参赛队员姓名：杨博为

中学：北京市十一学校

省份：北京

国家/地区：中国

指导教师姓名：韩江帆 范博昭

指导教师单位：北京市十一学校

论文题目：Study of Frisbee Flight and Rebound Dynamics Based on Frame Conversion and Vector Transformation

Study of Frisbee Flight and Rebound Dynamics Based on Frame Conversion and Vector Transformation

Yang Bowei
Instructor: Han Jiangfan Fan Bozhao
Beijing National Day School
Beijing, China

Innovation Statement

I claim that the paper submitted is my own accomplishment under the guidance of the instructor. To my best knowledge, the content in the paper does NOT include any results from other researchers' works, except as specifically noted and acknowledged in the text. If there is any dishonesty, I am willing to bear all relevant responsibilities.

Yang Bowei

Abstract

Frisbee is a kind of sports equipment in an airfoil shape that can glide in the air for throw and catch. The Frisbee-like unmanned aerial vehicles have potential as the next generation aircraft. This paper is the first one to investigate the touchdown dynamics of the Frisbee to the best of the author's knowledge. The numerical simulation of Frisbee is developed based on rigid body dynamics and aerodynamic database from existing experiments. The moment of inertia of Frisbee is determined using the three-wire pendulum method. The Frisbee is launched by a launcher and the motion is recorded by two cameras. A frame-by-frame analysis of recorded flight video using the software tracker provides the linear velocities, position, and other dynamic parameters before and after touchdown. A comparison between experimental measurement and numerical simulation results is done. The results demonstrate that the established model captures the influence of aerodynamic forces and self-spinning of Frisbee during flight and rebound. The future directions that this research might be extended and applied are discussed.

Key words: **Frisbee; Rebound; Rigid body dynamics; Attitude dynamics**

Contents

Variable Sheet

Variable	Representation
A_0	Initial amplitude
b	Damping coefficient
C_D	Drag coefficient
C_L	Lift coefficient
C_M	Pitching moment coefficient
d	The length of Frisbee
F	Force
g	Acceleration due to gravity
H	Distance between lower disk and top board
I	Moment of inertia
k	Spring constant
L	Angular momentum
m	Mass of Frisbee
m_0	Mass of the lower disc
(p, q, r)	Three components of angular velocity in embedded frame
q_∞	Dynamic pressure
r_{pc}	Displacement from center of Frisbee to impact point
S	Surface area of Frisbee
$(T_{i2e}, T_{e2s}, T_{s2v})$	Transformation matrix between four reference frames
T_0	period of oscillation of the lower board
T	period of oscillation of lower board with Frisbee
t	Time
t_1	The time Frisbee is in contact with the ground
(v, v', v'', v''')	Velocity in four frames
$\dot{v}$	Acceleration
w	The width of Frisbee
(x, y, z)	The three direction of position in inertial frame
α	The sideslip angle
β	The pitch angle
δ	Phase angle
(ϕ, θ, ψ)	Euler angles
$(\dot{\phi}, \dot{\theta}, \dot{\psi})$	Time derivative of Euler angles
σ	Angle between Frisbee surface and ground
τ	Torque
ρ	Density of air
χ	Coefficient of restitution
ω	Angular velocity in inertial frame
ω'	Circular frequency of the damping harmonic motion
ω_0	Natural circular frequency
$\dot{}$	Time derivative of a parameter
$\hat{}$	Unit vector

1 Introduction

1.1 Introduction to Frisbee

Generally speaking, Frisbee (also known as a flying disc) refers to a type of aircraft-like sports equipment. It is aerodynamically designed and obtains stability by spinning during flight. Frisbee is disk-shaped and formed an airfoil looking from its cross-section so that it could generate lift in the air. In addition, in recent years, Frisbee has become a popular sport. Over 300 million Frisbees have been sold worldwide [5]. The World Flying Disc Federation is founded in 1985 and has been organizing ultimate and freestyle Frisbee competitions ever since. Frisbee class is available at the author's school. It is quite entertaining as a kind of sport, so it has become one of the most popular sports classes at the author's school. In the class, many different ways to throw a Frisbee were taught to and mastered by students, including forehand, backhand, and hammer. The difference between throwing Frisbee with forehand and backhand is that the direction of rotation will be opposite for these two techniques. When Frisbee is thrown using the hammer technique, the Frisbee will be thrown vertically up onto the sky. It will flip upside down automatically in the air and glide slowly back to the ground. The three techniques are demonstrated in figure 1.

(a) (b) (c)

Figure 1: Three Frisbee throwing techniques (a):backhand throw (b):forehand throw (c):hammer throw

1.2 Discovery and potential application of the problem

When playing Frisbee at school in the class, we will deliberately throw Frisbee towards the ground to watch Frisbee rebound and glide for fun. However, sometimes, Frisbee does not rebound significantly. They will glide for a really short distance before falling to the ground. A successful rebound is presented in figure 2. The rebound phenomenon is complex and intriguing because it is related to many dynamic parameters.

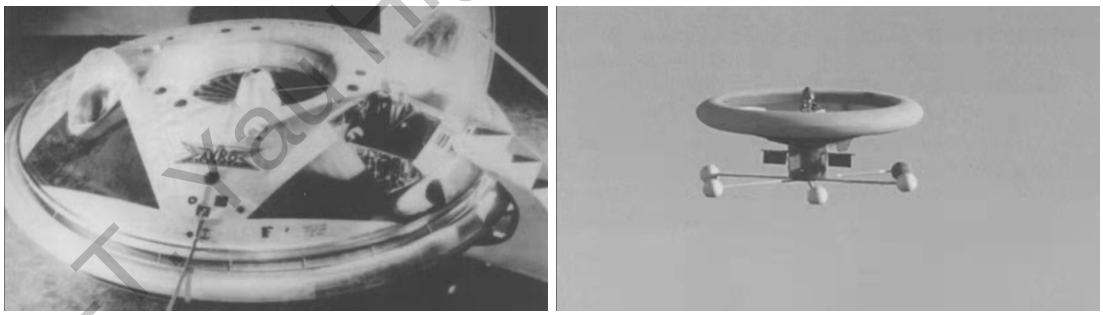
The study of the landing of Frisbee has many potential application prospects. It is known that Frisbee gains stability the way similar to many innovative types of aerial manned and unmanned vehicles. Method to study the landing and rebound process of Frisbee can be applied to the landing of many similar self-stabilized aircraft. As an example, a disc-wing aircraft prototype called the Avrocar is proposed in the 1950s. The lift is generated through a giant torborotor located at its center, and the torborotor also provides gyroscopic action to the aircraft[5]. Another example is the Turboplan, a radio-controlled toy sold in the early 1980s.

(a) (b) (c)

Figure 2: Sequence of a successful rebound (a): moment before rebound (b): during rebound (c): moment after rebound

The Turboplan shares more similarities with Frisbee than Avrocar. The rotor at the center provides lift and stability, and a rotating ring surrounding the rotor also provides a large portion of the stability [5]. Pictures of the Avrocar and the Turboplan [5] are provided in figure 3.

In addition, Frisbee-like unmanned aerial vehicle has certain advantages over the conventional unmanned aerial vehicles. Frisbee has a low-aspect-ratio wing, meaning that it has a relatively short wingspan compare to the width of the wing. A low-aspect-ratio wing is known to have high performance at low speed and a high attack angle [5]. Also, the intrinsic structural strength of the circular shape of Frisbee is quite appealing for aircraft design. The symmetric shape and the elegant design means that it is easier to carry compared with aircraft with high-aspect-ratio wings, which can suffer severe damage during packaging [5]. These characteristics of Frisbee suggest the prospect that a novel unmanned aerial vehicle could be designed based on the dynamics of Frisbee, and study over the landing process can provide insights for the design.



(a) (b)

Figure 3: (a): Avrocar; (b): Turboplan

1.3 Previous works on Frisbee

Researches on the dynamics and aerodynamics of Frisbee in flight can date back to 50 years ago. G. D. Stilley and D. L. Carstens studied the application of Frisbee flight principle on special deliveries [3]. The kinematic and aerodynamic parameters of Frisbee have been analyzed by numerically solving dynamic differential equations simplified at different levels. One paper explained the flight mechanics of Frisbee qualitatively including imposed aerodynamic forces and moments [1].

M. Hubbard and S. A. Hummel from UC Davis provided a model using the Euler angle to calculate the forces exerted on Frisbee in 2000. The aerodynamic coefficients were provided as a simplified fitting function [10]. In 2005, V. R. Morrison provided a simplified 2-dimensional model based on the assumption that Frisbee's lift force and drag force remain vertical and horizontal during flight. The paper also provided the deduction of the equations used to calculate the aerodynamic forces based on hydromechanics [16].

A thorough study over the simulation of Frisbee flight is conducted by W. J. Crowther and J. R. Potts from the University of Manchester in 2007. By calculating forces in different reference frames and unify the forces, the complete dynamic differential equations were provided for simulation. The aerodynamic coefficients used was based on experiment data [15]. In 2010, K. Baumback provided an extension to the model suggested by V. R. Morrison and described Frisbee's 3-dimensional motion using program simulation [11]. In 2019, one research was conducted, focusing on explaining Frisbee's flight using Newton's second law [14], which provides some qualitative analysis but not yet providing the full quantitative description in detail.

Several other studies focused on measuring the aerodynamic parameters of Frisbee using different methods. Using a MEMS triaxial gyroscope, Y. Weizman measured the angular velocity, the aerodynamic torque, and the decay of spin rate of Frisbee during flight [18]. A new method of identifying Frisbee's aerodynamic coefficients by real flight data was suggested by S. A. Hummel and M. Hubbard. This new method could be used as an alternative to wind tunnel tests [9]. Frisbee aerodynamic coefficients are obtained in wind tunnel tests by J. R. Potts and C. J. Crowther. The data indicates that the pitching moment coefficients become 0 when the attack angle is approximately 9° [4]. In 1999, K. Yasuda recorded the trajectory of Frisbee during free flight using two sets of cameras. By analyzing the videos, the frequency of occurrence of different initial parameters when Frisbee is thrown by hand is determined [12]. A comprehensive study over aerodynamic characteristics of flying dynamic of flying disc with different thickness, cavity, and edge shapes was reported using wind tunnel experiments [13]. Additionally, a Frisbee launching robot was built according to the flying principle of Frisbee. The robot can automatically control the initial parameters such as velocity and pitch angle of Frisbee [17].

The landing process of Frisbee is quite similar to the process of stone skipping. They both involve the process of a spinning object impact with a certain surface, and rebound is also observed in both processes. There are two highly renowned research papers that investigated the dynamic of stone skipping, and determined a "magic angle" of impact to maximize the numbers of times of skipping both experimentally [8] and theoretically [7]. By mimic, the process of stone skipping, the British army successfully designed a bouncing bomb during World War II to take down damps. The bomb was released near the surface of water away from the damp, and after skipping for several times, the bomb travels near the damp and explode [5].

1.4 Contribution of this research

After fully searching on this and the similar areas of study, no research papers are found to directly model and conduct experiments over the landing process of Frisbee. To provide a better understanding of the landing dynamics of Frisbee,

this work established a theoretical model used to calculate the rebound process. The model takes friction, the impulse from the ground, and different aerodynamic forces into consideration. A damping harmonic motion model is used to describe the impact. The ground is considered to be elastic, and Frisbee is considered to have no change in shape in this model. A program simulation is performed based on the dynamic differential equation derived in the theoretical part. Several sets of experiments were conducted with different initial launching parameters. The experimental result is compared with the simulation result to verify the proposed theoretical model. This research can inspire future works to determine the parameters that dedicate to the most stable landing process for similar aircraft.

This paper is organized as follows. Section 2 describes the theoretical model of Frisbee flight based on rigid body dynamics and vector transformation. Section 3 gives theoretical deduction for Frisbee's rebound process based on rigid body dynamics. Section 4 provides the flow chart for the program used to simulate Frisbee's flight and rebound, as well as Frisbee's typical flight path and rebound pattern base on the program. Section 5 gives detailed descriptions to the experiments conducted measuring coefficients and constants. Section 6 testifies the model suggested, and also analyzes the experiment data obtained. Conclusions to this study as well as modifications that could be carry out in future work are summarized in section 7.

2 Theoretical Model of Frisbee Flight

In this section, Frisbee dynamic during flight is described using rigid body dynamic knowledge. The forces and moments exerted on Frisbee is unified by transformation matrix. There are in total 12 degree of freedom (DOF) for Frisbee flight, including position (3DOF), velocity (3DOF), angular velocity (3DOF) and attitude (3DOF). The method of unifying forces and moments is based on the model suggested in previous work done by Potts and Crowther [15]. Totally, 12 dynamic differential equations are obtained as result for further simulation. The dynamic equations are presented at the end of this section.

The model established in section 2 is based on the model established in previous works done by Potts and Crowther [15]. Modification is done to improve to the original model. The linear velocity in the previous model is calculated in the embedded frame of Frisbee. With Frisbee continuously rotating, the embedded frame rotates with Frisbee. This means that the components of linear velocity is continuously oscillating with respect to time, which will be illustrate by program simulation in section 4. The linear velocity of aerial vehicle should be processed in the inertial frame for its clearance. For example, modern airplane provides information of its sink rate and the ground speed to the pilot. A continuously oscillating velocity could be confusing and thus conceal certain crucial information for flight control. Additionally, linear velocity in inertial frame is crucial for obtaining the rebound model in section 3. The velocity is therefore transformed into the inertial frame for calculation, which provides better clearance.

2.1 Forces during flight

Forcing model of Frisbee during flight could be separated into aerodynamic force and gravity. The aerodynamic characteristic of Frisbee varies with the attack angle, and is hard to determine qualitatively by calculation. Complete wind tunnel experiment was conducted to determine the three aerodynamic parameter, lift coefficient, drag coefficient, and pitching moment coefficient, of standard Frisbee in 2002 [4], and the data of the conducted experiment is presented in three figures in another research paper crafted by the same authors in 2007 [15, Figure 3]. The data is extracted from the research paper using an online software, while definition of the three parameter and the data is presented in the following sections.

2.1.1 Lift force

Lift force results from Frisbee's wing shaped cross section. The lift force could be calculated using equation (2.1) as suggested in previous work [13].

$$F_L = q_\infty S C_L \tag{2.1}$$

In the equation, S is the area of the surface of the airfoil. In this case, is the area of Frisbee, which is $0.05726m^2$. C_L is the lift coefficient of Frisbee, and q_∞ is the dynamic pressure. The dynamic pressure q_∞ is defined by equation (2.2) as suggested in previous work [13].

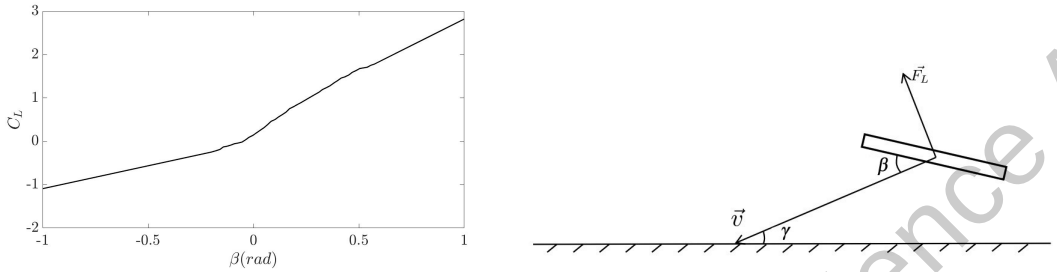
$$q_\infty = \frac{1}{2}\rho v^2 \tag{2.2}$$

In equation (2.2), ρ is the density of air and v is the instantaneous linear velocity. Combining equation (2.1) and (2.2), we get (2.3) to calculate the lift

force:

$$F_L = \frac{1}{2}\rho v^2 S C_L. \tag{2.3}$$

The variation of lift coefficient as a function of attack angle is presented in figure 4(a). The original data is extrapolated from the range of -0.2rad to 0.6rad to the range of -1rad to 1rad. The lift force is always in the direction perpendicular to the path of motion of Frisbee, as shown in figure 4(b).



(a) (b)

Figure 4: Lift force. (a) shows the data of lift coefficient, extrapolated from the range of -0.2rad to 0.6rad to the range of -1rad to 1rad. (b) shows the direction of the lift force.

2.1.2 Drag Force

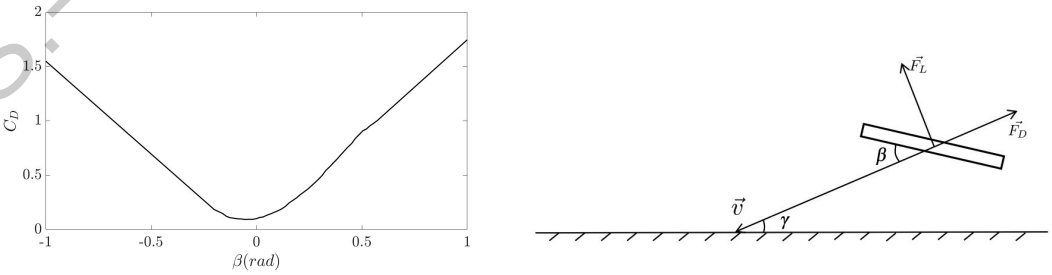
The drag force of Frisbee could be calculated using equation (2.4) as mentioned in previous work [13]:

$$F_D = q_\infty S C_D. \tag{2.4}$$

In the equation, S is the surface area of the Frisbee, C_D is the drag coefficient of Frisbee, and q is the dynamic pressure, which can be calculated using equation (2.2). Substituting q in equation (2.4) with equation (2.2), we get the equation for the drag force:

$$F_D = \frac{1}{2}\rho v^2 S C_D \tag{2.5}$$

Experiment data of drag coefficient, which also varies as a function of the attack angle, is presented in figure 5(a). The original data is extrapolated from the range of -0.2rad to 0.6rad to the range of -1rad to 1rad. The drag force is always with the direction of the path of motion of Frisbee, as shown in figure 5(b) below:



(a) (b)

Figure 5: Drag force. (a) shows the data of drag coefficient, extrapolated from the range of -0.2rad to 0.6rad to the range of -1rad to 1rad. (b) shows the direction of the drag force.

2.1.3 Gravity

During Flight, Frisbee also experiences a force due to gravity pointing toward positive z axis in inertial frame:

2.2 Moment

2.2.1 Pitching moment

During Frisbee flight, the lift force does not act directly on the center of mass of Frisbee. In fact, it exerts on a points slightly ahead the center of mass, and thus creates a torque that makes the Frisbee rotate around the y' axis in the embedded frame. In another word, with the action point of the lift force moving forward, the torque changes the pitch angle of Frisbee. A pitching moment coefficient C_M is thus defined as equation (2.6) to describe the effect [13].

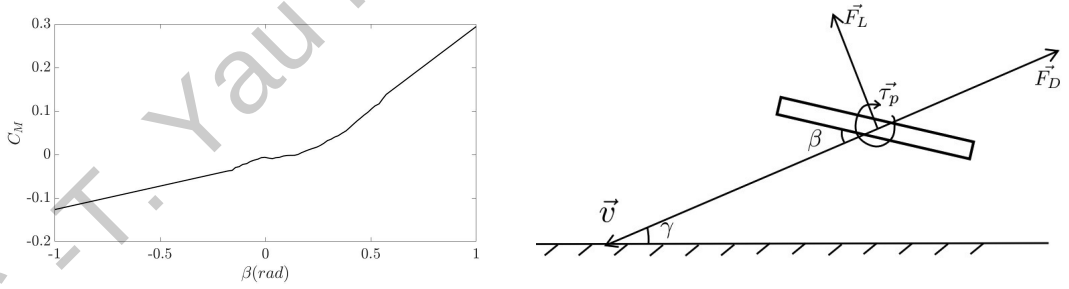
$$C_M = \frac{\tau_p}{q_\infty S d} \tag{2.6}$$

In the equation, τ_p is the pitching torque action on the Frisbee, S is the surface of Frisbee, d is the diameter of Frisbee, and q is the dynamic pressure in equation (2.2). The moment could then be find by substitute q in equation (2.2) into equation (2.6), which gives equation (2.7).

$$\tau_p = \frac{1}{2}\rho v^2 S d C_M \tag{2.7}$$

Using this equation, we can then calculate the value of the torque acting on the Frisbee if knowing the pitch angle and the pitching moment coefficient for the exact angle.

Experiment data of the pitching moment coefficient as a function of the attack angle from past research [4] is presented in figure 6(a). The original data is extrapolated from the range of -0.2rad to 0.6rad to the range of -1rad to 1rad. The way the moment act to rotate the Frisbee around y' axis is sketched in figure 6(b).



(a) (b)

Figure 6: Pitching moment. (a) shows the data of pitching moment coefficient, extrapolated from the range of -0.2rad to 0.6rad to the range of -1rad to 1rad. (b) shows the direction of the pitching moment.

2.3 Frame conversion of forces and moments

Four frames will be used to calculate motion, the inertial frame, Frisbee's embedded reference frame, the sideslip frame, and the velocity vector frame. This system of convert and unify vectors is first proposed by [15]. Definition for the frames are presented below.

2.3.1 Definition of four frames

The inertial frame has x and y axis lying on the surface of ground and z axis perpendicular to ground, assuming that the ground is perfectly horizontal. The embedded reference frame has the origin set at the center of mass of Frisbee, x' and y' axis lying horizontally on the plane formed by the Frisbee's upper surface, and z' perpendicular to the surface of Frisbee. Originally, the x axis point toward the direction of motion, as the Frisbee rotates, the embedded frame rotate along with Frisbee around z' axis.

The sideslip frame $x''y''z''$ is obtained by rotating the embedded frame with respect to the z' axis so that the x'' axis align with the direction of the projection of velocity on the $x'y'$ plane. The velocity vector frame $x'''y'''z'''$ is then obtained by rotating sideslip frame along y'' axis so that the x''' axis align with the velocity vector. The angle which distinct the embedded frame from sideslip frame is the sideslip angle α; the angle distinct the sideslip frame from velocity vector frame is the attack angle β. Sketch of the four frames are presented in figure 7

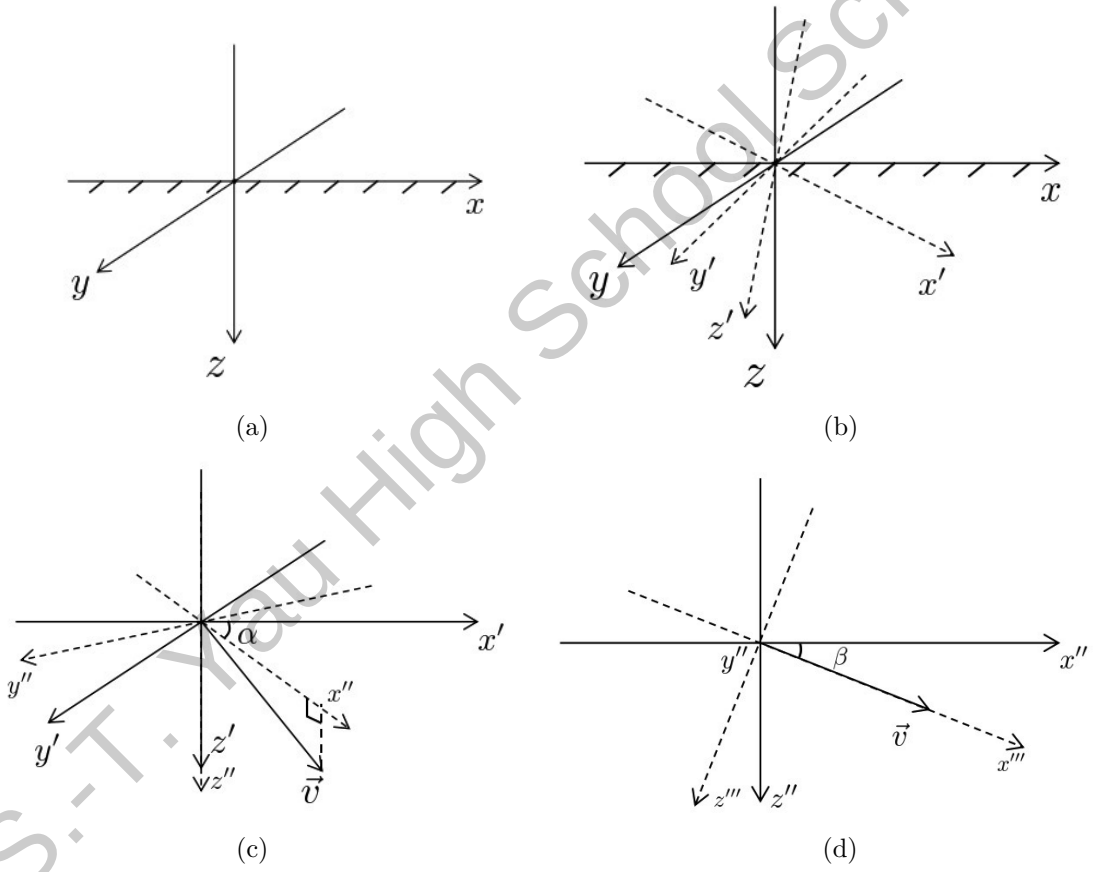


Figure 7: Illustration of the four frames and the relationship between them. (a): Inertial frame; (b): Embedded frame; (c): Sideslip frame; (d): Velocity vector frame;

The linear velocity of Frisbee is calculated in the inertial frame, because the inertial frame is generally fixed, and therefore Newton's law can be easily applied to calculate motion from forces in this frame. The angular velocity is calculated in the embedded reference frame.

2.3.2 Euler angles

Euler angles are three angles used to describe the orientation of an object by break the angle of orientation down into components with respect to different axis. There are multiple different conventions of describing the sequence of rotation [2]. The sequence used in this research is the zyx sequence, breaking the orientation down into rotation around z, y and x axis in succession. The Euler angles in this convention are commonly known as the yaw, pitch and row angle, labeled ϕ, θ and ψ respectively, which are widely used to describe the motion of aircraft. Figure 8 described the rotation sequence.

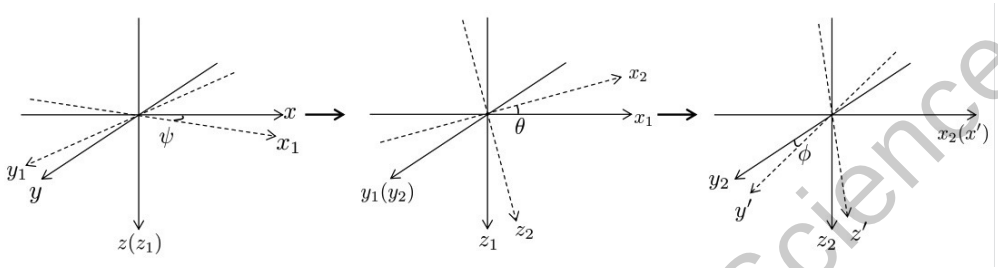


Figure 8: Euler angle rotation sequence. The original frame is rotated for ψ degree around the z axis. The obtained frame is then rotated for θ degree around the y axis. Finally, the frame is rotated for ϕ degree around the x axis.

In order to describe the attitude of Frisbee relative to the ground, a rotational matrix in terms of Euler angles is defined to transform inertial frame into the embedded frame. The matrix is decomposed into the multiplication of three rotational matrix in equation (2.8).

$$T_{i2e} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

The Euler angles are time dependent variables, and their derivatives are related to the angular velocity of Frisbee in the embedded frame, which has components p, q, and r, by equation (2.9) [2].

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos\theta} \begin{bmatrix} \cos\theta & \sin\phi\sin\theta & \cos\phi\sin\theta \\ 0 & \cos\phi\cos\theta & -\sin\phi\cos\theta \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.9}$$

2.3.3 Converting forces into inertial frame

Based on the definition, the lift force points toward the negative z''' axis in velocity vector frame, perpendicular to the direction of speed and the y''' axis, while drag force points toward negative x''' axis in velocity vector frame. The pitching moment points toward positive y''' axis in velocity vector frame. The aerodynamic forces and moment could all be transformed into the inertial frame by multiplying several rotational matrices. The rotational matrix transforming embedded frame to sideslip frame is presented by equation (2.10), and The rotational matrix for transforming sideslip frame to velocity vector frame is presented in equation (2.11)

$$T_{e2s} = \begin{bmatrix} \cos\alpha & \sin\alpha & 0 \\ -\sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

$$T_{s2v} = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix} \tag{2.11}$$

The property of rotational matrix indicates that a reverse transformation is described by the transpose of the original rotational matrix. Therefore, the reverse rotational matrix from velocity vector frame to sideslip frame and from sideslip frame to embedded frame is obtained as equation (2.12)

$$\begin{aligned} T_{s2e} &= T_{e2s}^T \\ T_{v2s} &= T_{s2v}^T \end{aligned} \tag{2.12}$$

The sideslip angle and attack angle could be represented by the components of velocity in embedded frame and sideslip frame as suggested by figure 7. Cosine and sine of the two angles could be calculated by following equation (2.13) and equation (2.14).

$$\begin{aligned} \cos\alpha &= \frac{v_{x'}}{\sqrt{v_{x'}^2 + v_{y'}^2}} \\ \sin\alpha &= \frac{v_{y'}}{\sqrt{v_{x'}^2 + v_{y'}^2}} \end{aligned} \tag{2.13}$$

$$\begin{aligned} \cos\beta &= \frac{v_{x''}}{\sqrt{v_{x''}^2 + v_{z''}^2}} \\ \sin\beta &= \frac{v_{z''}}{\sqrt{v_{x''}^2 + v_{z''}^2}} \end{aligned} \tag{2.14}$$

The components of velocity in embedded and sideslip frame could be obtained by multiply the velocity in inertial frame with respective transformation matrix:

$$\begin{bmatrix} v_{x'} \\ v_{y'} \\ v_{z'} \end{bmatrix} = T_{i2e} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{2.15}$$

$$\begin{bmatrix} v_{x''} \\ v_{y''} \\ v_{z''} \end{bmatrix} = T_{e2s} T_{i2e} \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{2.16}$$

2.4 Newton's second law

The linear motion of Frisbee is calculated by the differential form of Newton's second law in inertial frame:

$$\vec{F} = \frac{d(m\vec{v})}{dt} \tag{2.17}$$

By arranging this equation in matrix form, components of velocity and position of Frisbee could be calculated:

$$m \begin{bmatrix} \dot{v_x} \\ \dot{v_y} \\ \dot{v_z} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} \tag{2.18}$$

2.5 Angular Momentum Theorem

During Flight, Frisbee is spinning with a certain angular velocity, which gives the Frisbee an angular momentum. During Flight and Landing, torques are exerted on Frisbee, and Gyroscopic effect caused by the spinning could adjust the minor alter in attitude of Frisbee caused by torques. The following deductions analyze the influence spinning has on Frisbee's attitude.

The angular momentum of an object can be calculated as follows:

$$\vec{L} = I\vec{\omega} \tag{2.19}$$

which can be written in matrix form in term of their components as in equation 11:

$$\begin{bmatrix} L_x \\ L_y \\ L_z \end{bmatrix} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.20}$$

In the equation, the products of inertia I_{xy}, I_{xz}, I_{yx}, I_{yz}, I_{zx}, I_{zy} is determined to be 0 because the motion of Frisbee in air can be fully dissected into rotation in three directions. The equation therefore becomes:

$$\begin{bmatrix} L_x \\ L_y \\ L_z \end{bmatrix} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{2.21}$$

Torque can be related with angular acceleration $\dot{\omega}$ by the following equation:

$$\begin{aligned} \vec{\tau} &= I\dot{\vec{\omega}} \\ &= \frac{d\vec{L}}{dt} \end{aligned} \tag{2.22}$$

From equation of Coriolis [20, 2], we have:

$$\vec{\tau_e} = \frac{d\vec{L}_e}{dt} + \vec{\omega_e} \times \vec{L} \tag{2.23}$$

Substitute $\vec{L}$ with equation (2.19), and arrange the equation in matrix form, we have:

$$\begin{bmatrix} \tau_{x'} \\ \tau_{y'} \\ \tau_{z'} \end{bmatrix} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} L_{x'} \\ L_{y'} \\ L_{z'} \end{bmatrix} \tag{2.24}$$

Therefore, the derivative of angular velocity in embedded frame ω_e is:

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}^{-1} \begin{bmatrix} \tau_{x'} \\ \tau_{y'} \\ \tau_{z'} \end{bmatrix} - \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix}^{-1} \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} L_{x'} \\ L_{y'} \\ L_{z'} \end{bmatrix} \tag{2.25}$$

2.6 Dynamic Differential Equation

2.6.1 Position

The position of Frisbee as a function of time can be calculated through a set of integrals:

$$\begin{aligned} \int_{x_0}^{x} dx &= \int_{0}^{t} v_x dt \\ \int_{y_0}^{y} dy &= \int_{0}^{t} v_y dt \\ \int_{z_0}^{z} dz &= \int_{0}^{t} v_z dt \end{aligned} \tag{2.26}$$

The integrals above could be written into dynamic differential equation in matrix form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} \tag{2.27}$$

2.6.2 Linear velocity

Define an aerodynamic coefficient matrix in velocity vector frame:

$$C_a = \begin{bmatrix} -C_D \\ 0 \\ -C_L \end{bmatrix} \tag{2.28}$$

which gives us the magnitude and direction of the aerodynamic forces. The aerodynamic forces could therefore be determined in velocity vector frame as:

$$F_{va} = \frac{1}{2}\rho v^2 S C_a \tag{2.29}$$

Transform the forces into the inertial frame by multiplying the transformation matrix:

$$F_{ia} = T_{g2e}^T T_{e2s}^T T_{s2v}^T F_{va} \tag{2.30}$$

By adding gravity to the aerodynamic force in inertial frame, the total force exerted on Frisbee during flight is determined:

$$F_{if} = F_{ia} + \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \tag{2.31}$$

Substitute equation (2.31) into Newton's second law in equation (2.18) to calculate the acceleration, and expand the equation, we get the dynamic differential equation for the components of linear velocity in inertial frame:

$$
\begin{aligned}
m\dot{v_x} =& \frac{1}{2}(SC_D\rho(\cos\beta(\sin\alpha(\cos\phi\sin\psi - \cos\psi\sin\phi\sin\theta) \\
& - \cos\psi\cos\alpha\cos\theta) - \sin\beta(\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta)) \\
& (v_x^2 + v_y^2 + v_z^2)) - \frac{1}{2}(SC_L\rho(\sin\beta(\sin\alpha(\cos\phi\sin\psi \\
& - \cos\psi\sin\phi\sin\theta) - \cos\psi\cos\alpha\cos\theta) + \cos\beta(\sin\phi\sin\psi \\
& + \cos\phi\cos\psi\sin\theta))(v_x^2 + v_y^2 + v_z^2)))
\end{aligned} \tag{2.32}
$$

$$
\begin{aligned}
m\dot{v_y} =& -\frac{1}{2}((SC_D\rho(\cos\beta(\sin\alpha(\cos\phi\cos\psi + \sin\phi\sin\psi\sin\theta) \\
& + \cos\alpha\cos\theta\sin\psi) - \sin\beta(\cos\psi\sin\phi - \cos\phi\sin\psi\sin\theta)) \\
& (v_x^2 + v_y^2 + v_z^2)) - \frac{1}{2}(SC_L\rho(\sin\beta(\sin\alpha(\cos\phi\cos\psi + \sin\phi\sin\psi\sin\theta) \\
& + \cos\alpha\cos\theta\sin\psi) + \cos\beta(\cos\psi\sin\phi - \cos\phi\sin\psi\sin\theta)) \\
& (v_x^2 + v_y^2 + v_z^2)))
\end{aligned} \tag{2.33}
$$

$$
\begin{aligned}
m\dot{v_z} =& (mg + \frac{1}{2}(SC_D\rho(\cos\beta(\cos\alpha\sin\theta - \cos\theta\sin\phi\sin\alpha) \\
& - \cos\phi\sin\beta\cos\theta)(v_x^2 + v_y^2 + v_z^2)) - \frac{1}{2}(SC_L\rho(\sin\beta(\cos\alpha\sin\theta \\
& - \cos\theta\sin\phi\sin\alpha) + \cos\beta\cos\phi\cos\theta)(v_x^2 + v_y^2 + v_z^2)))
\end{aligned} \tag{2.34}
$$

2.6.3 Angular velocity

Similar to the aerodynamic forces Frisbee experiences, a moment coefficient matrix is also defined in velocity vector frame as:

$$
C_{moment} = \begin{bmatrix} 0 \\ C_M \\ 0 \end{bmatrix} \tag{2.35}
$$

which gives the magnitude and direction of the aerodynamic moment. The moment could then be determined in velocity vector frame:

$$
\tau_{va} = \frac{1}{2}\rho v^2 SdC_{moment} \tag{2.36}
$$

The aerodynamic torque could then be transformed into the embedded frame by multiplying the transformation matrix.

$$
\tau_{ea} = T_{e2s}^T T_{s2v}^T \tau_{va} \tag{2.37}
$$

Substitute equation (2.37) into equation (2.25) to calculate the angular acceleration, and we get the dynamic differential equation for the component of angular velocity in embedded frame:

$$
I_x\dot{p} = -(I_z qr - I_y qr + \frac{1}{2}(\rho(v_x^2 + v_y^2 + v_z^2)SdC_M\sin\alpha)) \tag{2.38}
$$

$$
I_y\dot{q} = I_z pr - I_x pr + \frac{1}{2}(\rho(v_x^2 + v_y^2 + v_z^2)SdC_M\cos\alpha) \tag{2.39}
$$

$$
I_z\dot{r} = I_x pq - I_y pq \tag{2.40}
$$

For Frisbee, because it's shaped in a circle, so the rotational inertia with respect to x and y axis is the same, and thus:

$$I_z\dot{r} = 0 \tag{2.41}$$

2.7 Summary of the equations

Based on the deductions in the previous sections, dynamic of Frisbee during flight is described by 12 differential equations, which dominates the 12 DOF. Equation (2.32), (2.33), and (2.34) describes the linear velocity; Equation (2.38), (2.39), and (2.41) describes the angular velocity in the embedded frame. The Euler angle is described by equation (2.9), which is a matrix containing derivatives for three angle. The position is described by equation (2.27).

3 Theoretical Model of Frisbee Rebound

The suggested rebound model in this section is the first model established to describe the landing process of Frisbee. The rebound model is based on the existing model for Frisbee flight. By adding in the impulse exerted on the Frisbee from the ground, the model described the exact motion of Frisbee when impact with the ground under the influence of aerodynamic forces, moments and the gyroscopic effect. A basic model for calculating the friction exerted on Frisbee is also suggested, which was not included in the dynamic equation to simplify the theoretical model.

In this research, the ground is considered elastic and equivalent to a damped spring. That means when the Frisbee touched down, the ground will exert force on the Frisbee the way a damped spring do. This assumption is made because the surface of the playground where Frisbee is typically played at is made of plastic rubber, which is more elastic than the plastic Frisbee. Additionally, by treating Frisbee as a rigid body, its motion during rebound can be better highlighted because deviations caused by small deformation is neglected. Moreover, when designing Frisbee-like self-spin UAVs, the damping harmonic motion model can also be applied the designed aircraft. The effect of shock absorber carried by the UVA on the landing gear is similar to the ground treated as elastic in this model.

3.1 Basic model of ground

The ground is treated as a damping spring in this model. The basic dynamic differential equation for damped harmonic motion [19] is:

$$m\frac{d^2z}{dt^2} = -b\frac{dz}{dt} - kz \tag{3.1}$$

By solving the equation, the position as a function of time [19] during damping harmonic motion is obtained as follows:

$$z = A_0 e^{\frac{-bt}{2m}} \cos(\omega' t + \delta) \tag{3.2}$$

The phase angle δ in the equation is simplified by neglecting the small difference between the initial equilibrium position of the ground and the established equilibrium position after Frisbee land on the ground, and approximated to be

$-\frac{\pi}{2}$ in this scenario. The negativity is because Frisbee comes from the negative part of z axis. Therefore, equation (3.2) becomes:

$$z = A_0 e^{\frac{-bt}{2m}} \sin \omega' t \tag{3.3}$$

Take the first derivative of the position function to obtain velocity function:

$$v = -A_0 e^{\frac{-bt}{2m}} (\frac{-bt}{2m} \sin \omega' t - \omega' \cos \omega' t) \tag{3.4}$$

When time $t = 0$, we have:

$$v_0 = A_0 \omega' \tag{3.5}$$

Assuming that during rebound, Frisbee is in contact with the ground for t_1 seconds. At time $t = t_1$, we then have:

$$v_1 = -A_0 \omega' e^{\frac{-bt_1}{2m}} \tag{3.6}$$

Define a coefficient of restitution χ as the initial rebound speed over the ultimate falling speed:

$$\begin{aligned} \chi &= \frac{\mid v_1 \mid}{v_0} \\ &= e^{\frac{-bt_1}{2m}} \end{aligned} \tag{3.7}$$

The damping coefficient b is obtained as:

$$b = -\frac{2m \ln \chi}{t_1} \tag{3.8}$$

The following equation is used to calculate the angular frequency ω' [19]:

$$\omega' = \omega_0 \sqrt{1 - (\frac{b}{2m\omega_0})^2} \tag{3.9}$$

Substitute ω_0 into equation , and square both side of the equation, we have:

$$\begin{aligned} \omega'^2 &= \frac{k}{m}(1 - \frac{m}{k}(\frac{b}{2m})^2) \\ &= \frac{k}{m} - \frac{\ln^2 \chi}{t_1^2} \end{aligned} \tag{3.10}$$

At time $t = t_1$, we have the relationship between the angular frequency as follows:

$$\begin{aligned} \sin \omega' t_1 &= 0 \\ \omega' t_1 &= \pi \\ \omega' &= \frac{\pi}{t_1} \end{aligned} \tag{3.11}$$

We have the following equation to calculate the spring constant:

$$k = m(\frac{\pi^2}{t_1^2} + \frac{\ln^2 \chi}{t_1^2}) \tag{3.12}$$

3.2 Determine Frisbee's initial attitude during rebound

3.2.1 Determine the angle between Frisbee and ground

The exact angle between Frisbee and ground is hard to be solved directly. However, this angle is the same as the angle between the z' axis and z axis, which could be solved through vector transformation. Define a unit vector $\hat{z'}$ pointing at positive z' direction in embedded frame first.

$$\hat{z'} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.13}$$

Representation of this unit vector in inertial frame $\hat{z'}_i$ could be calculated as:

$$\begin{aligned} \hat{z'}_i &= \begin{bmatrix} z'_{ix} \\ z'_{iy} \\ z'_{iz} \end{bmatrix} = C_{i2e}^T \hat{z'} \\ &= \begin{bmatrix} \sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta \\ \cos\phi\sin\psi\sin\theta - \cos\psi\sin\phi \\ \cos\phi\cos\theta \end{bmatrix} \end{aligned} \tag{3.14}$$

The direction of $\hat{z}$ is presented in the graph in figure 9.

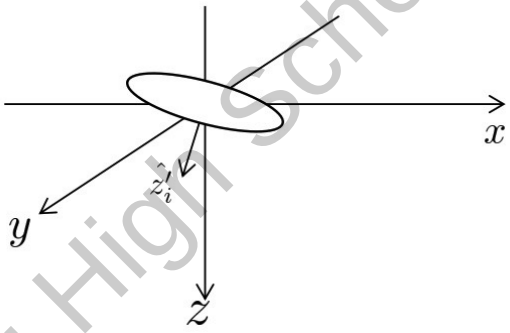


Figure 9: Direction of $\hat{z}$, the unit vector points at the positive z' direction in the embedded frame

Define another unit vector $\hat{z}$ pointing at positive z axis in inertial frame. By multiplying this vector with unit vector $\hat{z'}_i$, we have:

$$\hat{z} \cdot \hat{z'}_i = \cos\sigma = \cos\phi\cos\theta \tag{3.15}$$

In the equation, σ is the angle between $\hat{z}$ and $\hat{z'}_i$, and is also the angle between Frisbee surface and the ground.

3.2.2 Locating the point of impact with ground

The point of impact with the ground is the lowest point on the edge of Frisbee. With its location determined, we can calculate the torque on Frisbee exerted by the ground. To locate the point of contact with ground, a displacement vector $\vec{r}_{pc}$ is defined to be the position of the point of impact relative to the center of Frisbee. Representation of this vector in coordinate system is presented in figure 10.

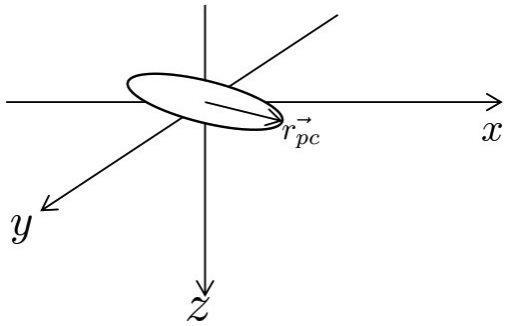


Figure 10: The displacement indicated by the vector $\vec{r}_{pc}$ represents the relative position of point of contact

The location could then be determined by adding the position of Frisbee's center in space to this vector $\vec{r}_{pc}$. The magnitude of this vector is simply the radius of Frisbee, which is $\frac{d}{2}$. The projection of this vector on the xy surface is:

$$r_{xy} = \frac{d}{2}\cos\sigma \tag{3.16}$$

This projection is the magnitude of the vector formed by the x and y component of $\vec{r}_{pc}$. This component vector is in the opposite direction as the vector formed by the x and y component of $\hat{z'}_i$. Normalize this formed vector of $\hat{z'}_i$, and multiply the normalized vector with the magnitude of the component vector, the component vector could be expressed as:

$$\begin{bmatrix} r_{pcx} \\ r_{pcy} \end{bmatrix} = -\frac{r_{xy}}{\sqrt{z'^2_{ix} + z'^2_{iy}}} \begin{bmatrix} z'_{ix} \\ z'_{iy} \end{bmatrix} \tag{3.17}$$

This is the x and y component of the displacement vector $\vec{r}_{pc}$. Because we have the angle between this vector and the ground, therefore, the z component of $\vec{r}_{pc}$ could easily be calculated as:

$$\begin{aligned} r_{pcz} &= \frac{d}{2}\sin\sigma \\ &= \frac{d}{2}\sqrt{1-\cos^2\sigma} \end{aligned} \tag{3.18}$$

Expand the equation for the x and y component, the displacement vector $\vec{r}_{pc}$ is therefore arranged in its components as:

$$\begin{bmatrix} r_{pcx} \\ r_{pcy} \\ r_{pcz} \end{bmatrix} = \frac{d}{2}\begin{bmatrix} -\cos\phi\cos\theta(\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta)/\sqrt{1-\cos^2\sigma} \\ \cos\phi\cos\theta(\sin\phi\cos\psi - \cos\phi\sin\psi\sin\theta)/\sqrt{1-\cos^2\sigma} \\ \sqrt{1-\cos^2\sigma} \end{bmatrix} \tag{3.19}$$

The location of the point of contact in the inertial frame is thus:

$$r_{pci} = \begin{bmatrix} r_{xi} \\ r_{yi} \\ r_{zi} \end{bmatrix} = \begin{bmatrix} x + r_{pcx} \\ y + r_{pcy} \\ z + r_{pcz} \end{bmatrix} \tag{3.20}$$

In this equation, r_{pci} represents the component of the displacement vector of the point of contact with respect to the origin. The velocity of this point of contact in space could then be determined by taking the derivative of r_{pci}:

$$v_{pci} = \dot{r}_{pci} = \begin{bmatrix} \dot{r}_{xi} \\ \dot{r}_{yi} \\ \dot{r}_{zi} \end{bmatrix} = \begin{bmatrix} v_x + \dot{r}_{pcx} \\ v_y + \dot{r}_{pcy} \\ v_z + \dot{r}_{pcz} \end{bmatrix} \tag{3.21}$$

The equation for the derivative of the component of r_{pci} is obtained by direct calculation, but is too complicated and will not be presented here. However, it is included in the MATLAB source code in Appendix B, represented as x_dot, y_dot, and z_dot respectively.

3.3 Condition for touch down

The previous part gives the location of the lowest point of Frisbee, which is the point of contact. Using the location displacement vector, this part gives the conditions under which Frisbee is considered landed and left the ground. The conditions will be used in program simulation to determine the status of rebound Frisbee.

The ground is set at level in the inertial Frame. That means $y = 0$ and $x = 0$. The height, which is the z component of the position of the lowest point on Frisbee is provided as:

$$r_{zi} = z + r_{pcz} = z + \sqrt{1 - \cos^2 \sigma} \tag{3.22}$$

When Frisbee landed, the lowest point on Frisbee becomes lower then the ground level. The height is less or equal to 0. The condition could not be set at exactly equals to 0. Limited accuracy of the simulation indicates that height equals to 0 rarely occurs. Therefore, when the condition is as shown in equation (3.23), the Frisbee is considered landed.

$$z + \sqrt{1 - \cos^2 \sigma} \leq 0 \tag{3.23}$$

Similarly, when Frisbee leaves the ground, the lowest point on Frisbee becomes higher then the ground level. The height is larger or equal to 0. Therefore, when the condition is as shown in equation (3.24), the Frisbee is considered left the ground.

$$z + \sqrt{1 - \cos^2 \sigma} \geq 0 \tag{3.24}$$

3.4 Forces during impact

During Impact, Frisbee experienced two forces from the ground: a normal force pointing upward, which could be calculated using the damping spring model, and a friction force. Combining the aerodynamic forces illustrated in section 2, a complete model of forces exerted on Frisbee is presented in figure 11.

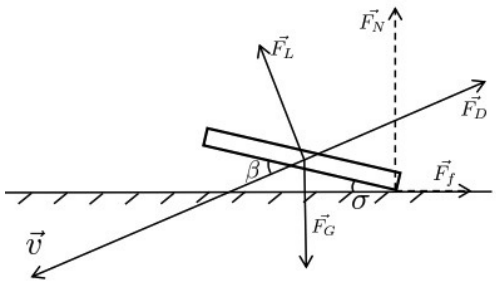


Figure 11: Complete force model during impact. $\vec{F_L}$ is the lift force, $\vec{F_D}$ is the drag force, $\vec{F_G}$ is the gravity, $\vec{F_f}$ is the friction, $\vec{F_N}$ is the normal force exerted on Frisbee by the ground.

3.4.1 Normal force

The normal force could be calculated using equation (3.1), and is arranged as follows with all the terms substituted:

$$F_{zi} = -b\dot{r_{zi}} - kr_{zi} \tag{3.25}$$

In this equaion, F_{zi} is the z component of the Force exerted by the ground in inertial Frame.

3.4.2 Friction

The Friction from the ground is of a determined value, and could be calculated by multiplying the normal force F_{zi} with the friction coefficient of the ground:

$$F_f = \mu F_{zi} \tag{3.26}$$

In this equation, μ is the friction coefficient. The friction force points at a certain direction opposite to the velocity of the point of contact relative to the ground. This relative velocity lies on the xy surface, and could be obtained by adding the x and y component of the velocity of the point of contact in space v_{pci} with the tangential velocity at the point of contact due to the rotation. The magnitude of the tangential velocity is:

$$|v_t| = r\frac{d}{2} \tag{3.27}$$

In this equation, v_t is the tangential velocity, and r is the z' component of Frisbee's angular velocity in the embedded frame, which is the rotational speed of Frisbee. This tangential velocity vector is perpendicular to $\vec{r}_{pc}$, the position of the point of impact relative to the center of Frisbee, which indicates that the tangential velocity is also perpendicular to the vector z_i'. The direction of this tangential velocity is linked with the direction of rotation, as shown in figure 12 and could be calculated by Equation (3.28). The magnitude of the tangential

velocity is calculated by multiplying angular velocity with the radius of Frisbee. The direction of the tangential velocity is determined by taking the direction perpendicular to the displacement vector $\vec{r}_{pc}$

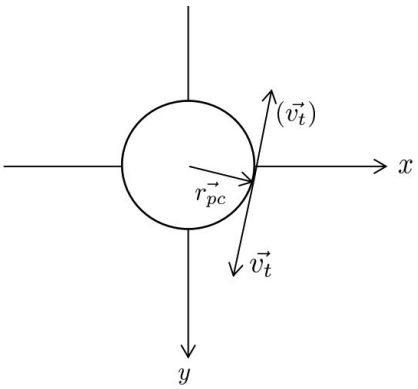


Figure 12: Direction of tangential velocity. v_t is the tengantial velocity, and it is always perpendicular to the displacement vector $\vec{r}_{pc}$

$$v_t = \text{sign}(r)\frac{d}{2}\frac{r}{\sqrt{z_{ix}'^2 + z_{iy}'^2}}\begin{bmatrix} z_{iy}' \\ -z_{ix}' \end{bmatrix} \tag{3.28}$$

In the equation, $\text{sign}(r)$ is a function of the sign of rotational speed r defined as follows:

$$\text{sign}(r) = \begin{cases} 1 & r > 0 \\ 0 & r = 0 \\ -1 & r < 0 \end{cases} \tag{3.29}$$

Combining all the equations, the relative velocity of the point of contact is determined as follows:

$$v_{rel} = v_t + \begin{bmatrix} \dot{r_{xi}} \\ \dot{r_{yi}} \end{bmatrix} \tag{3.30}$$

The friction could thus be calculated as:

$$F_f = \begin{bmatrix} F_{fx} \\ F_{fy} \end{bmatrix} = \mu F_{iz}\frac{-v_{rel}}{|v_{rel}|} \tag{3.31}$$

With the forces determined, the total force exerted by ground on Frisbee could be written in its component form as:

$$F_{ig} = \begin{bmatrix} F_{fx} \\ F_{fy} \\ F_{iz} \end{bmatrix} \tag{3.32}$$

3.5 Moment during impact

The moment exerted on Frisbee by ground during impact could be calculated by taking the cross product of the force arm, which is the displacement vector $\vec{r}_{pc}$, and the force exerted by the ground, F_i, as in equation (3.33). The moment could

then be transformed into the embedded reference frame by multiply the moment in inertial frame with the transformation matrix T_{i2e} as in equation (3.34).

$$\tau_{ig} = \begin{bmatrix} r_{pcx} \\ r_{pcy} \\ r_{pcz} \end{bmatrix} \times \begin{bmatrix} F_{fx} \\ F_{fy} \\ F_{fz} \end{bmatrix} \tag{3.33}$$

$$\tau_{eg} = T_{i2e}\tau_{ig} \tag{3.34}$$

In the equation, τ_{ig} is the moment from the ground, while τ_{eg} is the moment from the ground after being transformed into the embedded frame. A brief illustration of moments resulted from impact is presented in figure 13

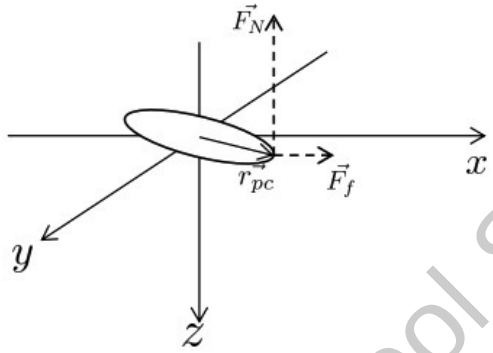


Figure 13: Moment during impact. The two forces (friction and normal force) are exerted on Frisbee by the ground, and can provide a moment.

3.6 Dynamic differential equation for rebound

The dynamic differential equation for the rebound process could be determined using the same process as for flight. By adding the aerodynamic force and the impact force together, we can get the equation for the total force, and therefore could determine the differential equation for linear velocity; by adding the aerodynamic moment and the impact moment together, we can get the equation for the total moment and therefore determine the differential equation for angular velocity. Equation (3.35) and equation (3.36) describes the total force and moment respectively. The total force is represented as F_i, while the total moment is represented as τ_e.

$$F_i = F_{if} + F_{ig} \tag{3.35}$$

$$\tau_e = \tau_{ea} + \tau_{eg} \tag{3.36}$$

However, after adding the equations up using the original version of MATLAB source code B (using the part noted in the code by %), the differential equation for the unsimplified model is found to be too complicated and would take about fifteen to twenty pages to write down all six equation. Beside, the computer used by the author has difficulties supporting such large amount of calculation, and would take a large amount of time to do the calculation. Therefore, a simplified model is considered, in which the friction is neglected. This simplified model has successfully reduced the length of the equation to an acceptable level. The differential equations for the linear velocity at x and y direction as well as for angular velocity r are the same as that in flight because no force is applied to

Frisbee in these directions. There are some adjustments to the differential equation for v_z and for p and q, and the new differential equation for v_z, p and q is presented as follows:

$$\begin{aligned} m\dot{v_z} =& -(k(z+(d(1-\cos\phi^2\cos\theta^2)^{\frac{1}{2}})/2)-gm+(b(2v_z(1 \\ & -\cos\phi^2\cos\theta^2)^{\frac{1}{2}}+dp\cos\phi\cos\theta^2\sin\phi+dq\cos\phi\cos\theta\sin\theta)) \\ & /(2(-\cos\phi^2\cos\theta^2+1)^{\frac{1}{2}})+(SC_D\rho(\cos\phi\sin\beta\cos\theta \\ & -\cos\alpha\cos\beta\sin\theta+\cos\beta\sin\alpha\cos\theta\sin\phi)(v_x^2+v_y^2+v_z^2))/2 \\ & +(SC_L\rho(\cos\beta\cos\phi\cos\theta+\cos\alpha\sin\beta\sin\theta \\ & -\sin\alpha\sin\beta\cos\theta\sin\phi)(v_x^2+v_y^2+v_z^2))/2) \end{aligned} \tag{3.37}$$

$$\begin{aligned} I_x\dot{p} =& (4I_zqr-4I_yqr+bd^2p\cos\phi^2\cos\theta^4-bd^2p\cos\phi^4\cos\theta^4 \\ & +d^2k\cos\phi\cos\theta^2\sin\phi-d^2k\cos\phi^3\cos\theta^4\sin\phi+ \\ & 4I_yqr\cos\phi^2\cos\theta^2-4I_zqr\cos\phi^2\cos\theta^2 \\ & +2SC_Md\rho v_x^2\sin\alpha+2SC_Md\rho v_y^2\sin\alpha+2SC_Md\rho v_z^2\sin\alpha \\ & +2bdv_z\cos\phi\cos\theta^2\sin\phi(1-\cos\phi^2\cos\theta^2)^{\frac{1}{2}} \\ & +2dkz\cos\phi\cos\theta^2\sin\phi(1-\cos\phi^2\cos\theta^2)^{\frac{1}{2}}+ \\ & bd^2q\cos\phi^2\cos\theta^3\sin\phi\sin\theta-2SC_Md\rho v_x^2\cos\phi^2\sin\alpha\cos\theta^2 \\ & -2SC_Md\rho v_y^2\cos\phi^2\sin\alpha\cos\theta^2-2SC_Md\rho v_z^2\cos\phi^2\sin\alpha\cos\theta^2) \\ & /(4\cos\phi^2\cos\theta^2-4) \end{aligned} \tag{3.38}$$

$$\begin{aligned} I_y\dot{q} =& I_zpr-I_xpr+(SC_Md\rho\cos\alpha(v_x^2+v_y^2+v_z^2))/2- \\ & (d\cos\phi\cos\theta(\cos\phi\cos\psi+\sin\phi\sin\psi\sin\theta) \\ & (\sin\phi\sin\psi+\cos\phi\cos\psi\sin\theta)(dk+2bv_z(1-\cos\phi^2\cos\theta^2)^{\frac{1}{2}} \\ & +2kz(1-\cos\phi^2\cos\theta^2)^{\frac{1}{2}}-dk\cos\phi^2\cos\theta^2+bdq\cos\phi\cos\theta\sin\theta \\ & +bdp\cos\phi\cos\theta^2\sin\phi))/(4(-\cos\phi^2\cos\theta^2+1)^{\frac{1}{2}}(\sin\phi^2+\sin\theta^2 \\ & -\sin\phi^2\sin\theta^2)^{\frac{1}{2}})+(d\cos\phi\cos\theta(\cos\phi\sin\psi-\cos\psi\sin\phi\sin\theta) \\ & (\cos\psi\sin\phi-\cos\phi\sin\psi\sin\theta)(dk+2bv_z(1-\cos\phi^2\cos\theta^2)^{\frac{1}{2}} \\ & +2kz(1-\cos\phi^2\cos\theta^2)^{\frac{1}{2}}-dk\cos\phi^2\cos\theta^2+bdq\cos\phi\cos\theta\sin\theta \\ & +bdp\cos\phi\cos\theta^2\sin\phi))/(4(-\cos\phi^2\cos\theta^2+1)^{\frac{1}{2}} \\ & (\sin\phi^2+\sin\theta^2-\sin\phi^2\sin\theta^2)^{\frac{1}{2}}) \end{aligned} \tag{3.39}$$

The equations are then used in MATLAB source code in Appendix C to model Frisbee's motion.

3.7 Summary of the equations

These three equations (3.37) (3.38) and (3.39) described in this section combining dynamic equations (2.32) (2.33) (2.34) (2.41) (2.9) (2.27) for v_x, v_y, r, ϕ, θ, ψ, x, y and z respectively described in section 2 constitutes the dynamic differential equation governing the rebound process. The 12 equations described the 12 DOF. The equations for rebound described in this section is simplified by neglecting the friction.

4 Program Simulation

Deriving the differential equation of Frisbee flight and rebound was done by MATLAB code, the source code is presented in Appendix A and B, which are for flight and rebound respectively. The derivation is done simply by following the logic of theoretical calculation and substitute certain equation by proper representations. The dynamic differential equation derived in section 2 and 3 are first order nonlinear inhomogeneous ordinary differential equations. They are too complicated to be solved analytically. Thus, they are solved using MATLAB ODE23s numerical solver, and program is coded to transfer these numerical solution into graphs containing dynamic variables such as linear velocity as a function of time. The source code for the dynamic simulation program is in Appendix C. Flowcharts for the coding logic of dynamic simulation program is presented below.

4.1 Flowchart of the dynamic simulation program

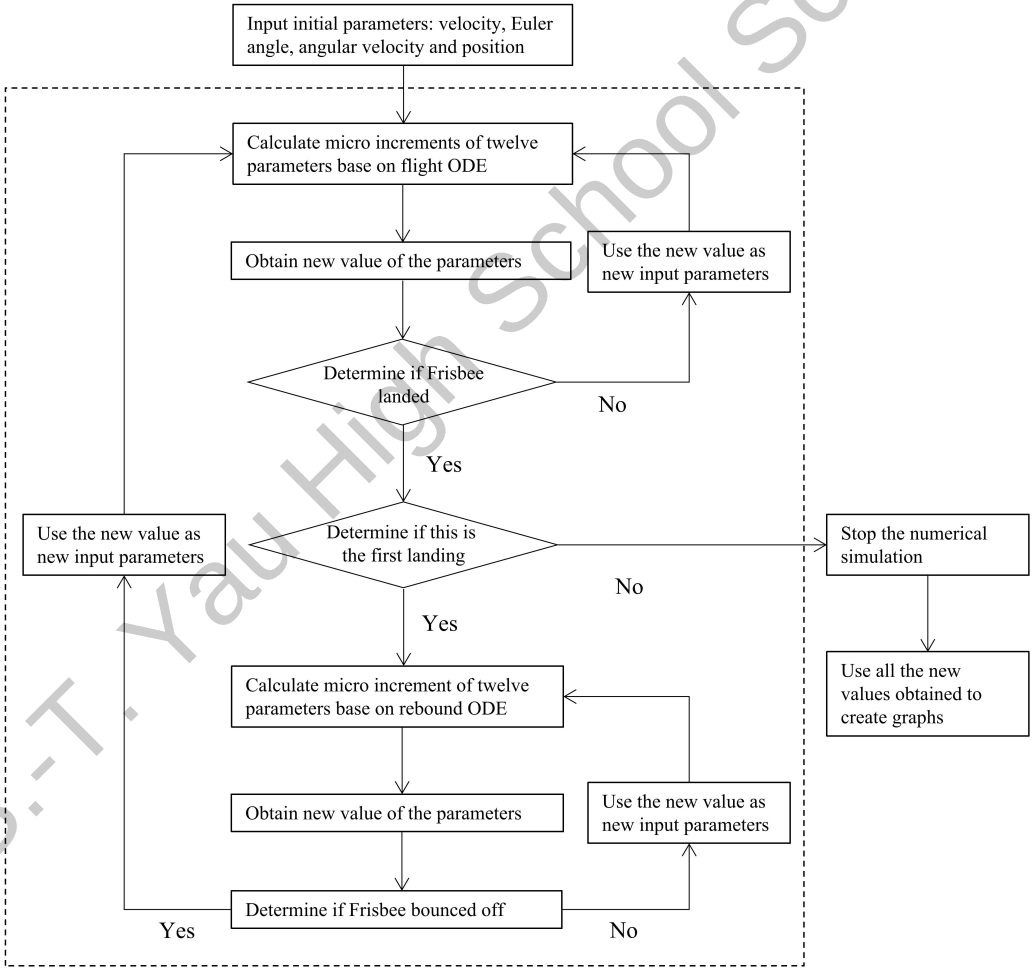


Figure 14: Flowchart for the numerical simulation of Frisbee dynamics.

4.2 Simulation of typical Frisbee flight path including rebound

In 1999, Kunio Yasuda measured the range of initial parameters for free Frisbee flights using camara sets. As indicated by analysis of the videos recorded, Yasuda determined that the most typical initial velocity, angular velocity and attack angle of Frisbee flight. A typical initial velocity is $10-10.5m/s$; a typical initial angular velocity is $350-450rpm$, which is about $37-47rad/s$; a typical attack angle is $10°$, which is about $0.175rad$ [12]. The initial launching height is arbitrarily to be $1m$, congruent with the height of human's hand. A pitch angle of $0.275rad$ is set. This indicates a $0.1rad$ angle make between the initial velocity and the ground, and can simulate the scenario that the Frisbee is thrown up into the air. Therefore, from these initial parameters, setting an accuracy at $1.0E-5$, the program can give us a calculated typical Frisbee flight path from the dynamic differential equation. The following graphs are the results, with different initial parameter sets labeled in the graph title. eight initial parameter sets are presented, the negativity of angular velocity indicates whether the Frisbee is thrown forehand or backhand (positive indicates forehand and negative indicates backhand). The spring constant and damping coefficient is set following the experiment result in section 5.

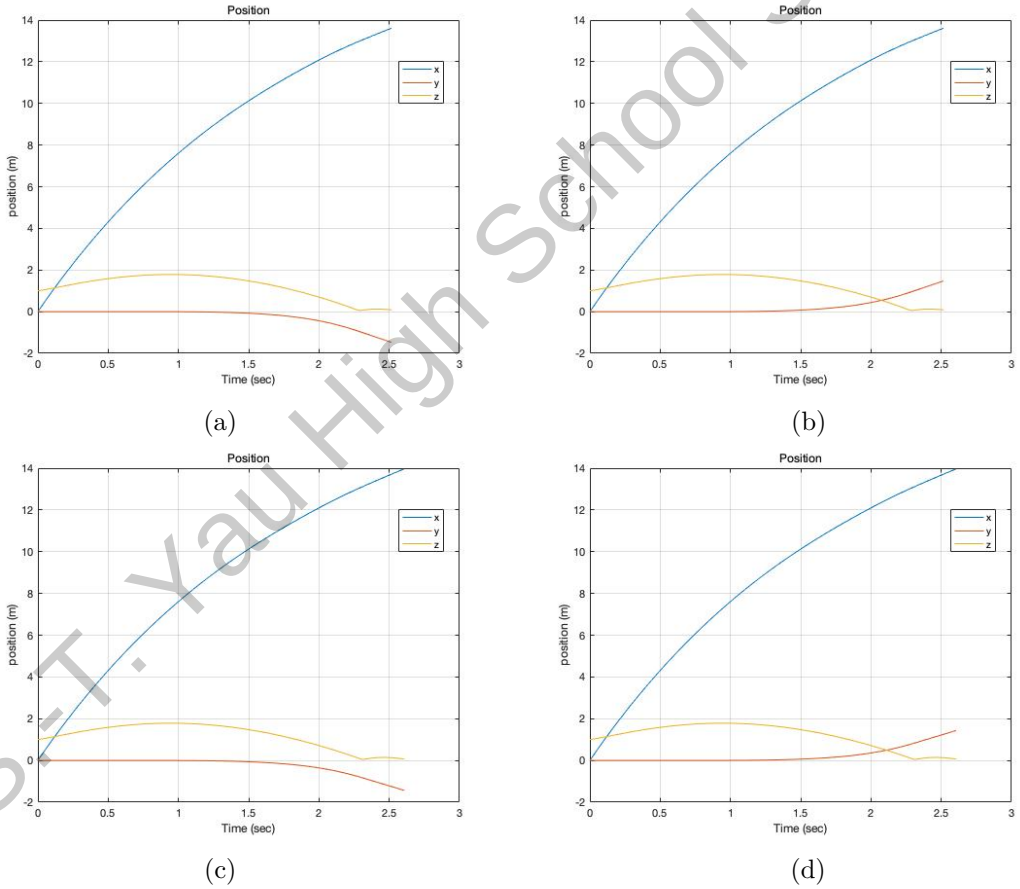


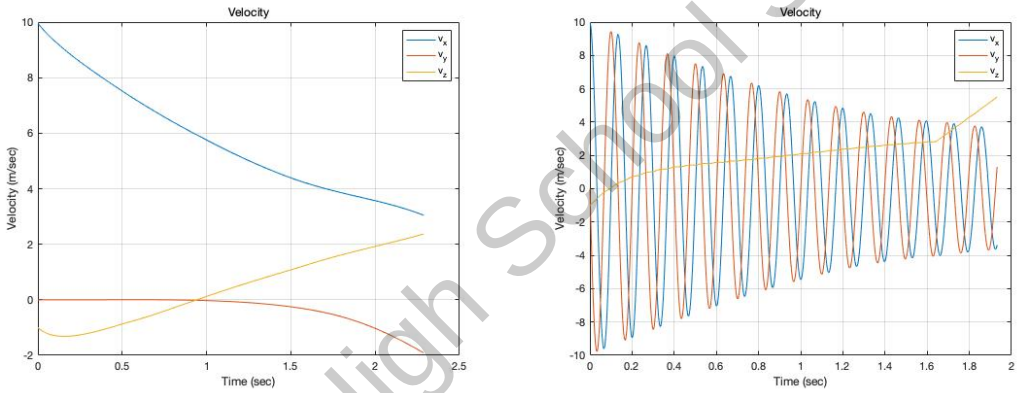
Figure 15: Position x, y, and x of Frisbee with different initial velocity, angular velocity an(a):$10m/s$, $37rad/s$, $0.175rad$ attack angle, $0.275rad$ pitch angle, $1m$ initial height; (b):$10m/s$, $-37rad/s$, $0.175rad$ attack angle, $0.275rad$ pitch angle, $1m$ initial height; (c):$10m/s$, $40rad/s$, $0.2rad$ head-up position; (d):$10m/s$, $-40rad/s$, $0.2rad$ head-up position

In the simulated result, the phenomenon that Frisbee shift to different y direction during flight according to the different direction of its initial spinning. Compare to a lower spin rate ($37rad/s$), Frisbee flight is more stable under high

spin rate ($47rad/s$). The yaw during flight is slightly reduced, presumably due to the larger initial angular momentum. The gradual deceleration during flight is also observed, as well as Frisbee's climb under the effect of lift force. The rebound of Frisbee is also included. Frisbee bounce up with reduced speed. Note that the position in vertical direction is inverted in negativity to help better understand the scenario. Thorough contrast between the theoretical model and the actual scenario is established by experiment.

4.3 Comparison of velocity in embedded frame and in inertial frame

The velocity is calculated in the inertial frame in this research, partially because velocity in inertial frame is clearer then in the embedded frame, as it was calculated in previous researches. A comparison between velocity in embedded frame and in inertial frame is provided in this part to illustrate the clarity mentioned. The velocity graph is obtained using two different programs, one is calculated in inertial frame, and the other is calculated in the embedded frame. The initial condition for the trial is $10m/s$, $0.275rad$ pitch angle, $0.175rad$ attack angle, $1m$ high, and $47rad/s$ angular velocity. The result is shown in figure 16.



(a) (b)

Figure 16: Comparison between velocity in inertial frame and in embedded frame. (a) shows velocity in inertial frame. (b) shows velocity in embedded frame. Velocity in embedded frame is continuously vibrating, which makes it hard to observe clear pattern.

The velocity in embedded frame is continuously oscillating, and it is very hard to obtain clear information from the graph. However, the velocity in the inertial frame is much clearer, since the velocity forms a smooth line. Therefore, no further vector transformation needs to be done to transform the velocity from the embedded frame to the inertial frame. The only drawback with the velocity dynamic differential equation in inertial frame is that the length of the equation is much longer then the dynamic differential equation in embedded frame. However, rely on computer, the length of the equation will not be a significant problem.

4.4 Simulation of rebound pattern

Typically, to achieve the degree of rebound author observed when engaging in Frisbee class, the Frisbee would have to be launched toward the ground. This way, the force exerted by the ground would be enough to invert the vertical velocity of Frisbee and adjust Frisbee's attitude, so that the lift generated would be enough to continue climbing up high. Typically, Frisbee won't fly up too high in rebound.

However, if a large force is applied to throw the Frisbee, giving the Frisbee a large initial velocity, Frisbee could climb up high during rebound. By setting the initial parameters to be $13m/s$, -60 rad/s, 0.1 rad attack angle, -0.1 rad pitch angle, and $0.1m$ from the ground, one typical rebound process (Frisbee do not fly up high) is established. The change in position, velocity, angular velocity and Euler angle for this high rebound is presented in figure 18. Additionally, By setting the initial parameters to be $15m/s$, 60 rad/s, 0.2 rad attack angle, -0.1 rad pitch angle, and $0.1m$ from the ground, one rebound process involving Frisbee climbing up high is established. The change in position, velocity, angular velocity and Euler angle for this high rebound is presented in figure 18.

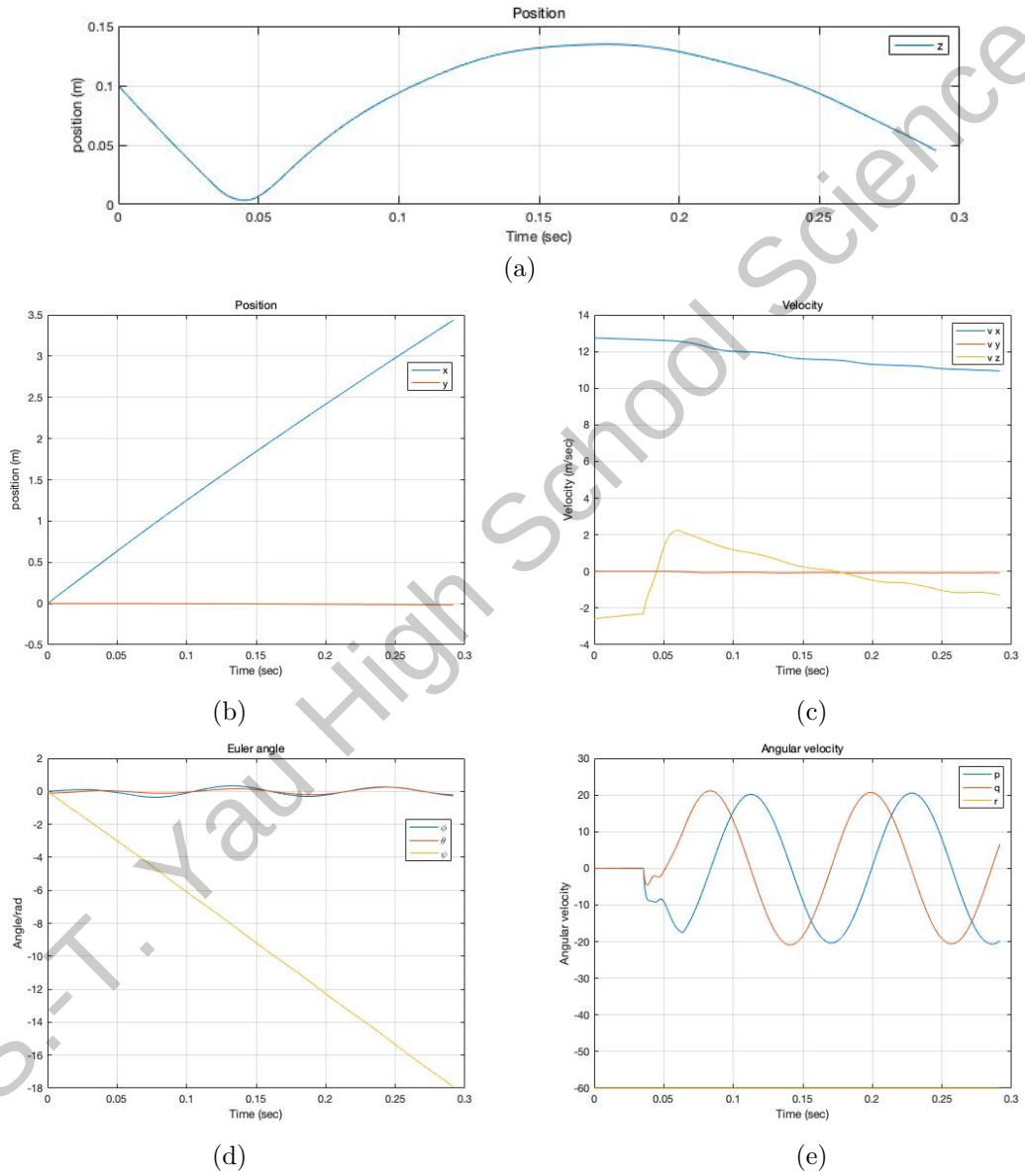


Figure 17: Initial parameters: position (0, 0, -0.1), velocity (13·cos(-0.2), 0, -15·sin(-0.2)), angular velocity (0, 0, -60), Euler angle (0, -0.1, 0); (a): z position of Frisbee during rebound process; (b):x and y position of Frisbee during rebound process; (c):velocity of Frisbee during rebound process; (d): Euler angle of Frisbee during rebound process; (e): angular velocity of Frisbee during rebound process

A clear pattern could be seeing in these five graphs, the rebound phenomenon on vertical direction could be identified. The impact caused a intense change

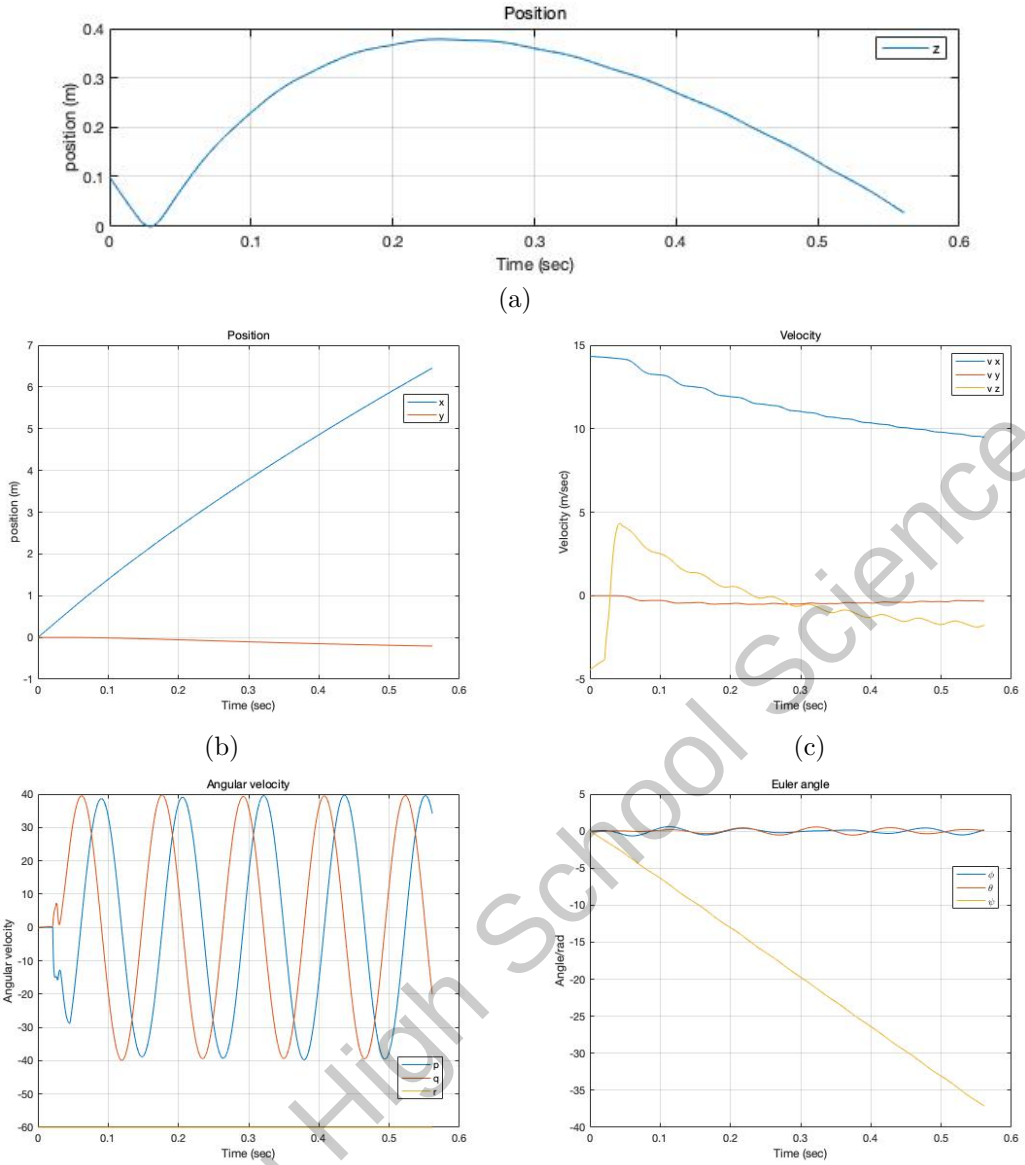


Figure 18: Initial parameters: position (0, 0, -0.1), velocity (15·cos(-0.3), 0, -15·sin(-0.3)), angular velocity (0, 0, -60), Euler angle (0, -0.1, 0); (a): z position of Frisbee during rebound process; (b):x and y position of Frisbee during rebound process; (c):velocity of Frisbee during rebound process; (d): Euler angle of Frisbee during rebound process; (e): angular velocity of Frisbee during rebound process

in Frisbee's attitude, result in drastic fluctuation of the two components of the angular velocity p and q. The vertical velocity has the direction shifted in a sudden, signaling the impact, and the noticeable fluctuation of the vertical velocity after rebound could be a result of the drastically fluctuating angular velocity.

5 Pre Experiments

5.1 Measuring the moment of inertia of Frisbee

5.1.1 Overview

In this experiment, a handmade three-wire pendulum was used to measure the moment of inertia of Frisbee. The moment of inertia of and object indicates the magnitude of its inertia when it rotates around a fixed axis, and this factor is used in the dynamic equation. I choose to use a three wire pendulum to conduct this experiment is because not only it's easier to build and operate, but also it's cheaper compare to a moment of inertia detector while obtaining results in an acceptable error bound.

5.1.2 Experiment Setup

The three-wire pendulum is made from a cardboard box. The structure of it is shown in figure 19(a). A circle is drew on the upper board as the upper disc and is fixed during the experiment. The lower disc is a piece of circle cardboard. Two discs are connected by three thin threads which have same length. The places where each thread is tied with the disc are equally set on the circumference. Threads are connected with the lower disc and the upper disc by toothpicks and adhesive tapes. Three holes were drilled on the both boards where threads should be connected with discs. Each threads were passed through the corresponding hole and tied it with a toothpicks tightly. Then, the toothpicks and threads were fixed on the both boards by adhesive tapes as shown in figure 19(b). By fixing the thread on discs in this way, the length of each thread could easily be adjusted by changing the place of the corresponding toothpick. The mass of threads and adhesive tapes are considered to be small enough to be ignored. Additionally, the side board of the box is found to be having a surprisingly impressive ability to prevent wind from getting inside the experiment area, so the experiment could be conducted out door without experiencing influence from wind.

(a) (b)

Figure 19: (a): Experiment setup for measuring moment of inertia. (b) Configuration of tooth-picks and adhesive tapes. The tooth picks are fixed on the board, and the threads are tied on the tooth picks.

In order to make sure that the lower disc of the pendulum is horizontal, a handmade gradienter was put on the disk and by observing if the bubble was locates in the middle point of the gradienter or not, as shown in figure 20, we

could determine whether the plane is horizontal. Then, we put the gradienter on the lower disc to confirm if it is perfectly horizontal.

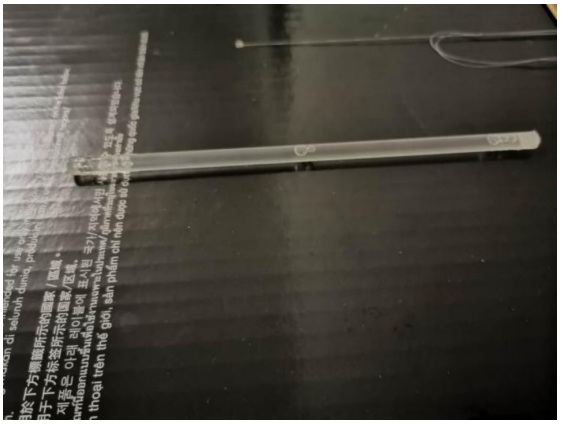


Figure 20: The handmade gradienter. The gradienter is used to make sure that the lower disc remains horizontal. Observe the location and the motion of the small bubble. If the small bubble is right at the middle of the gradienter and remains stable, then the board is considered to be horizontal.

Before the beginning of the experiment, some parameters of the setup should be measured. The moment of inertia of an object could be calculated in a three wire pendulum experiment using the following equation [21]:

$$I = I_1 - I_0 = \frac{gR_1R_2}{4\pi^2 H}[(m + m_0)T^2 - m_0T_0^2] \tag{5.1}$$

The parameters include the radius of the lower disc R_1, the radius of the upper disc R_2, the distance between two discs H, the mass of the lower disc m_0 and the mass of the object to be measured m, which is the mass of Frisbee. The acceleration due to gravity g is take as $9.8m/s^2$. The period of the oscillation of the lower board T_0 and the period of the oscillation of lower board with object on it T are the quantities to be measured through the experiment. The procedure for experiment is presented in the next section.

5.1.3 Experiment procedure

First, the period of oscillation of the lower board T_0 is measured. The lower board is rotated by a small angle and released as shown in figure 21. At the same time, a video is recorded until the lower board has oscillated ten times. The time the lower board spent to oscillate ten times is t_0. The process is repeated for six times. The average oscillation period is calculated as T_{0a}

Second, the period of oscillation T_1 is measured with Frisbee rotating horizontally. The Frisbee is put on the lower board, with the center of the Frisbee and the lower board aligned as shown in figure 22(a). Then, the lower board is rotate with a small angle and released. At the same time, a video is recorded until it has oscillated ten times. The time the lower board spent to oscillate ten times is t. Repeat this process for six times.

In addition, the period of oscillation T_2 is measured with Frisbee rotating vertically. the Frisbee is fixed vertically on the lower board with duct tapes to have the rotation axis passes the center of the lower board as shown in figure 22(b). The lower board is then twisted with a small angle and released. At the same time, a video is recorded to measure the time it spent oscillating ten times. The

Figure 21: Measuring the period of oscillation of the lower board. The lower board is twisted with a small angle, and set free to oscillate.

time spent for the board and the Frisbee to oscillate ten times is t. Repeat this process for six times.

(a)

(b)

Figure 22: (a): Set up of experiments to measure the period of the oscillation when the Frisbee rotates horizontally. The center of Frisbee is on the axis of rotation. (b): Set up of experiments to measure the period of the oscillation when the Frisbee rotates vertically. The Frisbee is fixed on the board vertically using duct tape. The center of Frisbee is on the axis of rotation.

5.1.4 Data Analysis

By previous measurement, parameters of the three-wire pendulum and Frisbee are obtained and organized in table 1. The moment of inertia of two different cases are calculated by putting the result into the formula (5.1). Calculate the average value of the moment of inertia obtained in each trial for both vertical and horizontal rotation. The result for each trial is organized in table 2, 3 and 4

R/m	r/m	H/m	m_0/kg	m/kg
0.150	0.075	0.400	0.089	0.175

Table 1: Parameters of three-wire pendulum and Frisbee

The final result is then obtained, the moment of inertia along z and y direction for the Frisbee paradigm is $0.0023kg/m^2$ and $0.0012kg/m^2$ respectively.

Trials	t/s	T_0/s	Average oscillation period T_{0a}/s
1	13	1.3	
2	13	1.3	
3	14	1.4	1.32
4	14	1.4	
5	13	1.3	
6	12	1.2	

Table 2: Period of oscillation of the lower board

Trials	t/s	T_1/s	$I_z/kg \cdot m^2$	Average $I_{zavg}/kg \cdot m^2$
1	14	1.4	0.0025	
2	13	1.3	0.0020	
3	14	1.4	0.0025	0.0023
4	14	1.4	0.0025	
5	13	1.3	0.0020	
6	13	1.3	0.0020	

Table 3: The moment of inertia of Frisbee-vertical rotation

Trials	t/s	T_2/s	$I_x(I_y)/kg \cdot m^2$	Average I_y $I_{xavg}(I_{yavg})/kg \cdot m^2$
1	12	1.2	0.0016	
2	11	1.1	0.0011	
3	10	1.0	0.00075	0.0012
4	12	1.2	0.0016	
5	11	1.1	0.0011	
6	11	1.1	0.0011	

Table 4: The moment of inertia of Frisbee-horizontal rotation

5.1.5 Error Analysis

In this section, the experimental percent Error for the experiment in the last section is evaluated, since the pendulum is build hand-made, and the accuracy cannot be directly ensured. Because Frisbee is in an irregular shape, so it is not possible to determine the value of rotational inertia completely by theoretical calculation. However, an error percentage bound could be set by determine two values for the rotational inertia of Frisbee, one known to be overestimated, and one known to be underestimated.

In order to obtain the two values, the Frisbee is separated into a loop and a disk, shown in figure 23. One value is obtained by treating the loop and the disk assuming they have thickness, and another value is obtained by assuming they are composed of thin plate. Because Frisbee's loop part is not a regular loop, instead have certain curvature that makes the real loop lighter then the theoretical thick loop, so the first treatment will output an overestimated value; and because in reality, Frisbee's loop part is thicker and heavier then the disk part, so the second treatment will output an underestimated value. By comparing the percent error obtained using both of the values, an error bound could be determined that gives a range of error margin.

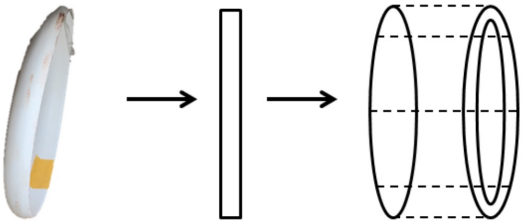


Figure 23: Theoretical model for Frisbee. Frisbee is considered composed of a disk and a loop.

The rotational inertia of the Frisbee along z direction is first calculated assuming that the loop and disk has thickness.

For the disk, the rotational Inertia along vertical axis is:

$$I_{zD} = \frac{1}{2} M_D R_1^2 \tag{5.2}$$

For the loop with thickness, we can calculate it as:

$$dI = r^2 dm \tag{5.3}$$

$$\begin{aligned} I_{zL} &= \int_{R_1}^{R_2} r^2 \rho w_L 2\pi r dr \\ &= \frac{1}{2} M_L (R_1^2 + R_2^2) \end{aligned} \tag{5.4}$$

In the equation, ρ is the density for the Frisbee.

$$M_D = \rho V_D = \rho \pi R_1^2 w_D \tag{5.5a}$$

$$M_L = \rho V_L = \rho \pi w_L (R_2^2 - R_1^2) \tag{5.5b}$$

$$\frac{M_D}{M_L} = \frac{R_1^2 w_D}{w_L (R_2^2 - R_1^2)} \tag{5.5c}$$

The factors were then measured:

$$\begin{aligned} M &= 0.175kg \\ R_1 &= 0.131m \\ R_2 &= 0.135m \\ w_D &= 0.001m \\ w_L &= 0.027m \end{aligned}$$

Substitute the factors into the equation, we get:

$$\frac{M_D}{M_L} = 0.597 \tag{5.6}$$

$$\begin{aligned} M_L &= 0.110kg \\ M_D &= 0.065kg \end{aligned}$$

Therefore, the rotational inertia is:

$$I_z = I_D + I_L = 0.0025kg \cdot m^2 \tag{5.7}$$

Calculate the percent error:

$$\%Error = \frac{|0.0023kg \cdot m^2 - 0.0025kg \cdot m^2|}{0.0025kg \cdot m^2} \times 100\% = 8.0\% \tag{5.8}$$

The rotational inertia through vertical axis is then calculated assuming the loop and disk is thin. The area of the loop and the plate is:

$$A_D = \pi R_2^2 \tag{5.9a}$$
$$A_L = 2\pi R_2 w_L \tag{5.9b}$$

$$\frac{M_D}{M_L} = \frac{A_D}{A_L} = \frac{R_2}{2w} = 2.5 \tag{5.10}$$
$$M_D = 0.125kg$$
$$M_L = 0.050kg$$

$$I_z = I_{zD} + I_{zL} = \frac{1}{2} M_D R_2^2 + M_L R_2^2 = 0.0021kg \cdot m^2 \tag{5.11}$$

$$\%Error = \frac{|0.0023kg \cdot m^2 - 0.0021kg \cdot m^2|}{0.0021kg \cdot m^2} \times 100\% = 9.5\% \tag{5.12}$$

Both the upper error percent and the lower error percent is below 10%, therefore the experiment determined value for rotational inertia about z axis can be considered as accurate.

Rotational inertia around horizontal axis would be calculated using the similar model. Consider a thin loop without thickness, the rotational inertia through central diameter is [6]:

$$dI = \frac{1}{2} r^2 dm + \frac{1}{12} w_L^2 dm \tag{5.13}$$

So for the thick loop in the model:

$$\begin{aligned} I_{xL} &= \int_{R_1}^{R_2} \frac{1}{2}(r^2 + \frac{w_L^2}{6}) \rho w_L 2\pi r dr \\ &= M_L(\frac{(R_1^2 + R_2^2)}{4} + \frac{w_L^2}{12}) \end{aligned} \tag{5.14}$$

And for the thick disk, the rotational inertia around diameter is:

$$\begin{aligned} I_{xD} &= \int_{0}^{R_1} \frac{1}{2}(r^2 + \frac{w_D^2}{6}) \rho w_D 2\pi r dr \\ &= M_D(\frac{R_1^2}{4} + \frac{w^2}{12}) \end{aligned} \tag{5.15}$$

Consider shift in center of mass with the disk adding on top of the loop with defining the center of mass of the loop to be the origin:

$$z_{cm} = \frac{\sum M_i zi}{\sum M_i} = \frac{0.065kg \cdot 0.013m}{0.175kg} = 0.0048m \tag{5.16}$$

As given by parallel axis theorem,

$$I = I_{cm} + Md^2 \tag{5.17}$$

In the equation, I is the result rotational inertia, and d is the distance between center of mass and the original axis. Therefore, with the distances determined:

$$d_L = 0.0048m$$
$$d_D = 0.0082m$$

we have:

$$\begin{aligned} I_x &= I_{xD} + I_{xL} \\ &= M_L(\frac{(R_1^2 + R_2^2)}{4} + \frac{w_L^2}{12}) + M_L d_L^2 + M_D(\frac{R_1^2}{4} + \frac{w^2}{12}) + M_D d_D^2 \\ &= 0.0013kg \cdot m^2 \end{aligned} \tag{5.18}$$

And the percent error is:

$$\%Error = \frac{|0.0012kg \cdot m^2 - 0.0013kg \cdot m^2|}{0.0013kg \cdot m^2} \times 100\% = 8.3\% \tag{5.19}$$

The same method is then applied to calculate the rotational inertia with thin loop and disk.

$$z_{cm} = \frac{\sum M_i zi}{\sum M_i} = \frac{0.125kg \cdot 0.0135m}{0.175kg} = 0.0096m \tag{5.20}$$

$$d_L = 0.0096m$$
$$d_D = 0.0039m$$

$$I_{xD} = \frac{1}{4}MR_2^2 + Md_D^2 \tag{5.21a}$$

$$I_{xL} = \frac{1}{2}M_L R_2^2 + \frac{1}{12}M_L w^2 + Md_L^2 \tag{5.21b}$$

$$\begin{aligned} I_x &= I_{xD} + I_{xL} \\ &= \frac{1}{4}M_D R_2^2 + M_D d_D^2 + \frac{1}{2}M_L R_2^2 + \frac{1}{12}M_L w^2 + Md_L^2 \\ &= 0.00104kg \cdot m^2 \end{aligned} \tag{5.22}$$

$$\%Error = \frac{|0.0012kg \cdot m^2 - 0.00104kg \cdot m^2|}{0.00104kg \cdot m^2} \times 100\% = 15\% \tag{5.23}$$

Although the error percentage for the model with thin loop and disk is a little bit high, however, the error percentages bound is still in an acceptable range, which indicates that the experiment for the rotational inertia around x and y axis is accurate.

5.2 Measuring the spring constant and damping coefficient

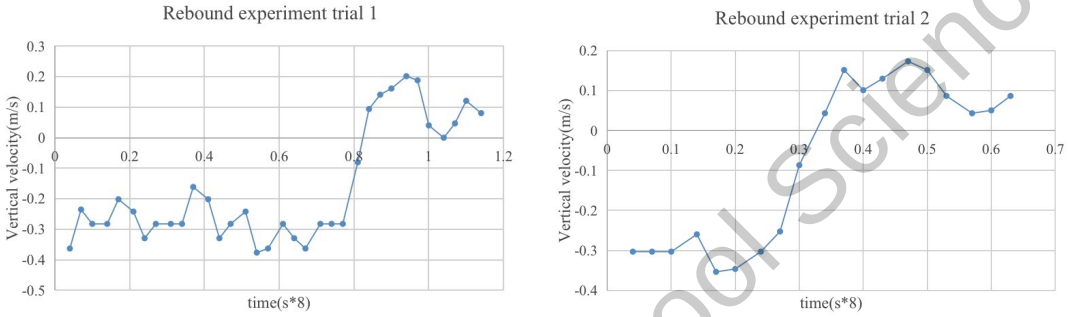
In this section, an experiment is conducted to measure the spring constant and damping coefficient using equation (3.12) and equation (3.8).

5.2.1 Experiment procedure

The Frisbee is held vertically above the rubber plastic pads on the ground. A camera (Apple 10, phone camera) is set aside the rubber plastic pads to shoot the process of falling of Frisbee using slow motion. The slow motion is set to be 240 fps, and 8 times slower than the normal speed. Open the camera and drop the Frisbee so that it hits the ground vertically and rebound with the same attitude. Convert the slow motion video into GIF and import to Tracker for analyzing.

5.2.2 Data analysis

The vertical velocity of Frisbee during rebound is analyzed by Tracker and imported in Excel to create graphs. The data is presented in figure 24



(a) (b)

Figure 24: Rebound experiment result. Both image shows that the velocity after rebound is about one third the velocity before rebound.

As could be seen from both graphs, while the magnitude of speed of falling is about $0.3m/s$ under 8 times slow motion, the speed after rebound is about $0.1m/s$ under 8 times slow motion. Therefore, the coefficient of restitution is determined to be approximately 0.33.

Additionally, the time duration Frisbee is in contact with the plastic pad is also determined using frame by frame analysis in Tracker. In the first videos recorded, image of Frisbee in contact with the ground is presented in 3 frames, counting $0.09s$ in total. In the second video recorded, image of Frisbee in contact with the ground is presented in 3 frames, counting $0.10s$ in total. Image of Frisbee in contact with the ground is shown in figure 25. Judging from the fact that the slow motion is 8 times slower then regular speed, the actual contact time is then approximately $0.011875s$, obtained by divide the average contact time in video by 8.

From all the parameters determined, the coefficient of restitution and the damping coefficient is calculated to be 13774 and 32.676 respectively.

6 Motion of Frisbee before and after touching down

6.1 overview

In this section, an experiment is conducted using equipment to record the motion of Frisbee in air and after touch down on a plastic pad. The Frisbee is launched with different linear velocity, angular velocity and initial attitude by a specifically made Frisbee launcher. Two cameras were used to shoot videos from

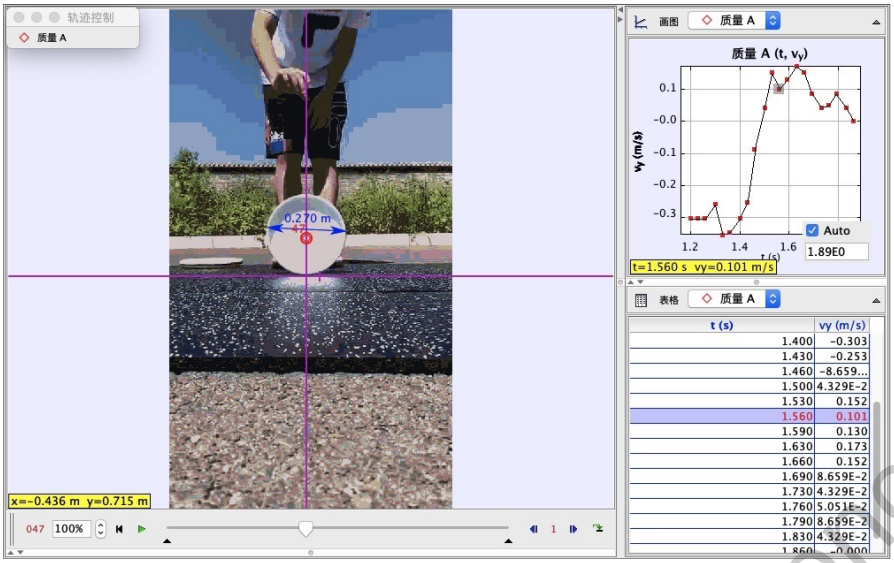


Figure 25: Image of Frisbee in contact with ground.

two different direction to capture the movement of Frisbee in three direction. The videos are then processed into GIF files and import into analyzing software Tracker to obtain the parameters during flight and after touch down. We can then compare simulation result corresponding the exact initial parameter with the experiment data to verify if the theoretical calculation is congruent with real life scenario.

6.2 Setup

A Frisbee launcher is built by imitate the current model launcher crafted[17] to control the initial parameters of the flight. This way, repeating experiments could be conducted to reduce error. A bicycle wheel is fixed while allowed to rotate through its central axis. A hand drill is used to accelerate the bicycle wheel to a high spinn rate. When Frisbee is placed in the arcuate chute aside the wheel, the friction between the Frisbee's edge and the rubber tyre would apply to make Frisbee accelerate in the chute while rotating. The wheel rotates in the direction that will result in Frisbee rotate the way as if it was throw using backhand, which means that the angular velocity points downward.

The spin rate of the wheel could be altered by adjusting the spin rate of the hand drill. A piece of plastic sheet is used to fix the speed adjusting button on the drill and hold the spinning speed still. The drill and wheel is compiled on a platform. The platform is allowed to rotated around one side so that the Frisbee could be launched at different initial pitch angle. The electricity Frisbee launcher requires were supplied by the plug base on a household electric vehicle. The voltage provided was large enough for operating the launcher. Pictures of Frisbee launcher from different views are presented in figure 26.

The white Frisbee used is nearly new with very few abrasions as shown in Figure 27. Five identical Frisbees were used, and launched repeatedly in this experiment.

To record the motion of the Frisbee, two cameras (apple 10 camera, 30fps) were set. The first one recording Frisbee's motion along x and z direction was put on the side of the experiment set up, perpendicular to Frisbee's trajectory. The second one recording Frisbee's motion along y direction was put in front of the experiment set up, parallel to Frisbee's trajectory. The second camera is set as

(a) (b)

Figure 26: (a) Top view and (b) side view of Frisbee launcher. The rotating wheel could be seen in (a), as well as the arcuate chute. (b) shows the way launcher is pitched in the experiment.

Figure 27: Frisbee used

near the ground and far from the setup as possible to help keep the video footage as realistic as possible, so that the reference length in the video will not be overly stretched or compressed.

In order to amplify the phenomenon of rebound, and to create an experiment condition closer to the assumptions of the theoretical model, same rubber plastic pads mentioned in the previous section were used for Frisbee to land on it and rebound after touching down. The theoretical model has excluded the influence of friction, and assumed that Frisbee lands on a elastic surface. Therefore, a rigid surface as the asphalt pavement do not fit the assumptions. The great friction from the ground (as asphalt pavement is much more rough then rubber plastic pad with smooth surface) would interfere with Frisbee's motion, and the surface is also not as elastic as the rubber plastic pad.

Figure 28 is the picture took at the experiment site to illustrate the set up. The camera used to shoot videos from the side is marked with red dot; the camera used to shoot videos at the front is marked with yellow dot; the launcher is marked with blue dot, while the rubber plastic pad is marked with green dot.

Figure 28: Experiment set up. The camera shooting videos from the side is marked with red dot. The camera shooting videos in front of the launcher is marked with yellow dot. The launcher is marked with blue dot, and the rubber plastic pad is marked with green dot.

6.3 Experiment procedure

2 different launching angle and 4 different launching velocity are available using the Frisbee launcher, which gives 8 different set of initial parameters that will be tested in the experiment. Before starting every formal experiment under different initial parameters, several sets of pre experiments are conducted by using the Frisbee launcher to launch the Frisbee with the selected parameter. The pitching angle of the launching plane is set using an electronic level, which can give the exact value of degree. Using the level, the pitch of the plane could be set at the required experiment condition. The electronic level at work is presented in figure 29. The purpose of pre experiments is to determine the approximate landing point, so that the rubber plastic pads could be placed around the landing area for the Frisbee to land on.

Figure 29: Electronic level used to measure the pitch angle established by the launcher

After having the pads placed properly, set the two cameras at their location. Turn on the Frisbee Launcher and start recording videos, put the five identical Frisbee in the chute one by one and launch the Frisbee carefully to avoid any tilt. There are cases which Frisbee did not landed on the pad, and these cases are considered as abnormalities and excluded from the trial. Keep repeating the experiment until there are in total 3 trials each initial parameter that the Frisbee lands on the pad. Stop the camera and the launcher after the experiment for each parameter set is done.

6.4 Video analysis

The videos recorded for the experiment are imported in tracker for analysis. Convert the videos from MOV file into GIF file while hold the same numbers of frames per second using software format factory. The GIF files are then opened in Tracker. Manually track Frisbee's trajectory in the series of images by import a point in tracker and click to track the center of Frisbee in each frame. After tracked the points, set the measuring scale based on the edge of the rubber plastic pad in the video, which has a side length of $1.5m$. Then set the coordinate, locate the origin at the bottom of the Frisbee launcher, and align the horizontal axis with the horizontal surface in the video. One of the processed file is presented in figure 30 below. In total, about 3000 points were marked by hand.

Figure 30: Tracker processed video. The red marks in the video is the location of Frisbee in each frame. The blue line on the right is the scale. The length of the scale in the video represents 1.5m, which is the side length of the rubber plastic pad. The purple cross is the reference frame. The relatively horizontal one is the x axis, and the relatively perpendicular one is the z axis. x axis points to the right, and z axis points towards the upside.

Eight sets of experiments are conducted. In each set of experiment, 6 videos are recorded for the three trials. From the 6 videos, the position of Frisbee in three direction as a function of time could be obtained, as well as the velocity in three direction. The data is copied from Tracker and imported into Excel to produce graphs. Even though the hand drill used on the Frisbee launcher has adjustable spinning speed, the speed is not shown. When assembled with the bicycle wheel, the spinning speed is even harder to determine. Therefore, the initial speed is obtained based on Tracker data. The spinning velocity of Frisbee could be determined by the relationship between linear velocity and angular velocity under condition of rolling without slipping. The equation for calculation is the same as equation (3.27).

The position of Frisbee in y direction is of a more complicated. It is very hard for the camera to be placed right in front of the launcher. In more common cases, the camera is somewhat slightly off the course, and cause an angle between its vision and Frisbee's initial trajectory. The case is illustrated in figure 31. This situation is in essence the same as if Frisbee is launched with an initial velocity along y direction. Therefore, when analyzing the position along y direction, the velocity on y direction is also obtained to supply the initial parameters.

The velocity and the angular velocity are then put in MATLAB source code to obtain the position of Frisbee predicted by the theoretical calculation as a function of time. The accuracy for the simulation is set to be $1.0e-5$. The result

Figure 31: Small deviation in the location of the camera. The solid rectangle on the left is the launcher. The solid rectangle on the right is where the camera should be if it's right in front of the set up. The dashed rectangle represents the actual location of the camera. The long arrow represents the direction of the initial velocity of the Frisbee v_0. The short arrow represents the equivalent y component of the velocity observe from the actual position.

from numerical simulation are then inserted in Excel to compare with measured data. Graphs are presented in the following sections, organized by initial parameter. Note that the negativity of y obtained in Tracker is opposite to previous definition because of Tracker's default setting. Therefore, the y position is invert in negativity when using MATLAB to create graphs.

6.4.1 Speed 1 (8m/s), 7 degree pitch, 52cm above ground

Speed 1 is determined by tracker analysis to be 8m/s. The x and z position as a function of time is presented in figure 32.

The initial velocity along y direction is determined to be about 0.05m/s. The y position as a function of time is presented in figure 33.

6.4.2 Speed 1 (8m/s), 14 degree pitch, 61cm above ground

The x and z position as a function of time is presented in figure 34.

The initial velocity along y direction is determined to be about 0.1m/s. The y position as a function of time is presented in figure 35.

6.4.3 Speed 2 (8.5m/s), 7 degree pitch, 52cm above ground

Speed 2 is determined to be 8.5m/s. The x and z position as a function of time is presented in figure 36.

The initial velocity along y direction is determined to be about -0.15m/s. The y position as a function of time is presented in figure 37.

6.4.4 Speed 2 (8.5m/s), 14 degree pitch, 61cm above ground

The x and z position as a function of time is presented in figure 38.

The initial velocity along y direction is determined to be about -0.1m/s. The y position as a function of time is presented in figure 39.

6.4.5 Speed 3 (9m/s), 7 degree pitch, 52cm above ground

Speed 3 is determined to be 9m/s. The x and z position as a function of time is presented in figure 40.

The initial velocity along y direction is determined to be about -0.3m/s. The y position as a function of time is presented in figure 41.

6.4.6 Speed 3 (9m/s), 14 degree pitch, 61cm above ground

The x and z position as a function of time is presented in figure 42.

The initial velocity along y direction is determined to be about -0.2m/s. The y position as a function of time is presented in figure 43.

6.4.7 Speed 4 (9.5m/s), 7 degree pitch, 52cm above ground

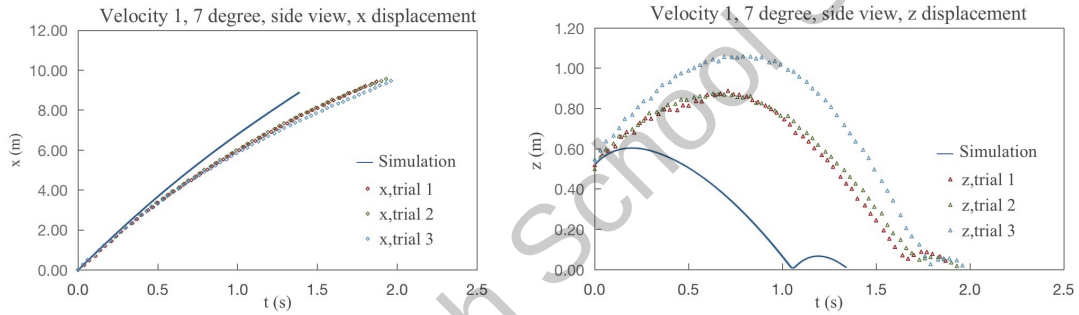
Speed 4 is determined to be 9.5m/s. The x and z position as a function of time is presented in figure 44.

The initial velocity along y direction is determined to be -0.1m/s. The y position as a function of time is presented in figure 45.

6.4.8 Speed 4 (9.5m/s), 14 degree pitch, 61cm above ground

The x and z position as a function of time is presented in figure 46.

The initial velocity along y direction is determined to be about 0m/s. Additionally, Frisbee launched in this group is observed to have a 1 degree row angle. The y position as a function of time is presented in figure 47.



(a) (b)

Figure 32: (a) x displacement and (b) z displacement. The initial condition for this set of experiment and simulation has speed 8 m/s, pitch angle of 7°, and height of 52cm above ground. The solid line is obtained from simulation and the markers are three trials obtained from experiments.

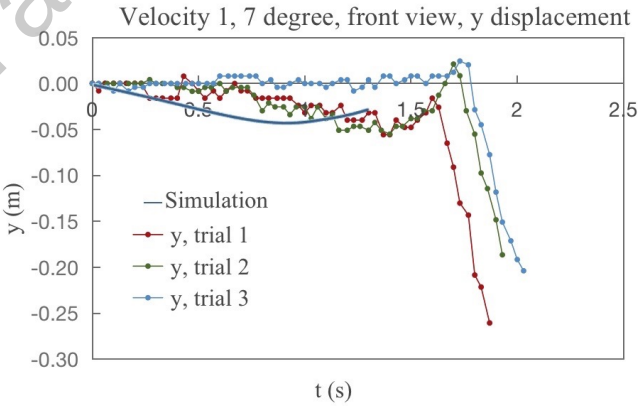
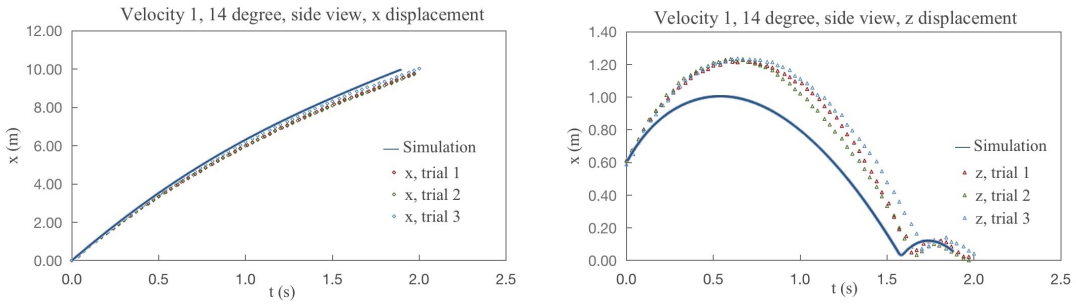


Figure 33: y displacement, y component of speed 0.05m/s



(a) (b)

Figure 34: (a) x displacement and (b) z displacement. The initial condition for this set of experiment and simulation has speed 8 m/s, pitch angle of 14°, and height of 61cm above ground. The solid line is obtained from simulation and the markers are three trials obtained from experiments.

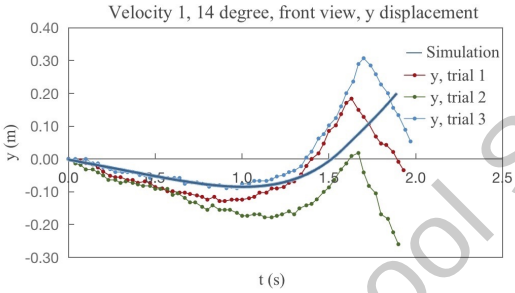
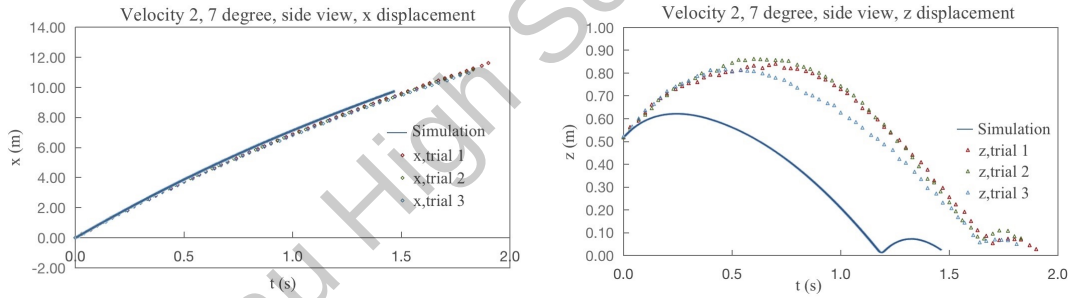


Figure 35: y displacement, y component of speed 0.1m/s



(a) (b)

Figure 36: (a) x displacement and (b) z displacement. The initial condition for this set of experiment and simulation has speed 8.5 m/s, pitch angle of 7°, and height of 52cm above ground. The solid line is obtained from simulation and the markers are three trials obtained from experiments.

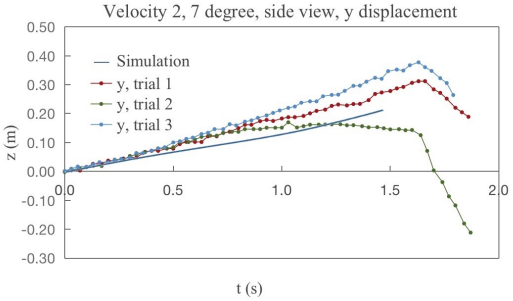
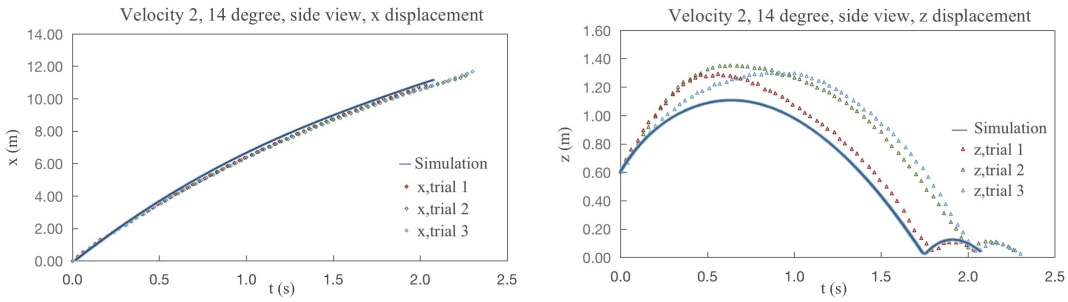


Figure 37: y displacement, y component of speed -0.15m/s



(a) (b)

Figure 38: (a) x displacement and (b) z displacement. The initial condition for this set of experiment and simulation has speed 8.5 m/s, pitch angle of 14°, and height of 61cm above ground. The solid line is obtained from simulation and the markers are three trials obtained from experiments.

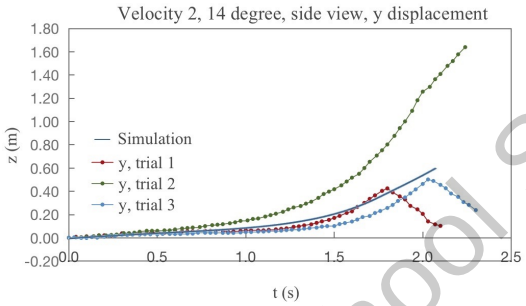
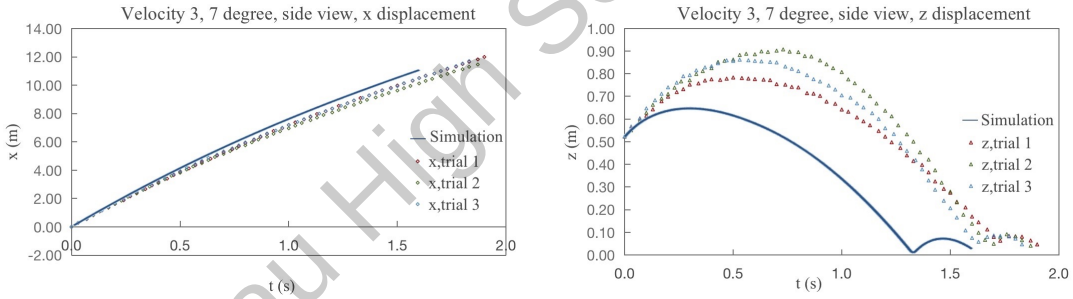


Figure 39: y displacement, y component of speed -0.1m/s



(a) (b)

Figure 40: (a) x displacement and (b) z displacement. The initial condition for this set of experiment and simulation has speed 9 m/s, pitch angle of 7°, and height of 52cm above ground. The solid line is obtained from simulation and the markers are three trials obtained from experiments.

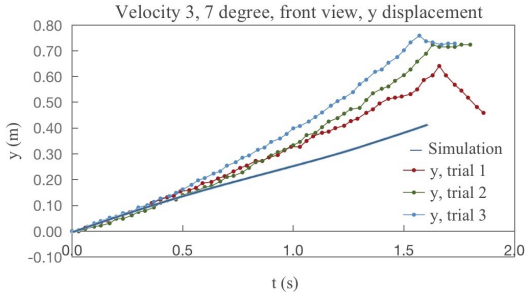
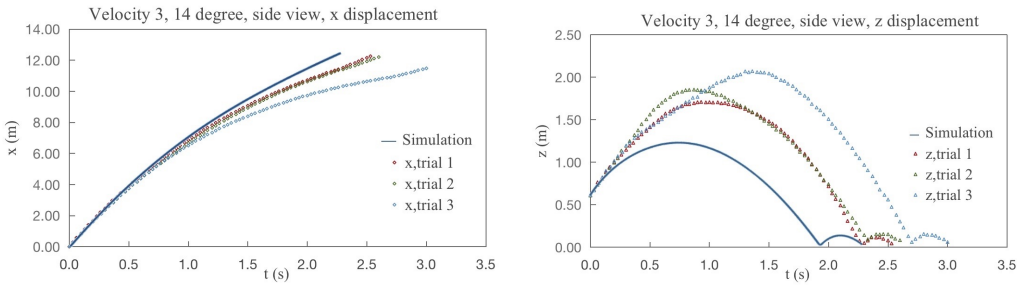


Figure 41: y displacement, y component of speed -0.3m/s



(a) (b)

Figure 42: (a) x displacement and (b) z displacement. The initial condition for this set of experiment and simulation has speed 9 m/s, pitch angle of 14°, and height of 61cm above ground. The solid line is obtained from simulation and the markers are three trials obtained from experiments.

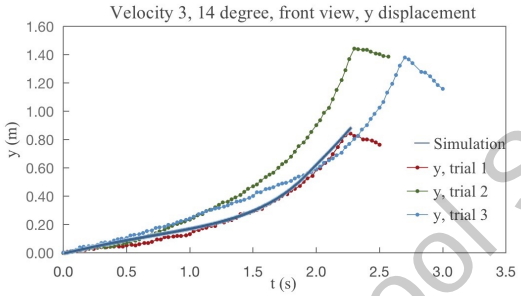
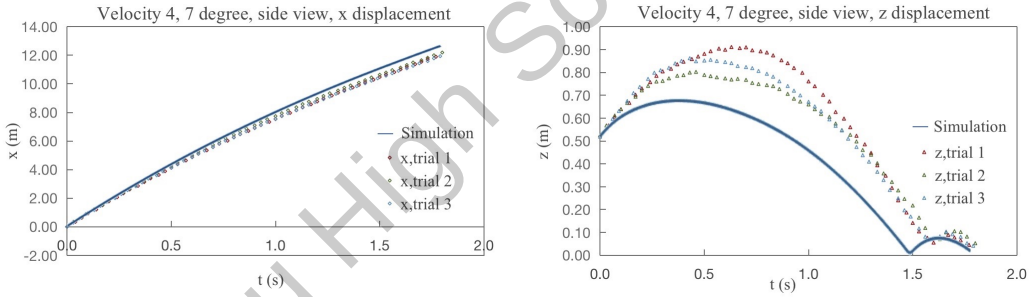


Figure 43: y displacement, y component of speed -0.2m/s



(a) (b)

Figure 44: (a) x displacement and (b) z displacement. The initial condition for this set of experiment and simulation has speed 9.5 m/s, pitch angle of 7°, and height of 52cm above ground. The solid line is obtained from simulation and the markers are three trials obtained from experiments.

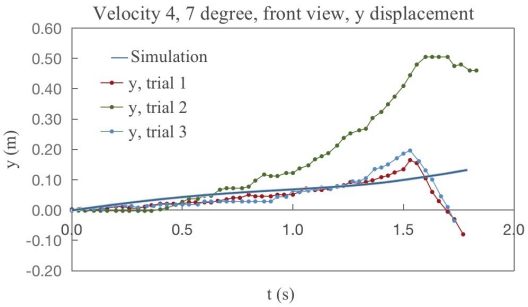
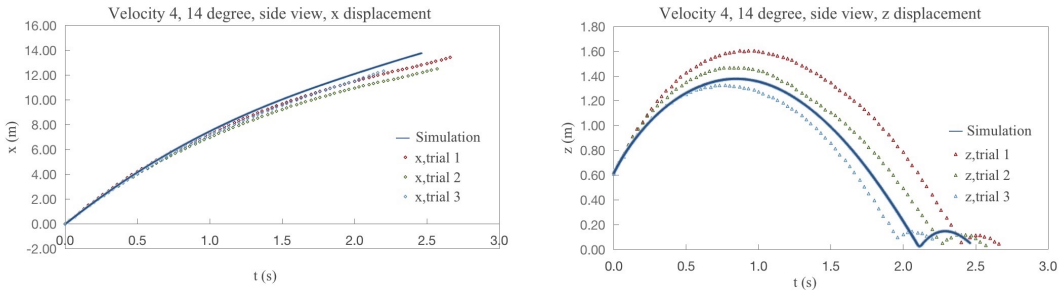


Figure 45: y displacement, y component of speed -0.1m/s



(a) (b)

Figure 46: (a) x displacement and (b) z displacement. The initial condition for this set of experiment and simulation has speed 9.5 m/s, pitch angle of 14°, and height of 61cm above ground. The solid line is obtained from simulation and the markers are three trials obtained from experiments.

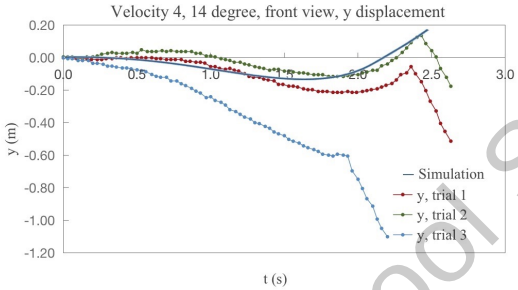


Figure 47: y displacement, y component of speed $0m/s$ (Frisbees launched in this set of experiment has an initial row angle of about 1 degree)

6.5 Data Analysis and Error analysis

6.5.1 general analysis

Based on the figures 32-47 in the previous part, the simulation result is consistent with the experimental measurements of actual flight and rebound trajectory. In particular, the simulated x and y position of Frisbee match the data obtained in experiments well. The measured y position changes drastically during the rebound. However, this pattern is not seeing on the theoretical model. This is because that the theoretical model used in this research has been simplified with friction force ignored. Although the friction of rubber plastic pad is significantly smaller then asphalt pavement, and the impact time is very short, because the instantaneous impulse is quite large, the normal force exerted on Frisbee is significant during rebound, and so is the friction. The complete version of the theoretical model suggested in 3, which takes friction into account, should be able to recreate this special pattern.

There is, however, visible discrepancy between the simulated z position of Frisbee and the measured z position of Frisbee, especially seen clearly in figure 32(b), figure 36(b), figure 40(b) and figure 42(b). This visible discrepancy is resulted under the influence of the wind. As recorded by an anemometer, The wind speed during the experiment is about between 0-2m/s, blowing against the direction of motion of Frisbee. The wind speed is not that high, however, because Frisbee's motion relies largely on the aerodynamic force, and that Frisbee does not fly at a high speed, going at a ground speed of 10m/s plus wind speed of 2m/s will give an air speed of 12m/s. Therefore, the wind, no matter how small, always has a

significant effect on altering Frisbee's motion in air. This is also supported by the experiment data. With wind blowing against the motion of Frisbee, the drag will be larger. This will result in Frisbee decelerating faster than as suggested in the theoretical model. The larger deceleration could be seen in almost all the $xvs.t$ graph. Additionally, with a higher air speed, the lift force will also be larger, and could support the Frisbee to fly higher, consistent with the pattern presented in the z-t graph.

To better avoid the influence of wind, Frisbee experiment will need to be conducted indoor. Yet, despite of all the influence Frisbee experienced during experiment (wind, friction and such), the simulated result matches the experiment data. The simulated vertical motion of Frisbee during and after rebound is highly similar to the result obtained by experiment, indicating that the "damping spring" model of rebound successfully describes Frisbee's rebound phenomenon.

6.5.2 Influence of initial parameters to the rebound process

The initial velocity before touch down is related to the rebound phenomenon in a complex way. For a higher initial velocity, the vertical velocity will also be higher when Frisbee is thrown toward the ground. With a higher vertical velocity before touch down, the vertical velocity after rebound will also be higher. Additionally, with a higher initial velocity before touch down, the resulted velocity after rebound will also be higher. Thus, Frisbee will experience a higher lift after rebound, and can be able to glide for a longer distance.

For different initial pitch angle, initial velocity have different relationship with the rebound phenomenon. If the Frisbee touches down with a head down attitude (pitch angle is negative), then the rebound will give Frisbee a moment to turn it into a head up position (pitch angle is positive). In this scenario, the pitch angle after rebound will be larger, result in a larger lift force. If the Frisbee touches down with a head up attitude (pitch angle is positive), then the rebound will give Frisbee a moment to turn it into a head down attitude (pitch angle is negative). With a head down attitude, combining an upward velocity, Frisbee will have a negative attack angle, result in a small, or even negative lift force.

If the initial angular velocity is small, then the change in attitude will be drastic, and the situation illustrated before is valid. However, Under different initial angular velocity, the situation changes. If the angular velocity is large enough, then the change in attitude as a result of the rebound will be relatively small. Under this circumstance, touch down with a head up attitude will result in a rebound with head up attitude, provide larger lift force for the Frisbee. Touch down with a head down attitude will result in a rebound with head down attitude, provide smaller lift force for the Frisbee. The difference between a head up rebound and a head down rebound is illustrated in 48.

Typically, Frisbee is thrown with a small angular velocity that is not large enough to maintain the attitude during rebound. The observed phenomenon in class is also congruent with the analysis before. A typical success rebound involve Frisbee touch down with a negative pitch angle and a larger velocity. The two simulated result in section 4 also support the analysis. Under a negative pitch angle, Frisbee rebound further and higher with a large initial velocity.

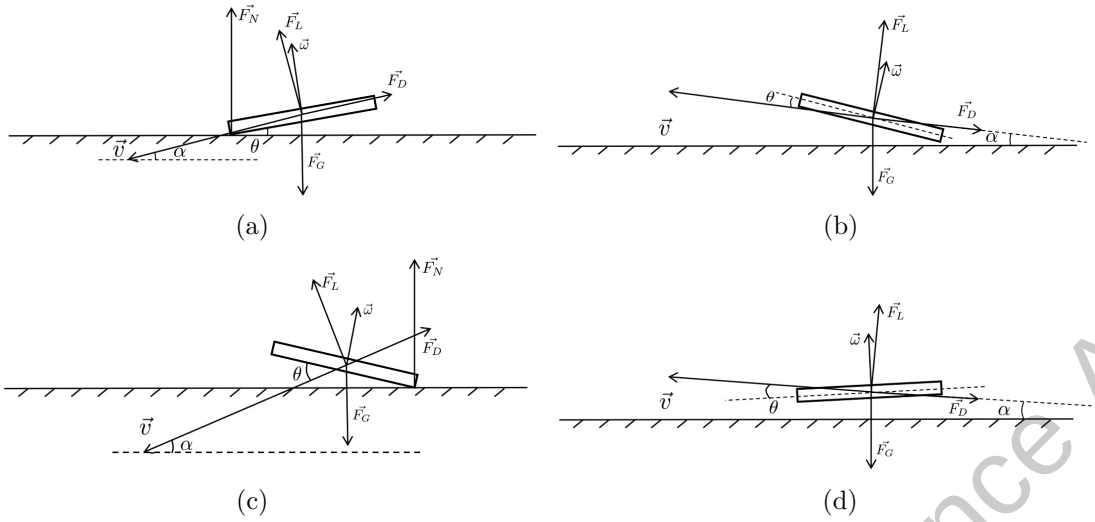


Figure 48: (a): head down attitude before touch down. (b): head down attitude after touch down. (c): head up attitude before touch down. (d): head up attitude after touch down

7 Conclusions and future work

7.1 Conclusion

This paper is the first one to investigate the touchdown dynamics of the Frisbee to the best of the author's knowledge. Dynamic differential equations of Frisbee flight have been established based on previous works with improvements, where moments and forces are unified into the same reference frame through vector transformation. Velocity of Frisbee is computed in the inertial Frame to provide better clearance and to support the deduction of rebound model, which is developed based on rigid body dynamics. The ground is modeled as a damping spring. Governing equations are then developed to calculate the damping coefficient and spring constant of the ground using experiment data. Dynamic differential equations of rebound are developed taking the friction and normal force into account, while the final model presented here is simplified by neglecting the friction for brevity.

MATLAB simulation for the equations governing flight and the simplified equations governing rebound is developed using ODE23s solver. The aerodynamic coefficients are based on database from existing experiments. Comparison between simulation result of velocity in inertial frame and in embedded frame illustrates the advantage of calculating velocity in inertial frame. Simulations for the typical Frisbee flight trajectories are done with the same initial parameters as these in previous work. Simulation results for two different rebound processes are also provided.

Experiments are conducted to validate the numerical simulation. The moment of inertia through horizontal axis and vertical axis for a specific Frisbee with known basic parameter is determined using the three wire pendulum method. Damping coefficient and spring constant of a rubber plastic pad is determined by video analysis following equations deducted. Identical Frisbees are then launched by a launcher for three times and the motion is recorded by a camera. Totally eight different initial parameter sets are tested in the experiment. A frame by frame analysis of recorded flight videos is done to get the position and velocity of Frisbee as a function of time. The comparison between experimental measurements

and numerical simulation results of respective initial parameters demonstrate that the model established captures the influence of aerodynamic forces, aerodynamic torques and self spinning of Frisbee on its flight and rebound pattern.

Further analysis provides the conditions needed for Frisbee to amplify the rebound phenomenon. The combination of a large velocity, negative pitch angle and a small spin rate would result in Frisbee having a significant rebound and flying higher and further after bouncing. The result and the theoretical model of this study could be applied on the landing of spin-stabilized unmanned aircraft.

7.2 Future works

This work opens up many promising future work direction, where primary direction is to simulate the rebound of Frisbee base on the complete rebound theory by adding in the friction. Because of the complexity of the equation, this will require better computational capacity. Simulation result of the complete rebound theory should give a more precise rebound pattern by taking friction into account. Additionally, the current aerodynamic force and moment model is simplified. Frisbee also experienced a force directed at its side due to the Magnus effect caused by the rotation, as well as aerodynamic torques that tilts the Frisbee and decreases Frisbee's rotational speed. Taking these parameters into account will result in a more realistic flight model.

Another follow up work that could be carry on is to repeat the experiments indoor to avoid the influence of wind. This should provide a better look at the original trajectory of Frisbee flight and rebound. The influence of wind could also be added into the numerical simulation as a modification term. This could give a more accurate result of Frisbee's motion under windy condition, and could be used in the development of a similar UAV.

Last but not the least, the launching strategy of Frisbee could be studied by run the MATLAB program in cycle. As an example, combination of parameters lead to a smooth landing could be studied. As observed in pre experiments, when Frisbee touchdown with perfect or near perfect horizontal attitude, the concaved bottom side of Frisbee would create an air cushion that buffers the impact. With the air cushion lifting the Frisbee, it normally would land smoothly without having any rebound. For the many variables involved in the theoretical model, at least, 100*100 cycles will need to be tested to obtain some results. This could be a promising area of study.

References

[1] Bloomfield L A. The flight of the frisbee. *Scientific American*, 280(4):132–132, 1999.

[2] Wie B. *Space vehicle dynamics and control*. Aiaa, 1998.

[3] Stilley G D Carstens D L. Adaptation of the frisbee flight principle to delivery of special ordnance. In *2nd Atmospheric Flight Mechanics Conference*, page 982, 1972.

[4] Potts J R Crowther W J. Frisbee (TM) aerodynamics. In *20th AIAA applied aerodynamics conference*, page 3150, 2002.

[5] Lorenz R D. *Spinning Flight: Dynamics of Frisbees, Boomerangs, Samaras, and Skipping Stones*. Springer, 2006.

[6] Miller I A Giancoli D C. *Physics: principles with applications*. Prentice Hall Upper Saddle River, NJ, USA:, 1998.

[7] Nagahiro S Hayakawa Y. Theoretical and numerical approach to "magic angle" of stone skipping. *Physical review letters*, 94(17):174501, 2005.

[8] Clanet C Hersen F, Bocquet L. Secrets of successful stone-skipping. *Nature*, 427(6969):29–29, 2004.

[9] Hubbard M Hummel S. Identification of Frisbee aerodynamic coefficients using flight data. volume 20, 2002.

[10] Hubbard M Hummel S A. Simulation of Frisbee flight. *5th Conference on Mathematics and Computers in Sport, University of Technology, Sydney*, pages 124–134, 2000.

[11] Baumback K. The aerodynamics of frisbee flight. *Undergraduate Journal of Mathematical Modeling*, 2010.

[12] Yasuda K. Flight and aerodynamic characteristics of a flying disk. *Japanese Soc. Aero. Space Sci*, 47(547):16–22, 1999.

[13] Kamaruddin N M. *Dynamics and performance of flying discs*. The University of Manchester (United Kingdom), 2011.

[14] Landell-Mills N. Newtons laws explain how Frisbees fly. *European Journal of Applied Physics*, 2(4), 2020.

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

[16] Morrison V R. The physics of Frisbees. *Electronic Journal of Classical Mechanics and Relativity*, 8(48), 2005.

[17] Sharma S Siddiqui Y, et. al. Design and manufacturing of Frisbee launching robot. *International Journal of Current Engineering and Technology*, 2017.

[18] Weizman Y Tan A M. Measurement of flight dynamics of a frisbee using a triaxial mems gyroscope. 49(1):66, 2020.

[19] Mosca G Tipler P A. *Physics for scientists and engineers*. Macmillan, 2007.

[20] Beard R W. Quadrotor dynamics and control. *Brigham Young University*, 19(3):46–56, 2008.

[21] Xia X. "comparison measurement" in the application of three-line pendulum experiment. *Physical Experiment of College*, page 05, 2011.

Acknowledgement

First of all, I would like to thank my two mentors, Fan Bozhao and Han Jiangfan, for the detailed and thorough instructions they provided for the writing and the revision of the thesis. Secondly, I would like to thank my physics teacher, Ma Qinghua. With his advices and suggestions, I was able to sort out the research problem. This project is inspired by the phenomenon observed in the Frisbee class. I would like to thank my friend, Wang Zhaojing. His experienced Frisbee throwing techniques inspired me during the early stage of the project. Last, but not least, I would like to thank my parents. They have provided unconditional support to me during this project. A special thanks to my physics teacher Dr.Kevin Coleman, who taught me vector transformation and frame conversion in the AP Physics C class.

A MATLAB source code for deriving governing equations during flight

```
syms phi psi theta bq alpha v1 v2 v3 p q r Ix Iy Iz rho;`1
syms cDq cLq;
syms cMq d S m g;

T_a12=[1, 0,0;
       0, cos(phi),sin(phi);
       0, -sin(phi),cos(phi)]*...
       [cos(theta), 0, -sin(theta);
           0,1,0;
       sin(theta), 0, cos(theta)]*...
       [cos(psi), sin(psi), 0;
       -sin(psi), cos(psi), 0;
       0,0,1];
T_a21=[cos(psi), -sin(psi), 0;
       sin(psi), cos(psi), 0;
       0,0,1]*...
       [cos(theta), 0, sin(theta);
           0,1,0;
       -sin(theta), 0, cos(theta)]*...
       [1, 0,0;
       0, cos(phi),-sin(phi);
       0, sin(phi),cos(phi)];
x_1_dot=[v1; v2; v3];
x_2_dot=T_a12*x_1_dot;
u=x_2_dot(1);
v=x_2_dot(2);
w=x_2_dot(3);
T_a23=[u/sqrt(u^2+v^2),v/sqrt(u^2+v^2),0;
       -v/sqrt(u^2+v^2),u/sqrt(u^2+v^2),0;
       0,0,1];

x_3_dot=T_a23*x_2_dot;
T_a34=[x_3_dot(1)/sqrt(x_3_dot(1)^2+x_3_dot(3)^2),0,...
       x_3_dot(3)/sqrt(x_3_dot(1)^2+x_3_dot(3)^2);
       0,1,0;
       -x_3_dot(3)/sqrt(x_3_dot(1)^2+x_3_dot(3)^2),0,...
       x_3_dot(1)/sqrt(x_3_dot(1)^2+x_3_dot(3)^2)];

I=diag([Ix, Iy, Iz]);

C_F_vec=[-cDq; 0; -cLq];
C_M_vec=[0; cMq; 0];

F_4=1/2*rho*(v1^2+v2^2+v3^2)*S*C_F_vec;
M_4=1/2*rho*(v1^2+v2^2+v3^2)*S*d*C_M_vec;

F_1_aero=T_a21*transpose(T_a23)*transpose(T_a34)*F_4;
M_2=transpose(T_a23)*transpose(T_a34)*M_4;

F_1=F_1_aero+[0;0;m*g];

x_1_ddot=F_1/m
theta_2_ddot=inv(I)*(M_2-cross(theta_2_dot,I*theta_2_dot))
```

B MATLAB source code for deriving governing equations during rebound

```
syms phi psi theta beta alpha v1 v2 v3 p q r Ix Iy Iz rho x y z miu;
syms cDq cLq;
syms cMq d S m g k b;

T_a12=[1, 0,0;
       0, cos(phi),sin(phi);
       0, -sin(phi),cos(phi)]*...
       [cos(theta), 0, -sin(theta);
           0,1,0;
       sin(theta), 0, cos(theta)]*...
       [cos(psi), sin(psi), 0;
       -sin(psi), cos(psi), 0;
       0,0,1];
T_a21=[cos(psi), -sin(psi), 0;
       sin(psi), cos(psi), 0;
       0,0,1]*...
       [cos(theta), 0, sin(theta);
           0,1,0;
       -sin(theta), 0, cos(theta)]*...
       [1, 0,0;
       0, cos(phi),-sin(phi);
       0, sin(phi),cos(phi)];
x_1_dot=[v1; v2; v3];
x_2_dot=T_a12*x_1_dot;
u=x_2_dot(1);
v=x_2_dot(2);
w=x_2_dot(3);
T_a23=[u/sqrt(u^2+v^2),v/sqrt(u^2+v^2),0;
       -v/sqrt(u^2+v^2),u/sqrt(u^2+v^2),0;
       0,0,1];
x_3_dot=T_a23*x_2_dot;
T_a34=[x_3_dot(1)/sqrt(x_3_dot(1)^2+x_3_dot(3)^2),0,x_3_dot(3)...
       /sqrt(x_3_dot(1)^2+x_3_dot(3)^2);
       0,1,0;
       -x_3_dot(3)/sqrt(x_3_dot(1)^2+x_3_dot(3)^2),0,x_3_dot(1)...
       /sqrt(x_3_dot(1)^2+x_3_dot(3)^2)];

T_r12=[1,0, -sin(theta);
    0, cos(phi), cos(theta)*sin(phi);
    0, -sin(phi), cos(theta)*cos(phi)];
theta_2_dot=[p;q;r];
E_dot=inv(T_r12)*theta_2_dot;
phi_dot=E_dot(1);
theta_dot=E_dot(2);
psi_dot=E_dot(3);

I=diag([Ix, Iy, Iz]);

C_F_vec=[-cDq; 0; -cLq];
C_M_vec=[0; cMq; 0];

F_4=1/2*rho*(v1^2+v2^2+v3^2)*S*C_F_vec;
M_4=1/2*rho*(v1^2+v2^2+v3^2)*S*d*C_M_vec;

F_1_aero=T_a21*transpose(T_a23)*transpose(T_a34)*F_4;
M_2_aero=transpose(T_a23)*transpose(T_a34)*M_4;

z2=[0;0;1];
zvec=T_a21*z2;
cosd=zvec(3);
rxy=cosd*d/2;
xyr=-[zvec(1);zvec(2)]*rxy/(zvec(1)^2+zvec(2)^2)^(1/2);
zr=(d/2)*(1-cosd^2)^(1/2);
r1=[xyr(1);xyr(2);zr];
rp=[x+xyr(1);y+xyr(2);z+zr];
xr_dot=(-d/2)*((phi_dot*(-sin(phi))*cos(theta)*zvec(1)+theta_dot*cos(phi)...
    *(-sin(theta))*zvec(1)+cosd*(phi_dot*cos(phi)*sin(psi)+psi_dot*sin(phi)...
```

```
    *cos(psi)+phi_dot*(-sin(phi))*cos(psi)*sin(theta)+psi_dot*cos(phi)...
    *(-sin(psi))*sin(theta)+theta_dot*cos(phi)*cos(psi)*cos(theta)))...
    *(1-cosd^2)^(1/2)-cosd*zvec(1)*(-(2*phi_dot*cos(phi)*(-sin(phi))...
    *(cos(theta))^2+2*theta_dot*cos(theta)*(-sin(theta))*(cos(phi))^2)...
    /(2*(1-cosd^2)^(1/2))))/(1-cosd^2);

yr_dot=(-d/2)*((phi_dot*(-sin(phi))*cos(theta)*zvec(2)+theta_dot*cos(phi)...
    *(-sin(theta))*zvec(2)+cosd*(phi_dot*(-sin(phi))*sin(psi)*sin(theta)...
    +psi_dot*cos(phi)*cos(psi)*sin(theta)+theta_dot*cos(phi)*sin(psi)...
    *cos(theta)-psi_dot*(-sin(psi))*sin(phi)-phi_dot*cos(psi)*cos(phi)))...
    *(1-cosd^2)^(1/2)-cosd*zvec(2)*(-(2*phi_dot*cos(phi)*(-sin(phi))...
    *(cos(theta))^2 +2*theta_dot*cos(theta)*(-sin(theta))*(cos(phi))^2)...
    /(2*(1-cosd^2)^(1/2))))/(1-cosd^2);

zr_dot=(d/2)*(-(2*phi_dot*cos(phi)*(-sin(phi))*(cos(theta))^2 ...
    +2*theta_dot*cos(theta)*(-sin(theta))*(cos(phi))^2)/(2*(1-cosd^2)^(1/2)));

vp=[v1+xr_dot;v2+yr_dot;v3+zr_dot];
F_1bz=-k*(z+zr)-b*vp(3);
%vp_rel=[vp(1);vp(2)]+[zvec(2);-zvec(1)]*r*(d/2)/(zvec(1)^2+zvec(2)^2)^(1/2);
%F_1bxy=miu*F_1bz*-vp_rel/norm(vp_rel);
%F_1b=[F_1bxy(1);F_1bxy(2);F_1bz];
F_1b=[0;0;F_1bz];
F_1=F_1_aero+F_1b+[0;0;m*g];

M_1b=cross(r1,F_1b);
M_2b=T_a12*M_1b;
M_2=M_2_aero+M_2b;

x_1_ddot=F_1/m
theta_2_ddot=inv(I)*(M_2-cross(theta_2_dot,I*theta_2_dot))
```

C MATLAB source code for dynamics simulation

```
clear all;
close all;
clc;

global g m d I rho S Ix Iy Iz b k;
g=9.8;
m=0.175;
d=0.27;
I = diag([0.0012 0.0012 0.0023]);
Ix=I(1,1);
Iy=I(2,2);
Iz=I(3,3);
rho=1.293;
S=0.05726;
b=32.676;
k=13774;
vel_Inertial=[cos(-0.3)*15;0;-sin(-0.3)*15];


%Omega=norm(vel_Inertial)/(d/2);
Omega=47;
q0=0;
p0=0;

Omega_Body=[p0;q0;Omega];
pos_Inertial=[0;0;-0.1];
E=[0;-0.1;0];

y0=[Omega_Body;vel_Inertial;pos_Inertial;E];
ode_option1 = odeset('RelTol', 1.0e-5, 'AbsTol', 1.0e-5,'Events', @touchdown);
ode_option2 = odeset('RelTol', 1.0e-5, 'AbsTol', 1.0e-5,'Events', @rebound);
dt=0.00001;
T=20;
tspan_1=0:dt:T;
[T_out_1,y_out_1]=ode23s(@f_dot, tspan_1,y0,ode_option1);
tspan_2=T_out_1(end):dt:T;
y_out_1_end=y_out_1(end,:);
y0_2=y_out_1_end;
[T_out_2,y_out_2]=ode23s(@f_dot_rebound, tspan_2,y0_2,ode_option2);
tspan_3=T_out_2(end):dt:T;
y_out_2_end=y_out_2(end,:);
y0_3=y_out_2_end;
[T_out_3,y_out_3]=ode23s(@f_dot, tspan_3,y0_3,ode_option1);

%%--------------------

T_out=[T_out_1;T_out_2;T_out_3];
y_out=[y_out_1;y_out_2;y_out_3];

omegaInertialPlot=y_out(:,1:3);
velPlot=y_out(:,4:6);
posPlot=y_out(:,7:9);
EPlot=y_out(:,10:12);

time=T_out;

figure(1)
plot(time,omegaInertialPlot(:,1),time,omegaInertialPlot(:,2)...
,time,omegaInertialPlot(:,3));
xlabel('Time (sec)')
ylabel('Angular velocity')
title('Angular velocity')
grid on
legend('p','q','r')
grid on

figure(2)
plot(time,EPlot(:,1),time,EPlot(:,2),time,EPlot(:,3));
xlabel('Time (sec)')
```

```
ylabel('Angle/rad')
legend('\phi','\theta','\psi')
title('Euler angle')
grid on

figure(3)
plot(time,velPlot(:,1),time,velPlot(:,2),time,-velPlot(:,3));
xlabel('Time (sec)')
ylabel('Velocity (m/sec)')
legend('v x','v y','v z')
title('Velocity')
grid on

figure(4)
plot(time,posPlot(:,1),time,posPlot(:,2),time,-posPlot(:,3));
xlabel('Time (sec)')
ylabel('position (m)')
legend('x','y','z')
title('Position')
grid on

function output=f_dot(time,input)

    Omega_2=input(1:3,:);
    vel_1=input(4:6,:);
    E=input(10:12,:);

    p=Omega_2(1);
    q=Omega_2(2);
    r=Omega_2(3);
    v1=vel_1(1);
    v2=vel_1(2);
    v3=vel_1(3);
    phi=E(1);
    theta=E(2);
    psi=E(3);

    global g m d rho S Ix Iy Iz;
    u=v1*cos(psi)*cos(theta) - v3*sin(theta) + v2*cos(theta)*sin(psi);
    v=v2*(cos(phi)*cos(psi) + sin(phi)*sin(psi)*sin(theta)) - ...
        v1*(cos(phi)*sin(psi) - cos(psi)*sin(phi)*sin(theta)) + ...
        v3*cos(theta)*sin(phi);
    w=v1*(sin(phi)*sin(psi) + cos(phi)*cos(psi)*sin(theta)) - ...
        v2*(cos(psi)*sin(phi) - cos(phi)*sin(psi)*sin(theta)) + ...
        v3*cos(phi)*cos(theta);

    dot_p=-(Iz*q*r - Iy*q*r + (S*intp3(atan(w/(u^2 + v^2)^(1/2)))*d*rho*v*(u^2 ...
        + v^2 + w^2))/(2*(u^2 + v^2)^(1/2)))/Ix;%checked
    dot_q=(Iz*p*r - Ix*p*r + (S*intp3(atan(w/(u^2 + v^2)^(1/2)))*d*rho*u*(u^2 + ...
        v^2 + w^2))/(2*(u^2 + v^2)^(1/2)))/Iy;%checked
    dot_r=0; %checked
    Omega_2_dot=[dot_p;dot_q;dot_r];

    v1_dot=-((S*intp2(atan(w/(u^2 + v^2)^(1/2)))*rho*v1*(v1^2 + v2^2 + ...
        v3^2)^(1/2))/2 + (S*intp1(atan(w/(u^2 + v^2)^(1/2)))*rho*(v1^2 + v2^2 + ...
        v3^2)^(1/2)*(v2^2*sin(phi)*sin(psi) ...
        + v3^2*sin(phi)*sin(psi) + v1*v2*cos(psi)*sin(phi) + ...
            v2^2*cos(phi)*cos(psi)*sin(theta) + ...
            v3^2*cos(phi)*cos(psi)*sin(theta) ...
        - v1*v3*cos(phi)*cos(theta) - ...
            v1*v2*cos(phi)*sin(psi)*sin(theta)))/(2*((v2*cos(phi)*cos(psi) - ...
            v1*cos(phi)*sin(psi) + v3*cos(theta)*sin(phi) ...
        + v1*cos(psi)*sin(phi)*sin(theta) + v2*sin(phi)*sin(psi)*sin(theta))^2 ...
            + (v1*cos(psi)*cos(theta) - v3*sin(theta) ...
        + v2*cos(theta)*sin(psi))^2)^(1/2)))/m;

    v2_dot=-((S*intp2(atan(w/(u^2 + v^2)^(1/2)))*rho*v2*(v1^2 + v2^2 + ...
        v3^2)^(1/2))/2 - (S*intp1(atan(w/(u^2 + v^2)^(1/2)))*rho*(v1^2 + v2^2 + ...
        v3^2)^(1/2)*(v1^2*cos(psi)*sin(phi) ...
        + v3^2*cos(psi)*sin(phi) + v1*v2*sin(phi)*sin(psi) - ...
            v1^2*cos(phi)*sin(psi)*sin(theta) - ...
            v3^2*cos(phi)*sin(psi)*sin(theta) ...
```

```
125         + v2*v3*cos(phi)*cos(theta) + ...
                v1*v2*cos(phi)*cos(psi)*sin(theta)))/(2*((v2*cos(phi)*cos(psi) - ...
                v1*cos(phi)*sin(psi) + v3*cos(theta)*sin(phi) ...
126         + v1*cos(psi)*sin(phi)*sin(theta) + v2*sin(phi)*sin(psi)*sin(theta))^2 ...
                + (v1*cos(psi)*cos(theta) - v3*sin(theta) ...
127         + v2*cos(theta)*sin(psi))^2)^(1/2)))/m;
128     v3_dot=-((S*intp2(atan(w/(u^2 + v^2)^(1/2)))*rho*v3*(v1^2 + v2^2 + ...
            v3^2)^(1/2))/2 - g*m + (S*intp1(atan(w/(u^2 + v^2)^(1/2)))*rho*(v1^2 + ...
            v2^2 + v3^2)^(1/2)*(v1^2*cos(phi)*cos(theta) ...
129         + v2^2*cos(phi)*cos(theta) + v2*v3*cos(psi)*sin(phi) - ...
                v1*v3*sin(phi)*sin(psi) - v1*v3*cos(phi)*cos(psi)*sin(theta) ...
130         - v2*v3*cos(phi)*sin(psi)*sin(theta)))/(2*((v2*cos(phi)*cos(psi) - ...
                v1*cos(phi)*sin(psi) + v3*cos(theta)*sin(phi) ...
131         + v1*cos(psi)*sin(phi)*sin(theta) + v2*sin(phi)*sin(psi)*sin(theta))^2 ...
                + (v1*cos(psi)*cos(theta) - v3*sin(theta) ...
132         + v2*cos(theta)*sin(psi))^2)^(1/2)))/m;
133     vel_1_dot=[v1_dot;v2_dot;v3_dot];
134
135     T_r12=[1,0, -sin(theta);
136     0, cos(phi), cos(theta)*sin(phi);
137     0, -sin(phi), cos(theta)*cos(phi)];
138
139     E_dot=inv(T_r12)*[p;q;r];
140
141     pos_1_dot=vel_1;
142
143     output=[Omega_2_dot;vel_1_dot; pos_1_dot; E_dot];
144
145 end
146 function output=f_dot_rebound(time,input)
147
148     Omega_2=input(1:3,:);
149     vel_1=input(4:6,:);
150     pos_1=input(7:9,:);
151     E=input(10:12,:);
152
153     p=Omega_2(1);
154     q=Omega_2(2);
155     r=Omega_2(3);
156     v1=vel_1(1);
157     v2=vel_1(2);
158     v3=vel_1(3);
159     z=pos_1(3);
160     phi=E(1);
161     theta=E(2);
162     psi=E(3);
163
164     global g m d rho S Ix Iy Iz b k;
165     u=v1*cos(psi)*cos(theta) - v3*sin(theta) + v2*cos(theta)*sin(psi);
166     v=v2*(cos(phi)*cos(psi) + sin(phi)*sin(psi)*sin(theta)) - ...
            v1*(cos(phi)*sin(psi) - cos(psi)*sin(phi)*sin(theta)) + ...
            v3*cos(theta)*sin(phi);
167     w=v1*(sin(phi)*sin(psi) + cos(phi)*cos(psi)*sin(theta)) - ...
            v2*(cos(psi)*sin(phi) - cos(phi)*sin(psi)*sin(theta)) + ...
            v3*cos(phi)*cos(theta);
168
169     dot_p=-(Iz*q*r - Iy*q*r + (S*intp3(atan(w/(u^2 + ...
            v^2)^(1/2)))*d*rho*((v3*cos(phi)*cos(theta) - v2*cos(psi)*sin(phi) + ...
            v1*sin(phi)*sin(psi)...
170         + v1*cos(phi)*cos(psi)*sin(theta) + v2*cos(phi)*sin(psi)*sin(theta))^2 ...
                + (v2*cos(phi)*cos(psi) - v1*cos(phi)*sin(psi)...
171         + v3*cos(theta)*sin(phi) + v1*cos(psi)*sin(phi)*sin(theta) + ...
                v2*sin(phi)*sin(psi)*sin(theta))^2 + (v1*cos(psi)*cos(theta)...
172         - v3*sin(theta) + v2*cos(theta)*sin(psi))^2)*(v2*cos(phi)*cos(psi) - ...
                v1*cos(phi)*sin(psi) + v3*cos(theta)*sin(phi)...
173         + v1*cos(psi)*sin(phi)*sin(theta) + ...
                v2*sin(phi)*sin(psi)*sin(theta)))/(2*((v2*cos(phi)*cos(psi) - ...
                v1*cos(phi)*sin(psi)...
174         + v3*cos(theta)*sin(phi) + v1*cos(psi)*sin(phi)*sin(theta) + ...
                v2*sin(phi)*sin(psi)*sin(theta))^2 + (v1*cos(psi)*cos(theta)...
175         - v3*sin(theta) + v2*cos(theta)*sin(psi))^2)^(1/2)) + ...
                (d*cos(phi)*cos(psi)*cos(theta)^2*(cos(psi)*sin(phi)...
```

```
176         - cos(phi)*sin(psi)*sin(theta))*(d*cos(phi)*cos(theta) - ...
                d*cos(phi)^3*cos(theta)^3 + 2*b*v3*(1 - ...
                cos(phi)^2*cos(theta)^2)^(1/2)...
177         + 2*z*cos(phi)*cos(theta)*(1 - cos(phi)^2*cos(theta)^2)^(1/2) + ...
                b*d*q*cos(phi)*cos(theta)*sin(theta)...
178         + b*d*p*cos(phi)*cos(theta)^2*sin(phi)))/(4*(- cos(phi)^2*cos(theta)^2 ...
                + 1)^(1/2)*(sin(phi)^2 + sin(theta)^2 ...
179         - sin(phi)^2*sin(theta)^2)^(1/2)) + ...
                (d*cos(phi)*cos(theta)^2*sin(psi)*(sin(phi)*sin(psi) + ...
                cos(phi)*cos(psi)*sin(theta))...
180         *(d*cos(phi)*cos(theta) - d*cos(phi)^3*cos(theta)^3 + 2*b*v3*(1 - ...
                cos(phi)^2*cos(theta)^2)^(1/2) + 2*z*cos(phi)*cos(theta)...
181         *(1 - cos(phi)^2*cos(theta)^2)^(1/2) + ...
                b*d*q*cos(phi)*cos(theta)*sin(theta) + ...
                b*d*p*cos(phi)*cos(theta)^2*sin(phi)))...
182         /(4*(- cos(phi)^2*cos(theta)^2 + 1)^(1/2)*(sin(phi)^2 + sin(theta)^2 - ...
                sin(phi)^2*sin(theta)^2)^(1/2)))/Ix;
183     dot_q=(Iz*p*r - Ix*p*r + (S*intp3(atan(w/(u^2 + ...
            v^2)^(1/2)))*d*rho*(v1*cos(psi)*cos(theta) - v3*sin(theta) + ...
            v2*cos(theta)*sin(psi))*((v3*cos(phi)*cos(theta)...
184         - v2*cos(psi)*sin(phi) + v1*sin(phi)*sin(psi) + ...
                v1*cos(phi)*cos(psi)*sin(theta) + ...
                v2*cos(phi)*sin(psi)*sin(theta))^2 ...
185         + (v2*cos(phi)*cos(psi) - v1*cos(phi)*sin(psi) + v3*cos(theta)*sin(phi) ...
                + v1*cos(psi)*sin(phi)*sin(theta) ...
186         + v2*sin(phi)*sin(psi)*sin(theta))^2 + (v1*cos(psi)*cos(theta) - ...
                v3*sin(theta) + v2*cos(theta)*sin(psi))^2))/...
187         (2*((v2*cos(phi)*cos(psi) - v1*cos(phi)*sin(psi) + ...
                v3*cos(theta)*sin(phi) + v1*cos(psi)*sin(phi)*sin(theta) ...
188         + v2*sin(phi)*sin(psi)*sin(theta))^2 + (v1*cos(psi)*cos(theta) - ...
                v3*sin(theta) + v2*cos(theta)*sin(psi))^2)^(1/2)) ...
189         - (d*cos(phi)*cos(theta)*(cos(phi)*cos(psi) + ...
                sin(phi)*sin(psi)*sin(theta))*(sin(phi)*sin(psi) ...
190         + cos(phi)*cos(psi)*sin(theta))*(d*cos(phi)*cos(theta) - ...
                d*cos(phi)^3*cos(theta)^3 + 2*b*v3*(1 - ...
                cos(phi)^2*cos(theta)^2)^(1/2)...
191         + 2*z*cos(phi)*cos(theta)*(1 - cos(phi)^2*cos(theta)^2)^(1/2) + ...
                b*d*q*cos(phi)*cos(theta)*sin(theta)...
192         + b*d*p*cos(phi)*cos(theta)^2*sin(phi)))/(4*(- cos(phi)^2*cos(theta)^2 ...
                + 1)^(1/2)*(sin(phi)^2 + sin(theta)^2 ...
193         - sin(phi)^2*sin(theta)^2)^(1/2)) + ...
                (d*cos(phi)*cos(theta)*(cos(phi)*sin(psi) - ...
                cos(psi)*sin(phi)*sin(theta))*(cos(psi)*sin(phi)...
194         - cos(phi)*sin(psi)*sin(theta))*(d*cos(phi)*cos(theta) - ...
                d*cos(phi)^3*cos(theta)^3 + 2*b*v3*(1 - ...
                cos(phi)^2*cos(theta)^2)^(1/2) ...
195         + 2*z*cos(phi)*cos(theta)*(1 - cos(phi)^2*cos(theta)^2)^(1/2) + ...
                b*d*q*cos(phi)*cos(theta)*sin(theta)...
196         + b*d*p*cos(phi)*cos(theta)^2*sin(phi)))/(4*(- cos(phi)^2*cos(theta)^2 ...
                + 1)^(1/2)*(sin(phi)^2 + sin(theta)^2 ...
197         - sin(phi)^2*sin(theta)^2)^(1/2)))/Iy;
198     dot_r=0;
199     Omega_2_dot=[dot_p;dot_q;dot_r];
200
201     v1_dot=-((S*intp2(atan(w/(u^2 + v^2)^(1/2)))*rho*v1*(v1^2 + v2^2 + ...
            v3^2)^(1/2))/2 + (S*intp1(atan(w/(u^2 + v^2)^(1/2)))*rho*(v1^2 + v2^2 + ...
            v3^2)^(1/2)*(v2^2*sin(phi)*sin(psi) ...
202         + v3^2*sin(phi)*sin(psi) + v1*v2*cos(psi)*sin(phi) + ...
                v2^2*cos(phi)*cos(psi)*sin(theta) + ...
                v3^2*cos(phi)*cos(psi)*sin(theta) ...
203         - v1*v3*cos(phi)*cos(theta) - ...
                v1*v2*cos(phi)*sin(psi)*sin(theta)))/(2*((v2*cos(phi)*cos(psi) - ...
                v1*cos(phi)*sin(psi) + v3*cos(theta)*sin(phi) ...
204         + v1*cos(psi)*sin(phi)*sin(theta) + v2*sin(phi)*sin(psi)*sin(theta))^2 ...
                + (v1*cos(psi)*cos(theta) - v3*sin(theta) ...
205         + v2*cos(theta)*sin(psi))^2)^(1/2)))/m;
206
207     v2_dot=-((S*intp2(atan(w/(u^2 + v^2)^(1/2)))*rho*v2*(v1^2 + v2^2 + ...
            v3^2)^(1/2))/2 - (S*intp1(atan(w/(u^2 + v^2)^(1/2)))*rho*(v1^2 + v2^2 + ...
            v3^2)^(1/2)*(v1^2*cos(psi)*sin(phi) ...
208         + v3^2*cos(psi)*sin(phi) + v1*v2*sin(phi)*sin(psi) - ...
                v1^2*cos(phi)*sin(psi)*sin(theta) - ...
                v3^2*cos(phi)*sin(psi)*sin(theta) ...
```

```
        + v2*v3*cos(phi)*cos(theta) + ...
            v1*v2*cos(phi)*cos(psi)*sin(theta)))/(2*((v2*cos(phi)*cos(psi) - ...
            v1*cos(phi)*sin(psi) + v3*cos(theta)*sin(phi) ...
        + v1*cos(psi)*sin(phi)*sin(theta) + v2*sin(phi)*sin(psi)*sin(theta))^2 ...
            + (v1*cos(psi)*cos(theta) - v3*sin(theta) ...
        + v2*cos(theta)*sin(psi))^2)^(1/2)))/m;
    v3_dot=(- b*(v3 + ...
        (d*(cos(phi)^2*cos(theta)*sin(theta)*((2*q*cos(phi))/(cos(phi)^2 + ...
        sin(phi)^2) - (2*r*sin(phi))/(cos(phi)^2 ...
        + sin(phi)^2)) + cos(phi)*cos(theta)^2*sin(phi)*(2*p + ...
            (2*r*cos(phi)*sin(theta))/(cos(phi)^2*cos(theta) + ...
            cos(theta)*sin(phi)^2) ...
        + (2*q*sin(phi)*sin(theta))/(cos(phi)^2*cos(theta) + ...
            cos(theta)*sin(phi)^2))))/(4*(- cos(phi)^2*cos(theta)^2 + ...
            1)^(1/2))) ...
        - k*(z + (d*(1 - ...
            cos(phi)^2*cos(theta)^2)^(1/2))/2))/m-((S*intp2(atan(w/(u^2 + ...
            v^2)^(1/2)))*rho*v3*(v1^2 + v2^2 + v3^2)^(1/2))/2 ...
        - g*m + (S*intp1(atan(w/(u^2 + v^2)^(1/2)))*rho*(v1^2 + v2^2 + ...
            v3^2)^(1/2)*(v1^2*cos(phi)*cos(theta) ...
        + v2^2*cos(phi)*cos(theta) + v2*v3*cos(psi)*sin(phi) - ...
            v1*v3*sin(phi)*sin(psi) - v1*v3*cos(phi)*cos(psi)*sin(theta) ...
        - v2*v3*cos(phi)*sin(psi)*sin(theta)))/(2*((v2*cos(phi)*cos(psi) - ...
            v1*cos(phi)*sin(psi) + v3*cos(theta)*sin(phi) ...
        + v1*cos(psi)*sin(phi)*sin(theta) + v2*sin(phi)*sin(psi)*sin(theta))^2 ...
            + (v1*cos(psi)*cos(theta) - v3*sin(theta) ...
        + v2*cos(theta)*sin(psi))^2)^(1/2)))/m;
    vel_1_dot=[v1_dot;v2_dot;v3_dot];

    T_r12=[1,0, -sin(theta);
    0, cos(phi), cos(theta)*sin(phi);
    0, -sin(phi), cos(theta)*cos(phi)];

    E_dot=inv(T_r12)*[p;q;r];

    pos_1_dot=vel_1;

    output=[Omega_2_dot;vel_1_dot; pos_1_dot; E_dot];

end

function [value, isterminal, direction]=touchdown(time, input)
ht=abs(input(9))-1/2*0.27*sqrt(1-(cos(input(10))*cos(input(11)))^2);
    if ht<=0 && input(6)>0
        value=0;
    else
        value=1;
    end
    isterminal=1;
    direction=0;
end
function [value, isterminal, direction]=rebound(time, input)
hp=abs(input(9))-1/2*0.27*sqrt(1-(cos(input(10))*cos(input(11)))^2);
    if hp>=0 && input(6)<0
        value=0;
    else
        value=1;
    end
    isterminal=1;
    direction=0;
end

function cLq=intp1(bq)

    bL = [-1 -0.20003535579531206 -0.17840202853875603 -0.1589209853218206 ...
        -0.14249379892496286 -0.12610638779783115 -0.10656789141351365 ...
        -0.09002137920747776 -0.07354999806647994 -0.05420595855637869 ...
        -0.03759757370854672 -0.021369263660319188 -0.002206422601192165 ...
        0.014273797488633666 0.030780534424943523 0.050151090781528795 ...
        0.06691857670826495 0.08308943358911042 0.10262351049901391 ...
        0.11954125855582626 0.1380632758249225 0.15538761552782338 ...
        0.17111210549285427 0.19017329864045918 0.20584917438693606 ...
        0.22332819569432705 0.24285343365540252 0.25884751155968777 ...
```

```
            0.2755796416911119 0.2955424076191739 0.31158509974201315 ...
            0.32783550716231075 0.34732980880248815 0.3639249352270781 ...
            0.3803477021495219 0.3997712921990752 0.41599960224730276 ...
            0.4332488108851655 0.4520227381958601 0.4687725462249402 ...
            0.48723711032665445 0.5040576299463586 0.5212405464680112 ...
            0.5400056348298778 0.5567466039101301 0.5731295955628477 ...
            0.5921819497616245 1];
259     cL = [-1.0919 -0.2486963953862198 -0.21860959296171129 -0.18802879299934916 ...
            -0.12905489645654783 -0.11545820231943604 -0.08631234758607709 ...
            -0.05566097654687879 -0.04142914271824105 -0.017623166132156087 ...
            0.03328210395903741 0.09098572111878678 0.13808015306082444 ...
            0.19458406191435806 0.24981769138483978 0.3141785134594727 ...
            0.3831264555284598 0.45795532066713196 0.5027914781504106 ...
            0.5630591777685425 0.6187632810845991 0.6754083320918052 ...
            0.7522837584587276 0.7972375344034 0.8411092205032502 ...
            0.8854278567563965 0.9407791046882719 0.986862017862323 ...
            1.0334389285742291 1.0898957900432054 1.1336498576816618 ...
            1.1947644102218284 1.23842438309117 1.2816609294994943 ...
            1.3405877786577383 1.400432051814853 1.457923955744093 ...
            1.4841763963271677 1.527507037504607 1.590033011581498 ...
            1.6320698496836066 1.6770001019360001 1.6916553622256552 ...
            1.7074632834369683 1.751923061843787 1.7781284550423035 ...
            1.8245877472928154 2.81905477];
260     cLq = interp1(bL,cL,bq,'linear',0);
261 end
262 function cDq=intp2(bq)
263     bD = [-1 -0.19980708036046615 -0.1787788396512768 -0.15911361813444613 ...
            -0.14238413026269575 -0.12562528505449455 -0.10642978092087675 ...
            -0.08954511855645785 -0.07344052256058883 -0.0539891902093283 ...
            -0.037419070735453724 -0.02091185983968799 -0.0021860730892830083 ...
            0.015109591985447118 0.0310799830146839 0.0503677530628614 ...
            0.06703852626171014 0.08311376492112837 0.10276640472233722 ...
            0.11921070703999409 0.13834330259550304 0.15443951078095752 ...
            0.17154225621615382 0.19050709556337247 0.2076391983350196 ...
            0.22428900200783208 0.24348870004665718 0.26035658679024704 ...
            0.2761969667580588 0.2955182880478944 0.312675554250785 ...
            0.3285620671758766 0.3478875823709194 0.3644828652760375 ...
            0.3815017326070887 0.4004497963334784 0.4171750903000215 ...
            0.43314128742405095 0.45163221548284954 0.46916693315439334 ...
            0.48810241516516123 0.5044167064213931 0.5218339947471338 ...
            0.5396706735936001 0.5571802278339004 0.5733015994505983 ...
            0.5929961783038797 1];
264     cD = [1.55016 0.18399887451593078 0.1653750749705671 0.14695563963777183 ...
            0.12207651810770585 0.10978800915196256 0.1047144454399384 ...
            0.09961422552627487 0.0984768942563329 0.09378540276780227 ...
            0.09499381724211976 0.09442515160714682 0.09998741234922585 ...
            0.11192939068365752 0.11609664353994358 0.13318326879077103 ...
            0.14489422671099494 0.15840003554160217 0.17418050691210116 ...
            0.19002317608642533 0.2154176508481855 0.24535256528918276 ...
            0.26751275425203436 0.29248961518811134 0.32157153117664916 ...
            0.34478019740398214 0.3725915011143773 0.4018066981111119 ...
            0.4300800426499226 0.465461707625895 0.5005146127816487 ...
            0.5433333580150015 0.5789638142062745 0.6137235011439953 ...
            0.6467149933729721 0.6888939897669804 0.7239024679200018 ...
            0.7585644154517115 0.7920357193101967 0.8386041035741524 ...
            0.8760205252752623 0.9087987678911242 0.9259831325479626 ...
            0.9570553782589057 0.979837545260009 1.003046211487342 ...
            1.0373349721961007 1.745938936447];
265     cDq = interp1(bD,cD,bq,'linear',0);
266 end
267 function cMq=intp3(bq)
268     bM = [-1 -0.19997534238130954 -0.17729855239230272 -0.15799985616389098 ...
            -0.14201760964934812 -0.12493809910307914 -0.10825311045585774 ...
            -0.08874071486546188 -0.07233928883318094 -0.05571183462957066 ...
            -0.035648752221754204 -0.019999383559532763 -0.0026198719858629704 ...
            0.016366494405802734 0.03294052376891701 0.049296744166932055 ...
            0.06582556789578042 0.08540782673913266 0.10212569221127465 ...
            0.1189052017301429 0.1381545827211737 0.15461354319706572 ...
            0.17116702454460458 0.19115202449323454 0.20665755704642824 ...
            0.22406994544501863 0.24336864167343036 0.26003719190819147 ...
            0.276627659683766 0.29282360556029297 0.3120401097264031 ...
            0.3290374282103706 0.34493748266261187 0.364622481583841 ...
            0.3821088428385028 0.3976472522166172 0.41721718225062415 ...
```

```
            0.43342956653961145 0.4505625019263764 0.46677899581847876 ...
            0.4856544029260374 0.5031613121962746 0.5192010931544286 ...
            0.5392559563560149 0.5554560118356568 0.5726711392847235 ...
            0.5909465443374806 1];
269         cM = [-0.126291185 -0.03983252030166611 -0.03738184502699651 ...
            -0.03621658005587482 -0.02947396609875078 -0.02766744350998075 ...
            -0.022515669556909805 -0.02066716847849688 -0.01588596326158389 ...
            -0.01291272816756655 -0.010272425922441118 -0.006921384421637741 ...
            -0.006126688186674001 -0.008067831449126414 -0.009157824645716024 ...
            -0.007027054412807779 -0.006241043382597747 -0.0031100270688881465 ...
            -0.0019346293588871355 -0.0016407799313868793 -0.001543795144970543 ...
            0.00043643153887353014 0.004502554897731714 0.008551307847082495 ...
            0.011424663086432266 0.014292228189279568 0.01723216999840771 ...
            0.021137617069322406 0.02598251378776255 0.03190003329328489 ...
            0.03575336913567738 0.04062432146837862 0.04344556547920618 ...
            0.04965548687810316 0.05461473879246703 0.06129800385044078 ...
            0.0699701807970123 0.07703848993240016 0.08380426443553408 ...
            0.09053819318790442 0.09730252015691272 0.10526395784780627 ...
            0.1119197197573933 0.11685581112574729 0.12777166596702516 ...
            0.138340112618155 0.14502482521025434 0.29464694182];
270         cMq = interp1(bM,cM,bq,'linear',0);
271     end
```