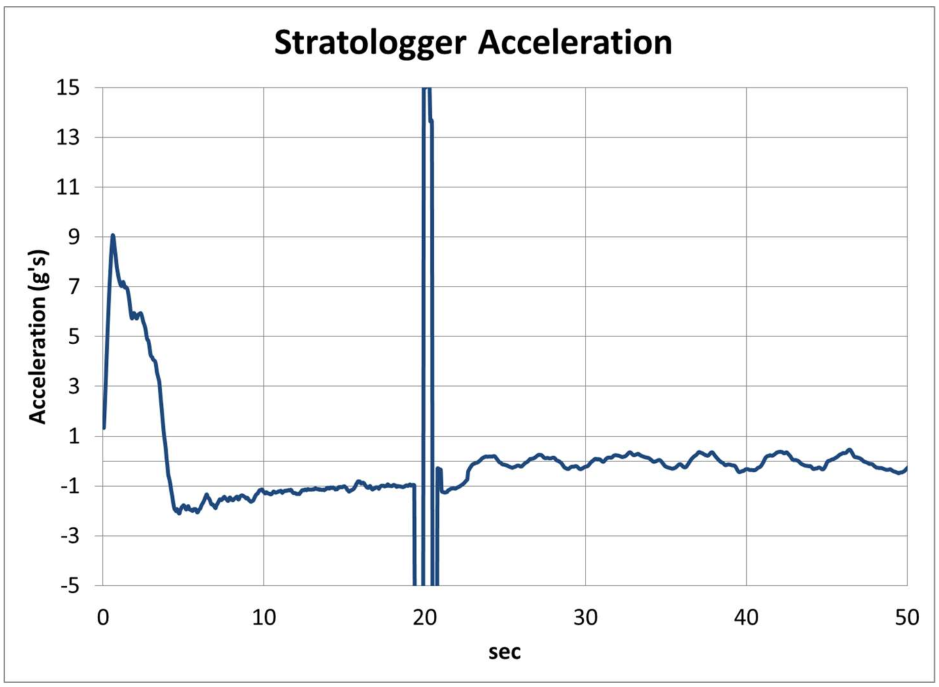

Figure 45 – Acceleration calculated from StratoLogger altitude data with three pass 11/33 dual Sample filter

Figure 46 shows the detail of the acceleration during the thrust portion of the flight. Note the acceleration takes almost a second to reach its peak. Remember, from Figure 34 (Part 3), the step response of the 11 sample three pass filter is about a half second, a little faster than the rise time in Figure 46, so the 11 sample filter is probably not significantly impacting the acceleration curve. Just to be sure, the filter length was reduced to 7 and then 5 samples in length. Reducing the filter length below 11 samples appeared to have minimal impact on the rise time of the acceleration, but the noise did increase, so the 11 sample filter looks like a good choice for the acceleration data as well as the velocity data.

The slow rise in acceleration is probably due to the time it takes for the motor to come up to full thrust. From observations at launches, there is a considerable variability in the speed at which AP motors come up to thrust. Also note in Figure 46, that once the motor burns out, the initial deceleration is about -2 G's, due to the force of both drag and gravity. As the velocity of the rocket slows and the force due to drag is reduced, the deceleration approaches -1G, and remains there until apogee is reached and the parachute is deployed. The large spike in the acceleration data is due to the ejection charge pressure spike, still visible even after the filter has been applied to the data. Once the parachute has been deployed, the acceleration centers about 0 as the average velocity is constant. But the same oscillation with a 5-10 second period can be seen clearly in this data as was seen in the horizontal velocity data. Again, this is likely due to the rocket swinging under its parachute.

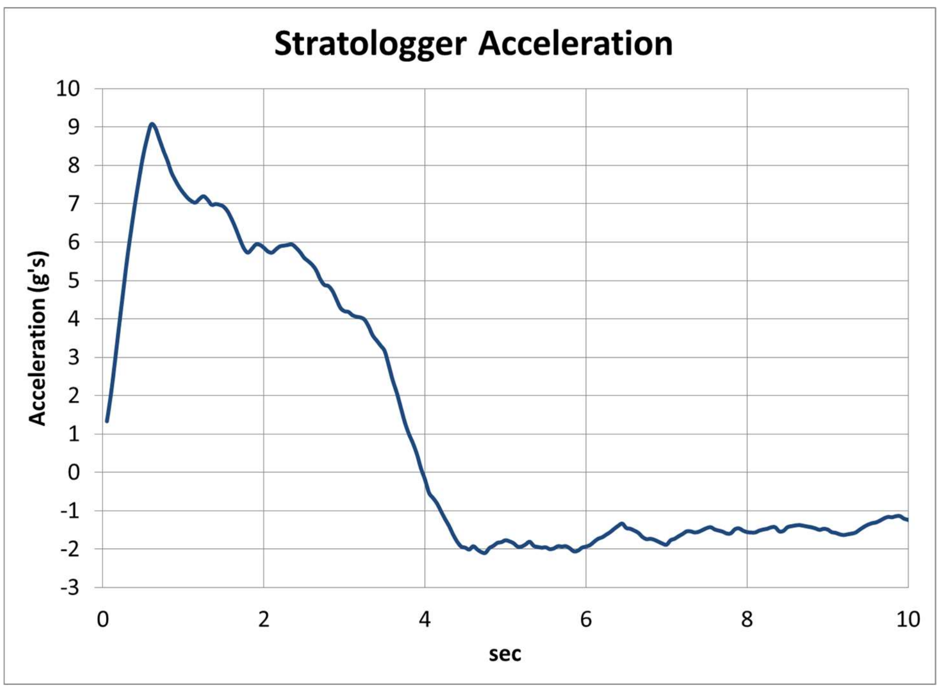

Figure 46 - Acceleration calculated from StratoLogger altitude data with three pass 11/33 dual sample filter

Adding RockSim Data

The last step of the flight analysis is to compare the flight simulation data to the actual flight trajectory measurements. Figure 47 shows the RockSim simulation results for Speedmotion 54 flying on the AeroTech J415W motor used for this flight. The RockSim simulation results can be exported to a comma separated, .csv file, just like the BeeLine GPS and PerfectFlite data. This, in turn, can be opened in Excel, and the simulation data added to the BeeLine GPS and StratoLogger data for comparison. The data is exported from RockSim using <File:Export:Simulation Data>, then selecting Time, y-Acceleration Total, y-Velocity, and Altitude as the data to export. Be sure to select y-Acceleration Total, and not y-Acceleration for both the export as well as on the plot in RockSim (Figure 47). The difference is y-Acceleration does not include the acceleration due to gravity, that is, it is offset by +1 G.

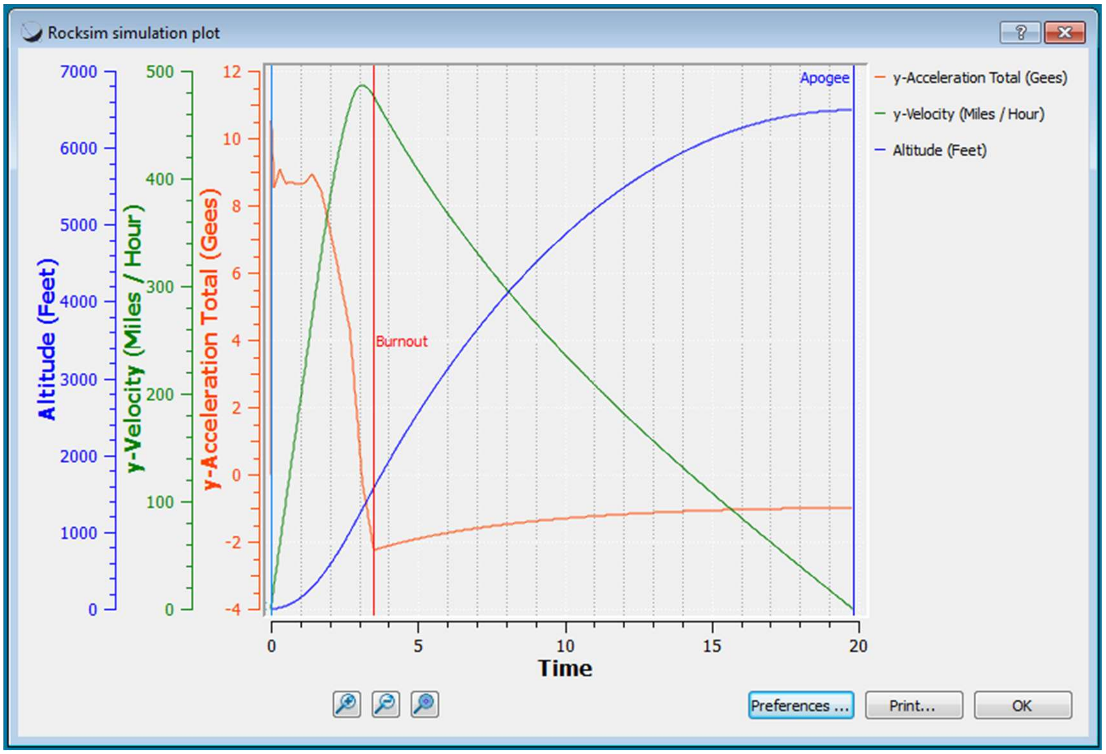

Figure 47 – RockSim simulation data

Figure 48 shows the RockSim data once imported into Excel.

Rocksim Data			
Time	y-Acceleration Total Gees	y-Velocity Miles / Hour	Altitude Feet
0.00000	0.000	0.000	0.000
0.00000	0.000	0.000	0.000
0.08375	8.889	18.088	1.164
0.16750	8.700	33.929	4.372
0.25125	8.935	50.133	9.546
0.33500	8.999	66.696	16.735
0.41875	8.827	83.070	25.951
0.50250	8.656	99.122	37.159
0.58625	8.675	115.043	50.326
0.67000	8.688	130.994	65.451
0.75375	8.693	146.961	82.537
0.83750	8.693	162.933	101.584
0.92125	8.691	178.901	122.593
1.00500	8.696	194.872	145.564
1.08875	8.709	210.858	170.497
1.17250	8.729	226.877	197.395
1.25625	8.803	242.966	226.264
1.34000	8.897	259.227	257.121
1.42375	8.925	275.637	289.984

Figure 48 – RockSim data imported into Excel

Figure 49 shows an overlay of the BeeLine GPS, corrected StratoLogger, and RockSim simulation data. Figure 50 shows the detail at apogee. The simulation peak is within about a hundred feet of the measured peak. For a flight to 6300 feet, this simulation is remarkably close to the actual flight data. To get results this close, I weighed each section of the rocket after building it and adjusted the weights for the respective components in RockSim. The wind settings in RockSim also have a significant impact on the predicted altitude, as would be expected. The stronger the wind, the more the rocket will angle away from vertical at the beginning of the flight, and the lower the resulting peak altitude. In fact, just re-running the same simulation can produce results that vary more than 100 feet from one simulation to the next as RockSim varies the wind randomly within the selected wind parameters. For this simulation, I choose “Slightly breezy (8-14MPH”) for the Wind Conditions setting found on the Launch conditions tab under <Simulation:Prepare for launch …>, and “Fairly constant speed (0.01)” for the Wind turbulence. These settings matched the actual wind conditions during the flight, but despite matching the weight and wind conditions, the simulation and measured flight do not always match this closely.

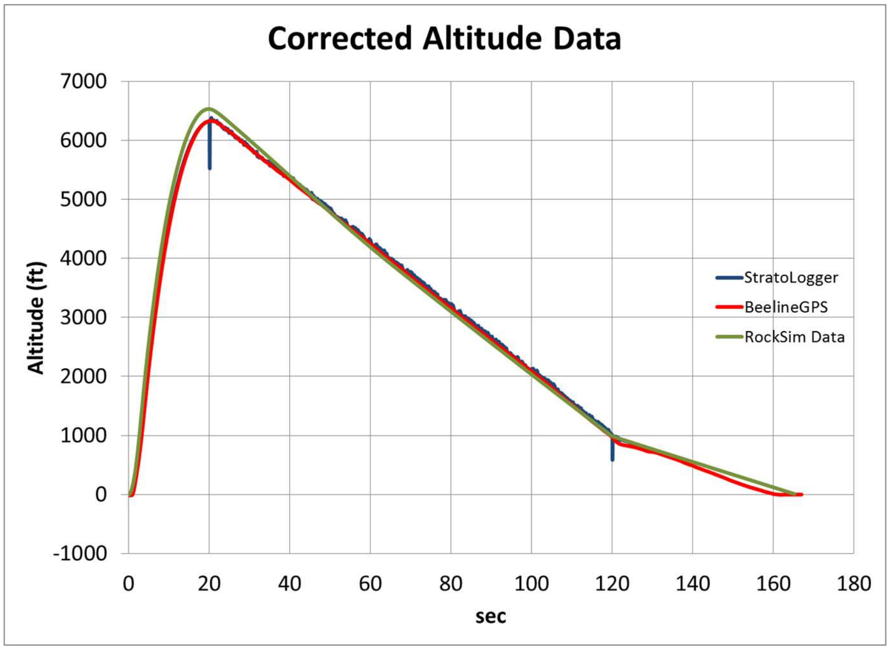

Figure 49- Actual flight altitude data vs. RockSim

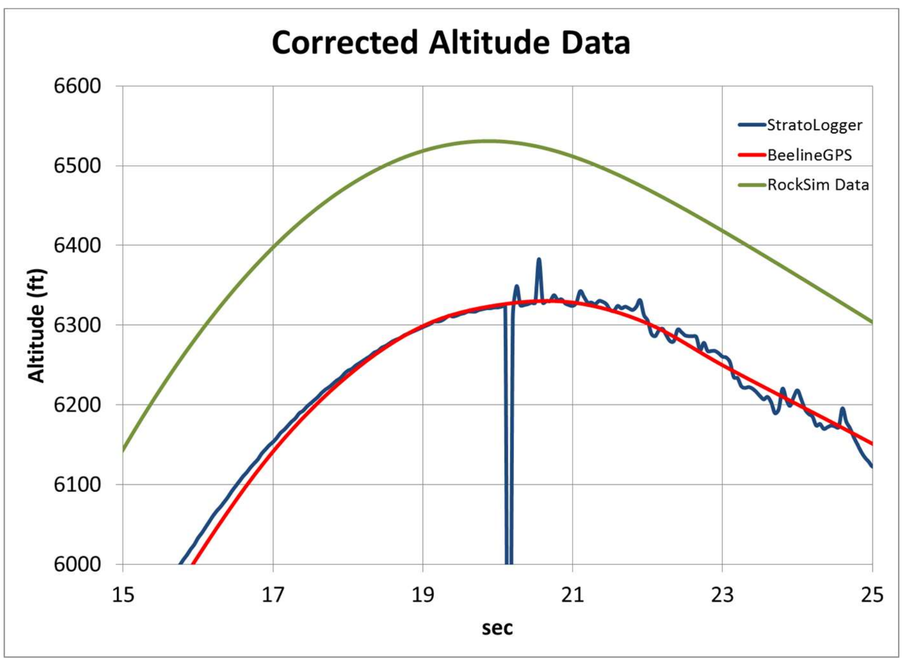

Figure 50 - Actual flight altitude data vs. RockSim

Figures 51 and 52 show a comparison of the three velocity curves, with Figure 52 showing the details at the peak velocity. Looking at Figure 52, the PerfectFlite and RockSim match quite well at the peak velocity. The PerfectFlite data shows the actual velocity peak was about 50 mph below the simulation, and the rate increase in velocity was a slower than the simulation. This could be due a slower ramp in the motor's thrust as the motor pressurized more slowly than the more ideal thrust curve used by RockSim. The velocity curves match very closely after motor burnout, which says the estimate of the drag coefficient for the rocket in RockSim is very accurate. The BeeLine GPS data lags the other two due to the low sample rate as described previously, but matches very closely.

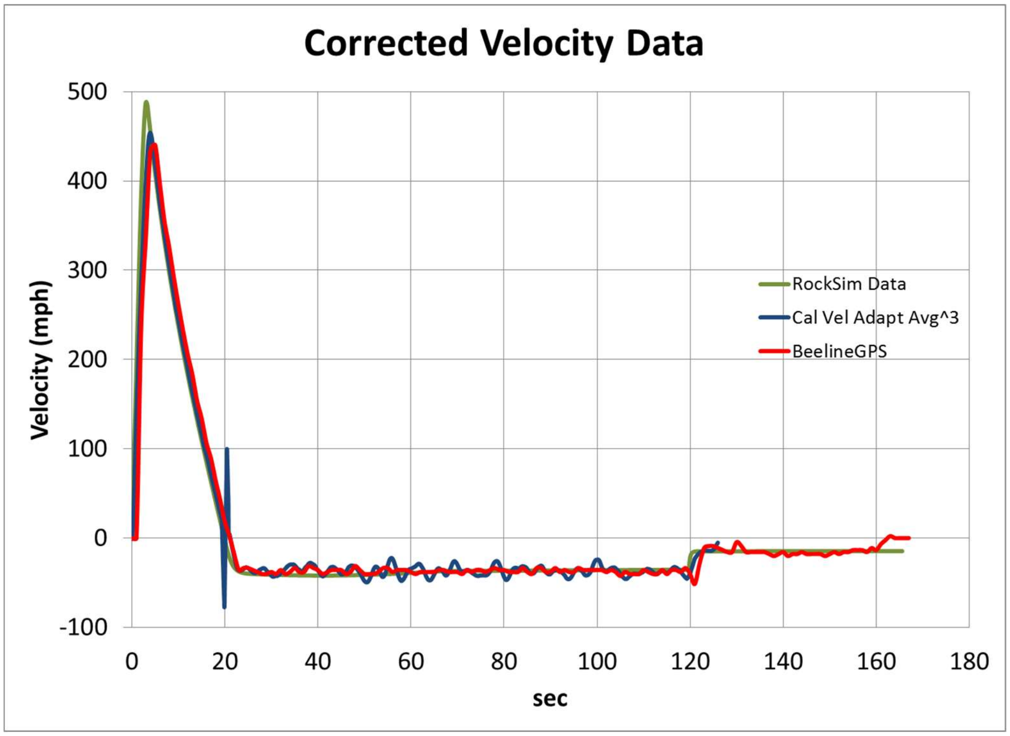

Figure 51 - Actual flight velocity data vs. RockSim

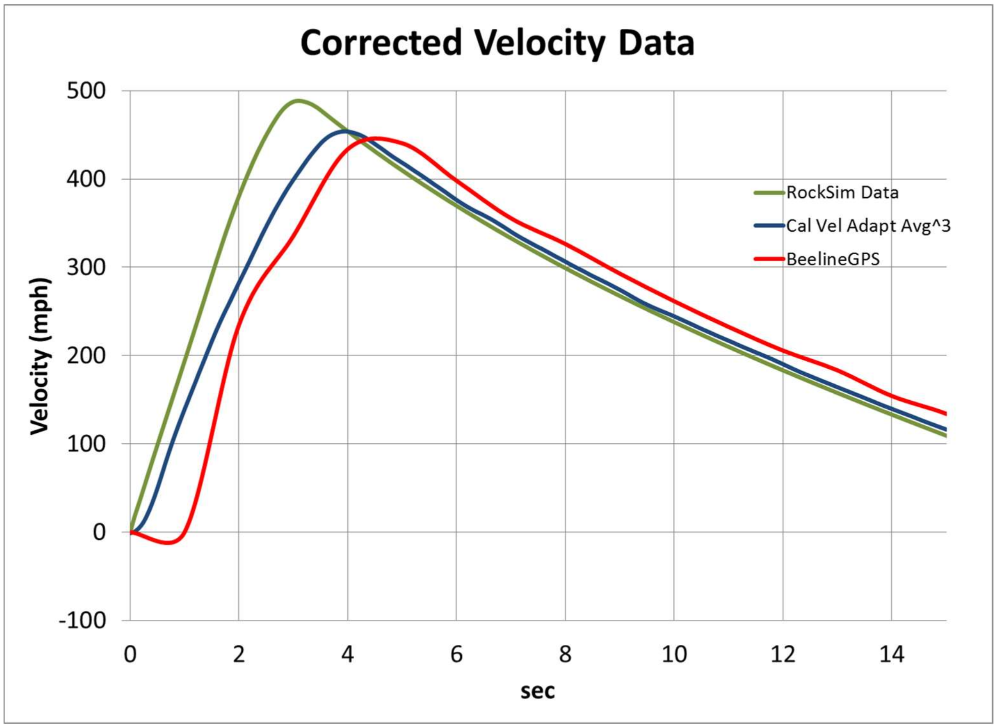

Figure 52 - Actual flight velocity data vs. RockSim

Figure 53 shows a comparison of the acceleration data. As would be expected from the velocity data, the RockSim predicted acceleration starts immediately at launch, but the PerfectFlite data shows a slower ramp for the acceleration. The smoothing filter could limit the step response of the PerfectFlite acceleration data, but it was verified that the 11 sample filter has minimal impact, so the slower ramp in acceleration is real and likely due to the actual ramp in the motor's thrust. Other than the delay, the RockSim and PerfectFlite acceleration curves actually match quite well. The BeeLine GPS, however, simply has too slow a sample rate, a 1 sample per second, to accurately catch the acceleration during thrust which lasts only 4 seconds.

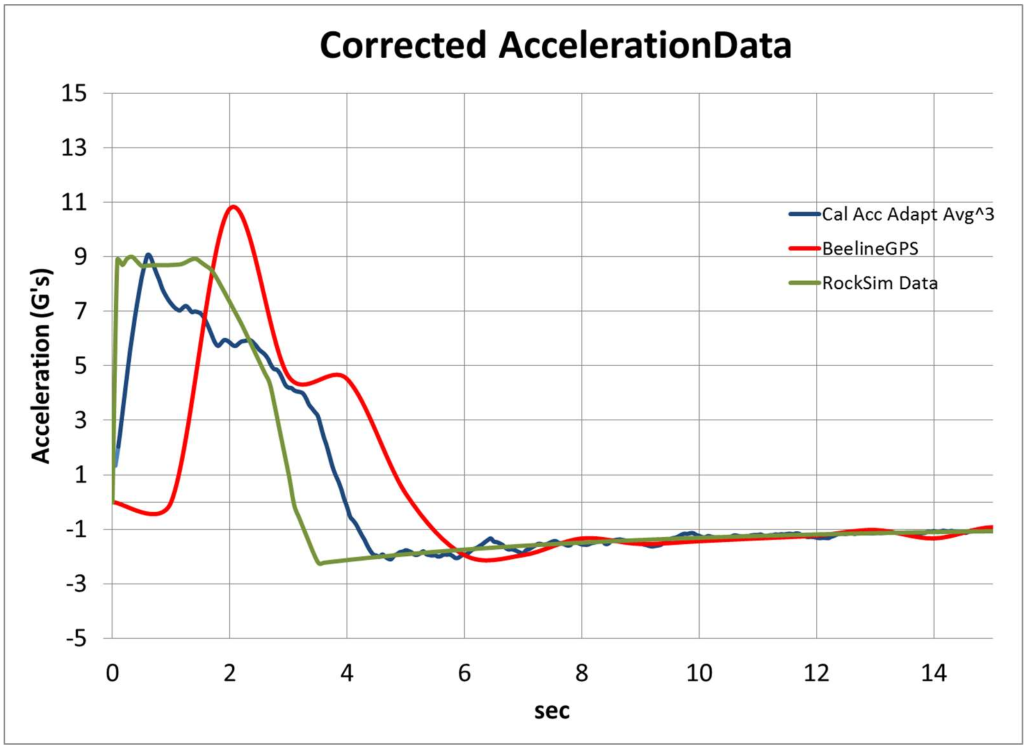

Figure 53 - Actual flight acceleration data vs. RockSim

Useful References

PerfectFlite DataCap: http://www.perfectflite.com/Download.html
BeeLine GPS Data Communicator: http://www.bigredbee.com/BeeLineGPS.htm
RockSim: http://www.apogeerockets.com/Rocket_Software
My Excel spreadsheet: http://www.Speedmotionrockets.com

Flight Data Analysis Using Excel - Part 5

Thomas B. Fetter
NAR 15551

Lesson on Using Backup Altimeters

Why did the StratoLogger stop logging before the end of the flight (see Figure 4 in Part 1)? Remember that the StratoLogger was set up as the back-up altimeter. Looking at the intended firing sequence, the HiAlt45 altimeter was set to fire at apogee. Then the StratoLogger was set to fire 2 seconds later (apogee delay). Then the HiAlt45 was to set fire at 1000 feet, and the StratoLogger at 800 feet. The HiAlt45 fired properly at apogee. This can be seen in Figure 4 as the positive pressure, negative altitude spike right at apogee. The pressure spike occurred because some of the pressure cause by the ejection charge leaked into the altimeter bay. But since apogee had already been detected by the StratoLogger, the pressure spike had no impact on its deploying properly, 2 seconds later. When the rocket reached 1000 feet, the primary HiAlt45 fired, and again, there was a positive pressure spike in the altimeter bay caused by the ejection charge leakage. This time, the pressure spike had an effect. Because the pressure spike looks like a negative change in altitude and was large enough, minus 400-500 feet, to indicate an altitude below 800 feet, that spike triggered the backup StratoLogger to fire immediately. The altimeter reading then returns to the actual altitude, 1000 feet, as the pressure spike dissipates. This apparent increase in altitude after the pressure spike fools the altimeter into thinking the rocket is no longer descending, and the data logging is stopped shortly thereafter. Apparently, the StratoLogger is programmed to ignore a pressure spike that occurs at the time it fires, but not at any other time.

A solution to the data logging ending early, short of a better seal on the ebay, is to reverse the altimeter's roles and use the StratoLogger as the primary altimeter set to fire at 1000 feet, and set the HiAlt45 altimeter as backup, set to deploy at a lower altitude. This way, the StratoLogger would ignore the pressure spike at the primary deployment, and when the backup charge fires once the lower altitude is reached, the rocket would already be open, and little pressure buildup would occur in the rocket, preventing the pressure spike in the ebay.

Zeroing in on the pressure spike in the StratoLogger data in Excel, it can be seen that the spike occurs for a single sample. Given the StratoLogger sample rate of 20 samples per second, the spike occurred and dissipated in 50 milliseconds or less. At first, I was skeptical that a pressure spike could occur and then dissipate that quickly, and I wondered whether there was some other mechanism, other than actual pressure leakage, causing the apparent pressure spike. I expected that the time constant for a pressure spike to vent from the ebay and return to the ambient pressure would be longer than 50 ms. Also, I have never seen any black powder residue inside any of my ebays, despite many flights using a similar configuration on the ebays on all my rockets. To test whether this rapid dissipation was actually possible, I placed my ebay, with the StratoLogger running, in the bell jar I use to test my altimeters, and evacuated the jar to over 1000 feet. Then I simply popped the bell jar off its base, giving a nearly instantaneously a drop in pressure. Looking at the logged data loaded into Excel and plotted in Figure 54, it can be seen that the altitude (pressure) dropped 1200 feet from one data sample to the next. The pressure spike and return to ambient could indeed have occurred within a single 50 ms sample of the StratoLogger.

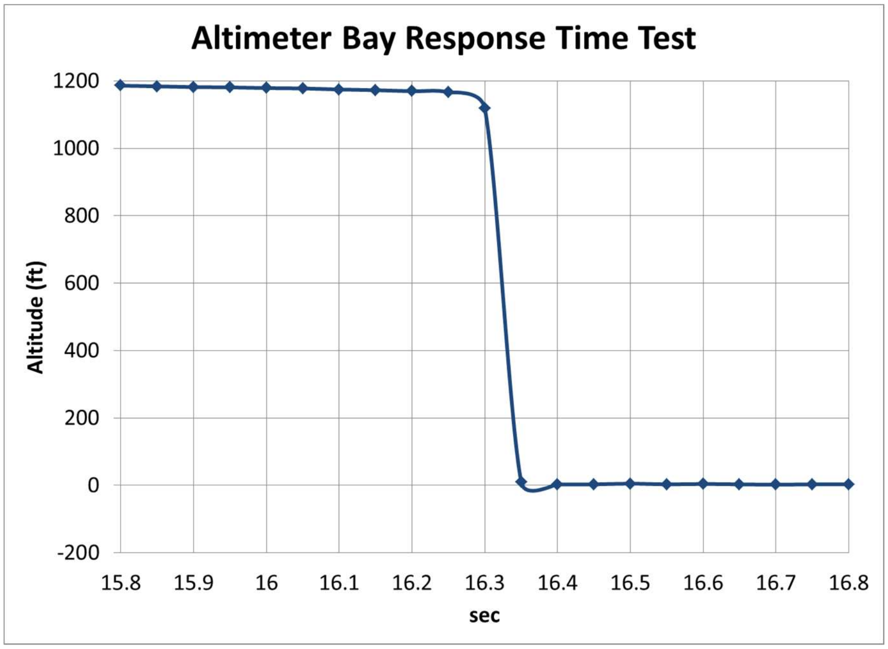

Figure 54 – Test of altimeter bay response time

I've always wondered whether the hole sizes used for altimeter bays are adequate to respond to the rapid decrease in pressure when the rocket is launched. The ebay would respond more quickly as the hole size is increased. I've used the common rule-of-thumb recommendations that are published by the altimeter manufacturers, but I've never actually measured the response time before this. A response time of 1200 feet in 50ms is equivalent to 24,000 feet/sec, which would be fast enough for any flight. In this case, I used two 3/16 in holes (0.055 sq. in) for a 3 in dia. x 8 in length (56 cu in volume) altimeter bay. The total hole size was larger than the rules-of-thumb give, but were necessary to access the altimeter switches. Still, these were small holes. Figure 54 clearly indicates that the response time is more than fast enough to keep up with the pressure decrease during launch.

Postscript – GPS Delay

I flew Speedmotion 54 two more times at the Tripoli Central October Skies launch in October, 2013. The weather for October Skies was perfect. There were beautifully blue skies, high temperatures in the mid 80's, and no wind during all three days of the launch. On Saturday, I chose the AeroTech K550W for the second flight of Speedmotion 54. I had changed the sequence of the two altimeters since its first flight in May, so that the StratoLogger would deploy the main at 1000 feet, and the HiAlt45 would follow up as the backup at 700 feet. This was done to eliminate the early termination of the StratoLogger data capture due to the misinterpretation of the main deploy pressure spike.

Figure 55 - Speedmotion 54 on the pad at October Skies

Figure 56 shows the results of the flight data, comparing the BeeLine GPS data to the StratoLogger data. The data in this graph has already been time aligned, and a gain correction of 1.05 has been applied to the StratoLogger data.

First, the change in the deployment sequence worked. The StratoLogger data continues all the way until the rocket reached ground level rather than shortly after main deploy, although it is hard to see because the graphs are so well aligned. Having data during the final portion of the flight actually makes it much easier to align the two traces in both time and altitude. In fact, it is easiest to get a time alignment between the two curves using the change in descent slope at main deploy. Here, the barometric and GPS data should be very close in time, since the rate of altitude change is slow, and the GPS data will be very accurate. The inflection point of the change in descent rates are aligned, as shown in Figure 57, and then the gain of the StratoLogger is changed until the two curves match closely over the entire length of the descent, as shown in Figure 56.

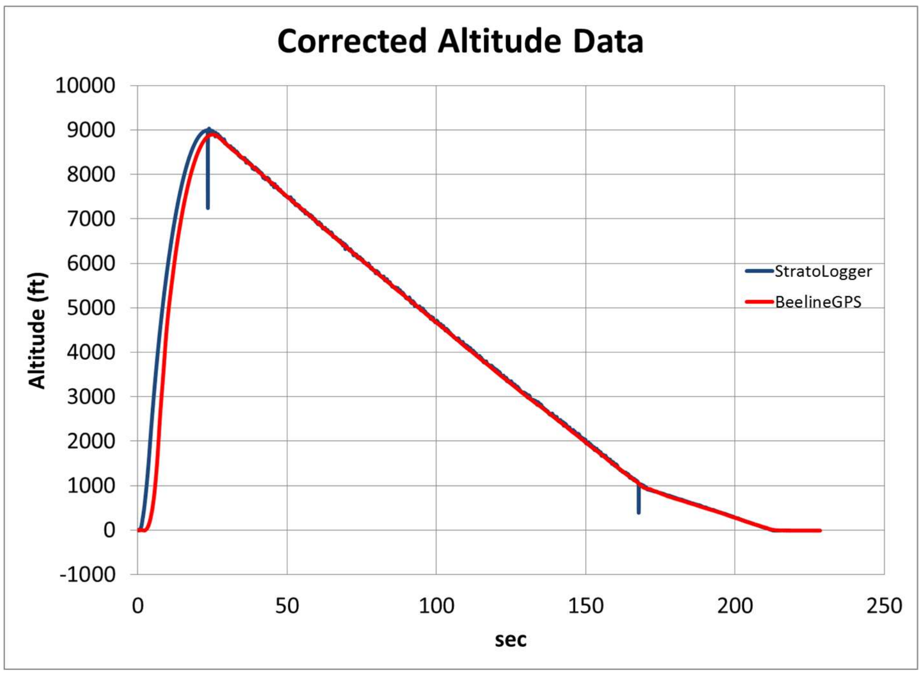

Figure 56 – Second flight of Speedmotion 54

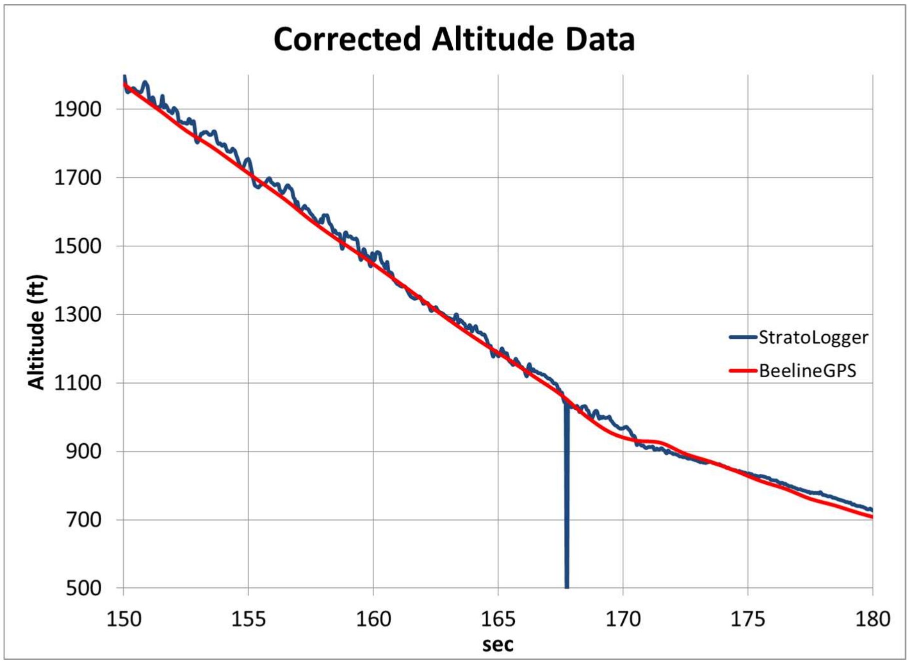

Figure 57 - Second flight of Speedmotion 54 – detail at main deploy

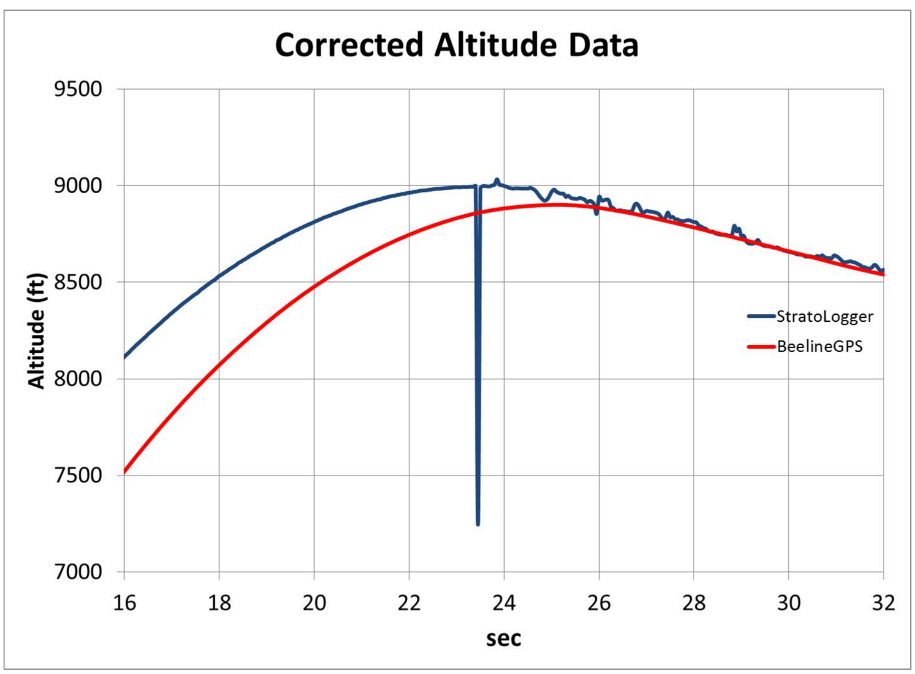

Figure 58 - Second flight of Speedmotion 54 – detail at apogee

The descent portion of the flight matches very closely in time and altitude, but there is an offset in the GPS data during the ascent. Figure 59 shows this offset more clearly. It starts at 3 seconds, and then reduces as the rate of ascent slows and converges just past the peak altitude.

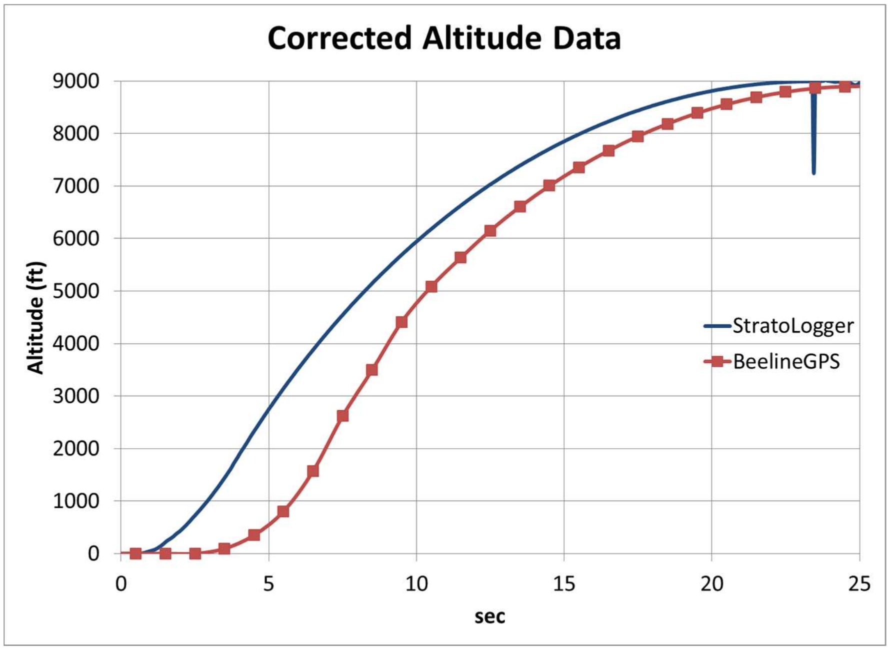

Figure 59 - Second flight of Speedmotion 54 – detail of ascent

Going back to the original flight of Speedmotion 54 and looking at the ascent portion of the flight in Figure 60, the BeeLine GPS data never falls behind the StratoLogger data. There is about a 0.2 sec fixed delay, but this is less than one sample interval. The initial acceleration of the first flight was around 9 g's, whereas the initial acceleration of the second flight was around 12 g's. The delay may be due higher rate of change in the altitude of the second flight and the GPS's inability to keep up with the more rapid change.

Because of this delay during the high velocity portion of the flight, the BeeLine GPS under predicts the peak altitude of the flight. Without the correction, the StratoLogger also under predicted the peak altitude. If the method of alignment used is correct, and the assumption that the GPS is the most accurate during the descent phase, then the combination gives a peak altitude that is higher than either would predicted alone for this flight. In this case, the difference between the GPS data and the corrected barometric data is 135 feet.

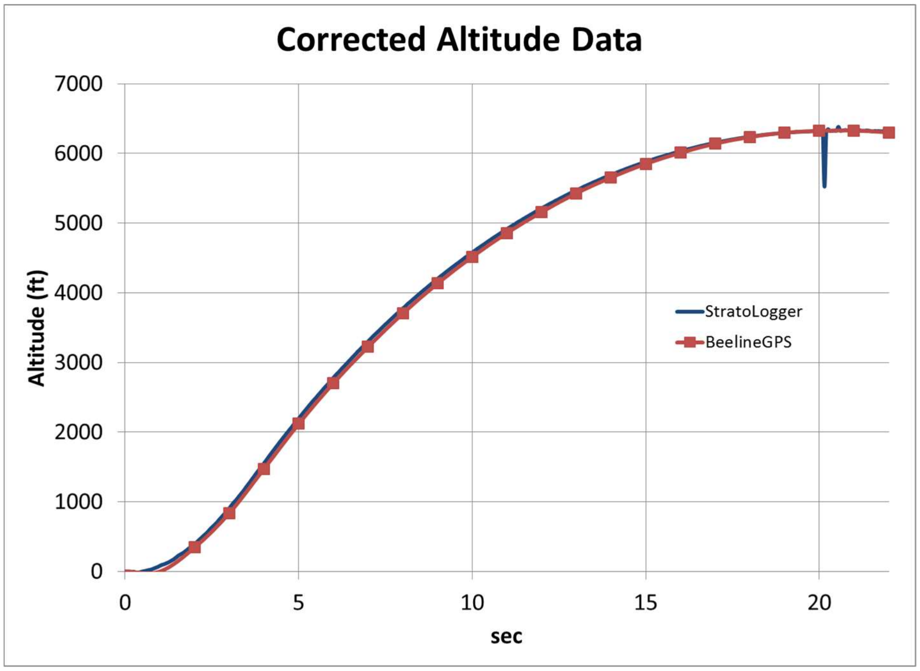

Figure 60 - - First flight of Speedmotion 54 – detail of ascent

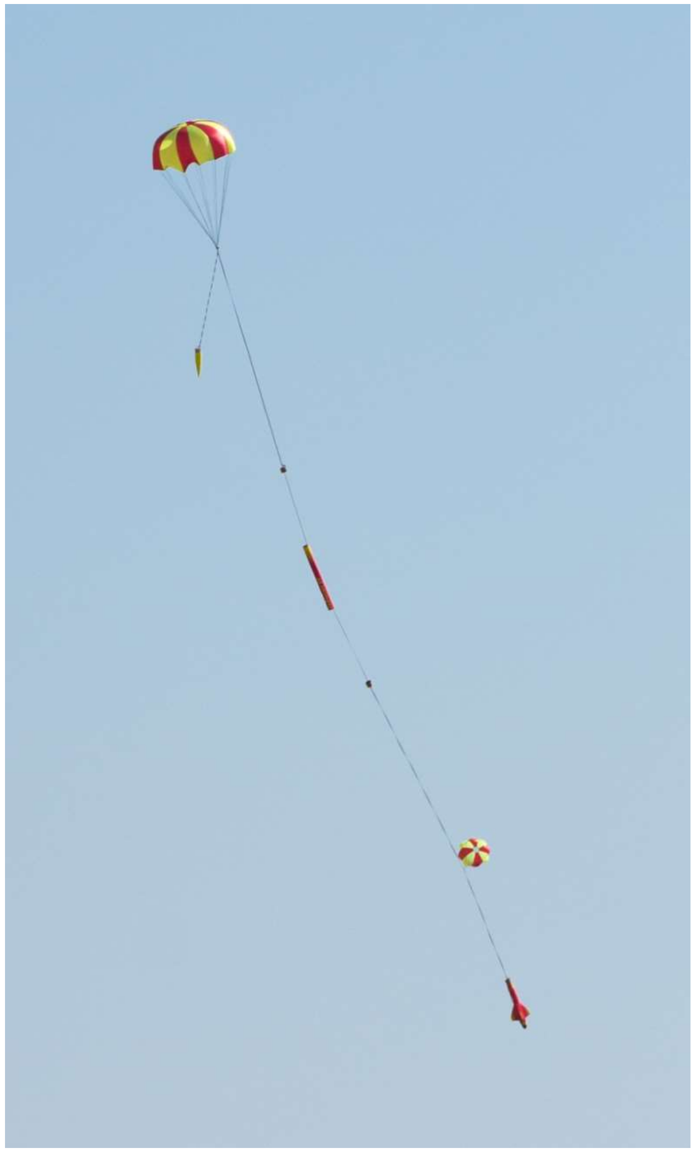

Figure 61 - Successful recovery of the second flight

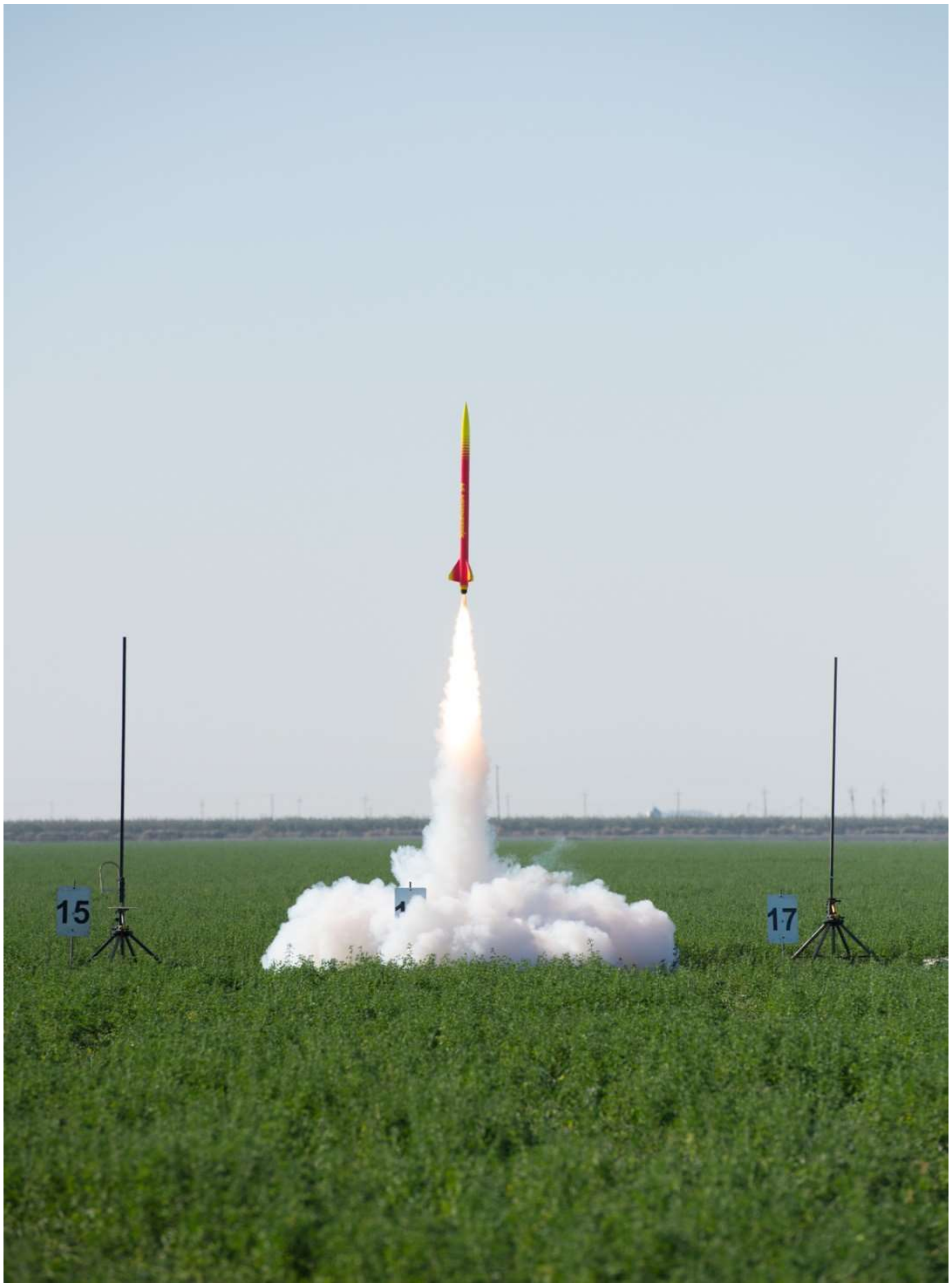

Figure 62 - Speedmotion on a K700 for its third flight

For the third flight of Speedmotion 55 on Sunday, I used an AeroTech K700W. This uses the 6 grain RMS-54/2560 casing. At 23 inches in length, this is the longest motor Speedmotion is designed to use. The flight was very impressive, straight up to 12,000 feet in the perfect calm. Figure 63 shows the aligned and corrected BeeLine GPS and StratoLogger data using the same technique of aligning the descent first in time, and then in altitude as a gain correction is applied to the StratoLogger data. The same value of StratoLogger gain, 1.05, worked for this flight.

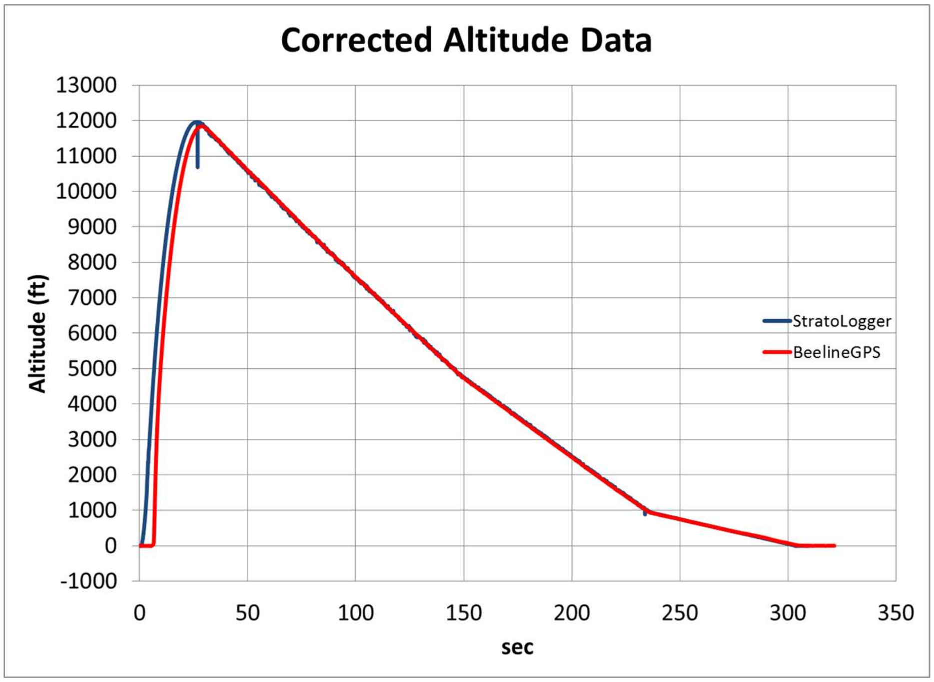

Figure 63 - Third flight of Speedmotion 54

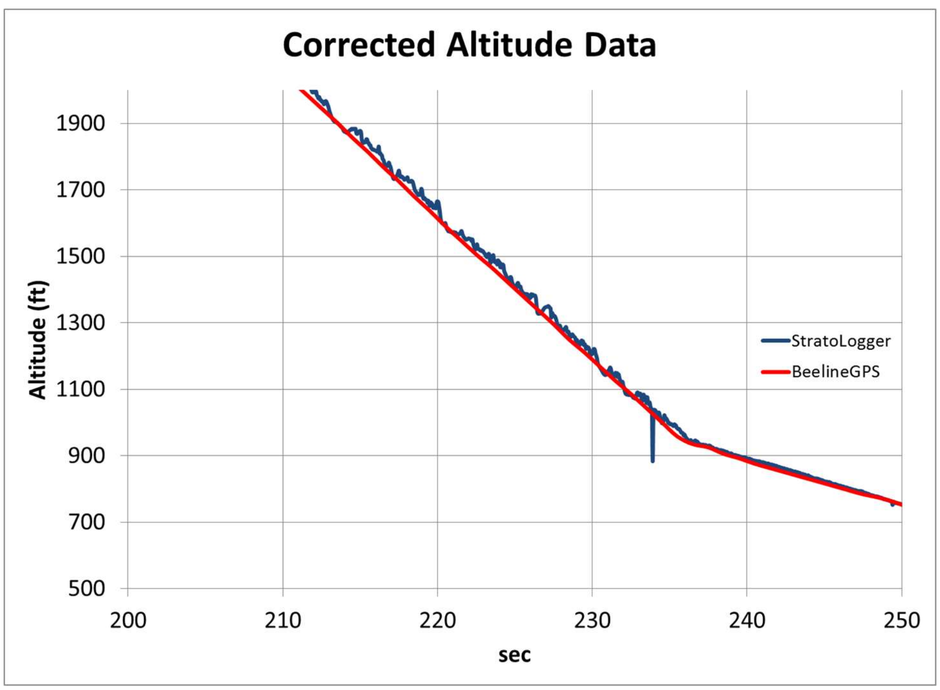

Figure 64 - Third flight of Speedmotion 54 – detail at main deploy

Looking at Figures 65 and 66, the BeeLine GPS delay is even greater than in the previous flight. The delay starts at 5 seconds, and then continuously converges until it catches the StratoLogger shortly after apogee. The initial acceleration of this flight was around 15 g's, so it appears the GPS delay is a function of the acceleration. At less than 10 g's, the BeeLine GPS can keep up, but above 10 g's, it falls behind by an increasing amount with increasing acceleration. But when used in conjunction with the StratoLogger, the ascent portion of the flight can be accurately captured.

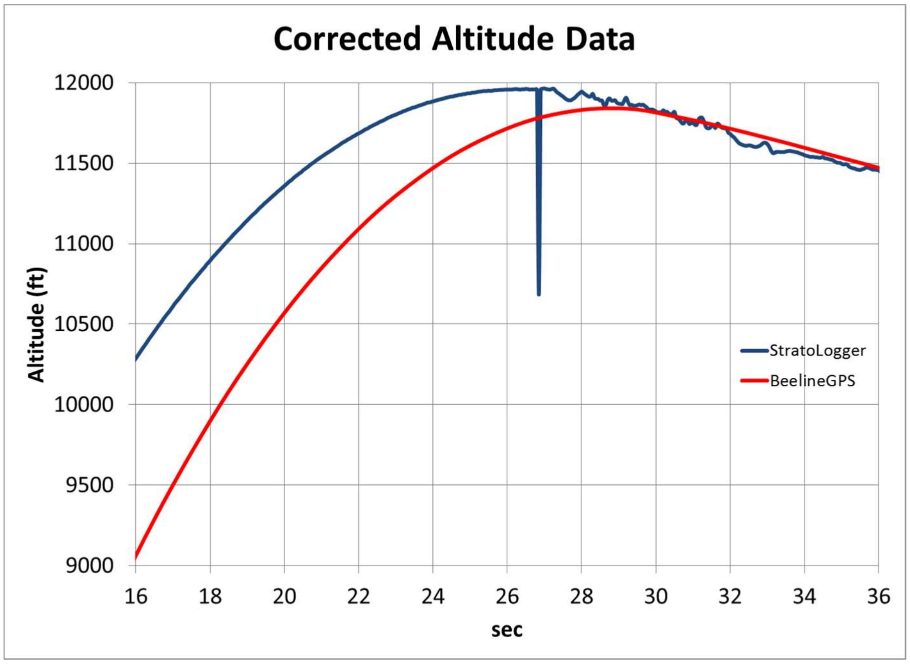

Figure 65 – Third flight detail at apogee

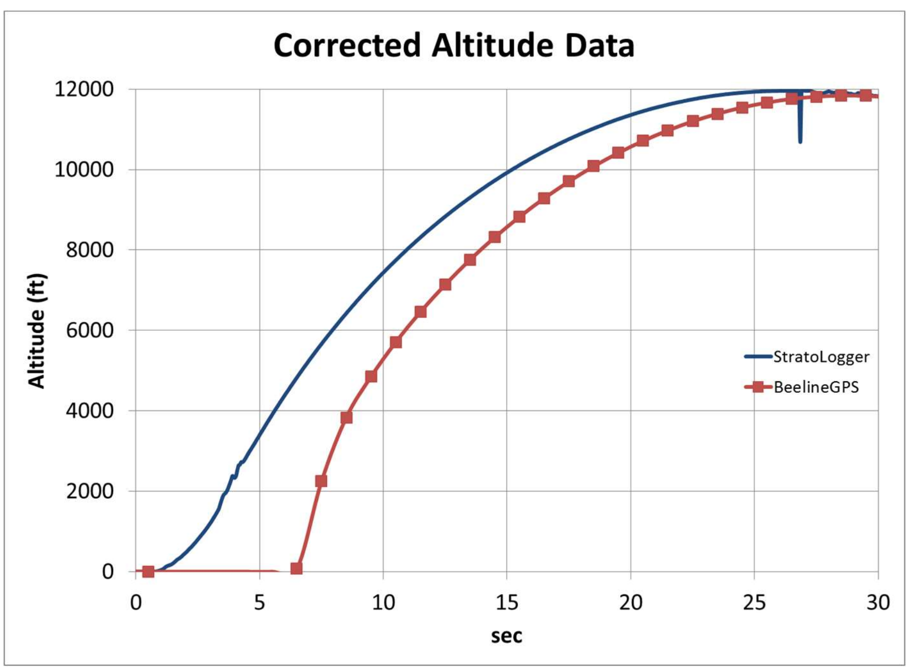

Figure 66 - Third flight of Speedmotion 54 – detail of ascent

Looking at Figure 63 closely, another anomaly can be seen. At 150 seconds, the slope of the altitude data decreases when the rocket reaches 5000 feet altitude, and remains on this new slope until the main deploys. What happened? At this point in the flight, the motor/fin section of Speedmotion 54 separated from the main body/nosecone section of the rocket. Speedmotion deploys the drogue rearward, separating at the motor/fin section, and the main parachute is deployed forward, separating at the nosecone (see figure 61). For this flight, drogue deployment occurred perfectly at apogee. The rocket descended to 5000 feet where the knot that attached the shock webbing to the u-bold of the forward bulkhead of the motor/fin section came loose, and the motor/fin section dropped free. Descending under drogue, but without the weight of the motor/fin section, the descent rate slowed. The main deployed normally, and the main section continued its descent normally. I was following the flight altitude real-time on the GPS readout on my Kenwood TH-D7A, and everything looked normal. The rocket had gone beyond visual range on the way up, but I knew what direction to look from the real-time waypoint on my Garmin. When the 48 inch main deployed, I picked up the rocket visually once again, at about a half mile in distance. When I arrived at the landing site, still not realizing anything had gone wrong, everything looked fine at first. Speedmotion 54 was laid out in a straight line in the alfalfa, as usual, with the drogue and main nicely deployed. I walked from the nosecone down the line, past the main section, drogue, and then to where the tail section should have been. My stomach sank. I saw the end of the shock cord, with its cleanly cut and sealed end, but without the motor/fin section attached, and not anywhere in sight.

Finding a rocket in a field of alfalfa is not easy, even though it is less than 2 feet tall. The alfalfa provides a nice cushion on landing, but rockets settle down into it below visual range. Realizing what had happened, I took the main section of the rocket back to camp, and downloaded both the StratoLogger and BeeLine GPS data. The StratoLogger data immediately showed the altitude at which the motor/fin section had let go. From the BeeLine GPS plot in Google Earth, I could see where rocket was when it was at 5000 feet altitude. This gave me the rough location where to look for the motor section, assuming it had dropped straight down from 5000 feet. I headed back out to where I expected to find the booster section. I was lucky, though, as another flier had already found my booster while looking for his rocket. Unfortunately, the booster had come in stably, and the front of the booster was damaged on impact. Luckily, the motor casing was undamaged, but the motor/fin section will have to be re-built.

What caused the knot to fail? I've never had a nylon webbing shock cord knot come loose before. I tie the webbing by wrapping the webbing back around itself multiple times and passing the end back under the loop formed at the first wrap. Then I lock the webbing at this point with CA glue, and tie an extra knot in the end of the webbing, just to be sure the end cannot pass back through the main knot in case the knot slips. I this case, I had not used enough CA glue to lock the knot, and I had failed to tie the safety knot. Lesson learned. I have now switched to using a Figure-Eight-Follow-Through knot, also known as the Flemish Bend, which is a much more secure knot.

Summary

In this series of articles, I have shown how to import the data recoded by a PerfectFlite StratoLogger and a BeeLine GPS into an Excel spreadsheet and analyze the flight trajectory of a rocket, and then compare the measured results to the RockSim flight simulation. Seeing the results overlaid on the same graph makes it much easier to align and compare the data. There is much more information in this data than can be seen in the basic default plotting tools, and once the data is in a spreadsheet like Excel, where it can be conveniently manipulated, filtered, and plotted, the data reveals a more detailed and accurate record of the rocket's flight.

Useful References

PerfectFlite DataCap: http://www.perfectflite.com/Download.html
BeeLine GPS Data Communicator: http://www.bigredbee.com/BeeLineGPS.htm
RockSim: http://www.apogeerockets.com/Rocket_Software
My Excel spreadsheet: http://www.Speedmotionrockets.com