University of Colorado, Boulder
For the Department of Applied Mathematics

APPM 2350 Project #1

Theo Lincke, Sam Carruthers

7 October 2019

BACK TO THE TNB FUTURE, BUZZ

Theo Lincke, Sam Carruthers

ABSTRACT In this paper, we analyze various properties of a travelling rocket's displacement path (in miles) as a function of time (in minutes). Using a vectored representation of the rocket's path, we approach the displacement and higher order rates visually as well as mathematically. In part two, we model the rocket's path and briefly touch on the rocket's speed to explore the rocket's displacement and the rocket's arc length. We find the rocket travels a total of roughly 299 miles and changes its position by a magnitude of roughly 31.3 miles. In part three, we analyze higher order rates represented by the rocket (i.e. velocity and acceleration as functions of time) and graphically display some of these properties. We find the rocket's max speed to be roughly 71 miles/minute at the time 9.9 minutes (593 seconds). We also find that the maximum normal acceleration is 1065 ft/s^2 and at this same point, the tangential component is 635 ft/s^2. In part four, we explore more complex quantities such as the rocket's curvature and torsion. We find the maximum curvature to be 2.5 miles^{-1} at roughly 511 seconds. Finally, in part five, we run into the scenario where a missile and rocket are moving concurrently. We find that the minimum distance between these two projectiles is 1.07 miles at a time of 493 seconds.

1. INTRODUCTION

A group of "space pirates" plan to steal a 1.21 gigawatt flux capacitor designed by specialized Docbrown native scientists. General Emmett, the commander of the space pirates, plans to escape in a 10 minute interval, and requires help from his crew to model the path of the space flight along the curve given by the vector valued formula in equ (1.1):

$$\overline{r}(t) = \begin{bmatrix} 10\sin(t) + 5\sin(5t) + 2.5\sin(2.3t) \\ 10\cos(t) + 5\cos(5t) + 2.5\cos(2.3t) \\ 0.0005\,t^4(3 + \cos(4\pi t)) \end{bmatrix} \tag{1.1}$$

Note that although we use the captain's original formula for r_3 because of the simpler computation, we simplify the original commander's model for a more easily approachable formula to solve by hand. We also consider two differentials on the curve of the space flight, and model them in equ (1.2) and equ (1.3):

$$\overline{r'}(t) = \begin{bmatrix} 10\cos(t) + 25\cos(5t) + 5.75\cos(2.3t) \\ -10\sin(t) - 25\sin(5t) - 5.75\sin(2.3t) \\ 0.002\,t^3(\cos(4\pi t) - \pi t\sin(4\pi t) + 3) \end{bmatrix} \tag{1.2}$$

$$\overline{r''}(t) = \begin{bmatrix} -10\sin(t) - 125\sin(5t) - 13.225\sin(2.3t) \\ -10\cos(t) - 125\cos(5t) - 13.225\cos(2.3t) \\ 0.002t^2(9 - 8\pi t\sin(4\pi t) - (4\pi^2 t^2 - 3)\cos(4\pi t)) \end{bmatrix} \tag{1.3}$$

Where the first derivative is representative of the air craft's velocity vector as a function of time, and the second order derivative is representative of the air craft's acceleration as a function of time (note that the acceleration is a single vector, not necessarily broken up into tangential / normal components).

2. MODELING THE PATH OF THE ROCKET

To visually model the space ship's flight trajectory, we utilize a graphical model of the path of the spaceship in Figure (1.1). This graphical model was generated using a MATLAB symbolic ("syms") representation for equation (1.1) on the closed interval [0, 10][1] minutes.

Visually, the starting point (the green sphere) and the ending point (the red sphere) are relatively close. Indeed, if we calculate the starting point at time = 0 minutes to be equ (2.1):

$$\overline{r(0)} = \begin{bmatrix} 0.0000 \\ 18.0000 \\ 0.0000 \end{bmatrix} \tag{2.1}$$

and the ending position of the spacecraft to be equ (2.2):

$$\overline{r(10)} = \begin{bmatrix} -9.0000 \\ -5.0000 \\ 20.0000 \end{bmatrix} \tag{2.2}$$

we can consider the distance between these two points (in miles) to be the norm of the vectors (2.1) and (2.2), shown in equ (2.3).

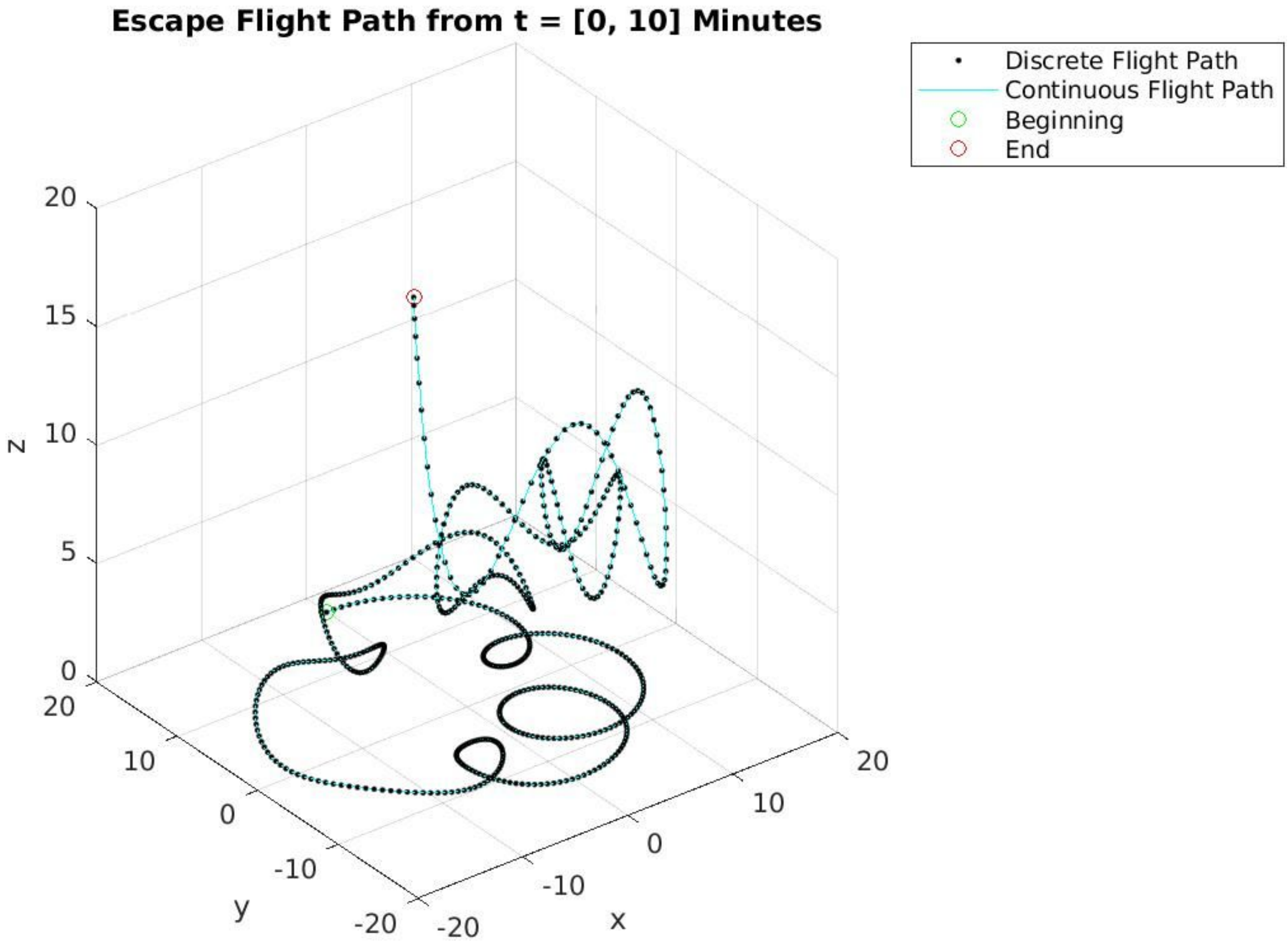

Figure 1.1 *- A Path of the space ship's flight trajectory from time = 0 minutes to time = 10 minutes. The Green point indicates the start and the red point indicates the end. Discrete time plot is in intervals of seconds.*

$$||r(0) - r(10)|| \tag{2.3}$$

$$= 31.3098 \text{ miles}$$

This quantity was calculated using symbolic MATLAB functions, so an accurate floating point approximation was used to find trigonometric identities[2]. A closer look at the graph shows that this distance is miniscule compared to the actual arc length the ship traveled. Because the ship burns fuel at a rate of 150 lbs per mile, we calculate the actual distance to be the definite integral of the ship's speed as a function of time from the closed time interval of [0, 10] minutes by substituting the value of (t) for 10 and the value of r' for (1.2) in equ (2.4).

$$l(t) = \int_0^t ||r'(u)||\, \mathrm{du} \tag{2.4}$$

$$l(10) \approx 299.1520 \text{ miles}$$

The quantity in equ (2.4) was calculated using a floating point approximation as well using a discrete anonymous MATLAB function.[3] Thus, we are able to confidently predict that the amount of trash to be burnt on the entire trip is **4.4873x10^4 lbs** which is approximately **22.4364 tons.**

3. MODELING HIGHER ORDER CHARACTERISTICS FROM THE FLIGHT

The general has provided the escape path; however, he is also interested in the speed and acceleration of the ship as a function of time. We purposefully included equ (1.2) and equ (1.3) because they both represent physical characteristics of the flight other than the displacement vs time, namely velocity and acceleration.

Equ (1.2), the first derivative of the ship, is representative of the rocket's *velocity* (change in displacement) as a function of time. Note that the velocity of the rocket is in fact a position and magnitude, but it does not represent the speed of the rocket. To find the speed of the rocket, we find the norm (length of the vector at any given point in time) of equ (1.2), shown in equ (3.1).

$$s(t) = ||v\overline{(t)}|| = ||\overline{r(t)}||$$
$$s(t) \in \Re \tag{3.1}$$
$$\overline{v(t)} \in \Re^3$$

It is important to note from equ (3.1) that the speed of the rocket is a scalar quantity, where the velocity is a vector quantity. Figure (3.1) shows this quantity visually from the closed interval of [0, 10] minutes. The maximum speed the rocket reaches is approximately **1.1863 miles/second** (71.1775 miles/minute) at the time

593 seconds. The maximum speed and plot of speed were generated using a discrete MATLAB anonymous function[4].

Equ (1.3), the second derivative of the ship's positio, is representative of the rocket's *acceleration* (change in velocity) as a function of time. Note that the acceleration of the rocket is not necessarily always affecting the rocket's speed. The acceleration is in fact a vector quantity, meaning it too, has a direction. If acceleration has a direction that is not invariant of the direction of the velocity, the acceleration at that point is both speeding up the rocket and changing the rocket's *path.* Thus, we can't simply take the norm of the acceleration like we did in equ (3.1). To accurately model the rocket's acceleration, we split it up into it's *normal* and *tangential* components in equ (3.2).

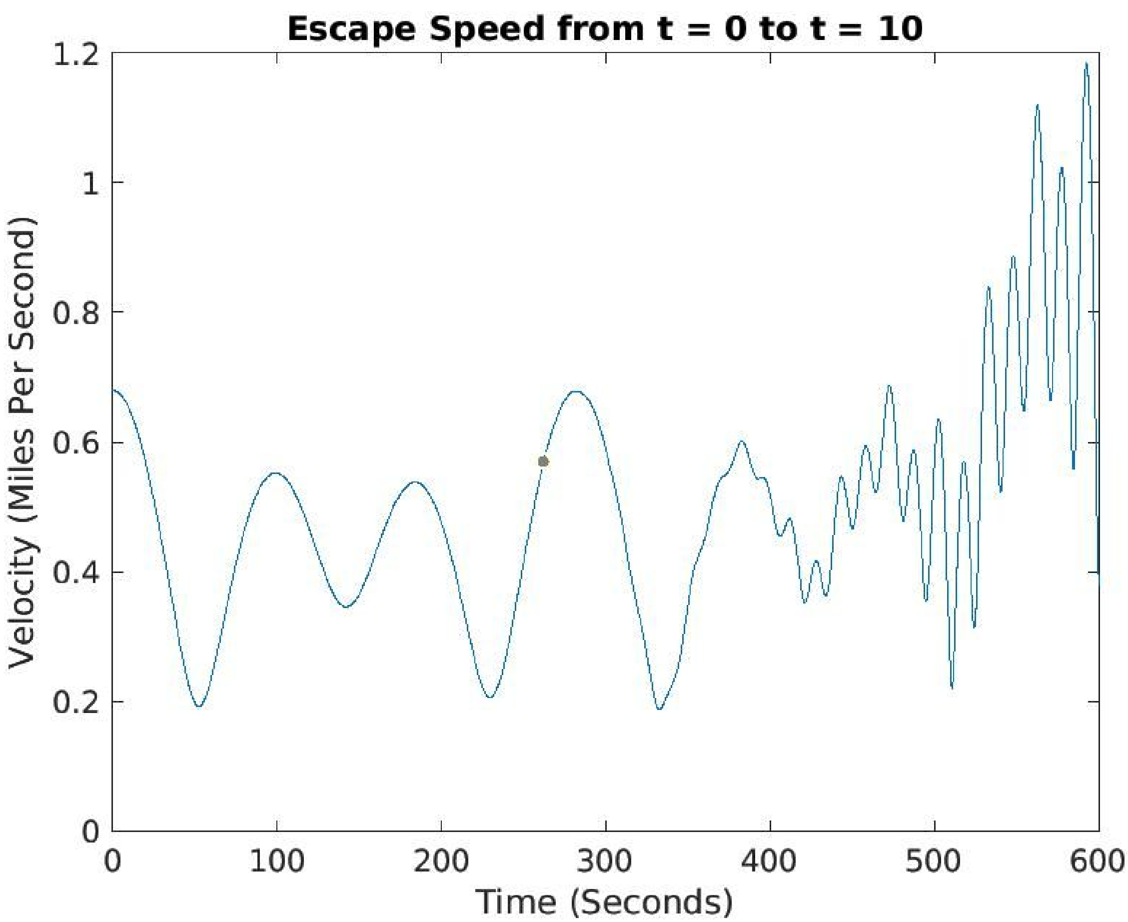

Figure 3.1 - *A plot of the ship's scalar speed as a function of time from the closed interval [0,600] seconds.*

$$\overline{a(t)} = \overline{r''(t)} = a_N \hat{N} + a_T \hat{T} \tag{3.2}$$

$$\overline{a(t)}, \overline{r(t)}, \hat{N}, \hat{T} \in \mathfrak{R}^3$$

$$a_N, a_T \in \mathfrak{R}$$

$$||\hat{T}|| = ||\hat{N}|| = 1$$

The ship's normal and tangential components can be calculated using a simple projection, however, we calculate them more easily using equ (3.3). These are derived using orthogonal and parallel projections, and the value of k is referenced later in equ (4.3).

$$a_N = k||v||^2$$

$$a_T = \frac{d}{\mathrm{dt}}\left|\left|v\right|\right| \tag{3.3}$$

The captain notes that one of his pirates is space sick when the normal component of acceleration is equal to 3g, where g is the acceleration of gravity on Docbrown (340 ft/s^2). Thus, the normal acceleration needs to be constrained by 1020 ft/s^2. Figure (3.2) shows the normal acceleration on the period [0, 10] minutes. The normal acceleration is considerably below 1020 from the interval [0, 10] minutes. However, there are two intervals that the captain must watch out for: **[585, 587]** and a miniscule period at two tenths of a second at the end of the flight **[599.86, 600]** seconds). In these two periods, the normal component of acceleration (shown in red) is above 1020 ft/s^2. The maximum normal acceleration happens at **585** seconds, where the rocket approaches **1064.9821** ft/s^2. At this point in time, the ship is **increasing** in speed at a rate of **634.8383** ft/s^2. The maximum values and plot for this normal component were found using a discrete MATLAB function[5].

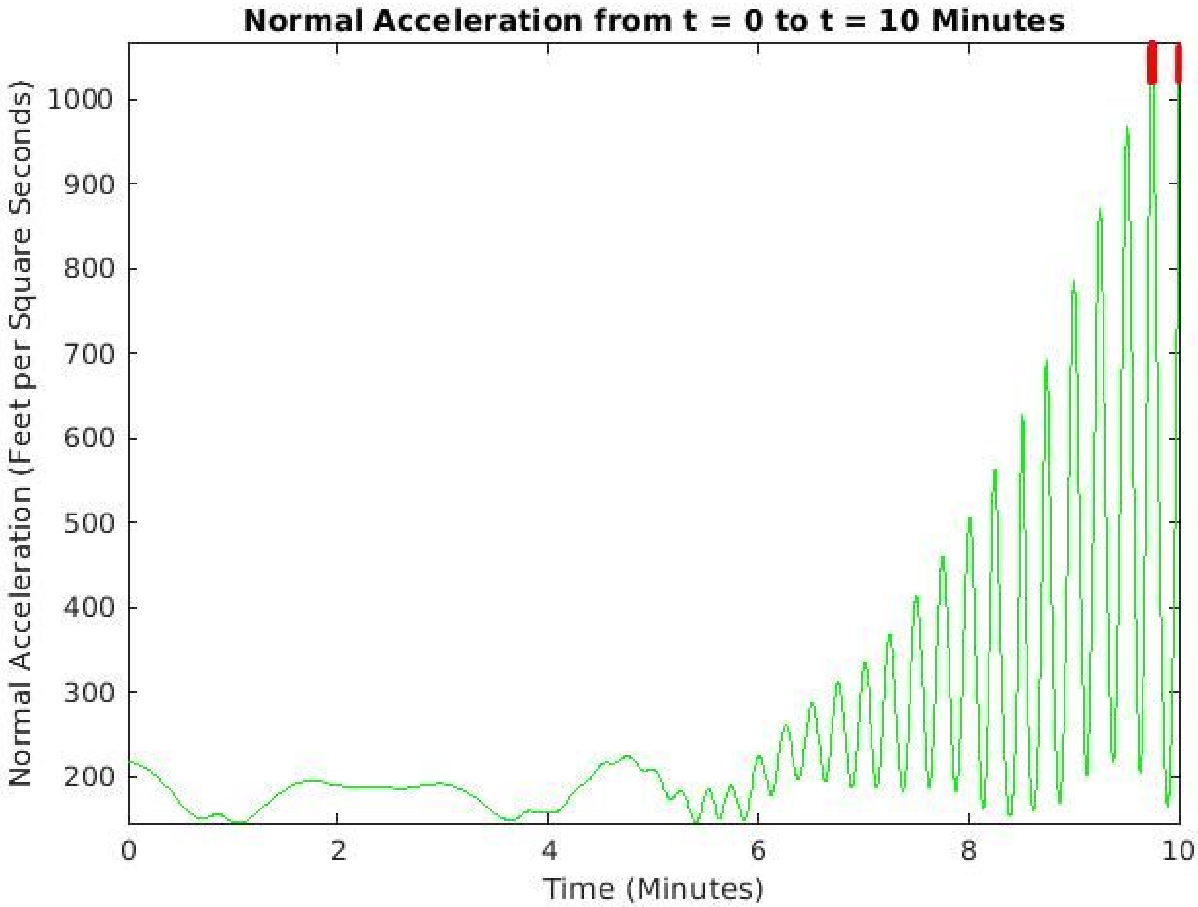

Figure 3.2 - *A plot of the ship's normal acceleration as a function of time. The Red Segments contain all values above 1020 ft/s^s. Note that the acceleration is in ft/s^2 and not miles/minute^2.*

4. TNB REFERENCE FRAME

The captain's interest in the TNB reference frame allows us to revisit the normal, tangential and, further, the binormal quantities for the curvature of the rocket. In equ (3.2) we referenced, but did not find the quantity of the unit tangent, normal and binormal vectors. For the sake of brevity, we omit the TNB vectors in this report and visualize both graphical models in the code reference[6][7]. These three vectors are mathematically described in equ (4.1). To visually represent these three unit vectors, we plot the three of them at time = 9.05 minutes in Figure (4.1).

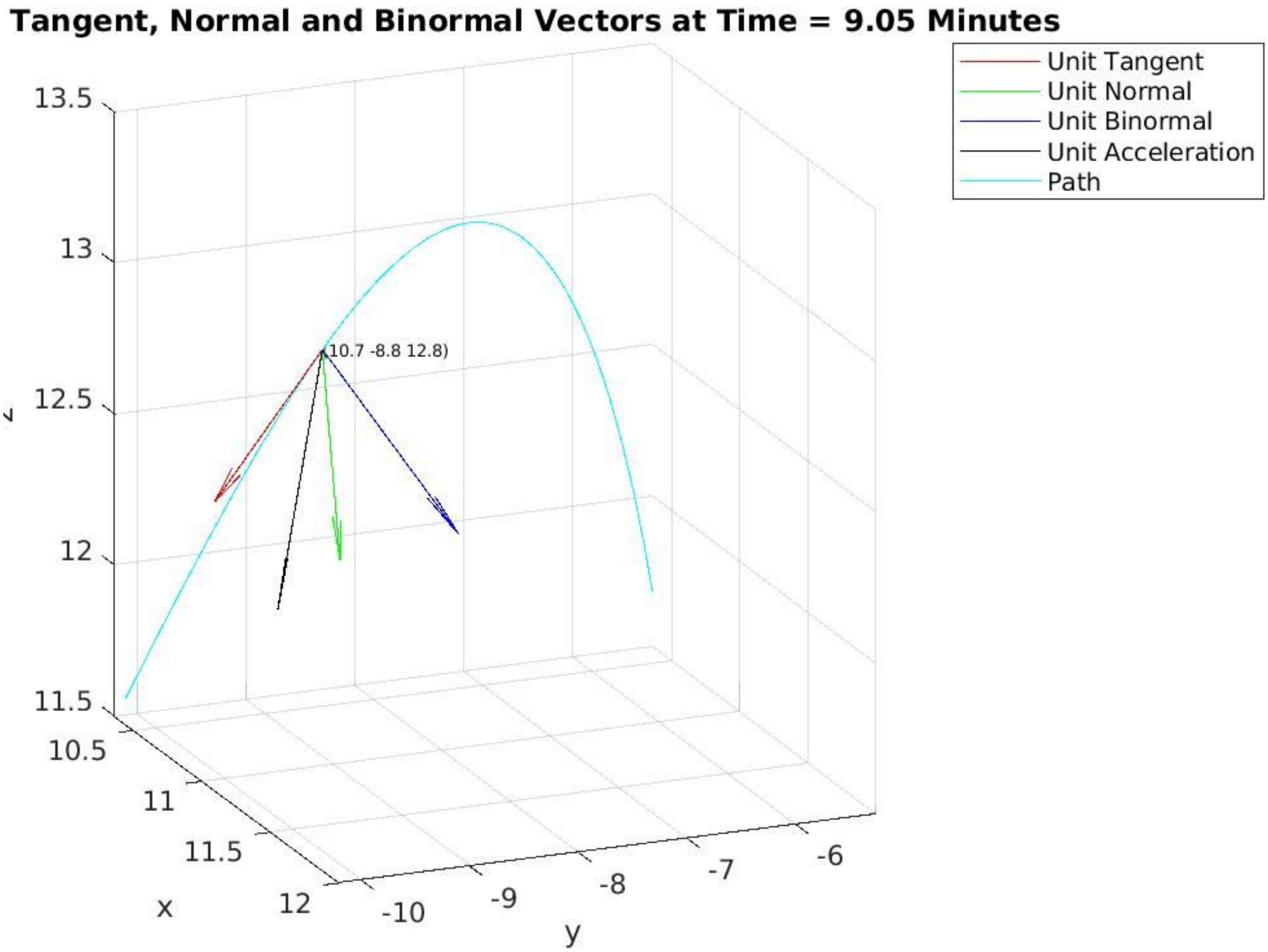

Figure 4.1 - *A plot of the Unit Normal, Tangent, Binormal and Acceleration vectors at time = 9.05 minutes as well as the path from the closed interval time = [8.9333, 9.08333] minutes*

In Figure (4.1), it is quite visually clear that the three unit vectors (T, N, B) are each orthogonal to one another. The final vector (unit acceleration) lies within the span of T and N, or the *Osculating plane*. Figure (4.2) shows the three of these vectors in a longer period of time (520 to 550 seconds).

$$T = \frac{\mathrm{d}}{\mathrm{d}t} r \left|\left| \frac{\mathrm{d}}{\mathrm{d}t} r \right|\right|^{-1} \tag{4.1}$$

$$N = \frac{\mathrm{d}}{\mathrm{d}t} T \left|\left| \frac{\mathrm{d}}{\mathrm{d}t} T \right|\right|^{-1}$$

$$B = T \times N$$

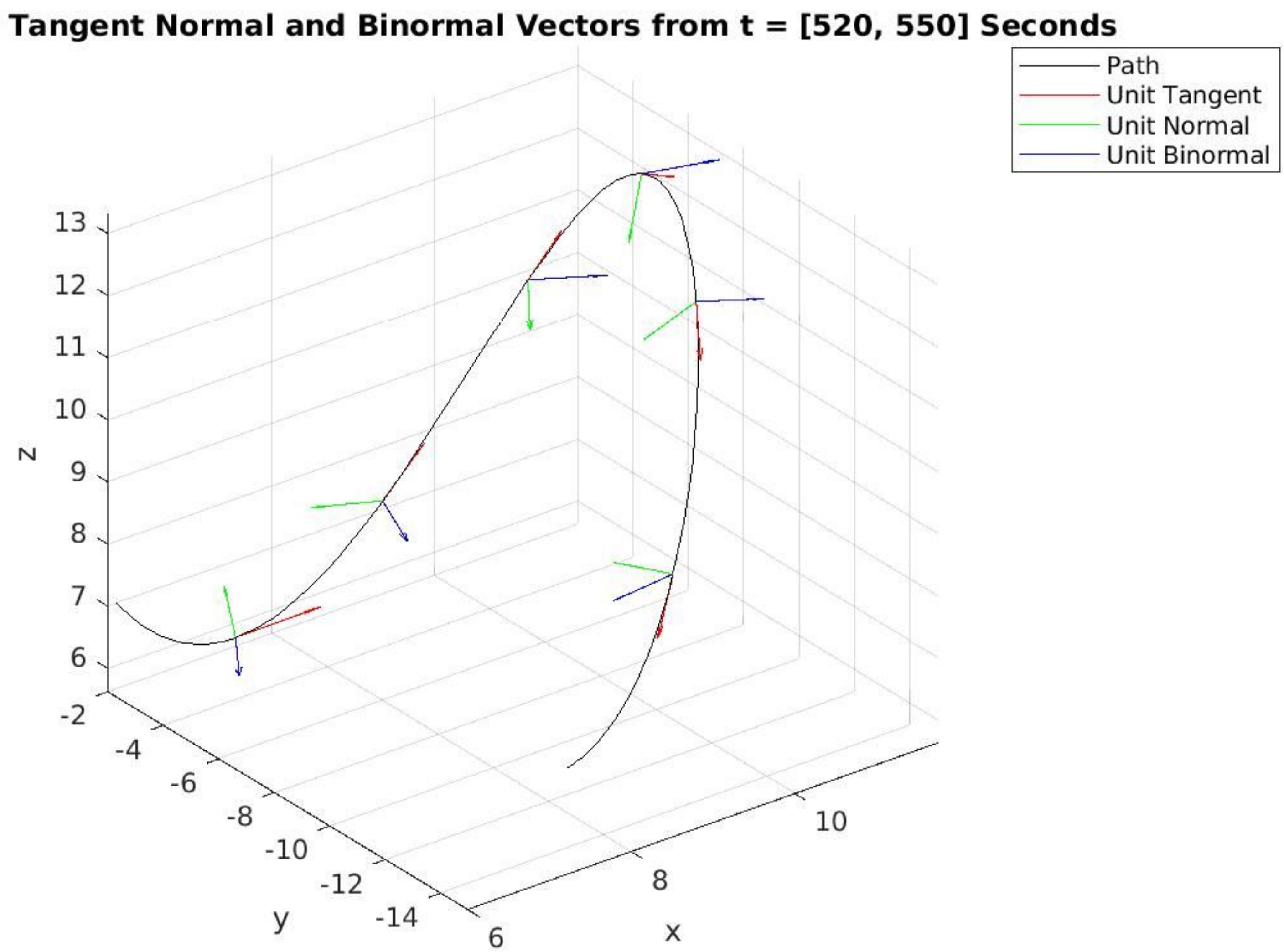

Figure 4.2 - *A plot of the TNB vectors on the closed interval time = [520, 550] seconds, one is plotted every 5 seconds.*

General Emmett points out a concern for the integrity of the ship withstanding curvature as well, in which we can revisit ideas of velocity and acceleration vectors to calculate it. The curvature is given by equ (4.3), which we can plot as a function of time with software[8].

$$\kappa = \frac{|\bar{v} \times \bar{a}|}{|\bar{v}|^3} \tag{4.3}$$

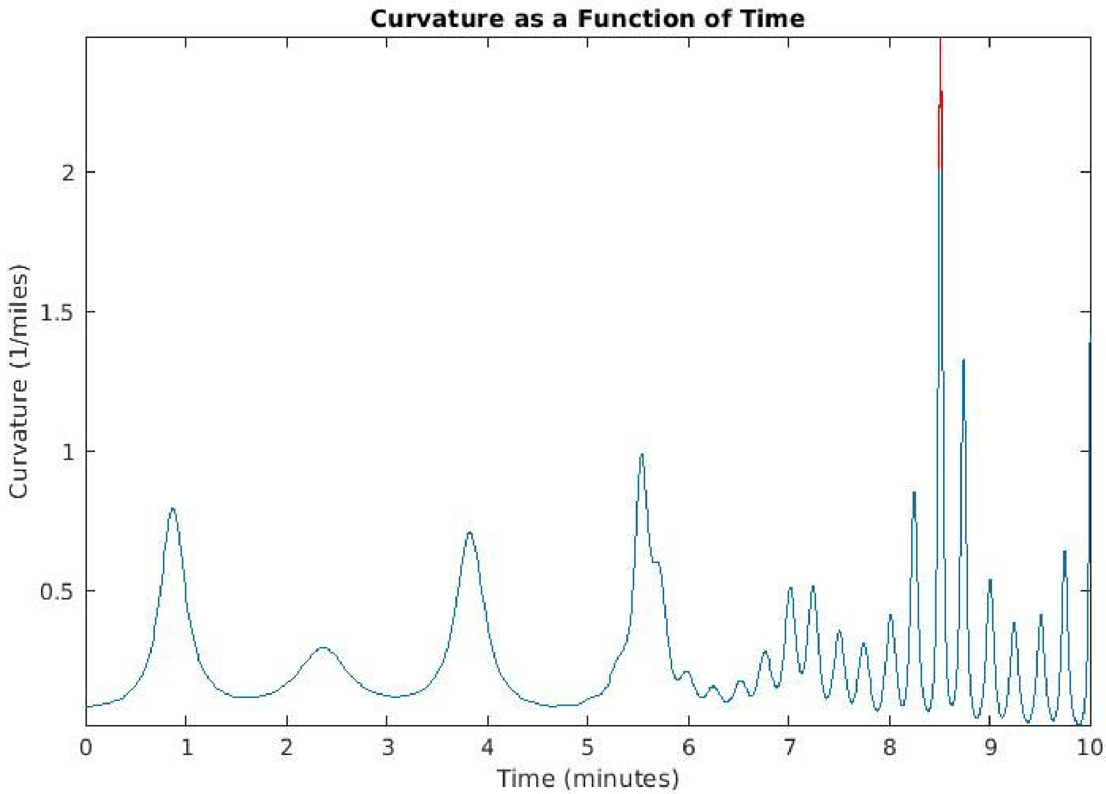

Figure 4.3 - *A plot of curvature as a function of time on the closed interval [0,10] minutes. Units of curvature are miles^{-1}. Red Segment is the period where all values are above 2 miles^{-1}*

In figure (4.3), we visually see that the curvature does exceed 2 miles^{-1} on the closed time interval **[510, 512] seconds**. The maximum curvature is **2.4796** miles^{-1}.

With the captain's interest in the subject, it is best to include the relationship between distance and torsion, which is defined as the rate of change of the osculating plane. The osculating plane is created by vectors B and N, and dB/ds is the change in the binormal vector with respect to arclength. We can calculate torsion by equ (4.2), and produce a plot with a MATLAB function[9]. As per the captain's orders, we include a figure for the displacement and distance.

$$\tau = - < N \mid B' > \tag{4.2}$$

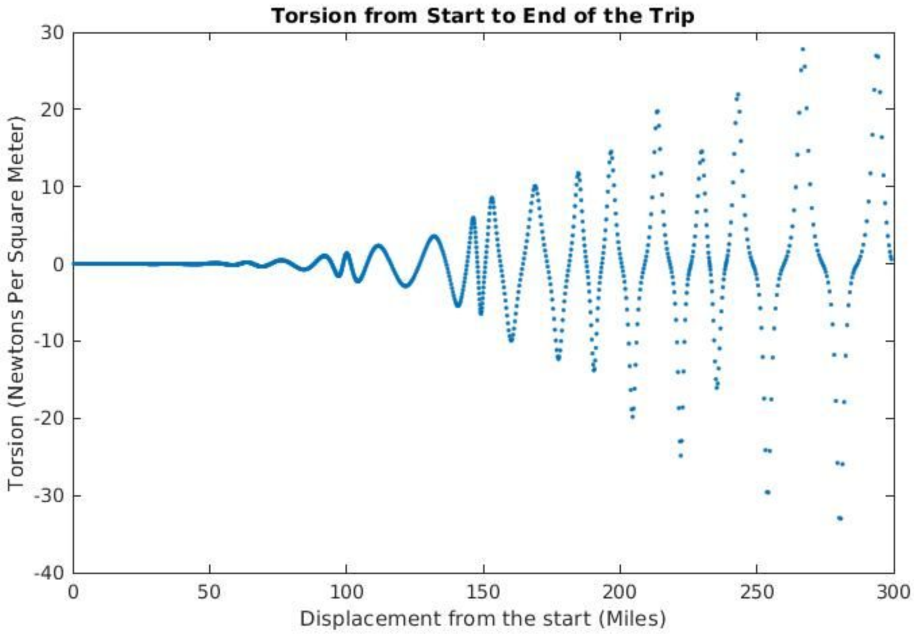

Figure 4.4 - *A plot of torsion as a function of time on the closed interval [0.10]. Note the units for torsion are N/m^2*

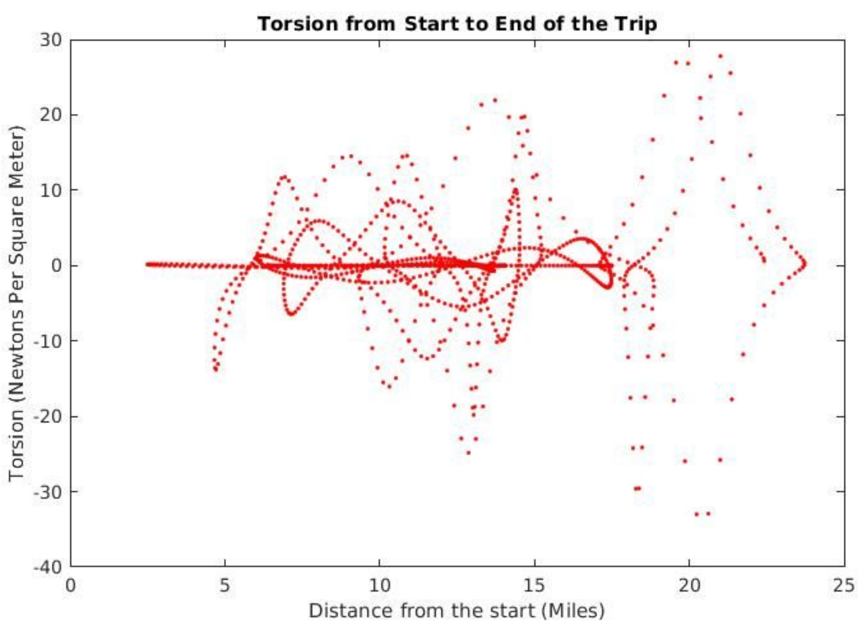

Figure 4.5 - *A plot of torsion vs distance (the norm of position from the starting point)*

5. A NEW FLIGHT TRAJECTORY OF AN INBOUND MISSILE

After modeling the space craft's flight path, there is a serious concern for an inbound missile following the path shown in equ (4.1)

$$\overline{r_m}(t) = \begin{bmatrix} 5 + \sin(t) \\ 2\cos(t) \\ 3\sqrt{t} \end{bmatrix} \tag{4.1}$$

The general's ship is unsafe when the missile approaches within 0.5 miles of the ship. Figure 4.4 shows the distance between the ship and the missile.

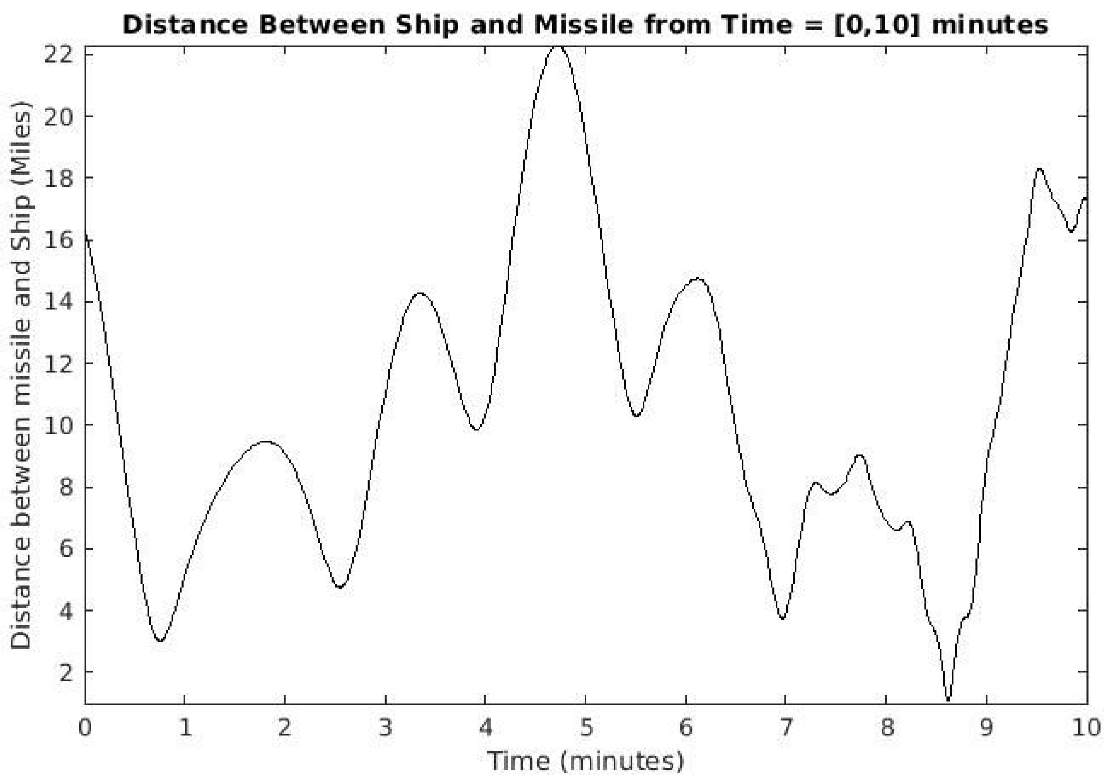

Figure 4.6 - *A Plot of the distance between the missile and the ship as a function of time in minutes.*

We calculate the minimum distance between the ship and the missile to be **1.0746 miles** at a time of **493 seconds.** Thus we can safely conclude that the missile will never reach within half a mile of the ship. To better visualize the path of the missile and rocket, we graph both trajectories from the closed interval of [0, 10] in Figure (4.5). Both plots can be generated with software[10].

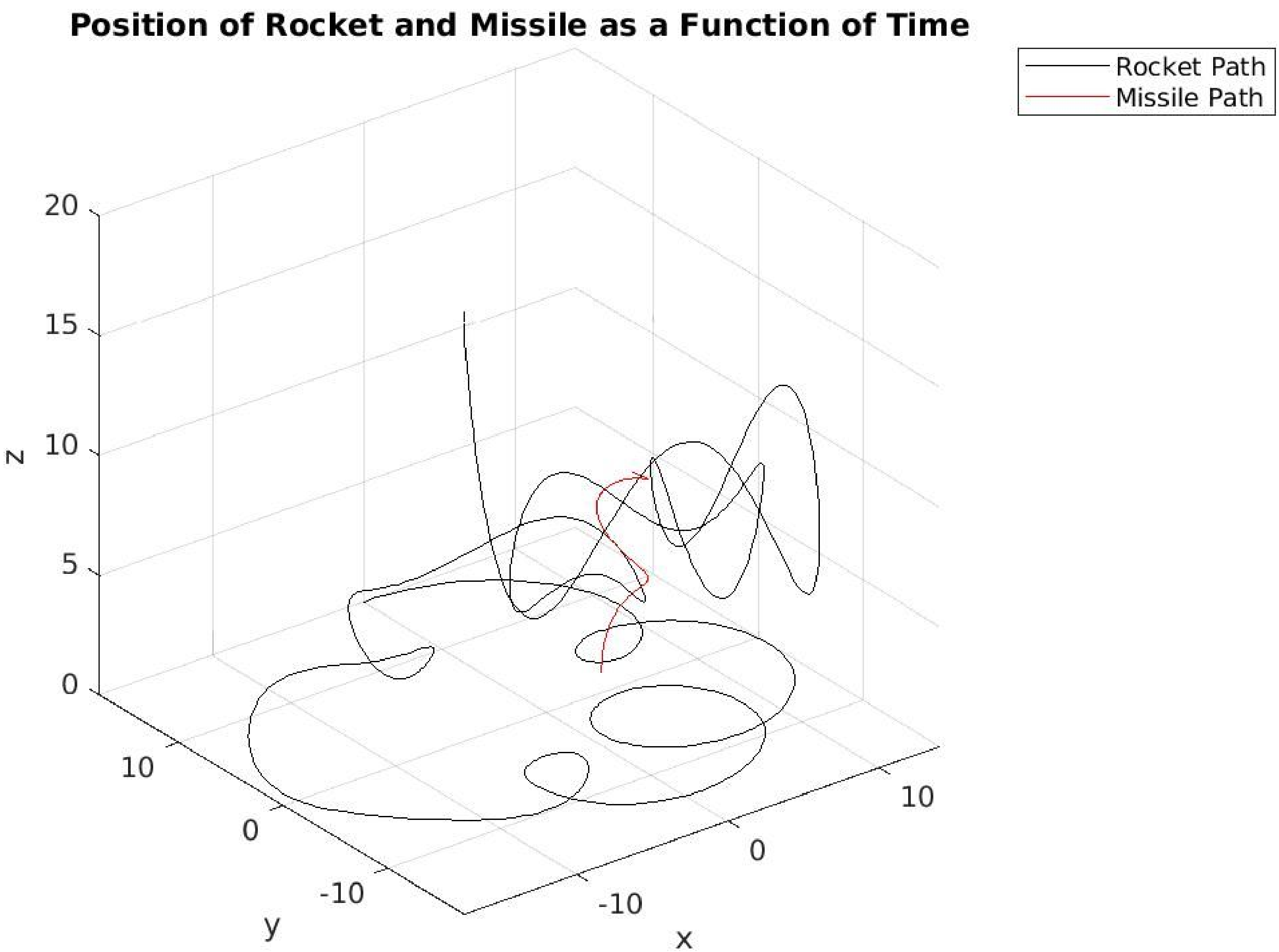

Figure 4.5 - *A plot of the Rocket (in black) and the Missile (in red) on the closed interval [0,10].*

CONCLUSION

In this paper, we found models for the flight path of an escape pod, as well as higher order models such as speed, acceleration, and the TNB frame to represent the pod throughout the mission. We avoided potential issues in the mission by analyzing flight information like the normal component of acceleration and the curvature at any given point along the path, as well as distance from a lethal missile. Our calculations and analysis led us to valid models of the escape pod's flight path as well as information of other factors in the situation that resulted in an overall successful mission as well as a strong understanding of the mathematical software behind some common problems.

References:

Code:

Code is referenced only to cut space - see the entire folder for modular code samples

[1] (2019) question1.m source code (Version 1.0) [Source code].
[2] (2019) question2.m source code (Version 1.0) [Source code].
[3] (2019) question3.m source code (Version 1.0) [Source code].
[4] (2019) question4.m source code (Version 1.0) [Source code].
[5] (2019) question5.m source code (Version 1.0) [Source code].
[6] (2019) question6.m source code (Version 1.0) [Source code].
[7] (2019) question6_part2.m source code (Version 1.0) [Source code].
[8] (2019) question7.m source code (Version 1.0) [Source code].
[9] (2019) question8.m source code (Version 1.0) [Source code].
[10] (2019) Question8part2.m source code (Version 1.0) [Source code].
[11] (2019) partb.m source code (Version 1.0) [Source code].
[12] (2019) partbpart2.m source code (Version 1.0) [Source code]

[13] (2019) main.mlx source code (Version 1.0) [Executable Source code]
[14] (2019) main.m source code (Version 1.0) [Source code]

Notes on the source code

Source code is organized by section. All questions inherit main.m, which initializes larger structures so that data isn't being repeatedly computed. The file main.mlx is the primary resource to execute, as it contains the logical flow of components in this project, generating every graph and every number with a single execution.

To run the code, simply run main.mlx. Every figure, equation and value is computed in one execution.