PATH PREDICTION FOR AN EARTH-BASED DEMONSTRATION BALLOON FLIGHT

DANIEL BEYLKIN

Mentor: Jerrold Marsden
Co-Mentors: Claire Newman and Philip Du Toit

August Progress Report

1. Progress

1.1. **Discrete Mechanics and Optimal Control.** NASA is planning on a mission to Titan and would like to use a Montgolfier balloon to help gather data. It is hoped that by optimally exploiting the winds on Titan, the balloon would be able to autonomously navigate different regions of Saturn's moon with minimal control or time of travel. Until the mission is launched, some experiments will be run on Earth to test methods that can be translated to solve the problem on Titan. If an approach is to succeed on Titan, it would be hoped that it succeeds on Earth, whose atmospheric models are much more robust and reliable due to a greater availability of accurate data and verifiable results.

Discrete Mechanics and Optimal Control (DMOC) is a tool that allows us to optimize trajectories satisfying a system's equations of motion. We are primarily concerned with the problem of a balloon autonomously navigating various regions of the Earth and appropriately exploiting the wind fields to minimize control requirements, fuel consumption, or time of travel. DMOC is the desirable tool to investigate such problem.

We are currently working on implementing DMOC to compute optimal trajectories for a balloon in a three-dimensional, time-dependent wind field. It is implemented in Matlab using fmincon, an optimization routine that finds a local minimum of a constrained, nonlinear multivariable objective function. It has the following inputs:

- Objective function to be minimized (also referred to as the cost function),
- Constraints which are dictated by the equations of motion for the balloon, and
- Initial guess of the optimal trajectory.

For the purposes of navigating a balloon, we are given its initial position, q_0, and initial velocity, p_0. We are also given the balloon's target destination, q_f, though we may or may not be concerned with its final velocity. In some circumstances we are primarily concerned with reaching the target and in other circumstances we are also concerned with staying there. Fortunately, these problems are very similar in nature and can be solved by the same approach.

We use a discrete trajectory with N time nodes between initial and final states such that $(N+1) \cdot h = t_f$, where h is a constant time step and t_f is the time of travel. Let us define q_i as the generalized position, p_i as the generalized impulse, and f_i as the control force. We want to find the set of $\{q_i\}$, $\{p_i\}$, and $\{f_i\}$ that satisfy the constraints and minimize the objective function. It is of note that we are choosing to not fix t_f so that the optimization will also search over trajectories with varying final time (i.e. it searches for an optimal h).

1.1.1. *Objective Function.* At the moment, we are primarily concerned with optimizing the amount of control required for the balloon to navigate from the initial to final point, however it is feasible to also optimize the time of travel or other parameters, depending on the problem at hand.

Currently, we are choosing to use the l^2-norm of $\{f_i\}$ as the measure of control, so we want to minimize

$$\sum_i f_i^2.$$

However, our approach is not limited to this problem. If we were to eventually choose a more sophisticated model of balloon dynamics, where fuel is an explicit parameter, we can instead try to minimize fuel consumption.

1.1.2. *Constraints.* Naturally, any optimal trajectory must obey physical laws and, thus, must satisfy the balloon's equations of motion. Initially, we choose a very simple model of balloon dynamics. The idea is that once it is successfully implemented, we will then employ a more realistic and sophisticated model.

The following describes an intuitive derivation of the model. If we consider the trajectory $\{q_0, q_1, \ldots, q_N, q_f\}$, we approximate the velocity of the balloon (in absence of wind) as

$$\frac{q_{i+1} - q_i}{h} = p_i,$$

where p_i are evaluated on a staggered grid, specifically at the midpoint between q_{i+1} and q_i. If we were to incorporate the influence of wind, we would naturally write

$$\frac{q_{i+1} - q_i}{h} = p_i + Wind(\frac{q_{i+1} + q_i}{2}, t_{i+\frac{1}{2}}).$$

Similarly, we approximate the control force as

$$\frac{p_{i+1} - p_i}{h} = f_{i+1}$$

where f_{i+1} are also evaluated on a staggered grid, at the midpoint between p_{i+1} and p_i (which one may note is where the q_i are located).

We are currently working on a more sophisticated model, which is described in detail in Section 3.

1.2. **Weather Research and Forecasting Model.** The influence of wind is a significant component of the constraint equations and heavily determine regions that the balloon can and cannot reach. Moreover, since we are investigating the problem of a balloon appropriately exploiting winds to reach high priority targets, it is important to obtain actual wind fields from the region of interest. To do so, we use the Weather Research and Forecasting (WRF) model. It is a numerical weather prediction system used for both research and forecasting applications. For my purposes, it is capable of providing wind velocities (i.e. speed and direction) at different latitudes and longitudes. As a simple example to demonstrate the WRF model, we consider a region of the Mojave Desert on July 5, 2005 at 12:00pm. We chose the Mojave Desert because it is the likely location of a balloon demonstration flight currently planned to take place in 2009. Figure (5.1) (see Appendix, Section 5) is a visualization of an output of WRF. It is a plot of the x-direction wind velocity at different sigma levels. Sigma is used as the vertical coordinate as opposed to altitude because it is preferred for ease of calculation in global climate models. It is the ratio of the pressure at a point in the atmosphere to the pressure of the surface of the Earth beneath it.

1.2.1. *Interpolating onto a Cartesian Grid.* Unfortunately, WRF gives the wind speeds on a curvilinear grid (Figure (5.2)), yet we would like it to have the wind speeds on a uniform, rectangular Cartesian grid (Figure (5.3)). To do this, we must first interpolate all the necessary data (which we will show are the wind speeds and the geopotential height) onto the grid. In this case, the interpolation is implemented as griddata in Matlab. We must also take into the account that the data is staggered on an Arakawa-C grid.

Additionally, we make a change of coordinates from latitude and longitude to Cartesian distance

$$x = \frac{\pi}{180^\circ} \cdot r \cdot \phi \cdot \cos(\lambda - \lambda_0), \tag{1.1}$$

$$y = \frac{\pi}{180^\circ} \cdot r \cdot (\phi - \phi_0), \tag{1.2}$$

where ϕ is latitude, λ is longitude, and $r = 6,378 \cdot 10^3 m$ is the radius of the Earth (for simplicity, we are assuming that the Earth is spherical). We note that (λ_0, ϕ_0) is the origin, which in the case of Mojave Desert, will be $(-116^\circ E, 36^\circ N)$.

Unfortunately, sigma coordinates complicate calculations for balloon navigation and thus it is necessary to perform a change of variable from sigma coordinates to altitude coordinates. This was done using interpolation, implemented as interp1 in Matlab. We use the geopotential height output from WRF divided by Earth's gravity to obtain the altitude. For a given latitude and longitude, we know the geopotential height (and hence altitude) and wind velocity at different sigma levels. We use interpolation to find the wind velocity at an altitude of our choice. For our purposes, we will need to perform such interpolation for many altitudes and for all times.

2. Problems

2.1. **The Initial Guess.** We note that fmincon is a local optimization routine and its convergence to "good" local minimum (if one exists) heavily depends on an appropriate initial guess. The initial guess not only impacts which local minimum the optimization routine finds, but moreover, could have bearing on whether the optimization routine even converges at all. It appears that this was the primary struggle faced early on in implementing DMOC.

Since it is possible to supply the initial and final positions for which there does not exist a trajectory, it is also important to distinguish the failure to converge due to a bad initial guess from the situation where there is no solution.

For our first implementation of DMOC, we chose an initial guess that linearly connects the initial and final position. Finite differences was used to compute approximate initial velocities and control forces. It is of note that this initial guess may not satisfy the constraints of the problem, however the optimization routine is sometimes capable of overcoming this issue and still find a minimum.

We provide a brief description of the difficulty we faced. Let us consider the following problem. We want to navigate the balloon from $(7, 0, 0)$ to $(10, 0, 2)$ with an initial velocity of $(0, 0, 0)$. For this example, we do not care what the final velocity is and the time is not fixed. The wind velocity is chosen to be $(1, 0, 0)$. In this case, DMOC successfully is able to minimize the objective function, and we get the trajectory and plot of control force over time in Figures (5.4) and (5.5) respectively.

Now, let us change the wind velocity from $(1, 0, 0)$ to $(x, 0, 0)$. If everything else is kept the same, the optimization routine is unable to find any trajectory that successfully satisfied the constraints let alone a minimal one. This seems to indicate that the problem is therefore related to the choice of initial guess. To verify this, we performed two experiments.

First, we used the working example and drastically changed the initial guess. In this circumstance, the optimization routine failed to find any trajectory that satisfied the constraints. This is to be expected but seems to support the idea that one cannot naively choose an initial guess.

Second, we changed the wind of the working example less drastically (using the linear initial guess) to see what happened. We chose winds of $(1 + \alpha \cdot x, 0, 0)$ where $\alpha \in [0.001, 1]$. The optimization succeeded for $\alpha \leq 0.6$. We then used the output results from $\alpha = 0.5$ as the initial guess for $\alpha = 1$. Whereas the optimization had failed, it now succeeded.

These two experiments seem to confirm that the issue had to do with the naive choice of the initial guess. As a result, we are currently investigating ways to generate a more reasonable initial guess.

3. Further Developments and Future Research Goals

3.1. **Newton's equations for Balloon Dynamics.** We are currently trying to implement a more sophisticated model that is derived using Newton's equations for the balloon. Although the current derivation uses some approximations, these can eventually be replaced by a more accurate model of balloon dynamics.

3.1.1. *Variables.* There are three forces acting on the balloon: lift, gravity and drag. In order to describe these forces, we introduce the following variables:

- ρ = density. We will say that ρ_{fluid} refers to the air and ρ_{gas} refers to the gas inside the balloon. Note that ρ_{fluid} depends on elevation, temperature, etc, however, to start, we will assume this is constant. In hot air balloons, ρ_{gas} is controlled by changing its temperature. For now, we will assume that we can influence ρ_{gas} directly.
- V = volume and A=cross-sectional area of the balloon. For now, we may assume the balloon is a sphere, in which case its cross-sectional area is $A = \pi R^2$ and its volume is $V = \frac{4}{3}\pi R^3$. Depending on the design, the volume of the balloon could depend on elevation, temperature, etc, however again, we will start by assuming this is constant.
- m_{eq} = mass of the equipment.
- C_d = drag coefficient .

- $\mathbf{v}$ is the velocity of the wind. Let us also consider the scalar quantity

$$\|\mathbf{v} - \dot{\mathbf{x}}\| = \sqrt{(v_x - \dot{x})^2 + (v_y - \dot{y})^2 + (v_z - \dot{z})^2},$$

where v_x, v_y, v_z are the components of the wind velocity and $\dot{x}, \dot{y}, \dot{z}$ are the components of the balloon velocity in the $e_x = (1, 0, 0)$, $e_y = (0, 1, 0)$, $e_z = (0, 0, 1)$ directions.

3.1.2. *Forces.* The buoyant force that pushes balloon up is $\rho_{fluid} g V$ and the gravitational force that pushes it down is $(m_{eq} + \rho_{gas} V)g$. From Newton's equations, we have (ignoring the contribution of wind)

$$(m_{eq} + \rho_{gas} V)a = (\rho_{fluid} V - m_{eq} - \rho_{gas} V)g.$$

Drag due to the wind pushes against the direction of motion of the balloon. Assuming quadratic dependence on relative velocity, we have (in a vector form)

$$\begin{aligned} \mathbf{F}_d &= -\frac{1}{2}\rho_{fluid}\|\mathbf{v} - \dot{\mathbf{x}}\|^2 C_d A \frac{\mathbf{v} - \dot{\mathbf{x}}}{\|\mathbf{v} - \dot{\mathbf{x}}\|} \\ &= -\frac{1}{2}\rho_{fluid}\|\mathbf{v} - \dot{\mathbf{x}}\| C_d A(\mathbf{v} - \dot{\mathbf{x}}). \end{aligned}$$

Combining all forces, we have

$$\begin{aligned} (m_{eq} + \rho_{gas} V)\ddot{\mathbf{x}} &= (\rho_{fluid} V - m_{eq} - \rho_{gas} V)g\mathbf{e}_z \\ &\quad -\frac{1}{2}\rho_{fluid}\|\mathbf{v} - \dot{\mathbf{x}}\| C_d A(\mathbf{v} - \dot{\mathbf{x}}). \end{aligned}$$

Let us introduce the following quantities

$$\eta = \eta(\mathbf{v}, \dot{\mathbf{x}}, \rho_{gas}) = \frac{\rho_{fluid}\|\mathbf{v} - \dot{\mathbf{x}}\| C_d A}{2(m_{eq} + \rho_{gas} V)}$$

$$\zeta = \zeta(\rho_{gas}) = \frac{(\rho_{fluid} V - m_{eq} - \rho_{gas} V)g}{m_{eq} + \rho_{gas} V}$$

If we set $\mathbf{q} = \mathbf{x}$ and $\mathbf{p} = \dot{\mathbf{x}}$, we get

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & \eta \end{bmatrix} \begin{bmatrix} \mathbf{q} \\ \mathbf{p} \end{bmatrix} + \begin{bmatrix} 0 \\ \zeta \mathbf{e_z} - \eta \mathbf{v} \end{bmatrix}$$

From this we get the following six equations:

$$\dot{q}_1 = p_1$$

$$\dot{q}_2 = p_2$$

$$\dot{q}_3 = p_3$$

$$\dot{p}_1 = \eta p_1 - \eta v_x$$

$$\dot{p}_2 = \eta p_2 - \eta v_y$$

$$\dot{p}_3 = \eta p_3 + \zeta - \eta v_z$$

3.1.3. *Discretization.* We use the implicit midpoint rule to discretize the equations of motion. To avoid confusion, we now refer to q_i and p_i as $q^{(i)}$ and $p^{(i)}$ respectively.

$$q_{n+1}^{(1)} = q_n^{(1)} + h \cdot (\frac{p_n^{(1)} + p_{n+1}^{(1)}}{2})$$

$$q_{n+1}^{(2)} = q_n^{(2)} + h \cdot (\frac{p_n^{(2)} + p_{n+1}^{(2)}}{2})$$

$$q_{n+1}^{(3)} = q_n^{(3)} + h \cdot (\frac{p_n^{(3)} + p_{n+1}^{(3)}}{2})$$

$$p_{n+1}^{(1)} = p_n^{(1)} + h \cdot (\eta \cdot \frac{p_n^{(1)} + p_{n+1}^{(1)}}{2} - \eta v_x)$$

$$p_{n+1}^{(2)} = p_n^{(2)} + h \cdot (\eta \cdot \frac{p_n^{(2)} + p_{n+1}^{(2)}}{2} - \eta v_y)$$

$$p_{n+1}^{(3)} = p_n^{(3)} + h \cdot (\eta \cdot \frac{p_n^{(3)} + p_{n+1}^{(3)}}{2} + \zeta - \eta v_z)$$

where

$$\eta = \eta(\mathbf{v}(\frac{\mathbf{q}_n + \mathbf{q}_{n+1}}{2}, nh + \frac{h}{2}), \frac{\mathbf{p}_n + \mathbf{p}_{n+1}}{2}, \rho_{n+\frac{1}{2}}^{gas})$$

$$\zeta = \zeta(\rho_{n+\frac{1}{2}}^{gas}).$$

3.1.4. *Control.* The control is due to changes in the density of the gas $\rho_{gas} = \rho_{gas}(t)$ inside the balloon. As an example of control, let us first consider an over-simplified problem. We minimize

$$\int_0^1 \dot{\rho}_{gas}^2 dt.$$

We discretize this integral on the time interval $[nh, nh+h]$ as

$$h \cdot (\dot{\rho}_n^{gas})^2$$

and, thus, we seek to minimize

$$\sum_{n=1}^{N} h \cdot (\dot{\rho}_n^{gas})^2.$$

To evaluate $\dot{\rho}_n^{gas}$, we define

$$\dot{\rho}_n^{gas} = \frac{\rho_{n+\frac{1}{2}}^{gas} - \rho_{n-\frac{1}{2}}^{gas}}{h}.$$

Therefore, we have

$$(\dot{\rho}_n^{gas})^2 = (\frac{\rho_{n+\frac{1}{2}}^{gas} - \rho_{n-\frac{1}{2}}^{gas}}{h})^2$$

and, thus, we minimize

$$\sum_{n=1}^{N} \frac{(\rho_{n+\frac{1}{2}}^{gas} - \rho_{n-\frac{1}{2}}^{gas})^2}{h},$$

which is equivalent to minimizing

$$\sum_{n=1}^{N} (\rho_{n+\frac{1}{2}}^{gas} - \rho_{n-\frac{1}{2}}^{gas})^2.$$

Note that we choose an initial ρ_{gas} and other parameters so that all forces acting on the balloon in the absence of wind are balanced, i.e.,

$$\rho_{fluid}V - m_{eq} - \rho_{gas}V = 0.$$

3.1.5. *A Simple Demonstration.* While the current implementation of this model has yet to be verified, we will demonstrate some preliminary results and further motivate the concerns with the initial guess. We also mention that we may require an alternative choice of cost function, perhaps one that also punishes higher derivatives (e.g. $\ddot{\rho}_{gas}$) or employs a different norm (e.g. l^1-norm), however this is an area that needs to be investigated further.

Let us consider the following problem. We want to navigate the balloon from $(0,0,0)$ to $(0,0,5)$ with no initial velocity or wind. We search for an optimal trajectory using two different initial guesses.

First, we try the same initial guess as described in Section 2.1 with the exception that the initial guess for the density of gas inside the balloon at each position is the same as it is at the initial position. In this case we get the trajectory and plot of gas density inside the balloon over time in Figures (5.6) and (5.7) respectively.

However, if we instead choose an initial guess which is entirely zero, the optimization converges slower and to a numerically worse local minimum. We get the trajectory and plot of gas density inside the balloon over time in Figure (5.8) and (5.9) respectively.

Intuitively, we would expect the optimal solution to be the balloon initially lowering the density of gas inside slightly and use the induced acceleration to reach the final destination. However neither of these local minima depict this scenario. Moreover, they depict completely different means of reaching the final destination, possibly indicating that as sophistication is added to the model, the more critical the initial guess becomes.

4. Interaction with Mentor

Professor Marsden conducts bi-weekly meetings where professors, post-docs, graduate students, and scientists from JPL give talks on topics related to the Titan mission. Myself and other SURF students then provide an update of our progress in our own projects. Also, I have met individually with Professor Marsden several times to further clarify details of my project and get feedback on some of my ideas. I have been mostly in email contact with Philip Du Toit regarding questions concerning DMOC and its implementation. Similarly, I have been in contact with Claire Newman regarding questions concerning the Weather Research and Forecasting model and Nick Heavens, who provided guidance in running the model and interpolating it onto a Cartesian grid. Claire Newman provided significant aid in the methodology described in Section (1.2.1).

5. Appendix

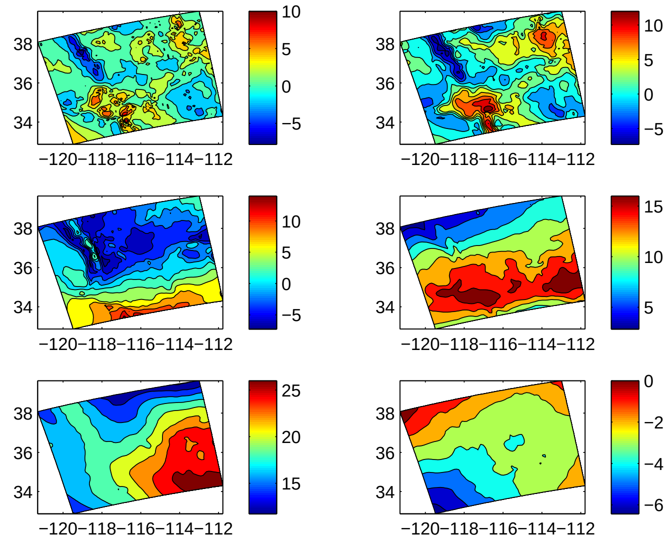

Figure 5.1. Plot of x-direction wind velocity at various different sigma levels

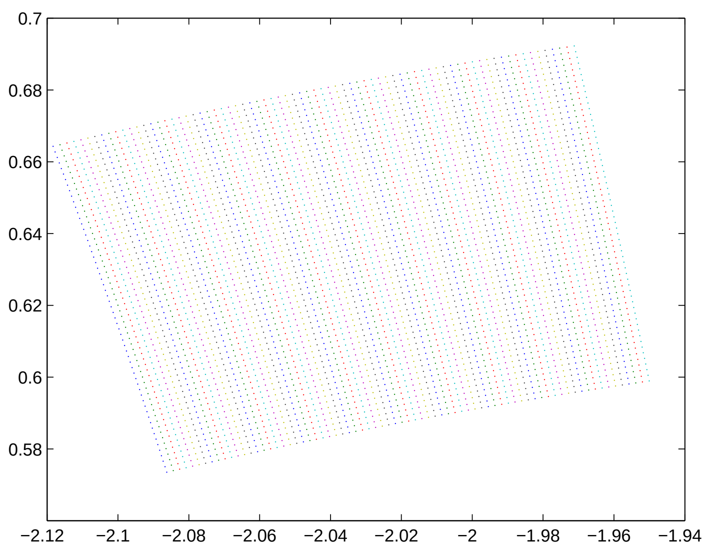

Figure 5.2. Curvilinear grid used by WRF model

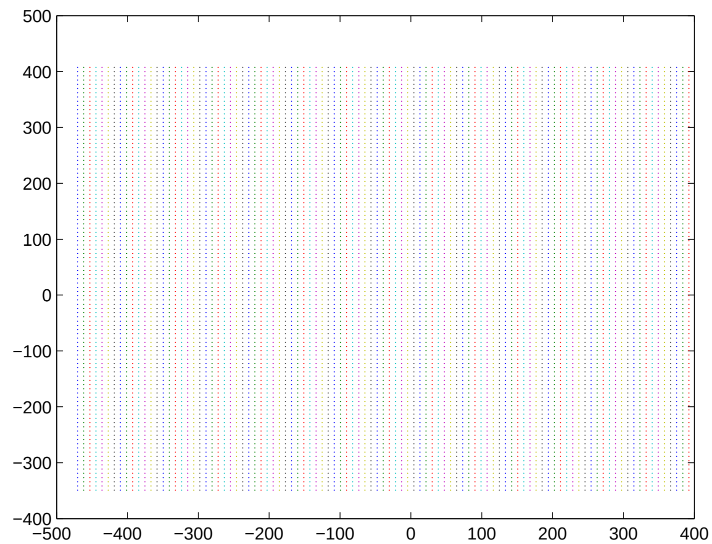

FIGURE 5.3. Uniform, rectangular Cartesian grid onto which we interpolate the WRF output

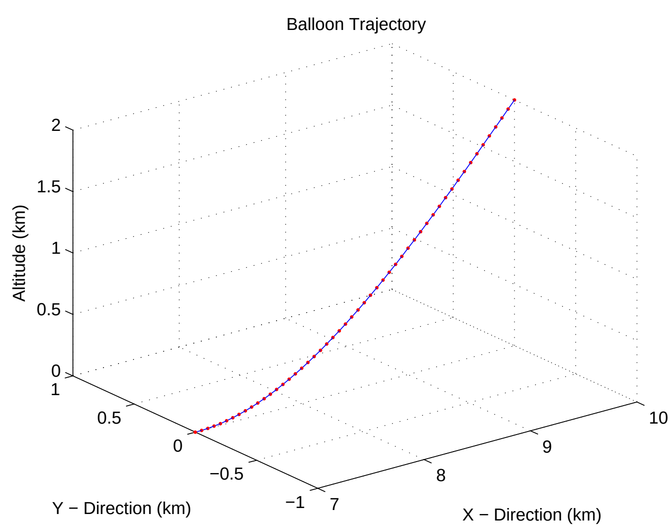

FIGURE 5.4. Optimal trajectory of balloon navigating from (7,0,0) to (10,0,2) in wind (1,0,0)

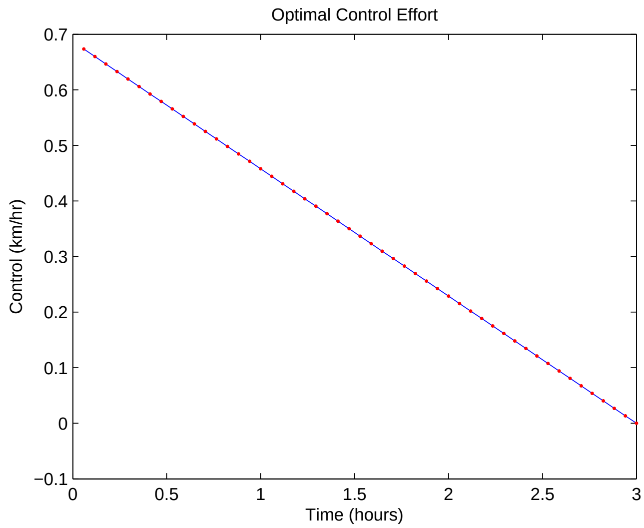

FIGURE 5.5. Control vs. Time of optimal trajectory for balloon navigating from (7,0,0) to (10,0,2) in wind (1,0,0)

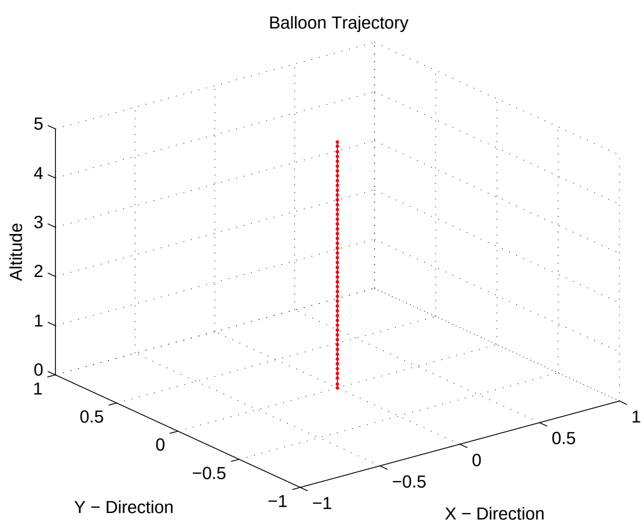

FIGURE 5.6. First initial guess: Optimal trajectory of balloon navigating from (0,0,0) to (0,0,5)

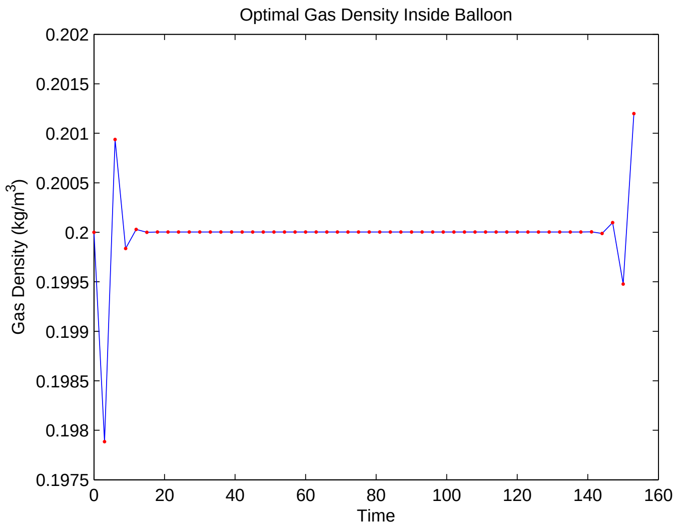

FIGURE 5.7. First initial guess: Gas density inside balloon vs. Time of optimal trajectory for balloon navigating from (0,0,0) to (0,0,5)

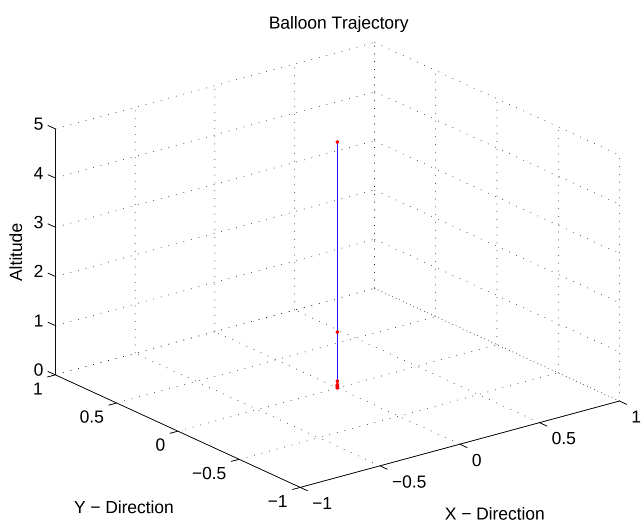

FIGURE 5.8. Second initial guess: Optimal trajectory of balloon navigating from (0,0,0) to (0,0,5)

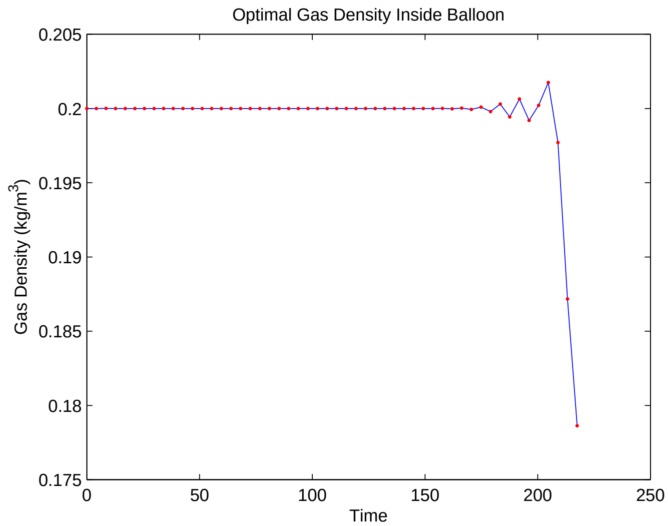

FIGURE 5.9. Second initial guess: Gas density inside balloon vs. Time of optimal trajectory for balloon navigating from (0,0,0) to (0,0,5)