MTE202 - Ordinary Differential Equations

University of Waterloo

Department of Mechanical and Mechatronics Engineering

Modeling the Flight of a Frisbee

Authors:
Steven Ha (ID:)
Brendan Johnston (ID:)
Niel Mistry (ID:)

Date: December 16, 2018

Contents

List of Figures

1 Nomenclature

α = angle of attack (rad)[2]
α_0 = angle of attack that causes the least lift (rad)[3]
A = area of Frisbee (m^2)
ac = aerodynamic center
C_D = coefficient of drag [2]
$C_D 0$ = form drag constant [3]
$C_D \alpha$ = induced drag constant [3]
C_L = coefficient of lift [3]
$C_L 0$ = lift equation y-intercept [3]
$C_L \alpha$ = lift equation slope
$\vec{F}_D$ = drag force (N)
$\vec{F}_L$ = lift force (N)
g = acceleration due to gravity m/s^2
I_z = mass moment of inertia about z (kg·m^2)
η = angle between aerodynamic center and x' (rad)
M_0 = zero lift pitching moment (N·m)
ρ = density of air (kg·m^{-3})
$\dot{p}$ = pitching rate (rad/s)
P = Pitching Moment (N·m)
$\dot{r}$ = rolling rate (rad/s)
R = Rolling Moment (N·m)
$\vec{v}$ = velocity of Frisbee (m/s)
x, y, z = unit vectors of reference frame of Frisbee parallel to global reference frame
x', y', z' = unit vectors of reference frame of Frisbee which rotates with Frisbee
X, Y, Z = unit vectors of the global inertial reference frame

2 Introduction

The Frisbee is a popular recreational device with over 300 million units being sold since its introduction over 40 years ago [4]. In sports such as Ultimate and Disc Golf, being able to predict the trajectory that is thrown is imperative to succeeding at the sport. Though this can be learned through practice , it is also possible to create a mathematical model to predict the trajectory of a Frisbee under certain conditions.

To understand this report, an understanding of the physics behind the flight of a Frisbee is needed. A Frisbee will experience lift, drag and gravity as it glides through the air. The lift force is produced by the pressure difference from air flowing over the top of the Frisbee faster than the air flowing the bottom of the Frisbee. Using the Bernoulli Principle a equation for the lift force can be derived [2]:

$$F_L = \frac{1}{2}\rho v^2 A C_L \tag{1}$$

It should be noted that the area in this equation does not change depending on the pitch of the Frisbee. In addition C_L is a function of the angle of attack which is given below (2). The angle of attack is the angle between the plane of the Frisbee and the velocity vector. The constants in equation (2) are determined based on the physical properties of the Frisbee ($C_L 0 = 0.15$ and $C_L \alpha = 1.4$) [2].

$$C_L = C_{L0} + C_{L\alpha}\alpha \tag{2}$$

The drag relationship used to calculate drag force is dependent on the Reynolds number. For an average Frisbee throw and sea level conditions, the Reynolds number is 2.59×10^5. This means that the Prandtl relationship for drag needs to be used [2]:

$$F_D = -\frac{C_D \rho A v^2}{2} \tag{3}$$

The coefficient of drag is also a function of the angle of attack which is defined below [2].

$$C_D = C_{D0} + C_{D\alpha}(\alpha - \alpha_0)^2 \tag{4}$$

All the constants in (4) depend on the physical property of the Frisbee with $C_{D0} = 0.08$, $C_{D\alpha} = 2.72$ and $\alpha_0 = -4°$ [2][3]. The figure below summarizes the forces on a Frisbee.

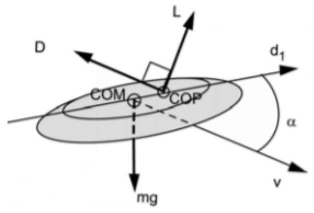


Figure 1: Lift and Drag acting on the CoP of a Frisbee [1]

In addition to the forces acting on the Frisbee, the flight of a Frisbee also depends on its rotation. The angular velocity in the Frisbee causes the gyroscopic effect which allows it to be very stable in the air. This means that it is resistant to the moments caused by the the lift and drag forces acting about the center of mass. These moments are due to the fact that the lift and drag forces act on a point called the center of pressure which is offset from the center of mass. These moments can be simply calculated by [2]:

$$\vec{\tau} = \vec{r} \times \vec{F} \tag{5}$$

3 Assumptions

A number of engineering assumptions were made to simplify and constrain the complexity of the trajectory of a Frisbee. Below, some trivial assumptions were made:

- g = 9.81 ms^{-2}
- r = 0.26 m [5]
- m = 0.175 kg [5]
- Frisbee is a rigid body

The next few paragraphs describe more involved assumptions that were made regarding the Frisbee. First, the throw is assumed to be instantaneous because the time to accelerate the Frisbee during the throw is small enough to neglect. In addition, it is assumed that the throw will impart enough angular velocity to the Frisbee such that the gyroscopic stability to glide through the air, and there will be minimal cyclical precession of the Frisbee rotational axes. For calculations of mass moment of inertia, the Frisbee is assumed to be a uniform circular cylinder of negligible height, as it is computationally difficult to calculate the mass moment of inertia for the shape of a Frisbee. Also, the negligible height is valid since the mass distribution of the Frisbee is weighted towards the top.

With regards to the flight of the Frisbee, it was assumed that the lift and drag forces act at the aerodynamic center. Although, in reality, the aerodynamic forces will act on the center of pressure and a moment would have to be applied to move the forces to the aerodynamic center, this moment was assumed to be negligible for the Frisbee. This was also done by Crowther in Simulation of Sports Stabilized Discs [6]. The function of the aerodynamic center is shown in figure 2.

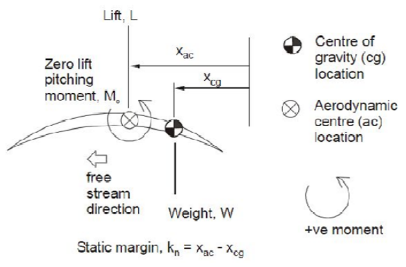

Lift, L
Zero lift pitching moment, M_o
x_{ac}
x_{cg}
Centre of gravity (cg) location
Aerodynamic centre (ac) location
free stream direction
Weight, W
+ve moment
Static margin, $k_n = x_{ac} - x_{cg}$

Figure 2: Aerodynamic Center

Another assumption that was made was that initial velocity of the Frisbee is in the direction of initial pitch and roll angles. This makes sense as when throwing a Frisbee you usually release the Frisbee at the pitch and roll angles and do not apply a velocity in any other direction. Finally it was assumed that the yaw rate (or the spin rate) of the Frisbee is equal to 37 rad s^{-1}[7], and that the spin rate would remain constant throughout flight. This assumption was made based of multiple papers stating that spin rate (for a normal Frisbee flight) has a minimal effect on aerodynamic loads [8]. Lastly, the Robins-Magnus force was neglected in this model because at low spin rates (in comparison to the translational velocity) there isn't a significant effect from the force [8].

4 Mathematical Model

The mathematical model was created using Newton's Second Law for the forces and some equations from "Simulation of a spin-stabilized sports disc" [6] for the pitching and rolling rate equations. In these calculations, three different reference frames are used. Reference frame XYZ is an inertial reference frame that is stuck to the ground. Reference frame xyz has axes that are always parallel to the XYZ, but the origin is coincident with the centroid of the Frisbee. Finally, reference frame $x'y'z'$ has an origin coincident with frame xyz, but rotated such that the z' axis is perpendicular to the plane of the Frisbee.

To account for the multiple reference frames, relative motion equations and transformation matrices were used. When moving from the global XYZ reference frame to the xyz reference frame, a simple translation of vectors was preformed. For example, as a part of including the wind values we had to first calculate the velocity of the Frisbee with respect to the wind for the appropriate lift and drag forces. However, for data output, the Frisbee's airspeed was converted to ground speed to ensure that the model predicted landing locations on the ground accurately.

A transformation matrix, seen on page 23 was used when calculating pitch and roll moments. Since calculating pitch and roll moments was much easier along the body coordinate system $x'y'z'$, the moments were calculated in that frame and transformed to the global coordinate system.

This mathematical model is one of intermediate complexity. It started with the ability to take into account simple projectile motion, as well as the lift and drag forces. It also started with the assumption that center of pressure (CoP) was coincident with the center of mass (CoM) and thus calculations were quite simple. Equations (6) through (10) show the model that was created up to this point. Note that all calculations for this stage of the model were done in the xyz coordinate system.

$$\vec{F}_L = \left|\vec{F}_L\right| \frac{\vec{v}\times\vec{y}'}{\left|\vec{v}\times\vec{y}'\right|} \tag{6}$$

$$\vec{F}_D = -\left|\vec{F}_D\right| \frac{\vec{v}}{\left|\vec{v}\right|} \tag{7}$$

$$\ddot{x} = \frac{F_{L_x} + F_{D_x}}{m} \tag{8}$$

$$\ddot{y} = \frac{F_{L_y} + F_{D_y}}{m} \tag{9}$$

$$\ddot{z} = \frac{F_{L_z} + F_{D_z}}{m} - g \tag{10}$$

Equation 6 determines the lift vector by multiplying the magnitude by the unit vector of the velocity vector crossed with the body y' vector. Similarly, since drag acts in the opposite direction of velocity, the drag vector is found by multiplying the magnitude of the drag force by the negative velocity unit vector in equation (7). It should be noted that even though $\dot{x}$, $\dot{y}$ and $\dot{z}$ do not appear in equations (8), (9) and (10), the drag and lift forces do still depend on them. This can be seen in (1) and (3) where v is $\dot{x}$, $\dot{y}$ and $\dot{z}$ depending on the subscript in the forces.

Then, it was decided to add complexity to the model by having the lift force and drag force act at the aerodynamic center and by removing the assumption that CoP = CoM. With the change in assumptions, the aerodynamic forces acted away from the center of mass which meant that moments were also being created. This resulted in the pitch and roll moments. The forces that act perpendicular to the plane of the Frisbee were initially calculated. Despite the change in assumption the differential equations derived above are still valid. Below in equation (11) shows the force of lift that is perpendicular to the plane of the Frisbee and (12) shows the force of drag that is perpendicular to the plane of the Frisbee.

$$\vec{F}_{L_{z'}} = proj_{\vec{z'}}\vec{F}_L \tag{11}$$

$$\vec{F}_{D_{z'}} = proj_{\vec{z'}}\vec{F}_D \tag{12}$$

The direction of the aerodynamic center was found by projecting the velocity vector onto the plane of the Frisbee. Then, the precise location was found by multiplying the distance of the aerodynamic center by a unit vector.

$$\vec{ac}_{dir} = \vec{v} - proj_{\vec{z'}}\vec{v} \tag{13}$$

$$\vec{ac}_{pos} = 0.12d\left|\vec{ac}_{dir}\right| \tag{14}$$

The angle that the position vector of the aerodynamic center makes to the x' axis is found using the scalar dot product formula. Equations (16) and (17) determine the perpendicular distance of the aerodynamic center to the x' and y' body axes.

$$\eta = \frac{\arccos(\vec{x'} \cdot \vec{ac_{pos}})}{\left|\vec{x'}\right|\left|\vec{ac}_{pos}\right|} \tag{15}$$

$$d_{x'} = \sin\eta \tag{16}$$

$$d_{y'} = \cos\eta \tag{17}$$

$$P = -\left|\mathrm{d}_{y'}(\vec{F}_{L_{z'}} + \vec{F}_{D_{z'}})\right| \tag{18}$$

$$R = \left|\mathrm{d}_{x'}(\vec{F}_{L'_z} + \vec{F}_{D_{z'}})\right| \tag{19}$$

The distances that are used in equations (18) and (19) are opposite of what is expected for a non-rotating body because of the gyroscopic nature of the Frisbee. The angular momentum vector of the Frisbee wants to chase the torque produced by the applied forces, so the axis precesses in a fashion that does not seem intuitive. The pitching and rolling rate equations are taken from Crowther [6].

$$\dot{p} = \frac{2P}{I_z * \dot{y}} \tag{20}$$

$$\dot{r} = \frac{2R}{I_z * \dot{y}} \tag{21}$$

The system of differential equations used to model the Frisbee's path through the air are expressed in equations (8), (9), (10), (20) and (21). The ode45 function used to numerically integrate the ordinary differential equation only supports first order differential equations, so the each second order differential equation was reduced to two first order equations. This can be seen in section 9.1.

5 Results

Due to the amount of parameters that the program can handle, this section will demonstrate the results of the program with changes in multiple parameters at a time. If the reader would like to experiment with the conditions, the repository is available at: `https://github.com/johnstonbrendan/202_Project`.

The first result shown is a baseline result. This is what the program produces when given the most basic initial parameters. These initial parameters are: startHeight=2m, Wind Speed=0, ThrowV=14ms^{-1}, r=0, p=0, spinRate=37rad s^{-1} With these initial parameters the 3D plot and the individual x,y,x with respect to t plots can be seen in Figures 3, 4, 5 and 6 in section 9.2. These initial conditions are those of a flat throw and thrown by an average Frisbee thrower [2].

Case 1 is one with a given initial pitch as well as a higher spin Rate and faster throw. The initial conditions that were changed from the base line and their values are: ThrowV=20ms^{-1}, p=10°, spinRate=40rad s^{-1}. This model would be more accurate for a more advanced Frisbee thrower and the plots can be seen from Figures 7, 8, 9 and 10 in section 9.2.

Case 2 is one with a given initial pitch as well as an initial roll. The initial conditions that were changed from the base line and their values are: p=10°and r=10°. The plots generated from these changed conditions can be seen in Figure 11, 12, 13 and 14 in section 9.2. Given these pitch and roll conditions, the Frisbee should be much higher than the baseline did and slowly steer to the left. This can be seen in Figure 11.

Case 3 is one with a given a wind speed and an initial roll. The initial conditions that were changed from the base line and their values are: Wind Speed = [0 1 -1]m/s and r=10°. The plots generated from these changed conditions can be seen in Figures 15, 16, 17 and 18 in section 9.2. This case demonstrates a person throwing a Frisbee with a roll angle of 10°into a cross wind of 1m/s in the y direction and -1m/s in the z direction. Intuitively, one would expect the wind to steer the Frisbee left and down which can be seen in Figure 15.

Case 4 is one with a given a greater start height, wind speed, initial pitch and roll. The initial conditions that were changed from the base line are: Start Height = 20m, Wind Speed = [4 0 0] m/s, p=10°and r=10°. The plots generated from these changed conditions can be seen in Figure 19, 20, 21 and 22 in section 9.2. This case demonstrates if a person were to throw a Frisbee from 20m high with wind blowing in the positive x direction.

6 Conclusion

The overall mathematical model was the solution to the system of ODEs previously proposed in the section 4. Though it was not analytically solved, numercal solutions are seen in section 5. Although the supposed model was a bit on the complex side, it was still solvable through MATLAB's ode45 function. The main limitations of this model come in it's assumptions. The assumption that the difference in location between the centre of pressure and the aerodynamic center is negligible means that an initial moment that is being created at the aerodynamic center is ignored. This initial moment potentially has trickle over effects into the solution equation and may have made the model less accurate. Although the model also does not account of any advanced aerodynamic equations it has quite a high predictive capability. Given a wind speed and direction and taking into account the angle at which the Frisbee would leave someones hand and the spin rate, the model is able to solve for the flight trajectory. These initial conditions that are set are the main factors that a normal Frisbee player would consider when making a Frisbee throw. The model itself is then able to account for the changes in angle of attack, velocity (direction and magnitude), pitch, roll, lift force and drag force in order to find the flight trajectory. These are most of the major factors that effect a Frisbee's flight and thus the model seems to be internally consistent. Additionally, there were a few factors that have assumed in the equation that may potentially allow for even more excitability in the future. Two of these factors are our mass and diameter of the Fribee. Although not tested, the model technically may be able to work for Frisbee's with a mass other than 0.175kg. This is important as not all Frisbee are designed with the same mass and this expandable feature may prove important if a real life test of this model was preformed. The same can also be said when discussing the diameter of the Frisbee, which was also taken to account in our calculations although the effect of changing it has not been tested.

7 Contribution

Research was distributed evenly among the group as everyone was required to perform their own research and to document findings in a Google Doc. When it came to creating the mathematical mode everyone worked together and discussed ideas as to how to relate the force equations with each other. Since Mistry was the most familiar with MATLAB he was responsible for writing the code while Ha and Johnston would aid Mistry in writing and reviewing the code. As Mistry and Johnston made final additions to the model, Ha became tasked with writing the report to ensure that it would be finished in time. After the model was finished, the rest of the report was split by having members work on sections that they were most familiar with.

8 Bibliography

[1] Sarah A. Hummel and Mont Hubbard. Identification of Frisbee aerodynamic coefficients using flight data. page 3. pages 3, 5

[2] V R Morrison. The Physics of Frisbees. 6:1–12, 2005. pages 4, 5, 9

[3] Kathleen Baumback, Masahiko Saito, and Scott Campbell. The Aerodynamics of Frisbee Flight. *Undergraduate Journal of Mathematical Modeling: One + Two*, 3(1):5, 2010. pages 4, 5

[4] Laurel Sheppard. Frisbee. `http://www.madehow.com/Volume-5/Frisbee.html`. pages 5

[5] Disc Testing Criteria. `https://www.usaultimate.org/assets/1/AssetManager/DSWG002-DiscTestingCriteria.doc`. pages 6

[6] W. J. Crowther and J. R. Potts. Simulation of a spinstabilised sports disc. *Sports Engineering*, 10(1):3–21, 2007. pages 6, 7, 8

[7] Ralph D. Lorenz. Flight and attitude dynamics measurements of an instrumented Frisbee. *Measurement Science and Technology*, 16(3):738–748, 2005. pages 6

[8] Preston Pozderac. Gone With the Wind: An Investigation into the Flight Dynamics of Discs. 2016. pages 6

9 Appendix

9.1 MATLAB Code

Main Code

```
function solver
    %%%%%%%% initial conditions %%%%%%%
     startHeight = 2; %m
     wind_speed = [0 0 0]; %x,y,z windspeed (m/s)
     throwV = 14; % magnitude of the throw velocity (m/s)
     initPitch =10*pi/180; % pitch of initial throw (rad)
     initRoll = 0*pi/180; %roll of initial throw (rad)
     spinRate =37; % spin throw rate of frisbee (rad/s)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    % time step setup
     tstart = 0; %seconds
     tend = 200; %this can be large as the calculation will stop
         when z = 0
     tstep = 0.0001;
     n = (tend-tstart)/tstep;
     tspan = linspace(tstart, tend, n);


     xInit = [0; throwV*cos(initPitch); 0; 0; startHeight;
        throwV*sin(initPitch); initRoll; initPitch]; %x, vx, y,
        vy, z, vz, r, p
     Opt = odeset('Events', @detectGround);
     [t, out] = ode45(@(t, out) discODEs(t, out, wind_speed,
        spinRate), tspan, xInit, Opt);

     x = out(:,1) +(t*wind_speed(1));
     vx = out(:,2) + wind_speed(1);
     y = out(:,3) + (t*wind_speed(2));
     vy = out(:,4) + wind_speed(2);
     z = out(:,5) +(t*wind_speed(3));
     vz = out(:,6) + wind_speed(3);
     r = out(:, 7).*180/pi; % roll state (degrees)
     p = out(:, 8).*180/pi; % pitch state (degrees)

   showPlots(t, x, y, z, vx, vy, vz, r, p, startHeight);
```

```matlab

end

function ddt = discODEs(t, out, wind_speed, spinRate)
    m = 0.175; %mass of frisbee (kg)
    g = 9.81; %gravity (m/s^2)
    CL0 = 0.15; %coefficient of lift (0)
    CLa = 1.4; %coefficient of lift (alpha)
    CD0 = 0.08; %coefficient of drag (0)
    CDa = 2.72; %coefficient of drag (alpha);
    alpha_0 = -0.0698; % 0 defined based on physcial aspects of
        frisbee (radians)
    rho = 1.225; %density of fliud (kg/m^3)
    rd = 0.137; %radius (m)
    Iz = 1/8*m*(rd*2)^2; % kg(m^2)



    % ddt order is [vx, ax, vy, ay, vz, az, rdot, pdot]
    x = out(1);
    vx = out(2);
    y = out(3);
    vy = out(4);
    z = out(5);
    vz = out(6);
    r = out(7);
    p = out(8);

    % frisbee position and velocity vector wrt XYZ
    pos = [x y z];
    v = [vx vy vz];
    v = v - wind_speed; % find free stream velocity, v frisbee
       rel to air

    % global to body transformation matrix
    globalToBody = transpose([1 0 0;0 cos(r) -sin(r);0 sin(r)
       cos(r)]*[cos(p) 0 sin(p); 0 1 0; -sin(p) 0 cos(p)]);

    % transform global axes to body axes
    bodyx = (globalToBody*transpose([1 0 0])).';
    bodyy = (globalToBody*transpose([0 1 0])).';
    bodyz = (globalToBody*transpose([0 0 1])).';

    %Calculate lift and drag unit vector
    lift_vect = cross(v, bodyy);
    lift_uvect = lift_vect/norm(lift_vect);
    drag_vect = -1*v;
    drag_uvect = drag_vect/norm(drag_vect);

    %Calculate lift and drag force
    lift_force = calc_lift_force(CL0, CLa, alpha(vx, vz, p),
       rho, rd, velocity(vx, vy, vz));
    lift = lift_force*lift_uvect;
    drag_force = calc_drag_force(CD0, CDa, alpha(vx, vz, p),
       alpha_0, rho, rd, velocity(vx, vy, vz));
    drag = drag_force*drag_uvect;

    %Get lift and drag force components
    liftX = lift(1);
    liftY = lift(2);
    liftZ = lift(3);
    liftBodyz = (dot(lift,bodyz)/(norm(bodyz))^2)*bodyz;

    dragX = drag(1);
    dragY = drag(2);
    dragZ = drag(3);
    dragBodyz = (dot(drag,bodyz)/norm(bodyz)^2)*bodyz;
    % X = [1 0 0], Y = [0 1 0], Z = [0 0 1]

    % PITCHING MOMENT SECTION %

    % assume clockwise rotation
    totalBodyzForce = norm(liftBodyz + dragBodyz);

    % project v vector onto plane
    acDirVector = (v - (dot(v, bodyz)/(norm(bodyz))^2)*bodyz);
    acPosVector = 0.12*rd*2*acDirVector/norm(acDirVector);

    % use scalar dot product formula to find angle to bodyx
       axis
    nu = acos(dot(bodyx, acPosVector))/(norm(bodyx)*norm(
       acPosVector));

    % find components of the position vector to ac on bodyx,
        bodyy axis
    acPosVectorx = acPosVector*cos(nu);
    acPosVectory = acPosVector*sin(nu);

    % find negative pitching moment by using sin(angle) as
        distance
    pMom = -norm((totalBodyzForce * acPosVectory)/globalToBody)
        ;

    % find rolling moment by using cos(angle) as distance
    rMom = norm((totalBodyzForce * acPosVectorx)/globalToBody);

    ddt = zeros(size(out));
    ddt(1) = vx;
    ddt(2) = (liftX + dragX)/m;
    ddt(3) = vy;
    ddt(4) = (liftY + dragY)/m;
    ddt(5) = vz;
    ddt(6) = (liftZ + dragZ - m*g)/m;
    ddt(7) = 2*rMom/(Iz * spinRate);
    ddt(8) = 2*pMom/(Iz * spinRate);
end

function showPlots(t, x, y, z, vx, vy, vz, r, p, startHeight)
    close all;
    %x,y,z plot
    figure('Name','x y z plot');
    plot3(x, y,z);
    x_limit = xlim();
    y_limit = ylim();
    z_limit = zlim();
    max_lim = max([z_limit(2) y_limit(2) x_limit(2)]);
    axis([0 max_lim 0 max_lim 0 max_lim]);
    xlabel('x (m)');
    ylabel('y (m)');
    zlabel('z (m)');
    %z vs time plot
    figure('Name','z t plot');
    plot(t,z);
    xlabel('t (s)');
    ylabel('z (m)');
    %x vs time plot
    figure('Name','x t plot');
    plot(t,x);
    xlabel('t (s)');
    ylabel('x (m)');
   %y vs time plot
    figure('Name','y t plot');
    plot(t,y);
    xlabel('t (s)');
    ylabel('y (m)');
   %roll vs time plot
    figure('Name', 'r t plot');
    plot(t, r);
    xlabel('t (s)');
    ylabel('roll (degrees)');
   %pitch vs time plot
    figure('Name', 'p t plot');
    plot(t, p);
    xlabel('t (s)');
    ylabel('pitch (degrees)');
end

function [value, isterminal, direction] = detectGround(t, out)
    value = (out(5) < 0);
    isterminal = 1;
    direction = 0;
end
```

Functions

Lift Force

```
function y = calc_lift_force(CL0,CLa,a,rho,r,v)
y = 1/2*(CL0+CLa*a)*rho*pi*power(r,2)*power(v,2);
end
```

Drag Force

```
function y = calc_drag_force(CD0, CDa, a, alpha_i, rho, r, v)
y = 1/2*(CD0+CDa*(a-alpha_i)^2)*rho*pi*(r^2)*(v^2);
end
```

Velocity

```
function y = velocity(x_velocity,y_velocity,z_velocity)
y = sqrt(x_velocity^2+y_velocity^2+z_velocity^2);
end
```

Angle of Attack

```
function y = alpha(x_velocity,z_velocity, pitch)
y = atan(abs(z_velocity)/abs(x_velocity)) + pitch;
end
```

9.2 Results Plots

9.2.1 Baseline Plots

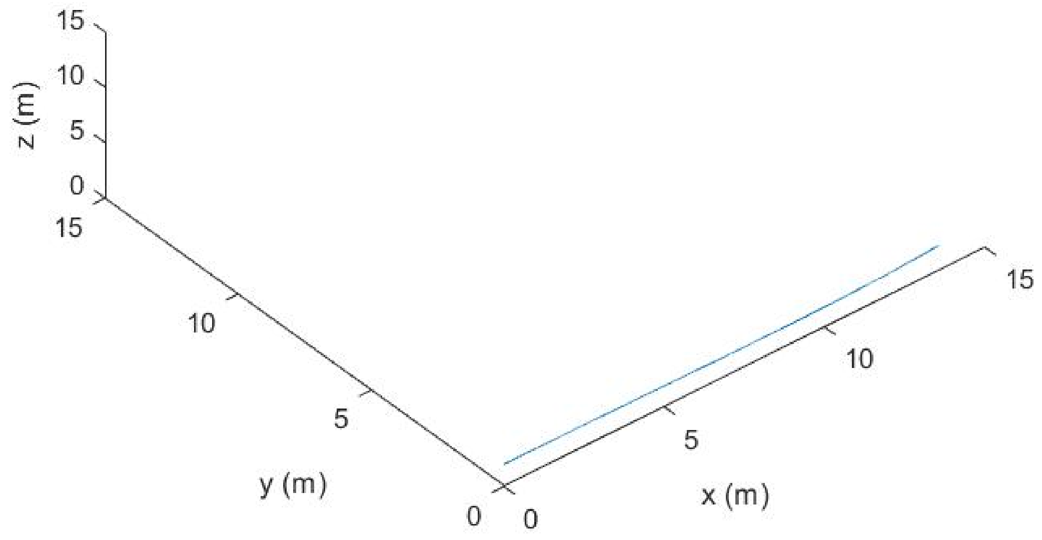


Figure 3: Baseline 3D Plot

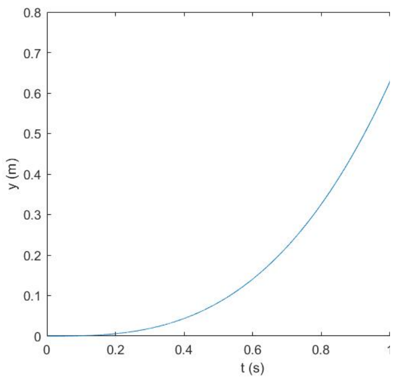


Figure 4: Baseline y Plot

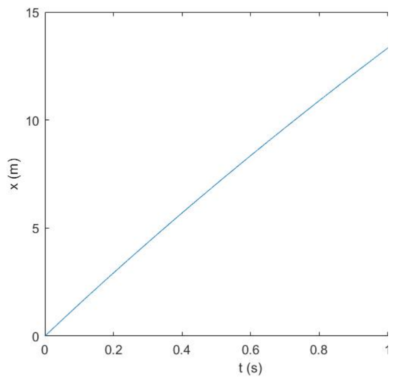


Figure 5: Baseline x Plot

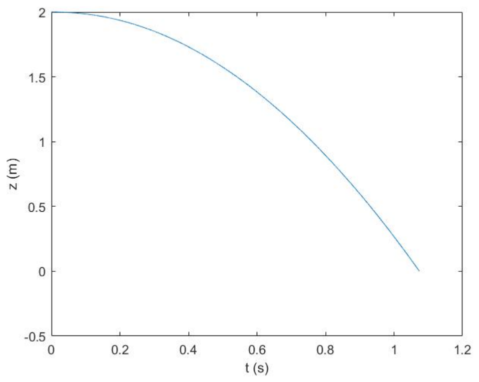


Figure 6: Baseline z Plot

9.2.2 Case 1 Plots

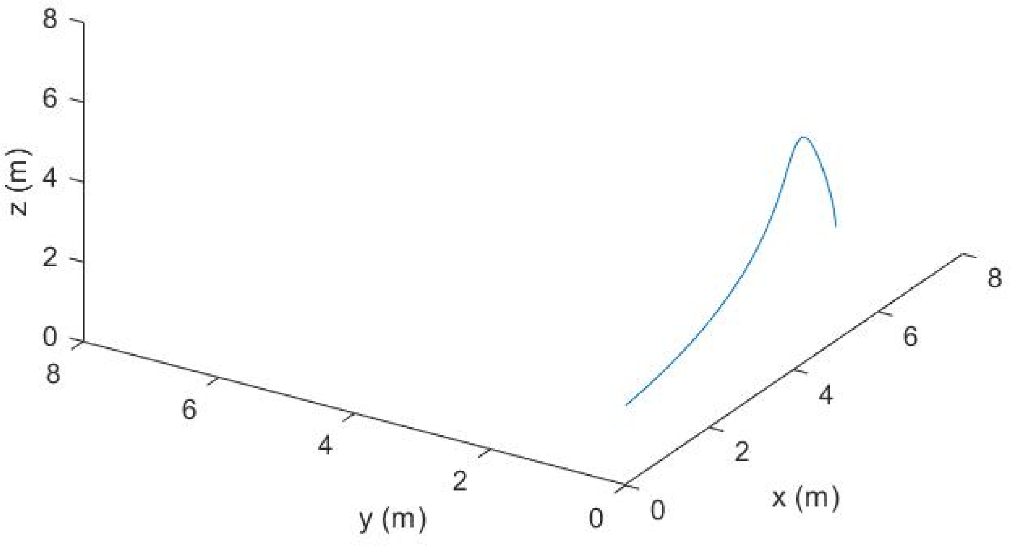
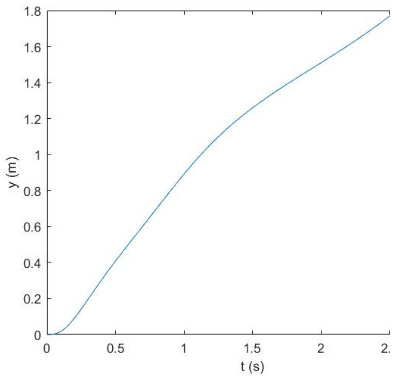
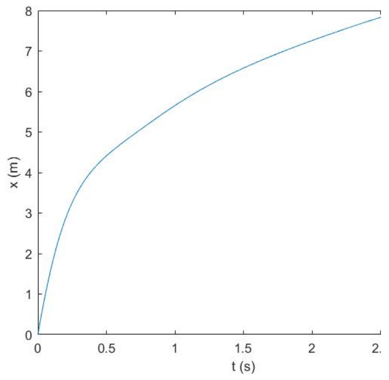


Figure 7: Case 1 3D Plot





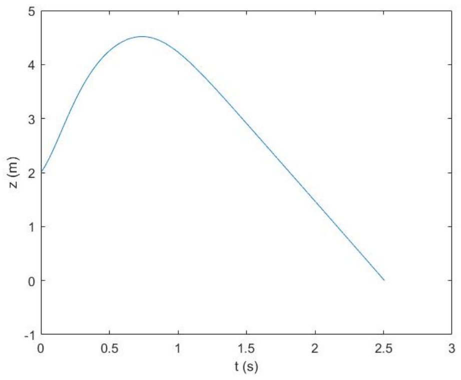


Figure 8: Case 1 y Plot **Figure 9:** Case 1 x Plot **Figure 10:** Case 1 z Plot

9.2.3 Case 2 Plots

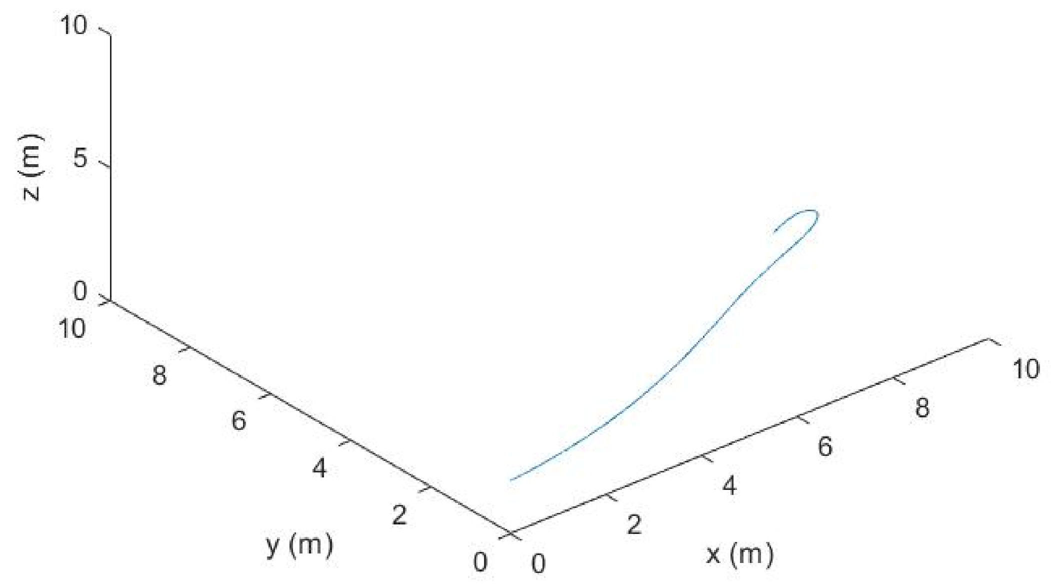


Figure 11: Case 2 3D Plot

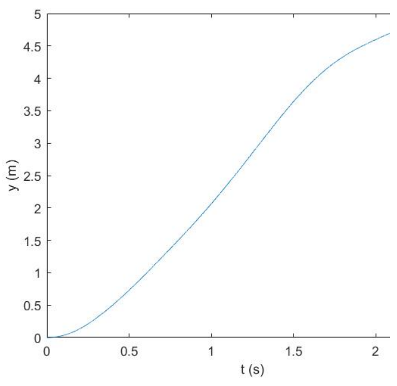


Figure 12: Case 2 y Plot

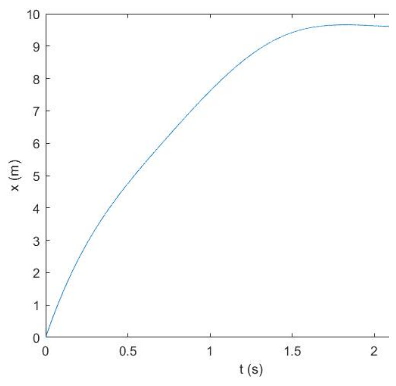


Figure 13: Case 2 x Plot

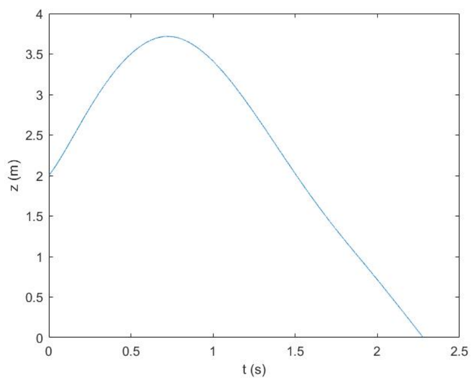


Figure 14: Case 2 z Plot

9.2.4 Case 3 Plots

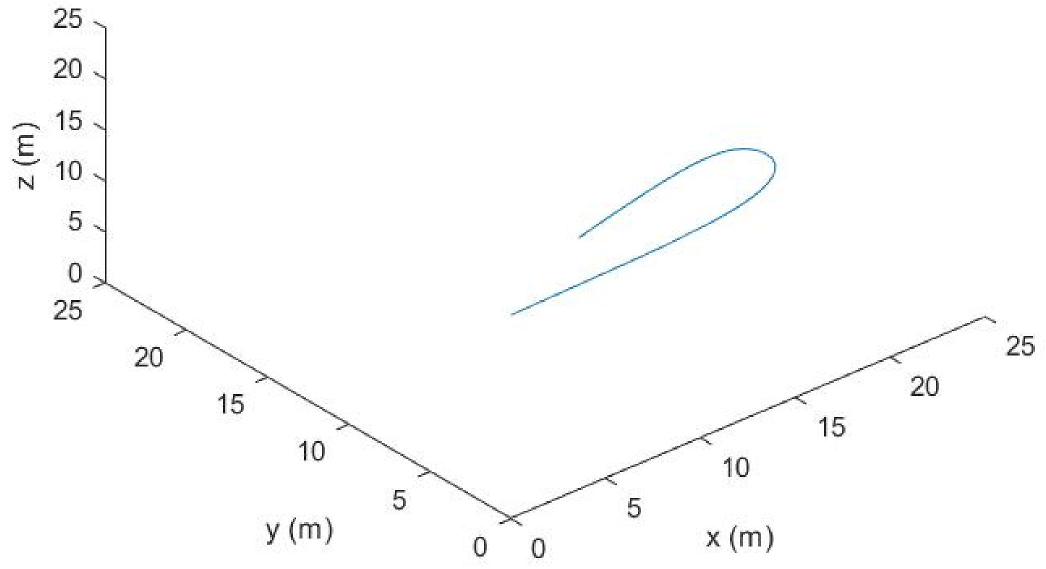


Figure 15: Case 3 3D Plot

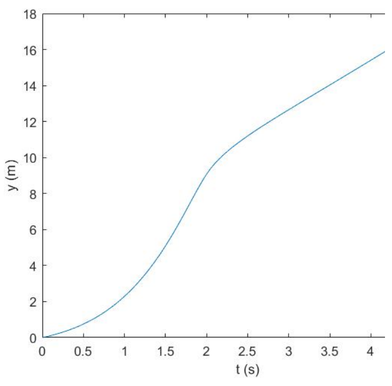


Figure 16: Case 3 y Plot

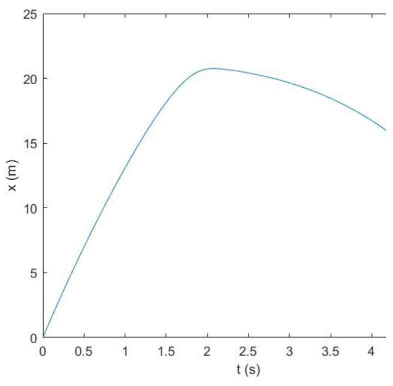


Figure 17: Case 3 x Plot

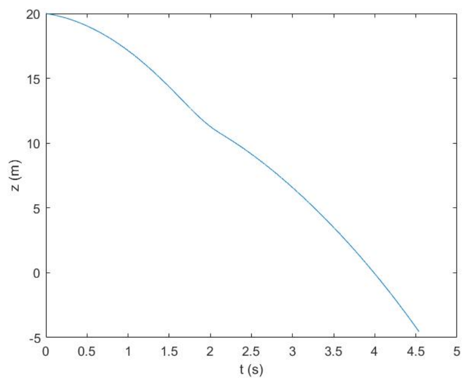


Figure 18: Case 3 z Plot

9.2.5 Case 4 Plots

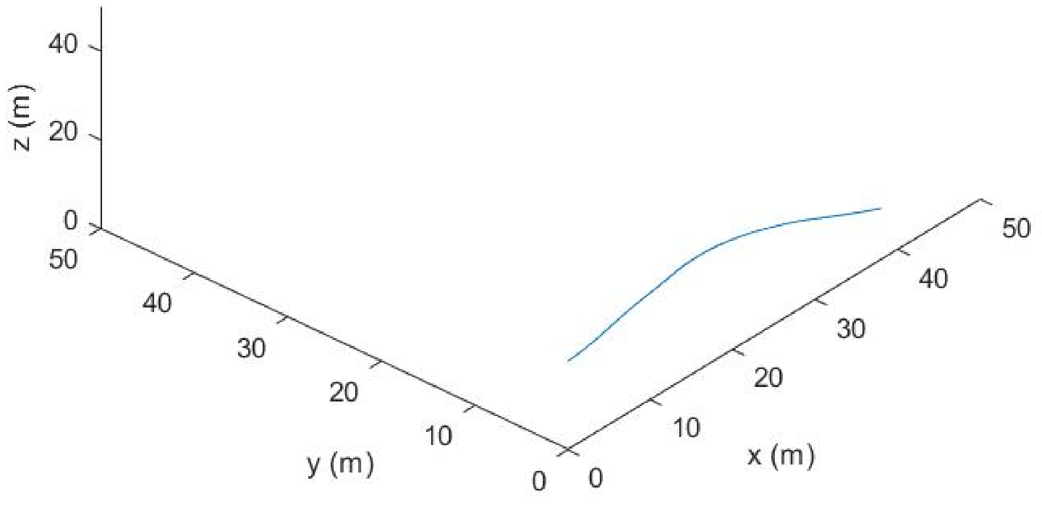


Figure 19: Case 4 3D Plot

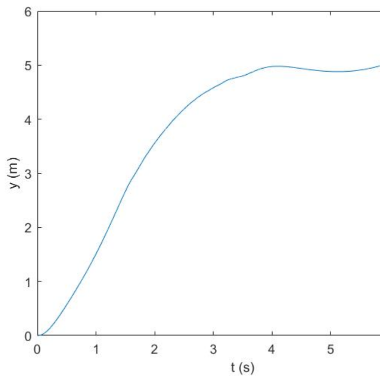


Figure 20: Case 4 y Plot

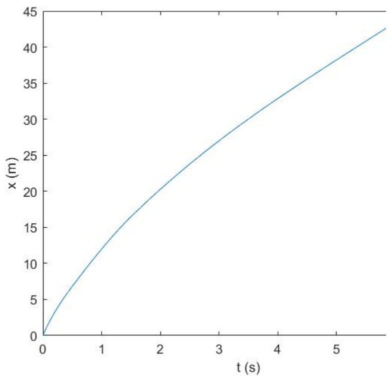


Figure 21: Case 4 x Plot

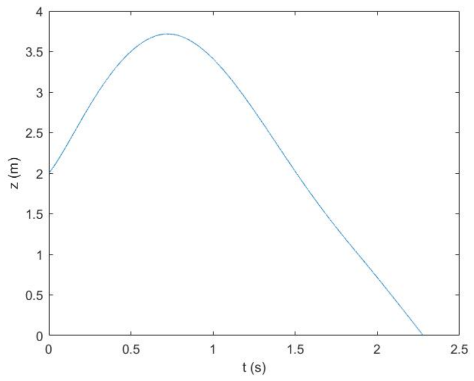


Figure 22: Case 4 z Plot

9.3 Mathematical Development

This is the transformation matrix used to convert the xyz frame to the XYZ frame:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & sin\phi & cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix} \tag{22}$$

where ϕ is the roll angle and θ is the pitch angle.