Flight Prediction Software for High Altitude Balloons

Aaron Ridley* Nathan Hamet Jason Xie

University of Michigan, Ann Arbor, MI, 48109-2143, USA

Paul Webb†

Warren, MI, 48088, USA

Software to predict the flight path of a high altitude balloon has been created and made available to the community. The code takes inputs including the size of the balloon, how much helium is added, the size of the payload, the parachute diameter, and the launch location. The code then finds the closest weather station and gathers predictions of the weather over the following 5 days. When the time of the launch is specified, the code calculates the lift and ascent rate of the balloon, and propagates it using the wind data from the weather model. The size of the balloon is calculated at each time step to see if it has expanded to its maximum capacity, at which point it bursts. The descent model takes into account the size of the parachute and determines the terminal velocity of the payload train. It propagates the package through the weather model also. There are a number of novel innovations: (1) ensembles can be run to determine how errors in the amount of helium, burst diameter and wind speeds can change the landing location; (2) a burst altitude or time-delay can be specified to allow for flight termination units; (3) the weather station location can be dynamically updated; (4) zero-pressure balloons can be simulated; and (5) the code can be run in real-time, allowing for continuous downloading of the balloon's real-time position, which specifies the wind speed as a function of altitude. This allows a more accurate specification of the landing location after burst. Examples of balloon flights are presented in order to give an idea of the functionality and accuracy of the model.

I. Introduction

High-altitude balloons have gained in populatity over the past few years. There are many reasons for this, including the ability of smart phones to, in essence, act as a balloon payload, with integrated Global Positioning Satellite (GPS) chips that can report their position; an integrated communication system; and a camera that can take both video and still pictures with very good quality. One of the main issues with flying a high-altitude balloon is that the balloons can travel over a hundred miles from the launch site, depending on many factors. This makes predicting the approximate possible landing site quite difficult.

Michigan is a state that has many great features, such as large lakes surrounding the state, many smaller lakes scattered around the state, forests that cover a large portion of the state, and a large metropolitan area with an international airport. All of these features are great for tourism, but are quite difficult to deal with when launching a high-altitude balloon. Indeed, there are only a few areas in the state that are really "safe" to land in, meaning that they are mostly farm land with very few trees. One such area is in the southeast section of Michigan.

In addition, the jet stream over Michigan tends to flow from west to east, which means that if a balloon were launched from Ann Arbor, it would most likely land in Lake Erie or Canada, neither of which are acceptable. Therefore, balloon launches have to occur on the western side of the state, with landings in the east. The University of Michigan does not have a launch site for balloons, such as a fixed airplane hanger or other large structure, which means that most launches tend to occur outdoors, primarily at elementary, middle and high schools in the western part of the state.

*Professor, Atmospheric, Oceanic and Space Science, 1416 Space Research Building, University of Michigan, Ann Arbor

†

With such limited places to land, it is quite important to choose a good launch site. This is quite difficult to do without having good flight prediction software. While some good balloon prediction software solutions exist, it was felt that many of them were too restrictive or too cumbersome to run the multiple simulations that are needed in preparing for a launch. Therefore, it was decided to create new balloon prediction software that could be run online. This paper describes the flight prediction software that was developed at the University of Michigan

II. Methodology

Flight prediction can be separated into a horizontal component and a vertical component, each of which can be treated roughly independently. In the vertical direction, a force balance equation is solved, providing the terminal ascent or descent velocity. In the horizontal direction, predicted winds are used to propagate the balloon in longitude and latitude. Results from a weather model are used to provide the pressure, temperature, wind speed and wind direction as a function of altitude.

A. Ascent Module

The ascent speed of a balloon is dependent on the amount of gas that is used to provide lift.[1] At the University of Michigan, we use Helium because of safety concerns with Hydrogen. Since Helium is delivered in K-sized tanks, the amount of Helium is input as the number of tanks. From measurements, K-sized tanks typically are pressurized to $1.45 \times 10^7 Pa$ and have a volume of 0.438m^3 [a]. Given a room temperature of $294.3K$, the number of atoms of helium in one K-sized tank is approximately 1.57×10^{27}, which is 10.5 kg of helium.

Figure 1 shows the balance of forces on ascent, which is described by:

$$\mathbf{F_b} + \mathbf{F_g} + \mathbf{F_d} = 0, \tag{1}$$

where $\mathbf{F_b}$ is the Buoyancy force given by:

$$\mathbf{F_b} = NM_{He}\mathbf{g} - NM_O\mathbf{g}, \tag{2}$$

with M_{He} being the mass of helium, M_{air} the mean mass of air, $\mathbf{g}$ being gravity and N the number of atoms of helium added to the balloon; $\mathbf{F_g}$ is the weight of the payload train and the balloon:

$$\mathbf{F_g} = (M_b + M_p)\mathbf{g}, \tag{3}$$

with M_b and M_p being the mass of the balloon and payload train, respectively; and $\mathbf{F_d}$ is the drag force:

$$\mathbf{F_d} = \frac{1}{2} A_b C_{db} \rho U_a^2, \tag{4}$$

with A_b begin projected area of the balloon, C_{db} being the drag coefficient of the balloon (set to 0.5), ρ begin mass density of the air, and U_a begin the ascent speed of the balloon. Equation 1 can be expanded with the above equations to derive:

$$N\mathbf{g}(M_{He} - M_O) + (M_b + M_p)\mathbf{g} + \frac{1}{2} A_b C_{db} \rho U_a^2 = 0. \tag{5}$$

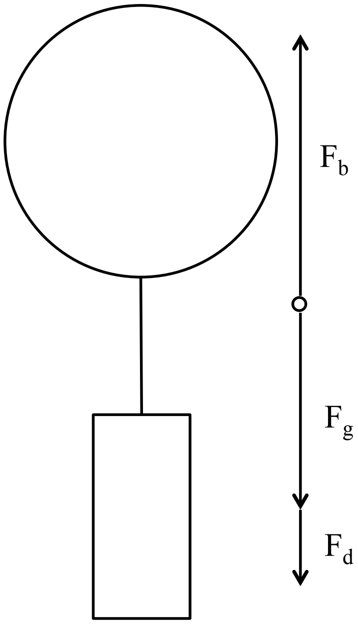


Figure 1. The force balance of the balloon on ascent.

The ascent speed can be solved for, to get:

$$U_a = \sqrt{\frac{2\mathbf{g}[N(M_O - M_{He}) - (M_b + M_p)]}{A_b C_{db} \rho}}. \tag{6}$$

[a] http://en.wikipedia.org/wiki/Gas_cylinder

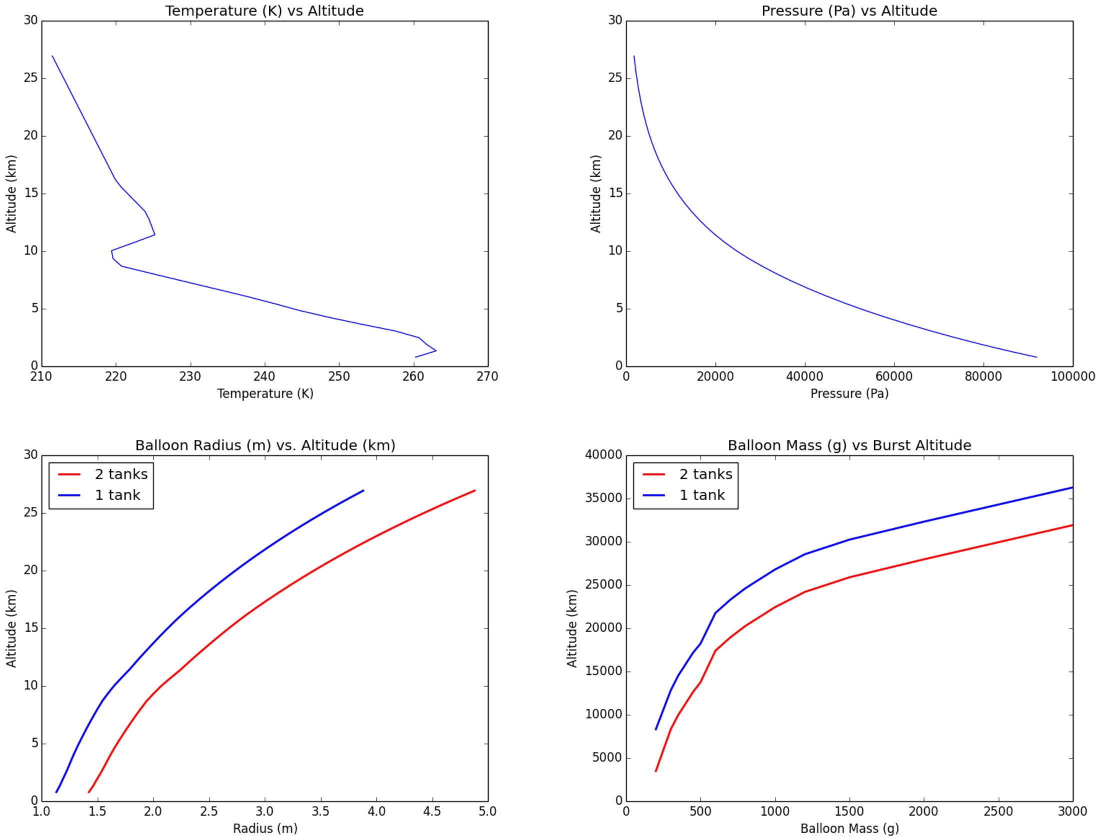


Figure 2. Atmospheric temperature (upper left) and pressure (upper right) predicted by the NAM model on February 15, 2015 at 15 UT over 42.31° latitude and -85.20° longitude. The radius of a balloon for these atmospheric conditions as a function of altitude with one or two tanks of helium added is shown in the lower left. The lower right shows the burst altitude of different sizes of balloons if one or two tanks of helium were added and the balloons were launched on this particular day at this particular time.

The only variables that change as a function of altitude in this equation are the projected area of the balloon and the mass density of air (ρ). The area of the balloon is derived from the volume of the balloon (V_b), specified by: $V_b = NkT/p$, where k is the Boltzmann Constant, T is the air temperature inside the balloon, assumed to be the local atmospheric temperature and p is the local atmospheric pressure. Figure 2 shows an example vertical profile of the atmospheric temperature and pressure from the North American Mesoscale Forecast System (NAM) model over Michigan. The volume of the balloon can be calculated as a function of altitude, given the atmospheric pressure as a function of altitude. The NAM atmospheric model is based on the Weather Research and Forecasting Model.[2,3] The model results are downloaded automatically from the Penn State University's Bufkit Data Distribution System.[b]

Given the atmospheric conditions, the radius of the balloon (r_b) can be determined, with the approximation that the balloon is a perfect sphere, such that $V_b = \frac{4}{3}\pi r_b^3$ or $r_b = (\frac{3V_b}{4\pi})^{0.33}$. The radius of a balloon filled with one (blue) or two (red) tanks of helium as a function of altitude is indicated in Figure 2 (lower left). This calculation assumes that the pressure inside the balloon and the pressure outside of the balloon are in perfect balance with each other. The radius is subsequently used to determine the projected area of the balloon on its ascent (i.e., $A_b = \pi r_b^2$) to determine the drag on the balloon and to determine whether the balloon has become larger than its burst radius, which is specified by the manufacturer. Table 1 provides the burst diameter (twice the radius) of many different balloon sizes that are typically used in balloon launches, as well as their prices. The lower right plot in Figure 2 shows the burst altitude of different sizes of balloons for the particular atmospheric conditions if one (blue line) or two (red line) tanks of helium were added to the balloons. This type of plot is very useful for determining the size of a balloon, given the altitude that

[b] e.g., http://www.meteo.psu.edu/bufkit

the user would like to reach.

Table 1. Burst diameters and costs of latex weather balloons

Balloon Mass	Burst Diameter	Estimated Price*
200g	3.00m	\$20
300g	3.78m	\$30
350g	4.12m	\$35
450g	4.72m	
500g	4.99m	
600g	6.02m	\$40
700g	6.53m	
800g	7.00m	\$80
1000g	7.86m	\$90
1200g	8.63m	\$110
1500g	9.44m	\$120
2000g	10.54m	\$250
3000g	13.00m	\$400

http://www.scientificsales.com/Meteorological-Weather-Sounding-Balloon-s/25.htm

Figure 3(top) shows the predicted ascent rate given Equation 6 as a red dashed line for a 1500g balloon with two tanks of helium carrying six pounds of payload for the atmospheric conditions shown in Figure 2. This plot shows that the ascent rate is predicted to start around 6.5 m/s. From experience, ascent rates greater than 5 m/s and lower than 7.5 m/s are considered optimal, since they balance the amount of time the balloon will spend in the jet stream and the altitude reached. With less helium added, the balloon will have a slower ascent rate and will ascend to a higher altitude, but will take more time to reach altitude and will spend more time in the jet stream. With more helium added, the balloon will pass through the jet stream faster and will reach its lower burst altitude faster. When the jet stream is not over Michigan, the balloon is filled less to reach a higher altitude. If the jet stream is strong over Michigan, extra helium is added to reduce the flight time.

The ascent rate shown in Figure 3(top) increases with time and altitude. This is because the mass density of the atmosphere decrease exponentially with altitude, and it is in the denominator of the vertical velocity equation. Also in the denominator is the area, which increases with altitude, but at a slower rate than the mass density. The ascent rate is predicted to almost double over the flight.

The thin blue line indicates the measured ascent rate for the balloon flight that occurred on February 15, 2014. This rate was determined by taking the difference in GPS measurements of the altitude every approximately 120 seconds. There is a significant amount of noise in the data, so a three-point smoother was applied (thick blue line). The balloon often passes into the jet stream below or around 10 km altitude and out of the jet stream at around 20 km altitude or so. As the balloon passes through these wind shears, it can experience massive amounts of turbulence which can cause spurious GPS measurements for some period of time. These could be caused by the main package being inverted at times, such that the GPS antenna is pointed towards the Earth instead of towards space. The GPS could (in theory) receive reflected signals off the Earth, which may confuse the navigation solution. The loss of GPS lock is often observed at burst also, since the packages can once again become inverted before the atmospheric drag torques can right the payload train.

The measured ascent rate is observed to increase with altitude for approximately 30 minutes. For the next 10-15 minutes, the ascent rate stays somewhat constant. 45 minutes into the flight, the ascent rate shoots up to greater than 10 m/s, then decreases to around 6 m/s, then starts to slowly climb again. It is clear that the simple force balance model (red dashed line) does not perfectly match the ascent rate profile. The dashed yellow line shows what the ascent rate would look like if it were held constant with time, which is closer to the truth in the later part of the flight, but is off in the middle of the flight. In the code that is operational on the web site, it was chosen to assume a constant ascent rate, since the predicted ascent rate is often well above the observed ascent rate.

It is unclear why the ascent rate may be significantly slower than the predicted ascent rate past 50 minutes of flight. One could imagine that the payload train may act as an increased drag on the balloon, but considering that the balloon would have a radius of approximately 3 meters or more (i.e., an area of approximately 30 m^2) at these heights, it is hard to imagine that the tiny payloads would add much to the drag. Also, it could be that the balloon is not a perfect sphere, and pancakes a bit, such that both the area and the drag coefficient increase. These would both cause a decrease in the ascent rate, but it is not clear why this might happen in the stratosphere only, instead of all the way up.

The second flight, shown in the bottom plot of Figure 3, shows a flight that took place on January 18, 2014. In this case, the measured ascent rate stayed almost constant for the first approximately 17 minutes of the flight (the first point is bad because the timing of the launch versus when the first point in the air was measured). The measurements are then quite random for the following 10 minutes. At around 27 minutes, the ascent rate increases gradually for about 10 minutes, then levels again at just over 8 m/s. When comparing the predicted and the actual ascent rates, differences are clearly observed, such as the actual ascent rate being relatively constant for long periods of time, but having some randomness to it. The ascent rate observed in this launch is also quite different than the ascent rate observed in the other launch. This once again provides some justification for using a constant ascent rate instead of attempting to predict the actual ascent rate.

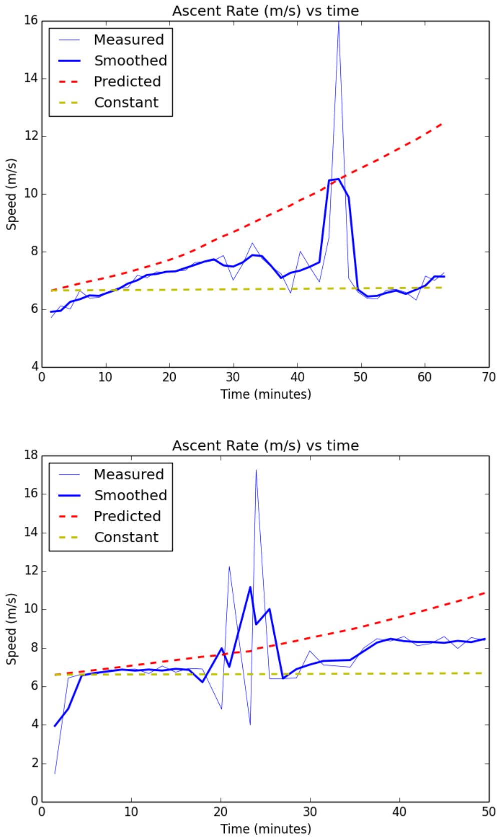


Figure 3. Ascent rate predictions (red dashed line) and measurements (blue solid lines) for two different launches. The top case is for the launch on February 15, 2014, while the bottom is for the launch on January 18, 2014. Both cases assume 2 tanks of helium with a six pound payload and a 1500g balloon. The thick blue line is the same as the thin blue line with a 3-point smoother applied. The dashed green line is assuming the ascent rate is constant.

For the flight prediction simulations, the ascent rate is calculated at the initial time, given the ground altitude and the atmospheric conditions. A vertical distance is calculated given a time-step of 10 seconds and the ascent rate. The balloon is moved upwards by this distance, and the steps are repeated until the balloon reaches a diameter that is larger than the published burst diameter. At that point, the code switches over to descent mode.

B. Descent Module

The descent module works through a force balance equation also. In this case, the terminal velocity (U_d) of the package is calculated assuming a parachute area (A_p):

$$U_d = \sqrt{\frac{2\mathbf{g}M_p}{A_p C_{dp} \rho}}, \tag{7}$$

where C_{dp} is the drag coefficient of a parachute, which is assumed to be 1.5 and r_p is the radius of the parachute, so that the area is given by $A_p = \pi r_p^2$. The mass density (ρ), mass of the payload (M_p) and gravity ($\mathbf{g}$) are as described above. One assumption that this formula makes is that the balloon shreds completely or is cut away from that parachute. If this is not the case, then an estimate needs to be made of how much of the balloon will be left once it has burst, and that mass should be included in the formula:

$$U_d = \sqrt{\frac{2\mathbf{g}(M_p + CM_b)}{A_p C_{dp} \rho}}, \tag{8}$$

where C is some fraction between 0-1. From experience, C is often less than 0.5, but is more than 0.0. The online code assumes C=0, which can add errors.

Figure 4 shows example descents for the launches shown in Figure 3. For the top case (February launch), the descent rate starts out at approximately -35 m/s (roughly 80 miles per hour), but rapidly slows to a rate of approximately -5 m/s. The prediction has a slower descent rate than the actual payload almost the entire time. This was noticed during most launches that were performed at the University of Michigan, and could be due to many different things, such as the balloon not completely shredding, as described above, and the parachute not fully deploying due the lines becoming tangled or the payload lines wrapping around the parachute during the turbulence surrounding the entry and exist of the jet stream and just after the balloon burst. In order to compensate for the faster than predicted descent rate, which was observed on almost all launches, the parachute area is reduced by a factor of three. When this is done, the yellow dashed line is derived. This curve matches the observed descent rate significantly better than the other prediction that uses the actual parachute area. A significant improvement in the landing site prediction has been observed since this was implemented.

The bottom plot shows the same trend - the prediction with the actual parachute area is significantly worse than the prediction with the reduction of the area. In this case, the actual descent rate started around -25 m/s and reduced to approximately -5 m/s, which is quite close to the predicted descent rate with the modified area. The actual descent rate appears to stay much larger than either predicted value for the first roughly 10 minutes of the descent. It is unclear why this would be the case, except that the GPS may have significant errors during this first part of the descent, due to the rapid rotation and tumbling of the packages immediately after the balloon burst. The uncorrected prediction (red line) dramatically underpredicts the descent rate, while the area-corrected prediction does a significantly better job.

The descent prediction works exactly the same as the ascent, in that the descent rate is determined, a 10 second time step is taken, the position is updated, and the atmospheric conditions are determined for that altitude. The descent rate is calculated again and the cycle continues until the position is below the launch altitude (i.e., the surface of the Earth, which, in Michigan, is approximately 300m, but is specified in the atmospheric results file).

C. Horizontal Transport

The wind speed (U) and direction (Θ) as a function of altitude are provided in the atmospheric weather model output. These are used to calculate the horizontal trajectory of the balloon, assuming that the balloon moves with the wind. The eastward (U_e) and northward (U_n) velocities are determined using the equations:

$$U_n = U \sin(\frac{3\pi}{2} - \Theta) \tag{9}$$

and

$$U_e = U \cos(\frac{3\pi}{2} - \Theta). \tag{10}$$

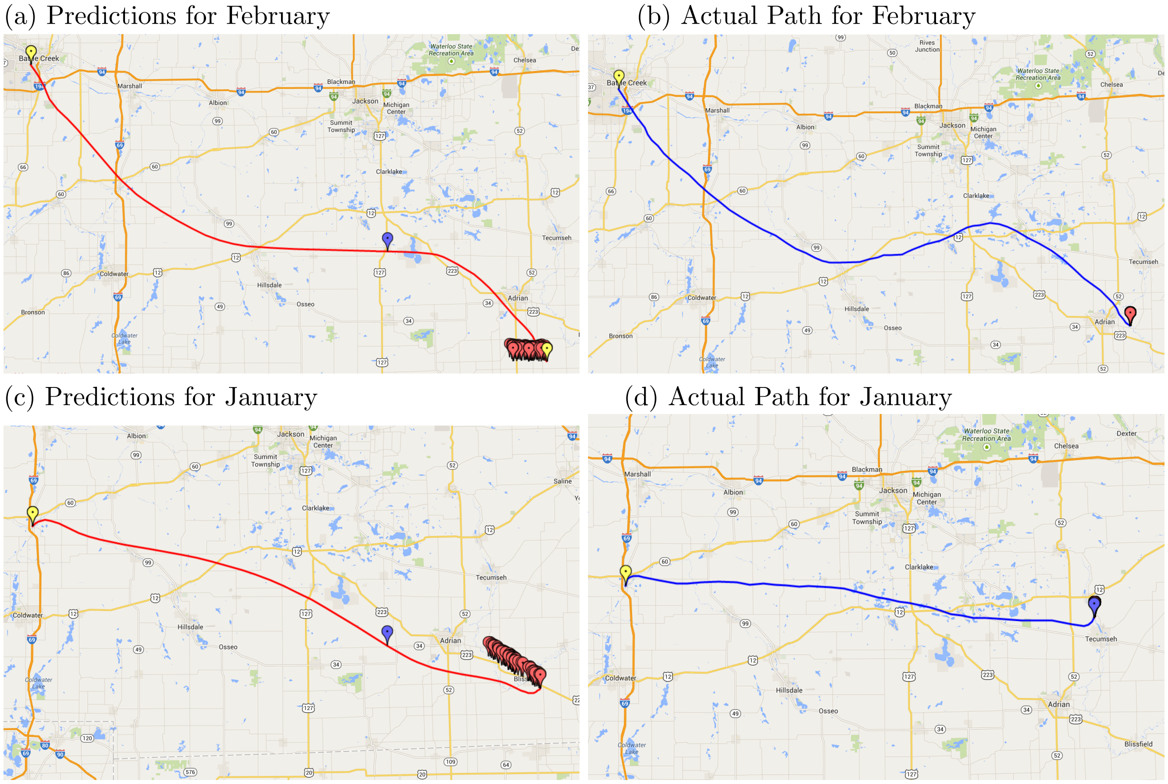
The latitude (ϕ) and longitude (λ) of the balloon at time i are then updated using the equations:

$$\phi_i = \phi_{i-1} + U_n \Delta t \frac{360°}{2\pi(6372 + H_i)} \tag{11}$$

and

$$\lambda_i = \lambda_{i-1} + U_n \Delta t \frac{360°}{2\pi(6372 + H_i)\cos\phi_i}, \tag{12}$$

where H_i is the current height of the balloon (in km).


(a) Predictions for February

(b) Actual Path for February

(c) Predictions for January

(d) Actual Path for January

Figure 5. Predictions (left) and actual flight paths (right) for the February 2014 (top) and January 2014 (bottom) launches. The yellow pin shows the launch site, the blue pin shows the burst location, and the cluster of red (and one yellow) pins show possible landing locations. On the right, the yellow pin marks the launch site, while the pin other pin indicates the actual landing site. These are direct outputs from the web page.

Figure 5 shows predictions and actual flight paths for the two flights that have been discussed within this study. For the February flight, the predictions showed that the balloon would most likely land approximately 5 miles south of Adrian. The balloon ended up landing just to the east of Adrian, so the error was approximately six miles. In January, the weather model predicted winds that were more southward-directed than actually occurred, so the balloon ended up significantly north of the predicted landing site. The error in this case was over 10 miles.

While these types of errors would make it impossible to find the balloon without tracking and communication hardware, these types of predictions inform the team greatly on optimum launch locations. From the initial predictions, it is clear that the team chose launch locations to land near Adrian, Michigan. In the February case, this meant launching in Battle Creek. In January, the launch location was in Tekonsha, which is southwest of Battle Creek (the balloon nearly flew over Techonsha in the February flight).

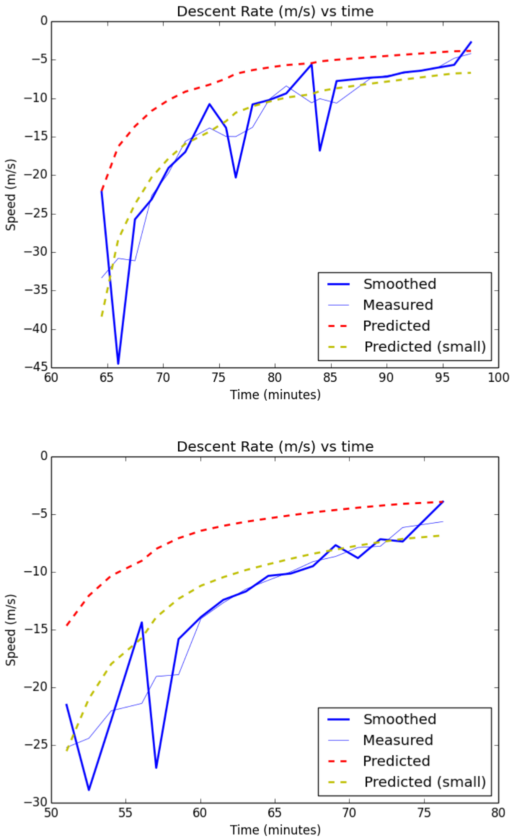


Figure 4. Descent rate predictions (red dashed line) and measurements (blue solid lines) for the two dif-

III. Additional Features

While the prediction software above is quite useful for determining a good launch site for the optimal landing position, there are many other features that were determined to be quite useful for balloon launches. In this section, these features are described.

A. Flight Termination

One of the first modifications of the code was to add the ability to artificially terminate the flight. This was a relatively simple modification of the code, in that the user can enter either a maximum altitude or a maximum time. If the code is still in the ascent mode when either criteria is met (time or altitude), the balloon “bursts” and the code switches to descent mode.

The main reason for adding this feature is that the University of Michigan has launched many balloons simply to test the reliability of a flight termination unit, which is a Federal Aviation Administration regulation for some flights, and comes in quite useful when the balloon does not follow the exact track of the prediction software and may move towards an area that would be negatively affected by a balloon landing (e.g., Detroit Metro Airport).

B. Dynamic Updating of the Weather Station

The atmospheric model results that are used for the flight prediction are available over the web at fixed weather station locations. In the past, the software used the station closest to the launch site to describe the atmospheric conditions. Recently, the ability to dynamically change which station is used has been implemented. The code does this by checking the list of stations each time-step in the prediction and using the closest station. If a new station is closer, the code automatically downloads the closest station and uses this for the conditions. This constant checking of the distances to the stations slows down the code a bit (so it can be switched off), but is very good for long-duration flights, since the wind patterns can change significantly over lengths of 50 miles or so.

C. Ensemble Predictions

There are many uncertain variables in the equations described above. Some examples include:

1. the wind speed is not truly know, since it is a model prediction;

2. the amount of helium added to the balloon is also questionable, since each tank is not precisely 10.5 kg of helium, but seems to vary by 10% or so, and if fractions of tanks are added, significant error can result;

3. the burst diameter of the balloon seems to have some variation, which is observed by the balloon bursting at altitudes that can be significantly different than the predictions; and

4. the parachute area is well known, but the amount of tangle in the lines and openness of the parachute can be uncertain, which can be “controlled” in the equations by putting an uncertainty on the area.

In order to capture some of this uncertainty, the software allows simulation of many balloon trajectories with different levels of uncertainty added to the quantities described above. This results in clusters of points near the landing site, as shown in the left two maps in Figure 5. The software on the website offers different levels of uncertainty, ranging from 5% to 20% and different numbers of simulations, ranging from 5 to 200.

One of the flaws in this ensemble calculation is that the errors are added as percentages on the original values. This is fine for always positive values, such as the area of the parachute or burst diameter of the balloon, but the wind speed (or one component of the wind speed) can be close to zero, which means that there is very little error added. For the wind speed, a random offset needs to be added that is not based on the actual wind speed. For example, a 2 m/s wind speed error could be added in a random direction. This needs to be implemented in the software.

Also, as can be observed from Figure 5, the ensemble coverage area doesn't actually include the landing sight, which means that there are larger errors than those described above. This could be a problem with way the error in the winds is handled, or it could be something unaccounted for. Further, it is clear that the ascent rate and descent rate are not perfectly modeled and possibly have significant variability that is not captured in the model. It is future goal to better describe the uncertainty in the calculation such that the landing site is often included within the scatter of points described in the ensemble calculation.

D. Zero Pressure Balloon

The model described above is specifically for latex balloons that are sealed, such that as they ascend into the atmosphere, they expand. At some point, the latex is stretched too thin and the balloon shreds. Another type of balloon is a zero pressure balloon. These balloons have a hole in the bottom and start out mostly empty. As the balloon ascends, the helium expands and fills the balloon.[4] At some altitude/pressure, the volume of the balloon is completely filled by the helium. As the balloon ascends higher than this, the helium starts to leak out of the hole in the bottom of the balloon, and the lift decreases. At some altitude/pressure, enough helium will leaked out of the balloon to reduce the lift to zero, at which point the balloon becomes neutrally buoyant. In theory, the balloon should float at this altitude for a significant length of time. Practically, there are small leaks in the balloon, either at the hole in the bottom or through microscopic tears in the membrane. Further, after dusk, the temperature of the air surrounding the balloon can drop rapidly, while the temperature inside the balloon can lag behind the external temperature drop. This will cause two things to simultaneously occur: (1) an increased pressure inside the balloon will result, leading to helium being expelled from the balloon, significantly reducing lift and (2) a decrease in the mass density of the air surrounding the balloon, which further decreases lift. These two combine to cause the balloon to rapidly descend.

Figure 6 shows predictions from the model for a zero pressure balloon launch that was performed by the University of Michigan on April 12, 2014. The code predicted that the balloon would rise to approximately 22 km altitude and stay there. Because it would be at a fixed altitude, the weather model was constant during this time, resulting in a very steady speed and direction. The zero pressure balloon, in this case, had a volume of $111m^3$, a radius of $4.65m$, and a mass of 2.7 kg. The payload mass was approximately 3.2 kg and 3.5 tanks of helium were added. For the model, the flight was expected to be terminated at 180 minutes into the flight (due to the fact that just of the right side of the map in Figure 6 is Lake Erie).

Figure 7 shows the actual flight path with the altitude profile of the zero pressure balloon. The first part of the flight path was predicted very well, but the second half was quite poorly predicted. The quality of the prediction dropped when the balloon roughly reached its peak altitude. In the predictions, this peak altitude was approximately 22 km, while in actuality, the peak altitude was 19.3 km. The difference between these two altitude is not very much, but the edge of the jet stream happened to be very near 19 km that day. This meant that the predicted flight path went above the jet stream, clearly seen in Figure 6 as a strong kink in the flight path about halfway through, with the predicted path moving north-east instead of south-east, as was observed in the actual flight. Comparing the predicted flight path with the actual flight path in real-time showed that if the balloon kept flying in the direction that it was headed, the balloon would most likely end up in Toledo or Lake Erie. Therefore, the flight was terminated. Interestingly, with a zero pressure balloon, the flight termination does not fully disconnect the balloon from the payload train (as is done with a latex balloon); it inverts the balloon and rips the top. This end up shredding the balloon such that very little of the balloon is left when the payload train reaches the ground. This process seems to take a while, so the balloon seems to slowly descend for a couple of minutes before dropping rapidly.

Modeling the ascent and descent of a zero pressure balloon is difficult at this point, since the University of Michigan has not performed enough of the launches to provide clear data on how the balloon performs. It is clear that the ascent is not accurately modeled by the simulation; the simulation shows an acceleration of the balloon, which is much less prevelant in the actual flight. It was unclear what type of area and drag coefficient to use with the zero pressure balloon, since it is extremely distorted when launched (as shown in Figure 8). Therefore, the same formula for the latex balloon was used for the zero pressure balloon (Equation 6), assuming that the balloon takes a spherical shape that can be determined by the external pressure, and that the drag coefficient is 0.5. Once more data is collected on zero pressure balloons, the formula can be refined, possibly having a dynamic drag coefficient or an increasing drag area.

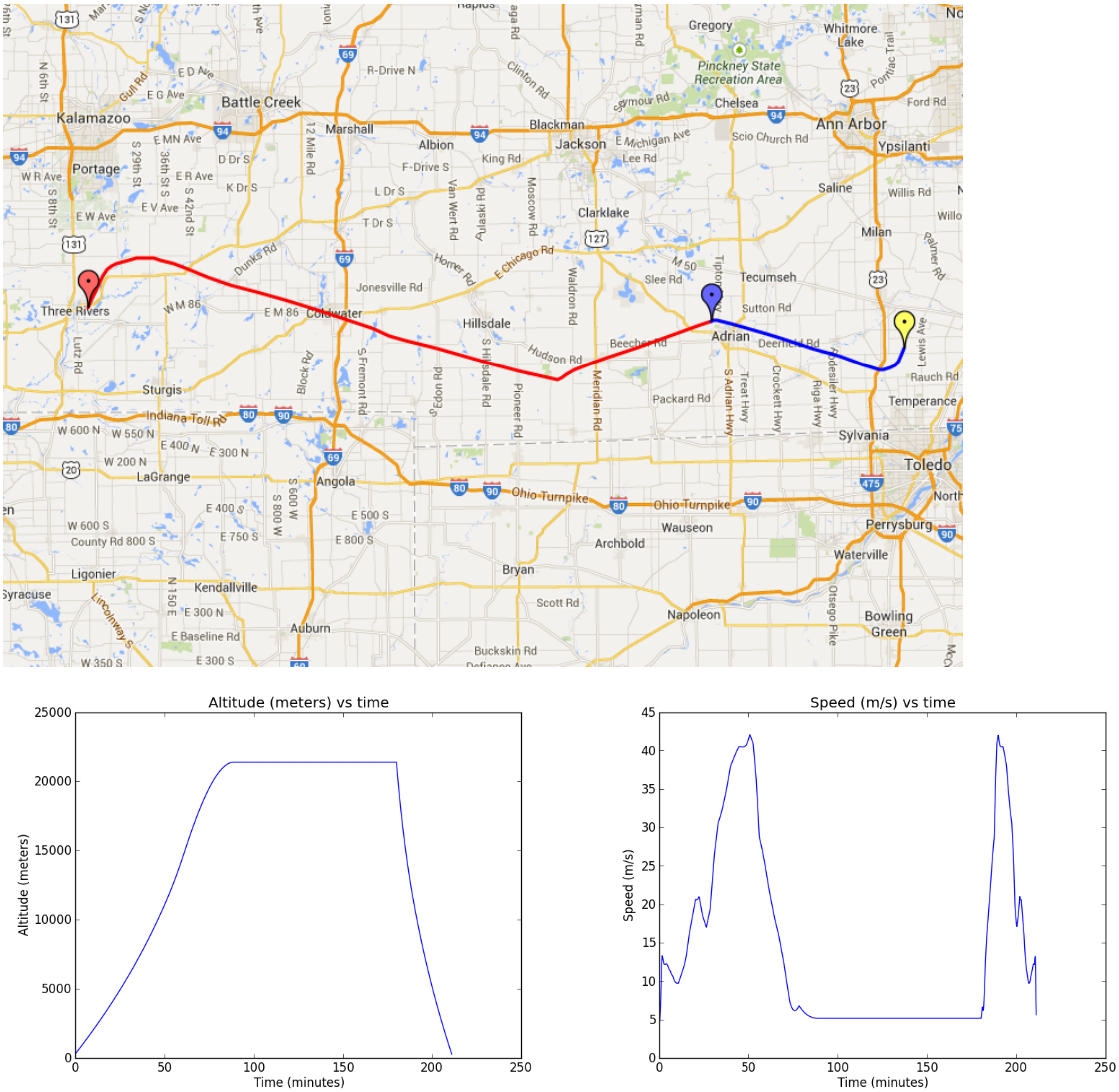


Figure 6. Predictions for a zero pressure balloon launch that was conducted on April 12, 2014. The top plot shows the horizontal path that was predicted, while the lower left plot shows the predicted altitude and the lower right plot shows the wind speed as a function of time that the balloon was predicted to encounter.

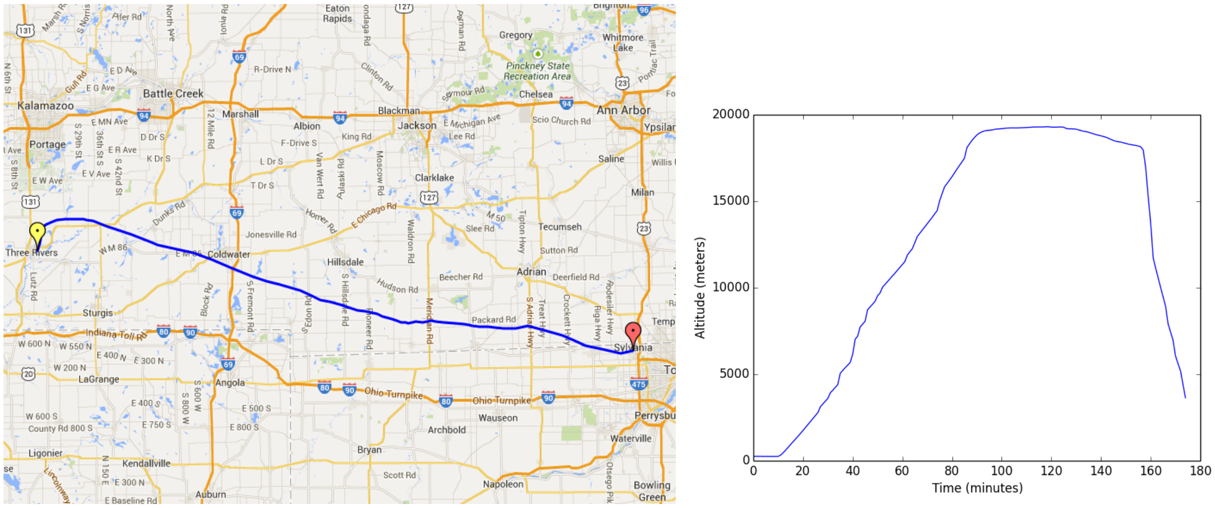


Figure 7. The left plot show the actual flight path of the zero pressure balloon launched on April 12, 2014. The right figure shows the altitude as a function of time for the balloon.

E. Real-Time Operations

Figure 8. A zero pressure balloon a couple of minutes after launch.

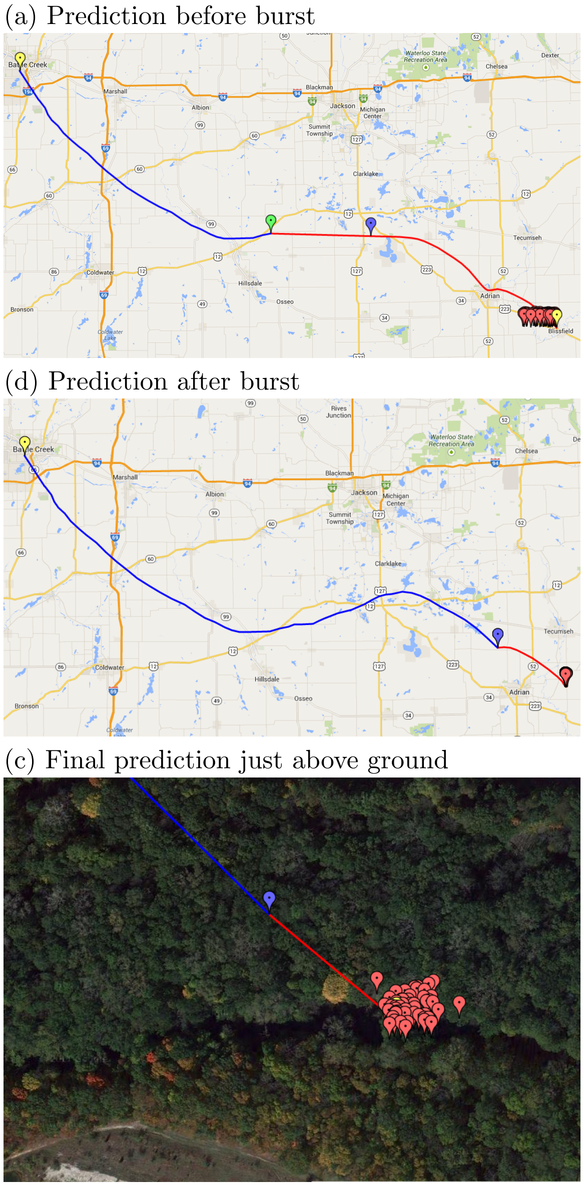
In the first version of the software (which was not easily available over the web), the ability to run in real-time was incorporated. The web-based software is being updated to include this, and it should be before the end of 2014.

As the balloon ascends and the user receives information about the position of the balloon, it is possible to improve the specification of the landing point in two ways: (1) with a better specification of where the balloon is currently located, errors in the landing site prediction are significantly reduced, especially towards the end of the flight, since the duration of prediction can be extremely short; and (2) by taking a time-history of the actual balloon positions, the wind speed and direction can be calculated and used for the rest of the prediction, which can improve the descent calculation of the model.

For the older software, two different processes were run at the same time: one process simply queried the http://aprs.fi/ website (APRS stands for Automated Position Reporting System) to get the most up-to-date position of the balloon every few seconds and stored this in a file; a second process read this file along with the weather model prediction and merged them to create a new atmospheric profile including wind speed and direction. The code also wrote out a full history of the balloon positions, simply so the prediction code could add these to the map. The prediction code read in the merged weather prediction as well as the balloon locations, and propagated the balloon through the rest of the flight. The methodology is currently being integrated into the web-based software, without two processes running and without the intermediate code to merge the weather prediction and balloon positions.

One of the complications in the real-time code is recognizing whether the balloon is ascending or descending. While this seems like a trivial task, it is complicated by the errors within the APRS system. Namely, packets from the balloon can get rebroadcast and reposted to the server, making the balloon seem like it has jump backwards in position. Further, during very turbulent times, the GPS position error can grow significantly, such that it is not clear whether the balloon has started its descent. Therefore, the software must observe three decreasing positions of the balloon before it triggers the descent mode in the real-time software.

Even with these caveats, the real-time prediction code has served to be extremely valuable. The primary reason for using it is to determine whether the flight should be terminated due to the balloon possibly landing in an undesirable location. This has occurred on at least three occasions. In addition, line-of-sign communication is often lost when the balloon descends to an altitude of about one km, depending on how close the chase team is to the balloon. When this occurs, the real-time prediction typically provides an extremely accurate location of the balloon, such that the recovery team can often walk into the field (if lucky) and recover with minimal searching.


(a) Prediction before burst

(d) Prediction after burst

(c) Final prediction just above ground

Figure 9. Real-time in-flight predictions based on balloon positions that were dynamically updated during the flight on February 15, 2014. The blue line in each plot indicates the path that the balloon had traveled, while the red path was the predicted path at the time. The green pin represents the position of the balloon at the time of prediction, while the blue pin represented the location of where the code switched to descent mode. In the last two predictions, the blue pin covers the green pin. The red pins represent potential landing sites given the ensemble simulation.

Figure 9 shows real-time predictions during the February 2014 launch, which can be compared directly to Figure 5. It is observed that for the prediction before the burst (a), the landing site is significantly closer to the landing site than the initial prediction. The prediction after burst (b), has moved the landing site a bit further away (northeast) of the actual landing site. Sadly, the prediction just above the ground (c) shows the balloon landing in a river bed surrounded by trees. The prediction was quite accurate and it took a couple of weeks to recover the balloon, with one of the University of Michigan staff climbing 100 feet up a tree to retrieve the balloon payload.

IV. Summary

This paper describes a flight prediction model for high altitude balloons. The model is available at http://vmr.engin.umich.edu/Model/_balloon. This model calculates the lift and ascent rate of the balloon given the size of balloon, amount of helium, mass of the payload train and atmospheric conditions. It calculates the descent rate based on these parameters and the diameter of a parachute. Horizontal motion is simulated using the wind speed and direction as a function of altitude provided by the atmospheric model. The model is accurate enough to allow determination of a good launch site for desired landing zone, but can sometimes be off by as much as 15 miles in the prediction, depending on many different factors. It should work anywhere on the Earth, but the accuracy is quite dependent on how close the flight path is to an available weather station. Therefore, it will work well in more populated regions of the planet, but might not be as accurate in which there are less people.

Additional features of the model include the ability to specify when a flight termination unit will detatch the balloon, use atmospheric simulation results that vary through the flight, use uncertainties in the parameters to determine uncertainties in the landing location, simulate zero pressure balloons and to operate in real-time in order to improve the specification of the landing site and make decisions on whether to terminate a flight.

This code has proved invaluable to the balloon flight program at the University of Michigan, which conducts between five and fifteen balloon launches per year, and is freely available to anyone to use to support their balloon launches.

References

[1] Warren, J., Smalley, J., and Morris, A., "Aerostatic Lift of Helium and Hydrogen in the Atmosphere," Tech. Rep. NCAR-TN/IA-69, National Center for Atmospheric Research, December 1971, NCAR Technical Notes.

[2] Michalakes, J., Dudhia, J., Gill, D., Klemp, J., and Skamarock, W., "Design of a next-generation regional weather research and forecast model," *Towards Teracomputing, World Scientific*, River Edge, New Jersey, 1998, pp. 117–124.

[3] Michalakes, J., Hacker, J., Loft, R., McCracken, M. O., Snavely, A., Wright, N. J., Spelce, T., Gorda, B., and Walkup, R., "WRF nature run," *Journal of Physics Conference Series*, Vol. 125, No. 1, July 2008, pp. 012022.

[4] Baginski, F., Collier, W., and Williams, T., "A Parallel Shooting Method for Determining the Natural Shape of a Large Scientific Balloon," *SIAM Journal on Applied Mathematics*, Vol. 58, 1998, pp. 961–974.