2017

Six Degree of Freedom Dynamic Modeling of a High Altitude Airship and Its Trajectory Optimization Using Direct Collocation Method

Pradens Pierre-Louis
University of Central Florida

SIX DEGREE OF FREEDOM DYNAMIC MODELING OF A HIGH ALTITUDE AIRSHIP AND ITS TRAJECTORY OPTIMIZATION USING DIRECT COLLOCATION METHOD

by

PRADENS PIERRE-LOUIS
B.S.A.E. University of Central Florida, 2014

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science in Aerospace Engineering
in the Department of Mechanical and Aerospace Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2017

Major Professors:
Yunjun Xu
Kuo-Chi Lin

ABSTRACT

The long duration airborne feature of airships makes them an attractive solution for many military and civil applications such as long-endurance surveillance, reconnaissance, environment monitoring, communication utilities, and energy harvesting. To achieve a minimum energy periodic motion in the air, an optimal trajectory problem is solved using basic direct collocation methods. In the direct approach, the optimal control problem is converted into a nonlinear programming (NLP). Pseudo-inverse and several discretization methods such as Trapezoidal and Hermite-Simpson are used to obtain a numerical approximated solution by discretizing the states and controls into a set of equal nodes. These nodes are approximated by a cubic polynomial function which makes it easier for the optimization to converge while ensuring the problem constraints and the equations of motion are satisfied at the collocation points for a defined trajectory. In this study, direct collocation method provides the ability to obtain an approximation solution of the minimum energy expenditure of a very complex dynamic problem using Matlab fmincon optimization algorithm without using Himiltonian function with Lagrange multipliers. The minimal energy trajectory of the airship is discussed and results are presented.

ACKNOWLEDGMENTS

The pursuit of my master has been a challenging journey that has not only pushed me to the limits, but also taught me to be grateful to those who have helped me. Foremost, I would like to thanks sincerely my advisor Dr. Yunjun Xu for his guidance, insight and patience.

My deepest gratitude and thanks go to Dr. Kuo-Chi Lin for serving as my second advisor. I extend my appreciations to all of the committee members for their time and valuable feedback. And I also want to extend my thanks to all my lab-mates especially August Mark who I am greatly indebted for helping me.

Finally, my deepest gratitude to my parents for their loyal and ever-lasting love and support through my entire life. I am also extremely grateful to the one who created all thing and Sovereign Lord of the universe, Jehovah God.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Airship Historic Background

The late of 18th century was the beginning of the airship industry technology, and it would take nearly half century before airships industry to firmly established. With the invention of steam powered engine, Henri Giffard built the first single propeller steam powered airship in 1852 [1]. Then, airships became the most advance form of transportation at that time. In the early 1900s, the industry of airships achieved great success with the launch of the Zeppelins which were not only used for long distance commercial flights, but they were also used during the First World War by the German's army [2], [3]

As airplane's technology started to develop in the late of 1930's, slowly airships became obsolete. A number of airships' accidents such as the infamous Hindenburg disaster in 1937 contributed to the abandonment of airships as a way of travelling in the 1940's [2].

As modern society face new challenges which require slow moving airborne and energy efficient vehicles, airships slowly have started to regain popularity due to their buoyant nature and slow dynamic.

1.2 Motivation

Currently, majority of papers used indirect collocation method based on Pontryagin's maximum Principle (PMP) to study airship trajectory optimization problem. This method can be very difficult to implement successfully because the problem is ill-conditioned due to extreme sensitivity of the co-state dynamic [4]. On the other hand, direct collocation method is inherently

slow and very difficult to converge. Motivated by that, a six degree of freedom of an airship trajectory optimization problem is solved using direct collocation method.

1.3 Thesis Contribution

Direct collocation method is used to solve the airship optimal trajectory problem. In the direct approach, the optimal control problem is converted into a NLP problem. Control variables and performance index are calculated at each grip point. A differentiation matrix is used to estimate the derivative of the state variables. Trapezoidal and Hermite-Simpson discretization methods are used due to the fact that they retain the structure of the original problem while both states and controls are discretized at the selected numbers of collocation nodes [4]. Graphical and numerical results are presented to illustrate the effectiveness of the direct collocation and discretization methods used in solving this nonlinear 6DOF trajectory optimization problems.

1.4 Thesis Outline

In this paper, CHAPTER 2 discusses the basic coordinate system used for the simulations. In CHAPTER 3, the nonlinear dynamics model of the airship is presented including all the external forces and moments acting on the airship as well as its virtual mass and inertial terms. CHAPTER 4, discusses model validation by using Simulink/Matlab. CHAPTER 5, discusses straight level flight condition. CHAPTER 6, discusses the collocation methods and problem formulation. In CHAPTER 7, simulation results are presented and the effectiveness of each discretization method is discussed. CHAPTER 8 concludes this thesis and provides an overview of the work done and recommendation for future work.

CHAPTER 2: COORDINATE SYSTEM

2.1 Reference Frames

Generally, a rigid body consists of a finite rotation about its axes and a finite translation along some vectors [5]. Therefore, a reference frame is very necessary to convey the motion of a rigid body in space as well as to describe its position relative to its close environment and its datum line.

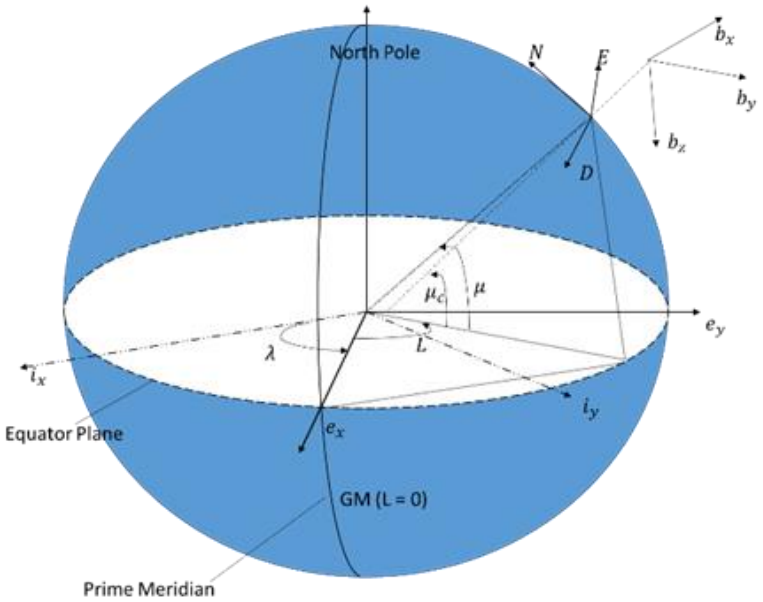


Figure 2.1.1 Earth geodetic datum

The airship's motion can be described in space by a set of coordinate system which can be categorized in two frames such as body and inertial axes.

2.2 Body and Inertial Reference frame

The body frame is the coordinate system that is aligned and attached to the airship and points towards the orthogonal O_x, O_y and O_z direction. As can be seen in figure 2.2.1, the $x-axis$

points toward the nose of the airship, the $y-axis$ points toward the right side of the airship, and the $z-axis$ points downward vertically.

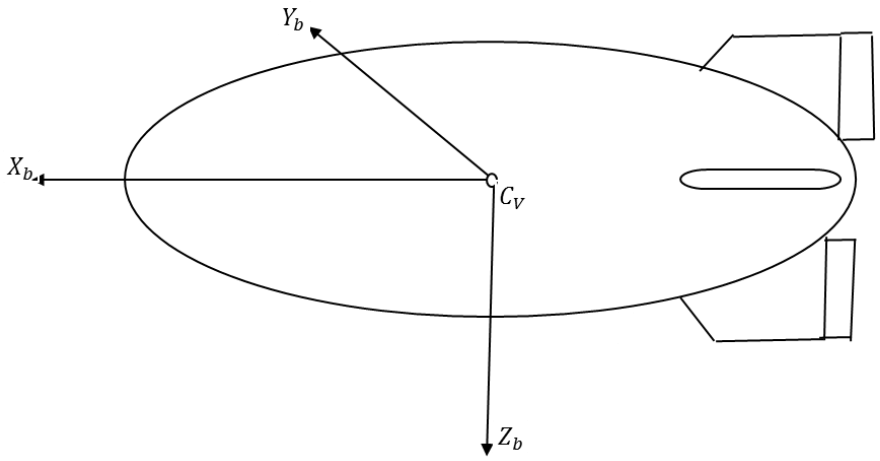


Figure 2.2.1 Airship body axes

Unlike aircraft, the airship's body axes system are defined at the center of its volume C_v, not at the center of its gravity C_g. By placing the body axes system at its C_v, allows the simplification of the equations of motion as well as the calculation for both virtual mass and inertial terms.

2.3 Body to Inertia Rotational Sequence

The airship's orientation and position with regard to its inertia or body frame can be best described by the Euler 3-2-1 rotation sequences [6]. The transformation between the frames is accomplished through the Euler angles, ψ, θ, φ. These rotation sequences are not commutative, therefore, they must be made in the specified order to achieve the desired vehicle position and orientation [14]. It can be represented as such:

$$R_{ib}^{i} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & s\phi s\theta c\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & c\theta s\psi & c\theta c\phi \end{pmatrix} \qquad (2.3.1)$$

$$-\pi \le \psi \ge \pi \quad or \quad 0 \le \psi \ge 2\pi$$

$$-\pi/2 \le \theta \ge \pi/2$$

$$-\pi \le \phi \ge \pi \quad or \quad 0 \le \phi \ge 2\pi$$

Where, $c = \cos$, $s = \sin$, and $t = \tan$

It should be noted here that singularities can result from Equation 2.3.1. Therefore, in order to avoid these ambiguities, the range of the Euler angles should be limited.

2.4 Positions and Altitude Kinematics

Position kinematic equations relate the position of the airship in body-fixed reference frame to the inertial reference frame. The rate of change of the position in the inertial frame can be represented as follow:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \end{bmatrix} = \begin{pmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & s\phi s\theta c\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & c\theta s\psi & c\theta c\phi \end{pmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \qquad (2.4.1)$$

where, $[u, v, w]$ are linear velocities

The relationship between angular velocity and a rigid body coordinate system is well defined in Ref. [7]. it shows that the angular velocity between the inertial and body-fixed coordinate system are related through the three Euler angles rotations [20]:

$$\begin{pmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{pmatrix} = \begin{bmatrix} 1 & \mathrm{s}\phi\,\mathrm{t}\theta & \mathrm{c}\phi\,\mathrm{t}\theta \\ 0 & \mathrm{c}\phi & -\mathrm{s}\phi \\ 0 & \mathrm{s}\phi/\mathrm{c}\theta & \mathrm{c}\phi/\mathrm{c}\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{2.4.2}$$

Where, $[p, q, r]$ are angular velocities

The altitude Kinematic equations become singular for the pitch angle when $\theta = \pm 90^{\circ}$ *or* $\pm \frac{\pi}{2}$.

However, under normal airship operation, this singularity does not occur [15].

CHAPTER 3:
AIRSHIP MATHEMATICAL MODEL

3.1 Airship Geometry and Parameters

The airship model consists of an axisymmetric hull which is formed by a lightweight skeletal structure. The hull is modeled as a double ellipsoid with equal semi-minor axis, b and different semi-major axes a_1 and a_2. This ellipsoid configuration facilitates the derivation of the airship aerodynamic equations [8]. The lift capability of the airship is basically generated by the volume of the hull.

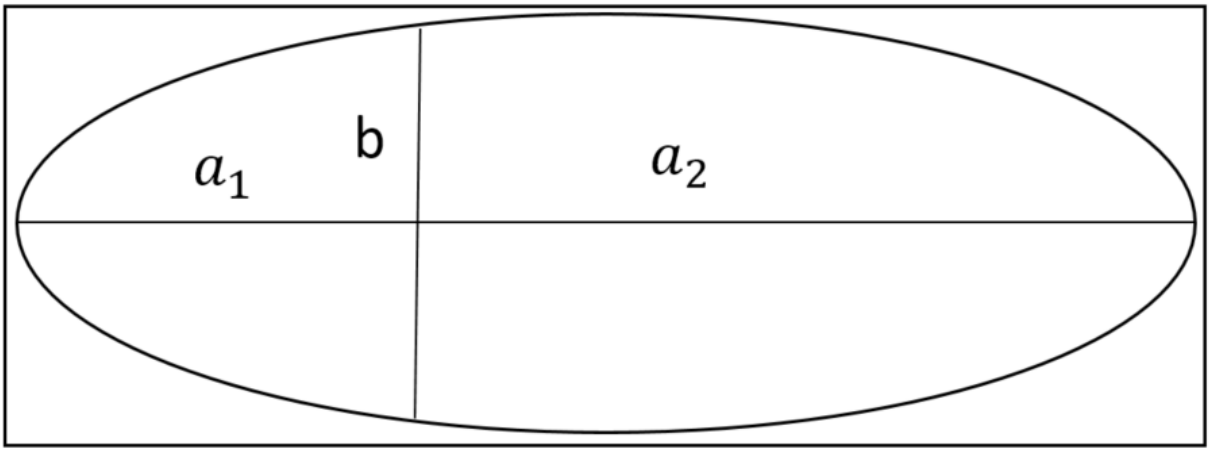


Figure 3.1.1 Double ellipsoid hull

The parameters of the airship can be defined as such: L_h, total length of the airship, b, semi-minor axis, V_h, volume of the hull, and S_h, surface area.

$$a_1 = L_h / 3 \quad (3.1.1)$$

$$a_2 = 2L_h / 3 \quad (3.1.2)$$

$$b = D / 2 \quad (3.1.3)$$

$$V_h = 2 / 3\pi L_h b^2 \quad (3.1.4)$$

$$S_h = V_h^{2/3} \quad (3.1.5)$$

$$d_{cv} = a_1 + 3 / 8L_h \quad (3.1.6)$$

The parameters of the airship hull are carefully chosen because of the need for the airship to operate at a prescribes altitude. The parameters contain in Table 3.1.1 provide a suitable volume that met the airship operation requirement [8].

Table 3.1.1 Airship Parameters

Parameter	*Symbol*	*Value*
Pressure Altitude	h_p	$21\ km$
Length	L_v	$250\ m$
Diameter	D	$75\ m$
Volume	V_h	$736311\ m^3$
Surface Area	S_h	$8150\ m^2$
Max Thrust	T_{max}	$15\ kN$
Max Propulsion Power	T_{max}	$200\ kW$
Max Speed	V_{max}	$45\ m/s$

As depicted in Figure 3.1.1, the airship has an ellipsoid shape and four fins that are mounted in the aft section in a cross '+' configuration. These include two elevators (left and right) for longitudinal motions and two rudders (top and bottom) for lateral motions. In addition, tree thrusters mounted on the bottom of the airship.

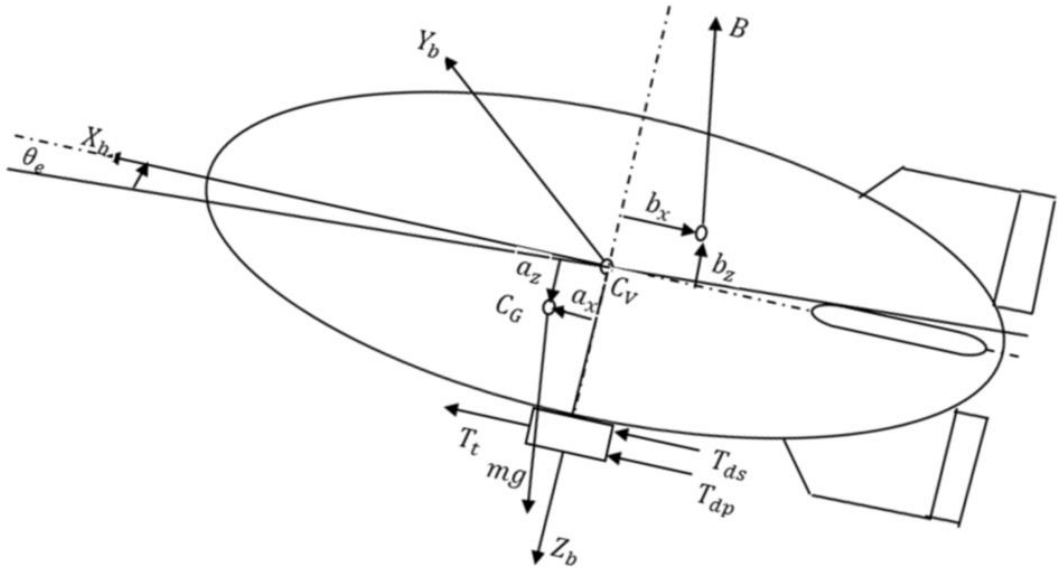


Figure 3.1.2 Airship configuration

3.2 Aerostatics

In this paper, aerostatic force that acts on that airship is considered. Several assumptions are made such as the entire hull volume is filled with helium gas, and it is fixed at any altitude. Furthermore, the net lift force of the airship is assumed to be equal to the total weight of the airship structure which includes power, propulsion, and payload system [8].

3.3 Virtual Mass and Inertia

The interaction of the surrounding air with the airship is called virtual mass. This phenomenon impacts the airship directly through the air stream velocities around the airship [9]. In fact, any moving object through gas (i.e. air) or fluid, encounters a virtual mass effects. However, in many cases that phenomenon assumes to be negligible. In the case of an airship, those terms cannot be neglected [10]. Therefore, mass matrix of the airship can be expressed as such:

$$[\mathbf{\Pi}] = m_{air}\begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & k_2 \end{bmatrix} \qquad (3.3.1)$$

$$[\mathbf{\Gamma}] = m_{air}\begin{bmatrix} 0 & 0 & 0 \\ 0 & k_3 & 0 \\ 0 & 0 & k_3 \end{bmatrix} \qquad (3.3.2)$$

Due to the fact that airships move very slow, it is reasonable to assume that longitudinal-lateral coupling effects are insignificant [11]. Therefore, some of the virtual mass and inertia terms are assumed to be zero, whereas others can be estimated as given in ref [8]. k_1, k_2 , and k_3 are ellipsoid inertia factors and these constants can be obtained through experiment of potential flow [8].

3.4 Equation of Motion

The general nonlinear 6DOF dynamic equations for the airship are derived. Many assumptions are utilized such as the following: rigid, no aero-elastic effect, constant mass and volume, equilibrium disturbances nearly zero, hull is symmetric $I_{yz} = I_{xy} = 0$. In addition,

many other assumptions are made such as: body axes move along with the airship, body axes origin coincides with C_v, and C_g and center of buoyancy C_b are lied in the plane of symmetry $\left(i.e.\, a_y = b_y = 0\right)$.

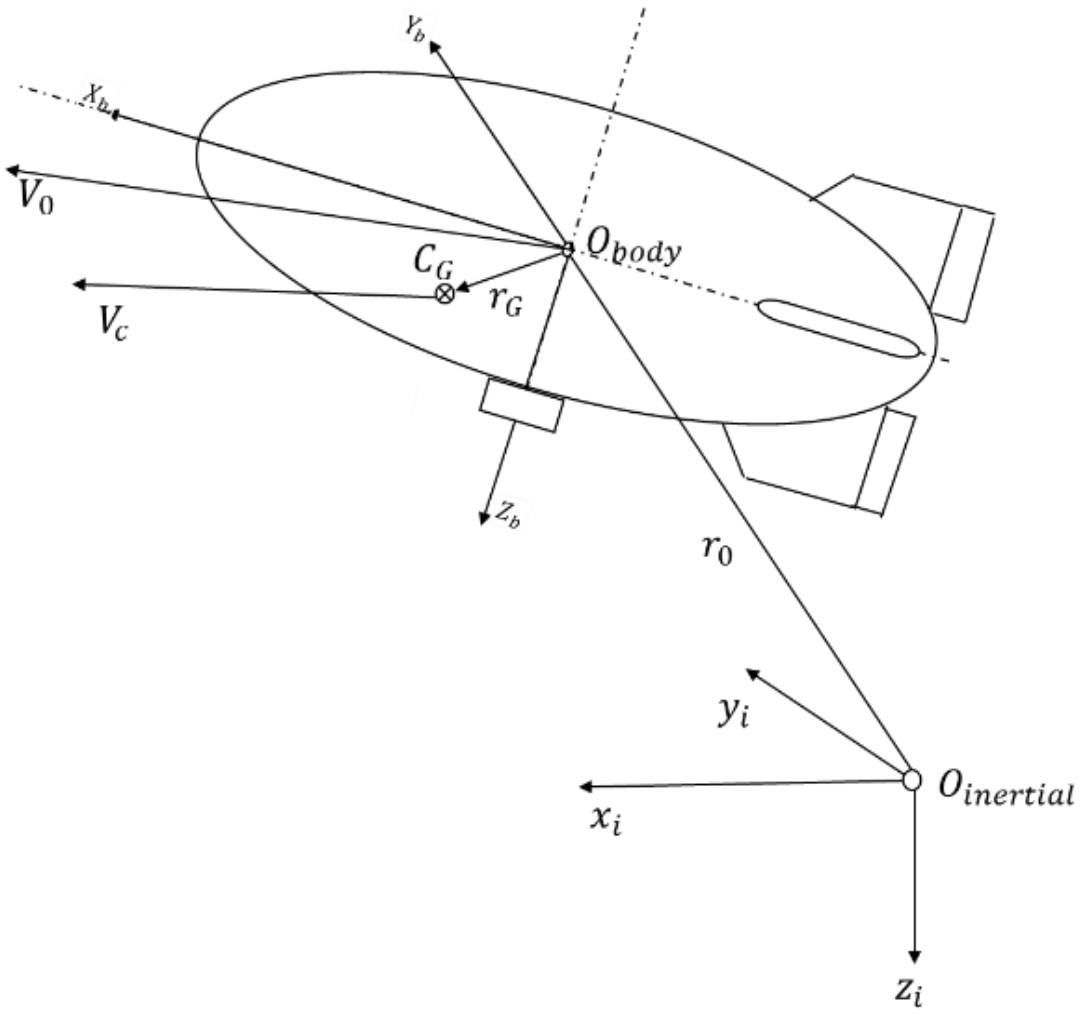


Figure 3.4.1 Airship body and inertial frame

The equations of motion for the airship can be written as follows:

$$m\dot{\vec{v}} = \sum \vec{F} \tag{3.4.1}$$

$$\mathbf{I}\dot{\vec{\omega}} = \sum \vec{T} \tag{3.4.2}$$

Equation(3.4.1) and Equation(3.4.2) can be expended to obtain the translational and rotational equations of motion in body frame [8].

$$m\left[\dot{\vec{v}}_B + \vec{\omega}\times\vec{v}_B - \mathbf{r}_G\times\dot{\vec{\omega}} + \vec{\omega}\times(\vec{\omega}\times\mathbf{r}_G)\right] = \sum \vec{F} \tag{3.4.3}$$

$$\mathbf{I}\dot{\vec{\omega}} + \vec{\omega}\times(\mathbf{I}\vec{\omega}) + m\mathbf{r}_G\times(\dot{\vec{v}}_B + \vec{\omega}\times\vec{v}_B) = \sum \vec{T} \tag{3.4.4}$$

The above equations can be re-written as such:

$$\begin{bmatrix} m\mathbf{E} & -m\mathbf{r}_\mathbf{G}^\times \\ m\mathbf{r}_G^\times & \mathbf{I} \end{bmatrix}\begin{bmatrix} \dot{\vec{v}}_B \\ \dot{\vec{\omega}} \end{bmatrix} + \begin{bmatrix} m\left(\vec{\omega}\times\vec{v}_B + \vec{\omega}\times\left(\vec{\omega}\times\mathbf{r}_G\right)\right) \\ \vec{\omega}\times\left(\mathbf{I}\vec{\omega}\right) + m\mathbf{r}_G\times\left(\vec{\omega}\times\vec{v}_B\right) \end{bmatrix} = \begin{bmatrix} \sum\vec{F} \\ \sum\vec{T} \end{bmatrix} \tag{3.4.5}$$

After adding the virtual terms to the left side of Equation (3.3.5), the 6DOF nonlinear dynamic equations of motion for the airship is given as:

$$\begin{bmatrix} m\mathbf{E}+\left[\mathbf{II}\right] & -m\mathbf{r}_\mathbf{G}^\times \\ m\mathbf{r}_G^\times & \mathbf{I}+\left[\mathbf{\Gamma}\right] \end{bmatrix}\begin{bmatrix} \dot{\vec{v}}_B \\ \dot{\vec{\omega}} \end{bmatrix} + \begin{bmatrix} m\left(\vec{\omega}\times\vec{v}_B + \vec{\omega}\times\left(\vec{\omega}\times\mathbf{r}_G\right)\right) \\ \vec{\omega}\times\left(\mathbf{I}\vec{\omega}\right) + m\mathbf{r}_G\times\left(\vec{\omega}\times\vec{v}_B\right) \end{bmatrix} = \begin{bmatrix} \sum\vec{F} \\ \sum\vec{T} \end{bmatrix} \tag{3.4.6}$$

$$\begin{bmatrix} m\mathbf{E}+\left[\mathbf{II}\right] & -m\mathbf{r}_\mathbf{G}^\times \\ m\mathbf{r}_G^\times & \mathbf{I}+\left[\mathbf{\Gamma}\right] \end{bmatrix} = \begin{bmatrix} m_x & 0 & 0 & 0 & md_z & 0 \\ 0 & m_y & 0 & -md_z & 0 & md_x \\ 0 & 0 & m_z & 0 & -md_x & 0 \\ 0 & -md_z & 0 & J_x & 0 & -J_{xz} \\ md_z & 0 & -md_x & 0 & J_y & 0 \\ 0 & md_x & 0 & -J_{xz} & 0 & J_z \end{bmatrix} \tag{3.4.7}$$

Where, m is the actual mass of the airship, $\left[\mathbf{II}\right]$ is the virtual mass matrix, and $\left[\mathbf{\Gamma}\right]$ is the virtual inertial and

$$\mathbf{r}_G^{\times} = \begin{bmatrix} 0 & d_z & d_y \\ d_z & 0 & -d_x \\ -d_y & d_x & 0 \end{bmatrix} \qquad (3.4.8)$$

3.5 Airship Forces and Moments

The dynamic equations of the airship model is a 6×1 column matrix depending on the linear and angular velocities. This vector can be expressed as:

$$\vec{f}_d = \begin{bmatrix} -m_z wq + m_y rv + m(d_x(q^2+r^2) - d_y qp - d_z pr) \\ -m_x ru + m_z pw + m(d_y(p^2+r^2) - d_x qp - d_z qr) \\ -m_y pv + m_x uq + m(d_z(p^2+q^2) - d_x pr - d_y qr) \\ -J_{xy} pr + J_{xz} pq - J_{yz}(r^2-q^2) - rq(J_z - J_y) - m(d_y(pv-qu) + d_z(ru-qw)) \\ -J_{yz} pq + J_{xy} qr + J_{xz}(r^2-p^2) - pr(J_x - J_z) + m(d_x(pv-qu) - d_z(qw-rv)) \\ -J_{xz} qr + J_{yz} pr - J_{xy}(q^2-p^2) - pq(J_y - J_x) + m(-d_x(ru-pw) + d_y(qw-rv)) \end{bmatrix}$$

(3.5.1)

The gravitational force W and the buoyancy force B are two static forces that act on the airship in opposite directions; therefore, they can be combined as:

$$\vec{f}_{gb} = \begin{bmatrix} -\sin(\theta+\theta_0)(W-B) \\ \sin(\phi)\cos(\theta+\theta_0)(W-B) \\ \cos(\phi)\cos(\theta+\theta_0)(W-B) \\ -\sin(\phi)\cos(\theta+\theta_0)(d_z W + b_z B) \\ \cos(\phi)\cos(\theta+\theta_0)(d_z W + b_z B) - \sin(\theta+\theta_0)(d_x W + b_x B) \\ \sin(\phi)\cos(\theta+\theta_0)(d_x W - b_x B) \end{bmatrix} \tag{3.5.2}$$

$$B = V\rho g \tag{3.5.3}$$

$$W = B + Hg \tag{3.5.4}$$

The aerodynamic forces and moments are can be derived from the airship geometry. In this work, the aerodynamic model of the airship is adopted from ref [8]. It is a 6×1 matrix which includes elevators and rudder deflections, as shown in Figure 3.5.1, and it can be expressed as:

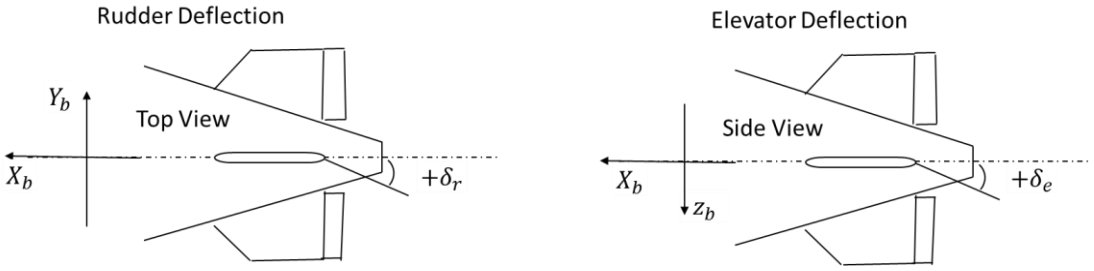


Figure 3.5.1 Airship control surfaces

$$\vec{f}_a = 0.5\rho V_T^{\,2} \begin{bmatrix} \left(C_{X1}\cos(\alpha)^2\cos(\beta)^2 + C_{X2}\sin(2\alpha)\sin(\alpha/2)\right) \\ C_{Y1}\cos(\beta/2)\sin(2\beta) + C_{Y2}\sin(2\beta) + C_{Y3}\sin(\beta)\sin(|\beta|) + 2C_{Y4}\delta_r \\ C_{Z1}\cos(\alpha/2)\sin(2\alpha) + C_{Z2}\sin(2\alpha) + C_{Z3}\sin(\alpha)\sin(|\alpha|) + 2C_{Z4}\delta_e \\ C_{l1}(\delta_r - \delta_r + \delta_e - \delta_e) + C_{l2}\sin(\beta)\sin(|\beta|) \\ C_{m1}\cos(\alpha/2)\sin\left(2\alpha\right) + C_{m2}\sin(2\alpha) + C_{m3}\sin(\alpha)\sin(|\alpha|) + 2C_{m4}\delta_e \\ C_{n1}\cos(\beta/2)\sin\left(2\beta\right) + C_{n2}\sin(2\beta) + C_{n3}\sin(\beta)\sin(|\beta|) + 2C_{n4}\delta_r \end{bmatrix}$$

(3.5.5)

$$V_T = \sqrt{u^2 + v^2 + w^2} \tag{3.5.6}$$

$$\alpha = \tan^{-1} w/u \tag{3.5.7}$$

$$\beta = \tan^{-1} v\cos(\alpha)/u \tag{3.5.8}$$

The non-dimensional aerodynamic coefficients are given in ref [8].

The propulsion system of the airship can be best described by Figure 3.5.1. It has tree thruster units, two of them are symmetrically mounted, and one is mounted in the middle and parallel to the two other thrusters. Their locations are carefully chosen in order to avoid unnecessary moments on the airship.

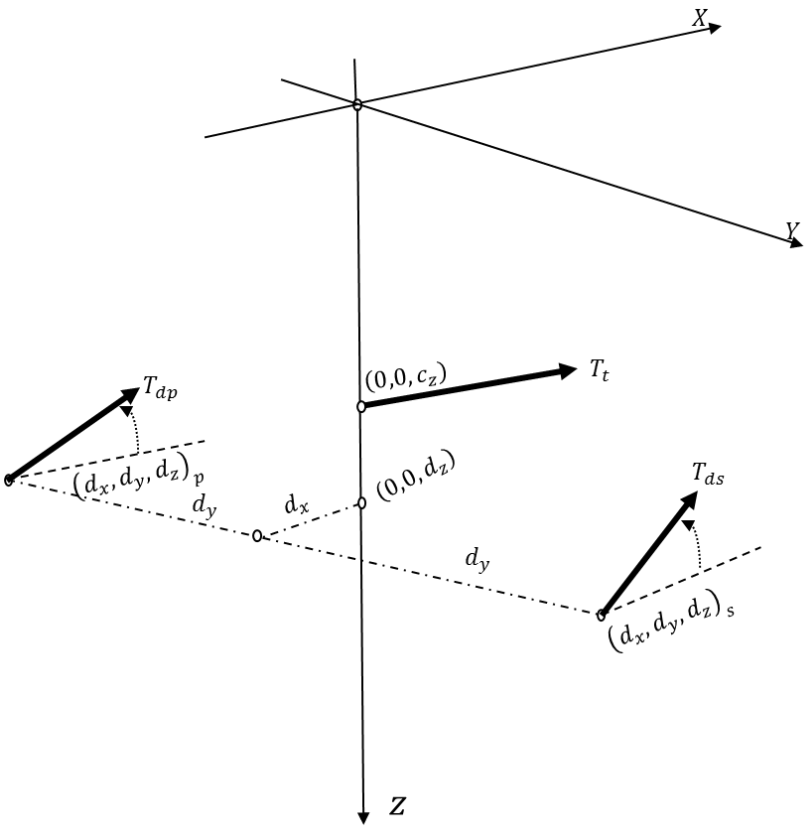


Figure 3.5.1. Airship propulsion system configuration

The propulsion forces and moments of the system can be written as a 6x1 column matrix as follows:

$$\vec{f}_p = \begin{bmatrix} T_t + (T_{ds} + T_{dp})\cos(\mu) \\ 0 \\ -(T_{ds} + T_{dp})\sin(\mu) \\ d_y(T_{dp} - T_{ds})\sin(\mu) \\ c_z T_t + (T_{ds} + T_{dp})(d_z\cos(\mu) - d_x\sin(\mu)) \\ d_y(T_{dp} - T_{ds})\sin(\mu) \end{bmatrix} \tag{3.5.9}$$

CHAPTER 4:
SIMULINK MODEL DEVELOPMENT

4.1 Simulink Model Development

In this section, the airship dynamic system is implemented in Matlab/Simulink software. Some of the Simulink blocks and results are presented. The subsystem blocks are divided in five categories as follow: dynamic block, gravitational and buoyancy block, aerodynamic block, propulsion block, and kinematic block. A picture of each of the subsystems are briefly described below.

4.2 Dynamics Simulink Block

This subsystem calculates the forces and moments acting on the airship due to normal and tangential acceleration and centrifugal effects. The Simulink block diagram is shown in Figure 4.2.1

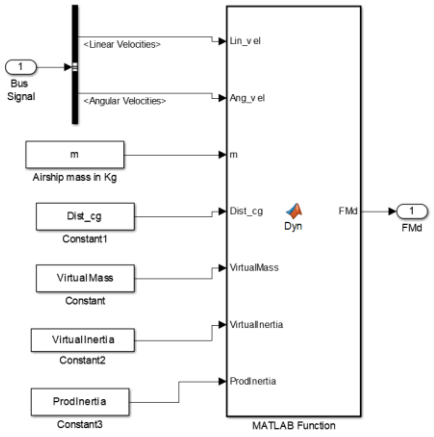


Figure 4.2.1 Dynamics block

The subsystem inputs are linear and angular velocities, mass, distance of C_g relative to C_v, virtual mass and inertial terms. Then, its output are the forces and moments in vector format.

4.3 Gravitational and Buoyancy Simulink Block

Gravity and Buoyancy Simulink block is given in Figure 4.3.1. This block determines the effects of aerostatic and gravitational force and moment have on the airship.

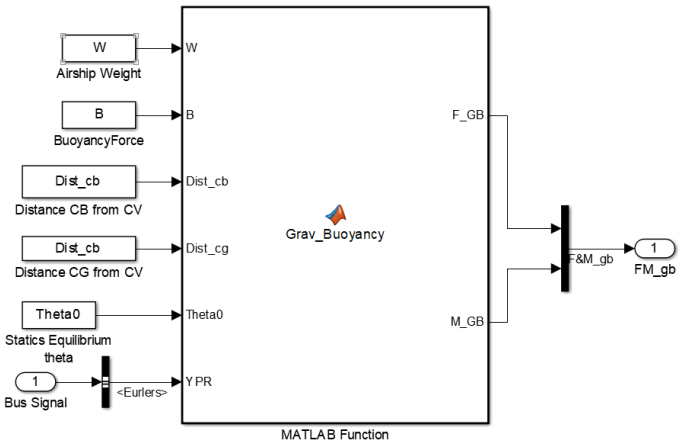


Figure 4.3.1 Gravity and buoyancy block

The input variables are the following: weight, buoyancy force, distance of C_g relative to C_v, and Euler's angles. The output of the block contains the forces and moments that are acting on the airship.

4.4 Aerodynamics Simulink Block

The aerodynamics block is shown in Figure 4.4.1. This block calculates the forces and moments acting on the airship due the change in air dynamics pressure and surface controls such as elevators and rudders.

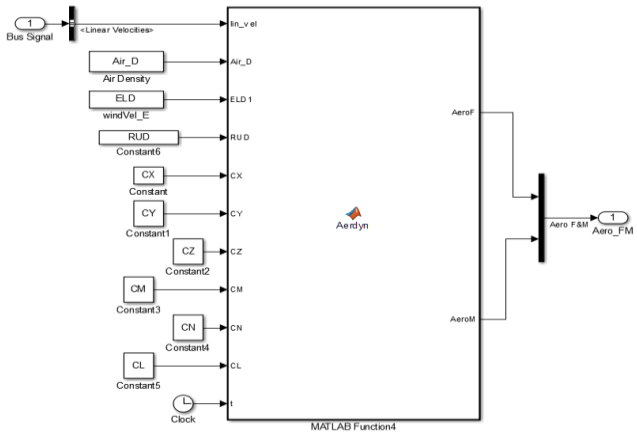


Figure 4.4.1 Aerodynamics block

The input variables are the following: air density, elevator deflection, aerodynamic coefficients, and linear velocities. All those variables are pre-calculated in the initialization Matlab script. The output of the block contains the aerodynamic forces and moments acting on the airship.

4.5 Propulsion Simulink Block

The propulsion block determines the forces and moments that are acting on the airship due the airship's thrusters. This block is shown in figure 4.5.1. The airship has three thrusters, therefore, each of them contributes to the propulsive forces and moments.

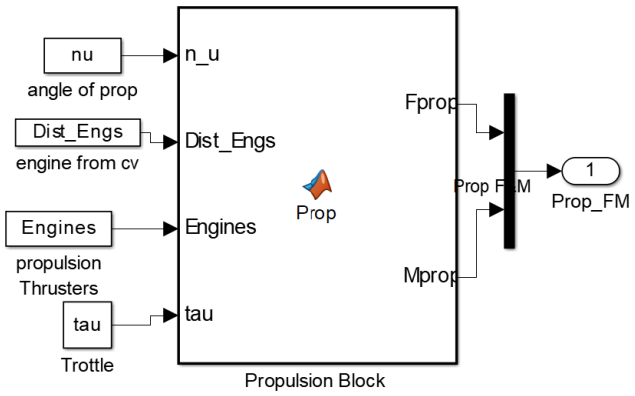


Figure 4.5.1 Propulsion block

4.6 Kinematics Position and Altitude Blocks

The kinematic blocks are shown in Figure 4.6.1. These blocks used to determine the rate of change of the airship position in inertia frame as well as the rate of change of the Euler rotational angles.

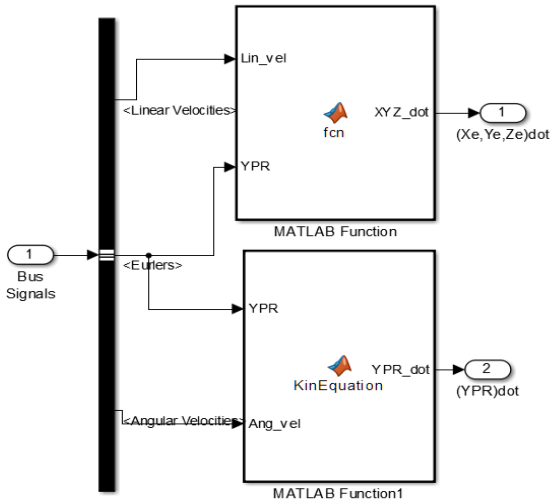


Figure 4.6.1 Kinematics block

The inputs are linear and angular velocity, and Euler angles. The outputs of the blocks are later integrated and used as a feedback signal throughout the simulation.

All the subsystem blocks presented above are integrated to form the main block diagram of the simulation as show in Figure 4.6.2.

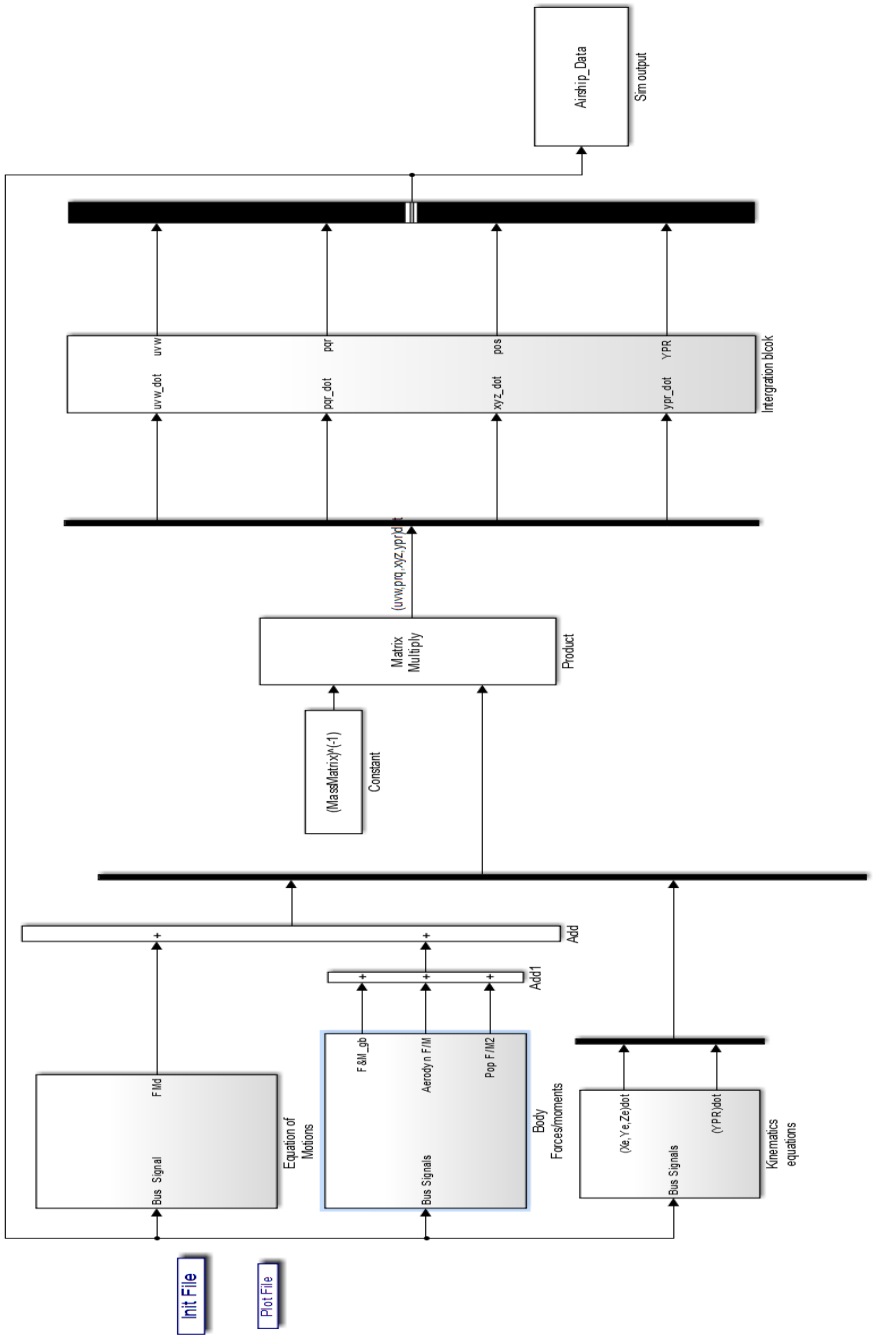


Figure 4.6.2 Simulation block

The Simulink model of the airship is presented in Figure 4.6.2; it contains all the subsystems shown previously. The mass matrix is calculated in the initialization Matlab script and called by Matlab/Simulink software upon running. All the airship data is saved in Matlab workspace environment using SimOutput Simulink block. A Matlab script extracted all the data from SimOutput and plot them automatically. To setup the simulation, all parameters are calculated from the initialization Matlab script including model geometry parameters, aerodynamic coefficients, and mass matrix. The thrusters are assumed to be symmetrically and synchronously control. In addition, they are assumed to be constant and aligned with the airship longitudinal $x-axis$. Neutral gravitational and buoyancy effects is assumed to be zero. During model validation, the running time of the simulation is 500 seconds with an initial flight speed of $20\ m/s$ and an altitude of $21km$. Rudder and elevator deflection angles are set to a constant variable as well as the thrusters' angle. The airship responses to the input have been analyzed and plotted.

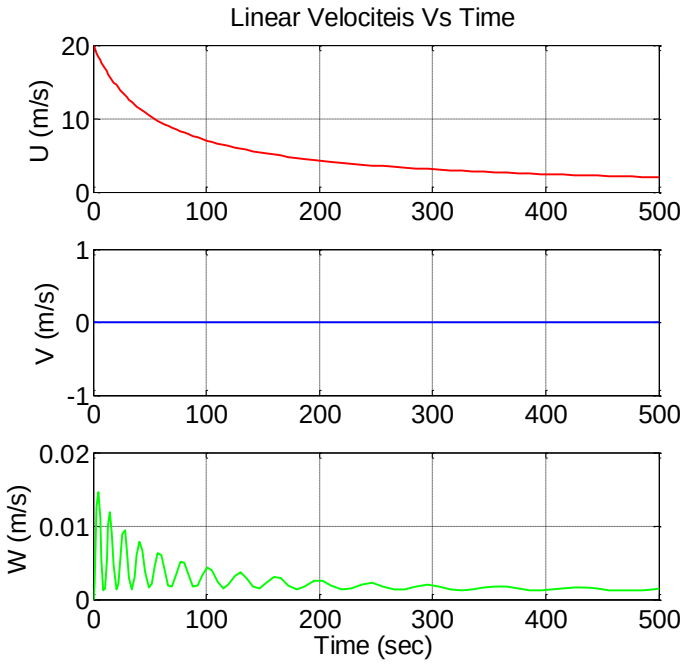


Figure 4.6.3 Linear velocities

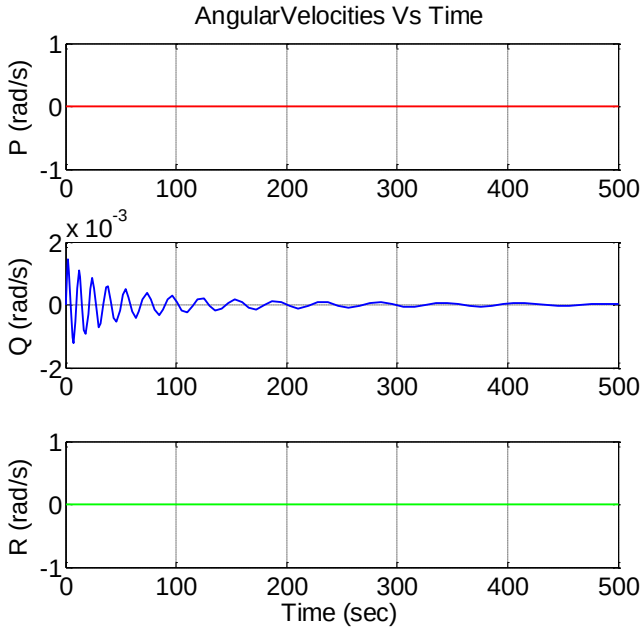


Figure 4.6.4 Angular velocities

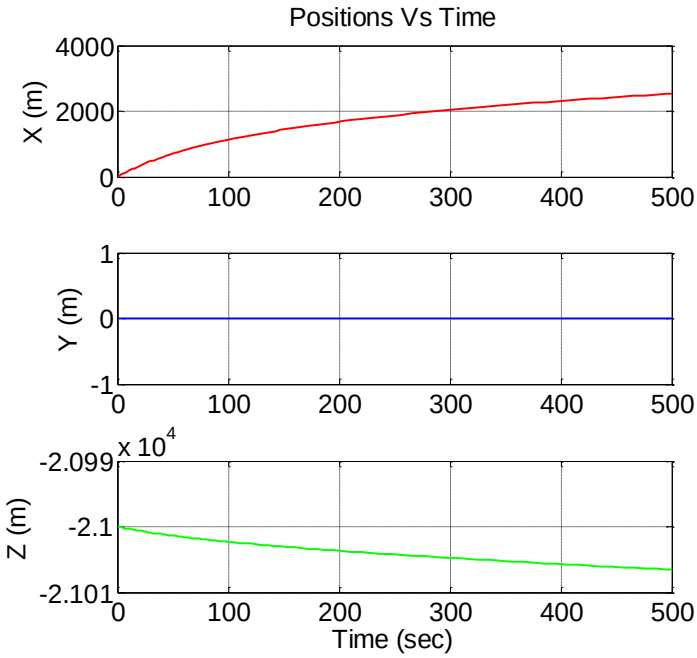


Figure 4.6.5 Position

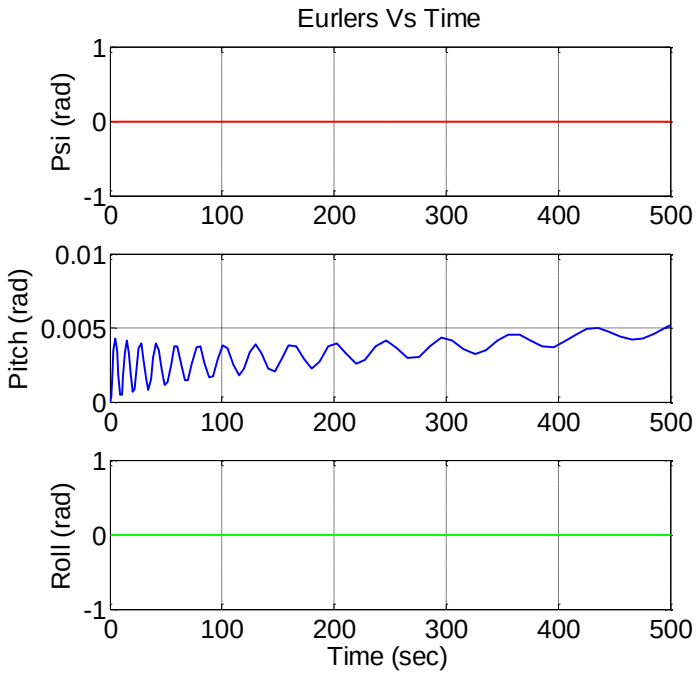


Figure 4.6.6 Euler

Figure 4.6.2 to Figure 4.6.6 represent the exact dynamic behavior of the airship. As shown in the figures above, there is only translational motions and no lateral motion. The results demonstrate that the airship is quite stable. However, there is a small velocity along the $z-axis$ and the time taken for the transient to settle is about 300 seconds. Figure 4.6.4 show there is a very small oscillatory motion in the pitch angle and it appears as well in the angular velocity along the $y-axis$. The magnitude of the oscillatory motion is so small therefore it can be neglected.

CHAPTER 5:
STRAIGHT LEVEL FLIGHT

5.1 Condition for Trim Flight

Trimmed flight condition can be defined as the ability to maintain a level flight with fixed control [12], [13]. To trim the airship model, the sum of all the forces and moments acting on the airship should be zero [8]. This includes contributions from all external forces. In addition, symmetric elevator deflections, throttle, and propeller pitch angles are needed to establish level flight. To find the trim condition of the airship, the rate of change of the states are set to zero as follow:

Table 5.1.1 Trim initial condition

$\dot{u} = 0$	$\dot{p} = 0$
$\dot{v} = 0$	$\dot{q} = 0$
$\dot{w} = 0$	$\dot{r} = 0$

5.2 Steady straight flight

For the steady-state flight condition, the velocity along the $x-axis$ is set to a constant and as well as the airship's altitude. Table 5.4.1. shows the rest of the states.

Table 5.2.1 Initial condition

$u = 5\ m/s$	$p = 0$	$\phi = 0$
$v = 0$	$q = 0$	$\theta = \theta_0$
$w = 0$	$r = 0$	$\psi = 0$

Matlab/Simulink model presented above is used to simulate a straight level flight for the airship and the results are plotted. As it was expected, the airship maintains its linear velocity along the $x-axis$ and its altitude as shown in Figure 5.2.1. The reason for that is the assumption made while solving for the trimmed condition. Some of those assumptions are the following: drag is equal to the thrust and net lift is equal to the weight. On the other hand, Figure 5.2.1 shows that angular velocities and Euler's angles stayed zero throughout the flight, and that was expected because one of the features of straight and level flight is that the sum of the net forces and moments should be zero.

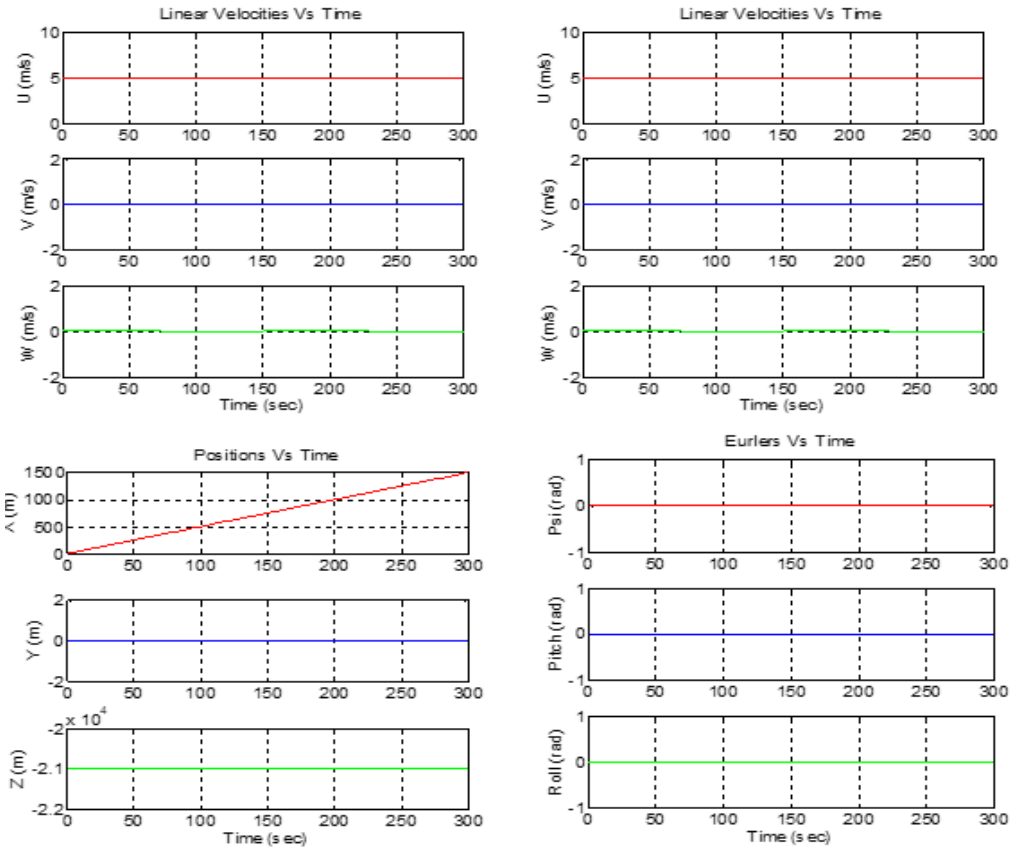


Figure 5.2.1 Straight level flight

CHAPTER 6:
DIRECT COLLOCATION AND NONLINEAR PROGRAMMING

6.1 Nonlinear Direct Collocation

Direct collocation with nonlinear programming can be used to solve any type of nonlinear optimal control problem. It provides a numerical approximated solution to the problem by discretizing the states and controls into a set of equal nodes while ensuring the problem constraints and the equations of motion are satisfied at the collocation points [14].

6.2 Problem Formulations

A typical optimal control problem can be expressed by a system of dynamic variables consisting of the state and the control variables [15].

$$\vec{x}(t) = \left[u, v, w, p, q, r, x_i, y_i, z_i, \phi, \theta, \psi\right]^T \tag{6.2.1}$$

$$\vec{u}(t) = \left[T_{dp}, T_t, T_{ds}, \delta_e, \delta_r\right]^T \tag{6.2.2}$$

the nonlinear dynamic system can be represented as a first order model as:

$$\dot{\vec{x}}(t) = f(\vec{x}(t)) + B\vec{u}(t) \tag{6.2.3}$$

Where,

$$\vec{u}(t) = B^{-1}(\dot{\vec{x}}(t) - f(\vec{x}(t))) \tag{6.2.4}$$

$$B^{-1} = (B^T B)^{-1} B^T \tag{6.2.5}$$

The optimization problem statement can be summarized:

$$J = \int_{t_0}^{t_f} \vec{c}^{\,2}(t)dt \tag{6.2.6}$$

$$\vec{c}(t) = [T_{dp}, T_t, T_{ds}]^T \tag{6.2.7}$$

Subject to the nonlinear dynamic equation

$$\dot{\vec{x}}(t) = f(\vec{x}(t)) + B\vec{u}(t) \tag{6.2.8}$$

the end-point conditions

$$\xi_{L_0} \le \xi_0(\vec{x}(t_0), t_0) \le \xi_{U_0} \tag{6.2.9}$$

$$\xi_{L_f} \le \xi_f(\vec{x}(t_f), t_f) \le \xi_{U_f} \tag{6.2.10}$$

the mixed state-control path constraints

$$g_{lb} \le g(\vec{x}(t), \vec{u}(t), t) \le g_{up} \tag{6.2.11}$$

and box constraints

$$x_{lb} \le \vec{x}(t) \le x_{up} \tag{6.2.12}$$

$$u_{lb} \le \vec{u}(t) \le u_{up} \tag{6.2.13}$$

$$c(x) = 0 \tag{6.2.14}$$

$$c(x) \le 0 \tag{6.2.15}$$

6.3 Discretization Methods

The airship optimal control problem is numerically solved by applying the Sequential Quadratic Programming method in Matlab using fmincon optimization algorithm. The two discretization methods are Trapezoid and Hermite-Simpson. The trapezoid method is used to integrate the dynamic equations. By applying this method, dynamic and control are assumed to

be linear between grid points. The decision variables for the trapezoid rule are the states and the controls at $N+1$ node points [14]. The cost function is approximated as such:

$$J \cong M(x_N, t_N) + \sum_{k=0}^{N-1} \frac{h_k}{2} \left\{ L(x_k, u_k) + L(x_{k+1}, u_{k+1}) \right\} \tag{6.3.1}$$

Where,

$$M(x_N, t_N) = 0 \tag{6.3.2}$$

$$L(u_k) = \frac{1}{2} \sum_{k=0}^{N-1} \vec{u}_k^2 \tag{6.3.3}$$

$$h_k = \frac{t_f - t_0}{N} \tag{6.3.4}$$

the cost function can be re-written as such.

$$J \cong \sum_{k=0}^{N-1} \frac{h_k}{2} \left\{ L(u_k) + L(u_{k+1}) \right\} \tag{6.3.5}$$

The second discretization method is the Hermite-Simpson method. The decision variable for the Hermite Simpson rule are the states and controls at $N+1$ node points [14]. And by using this method, dynamic and control are assumed to be quadratic between grid points. The cost function is approximated as such:

$$J \cong M(x_N, t_N) + \sum_{k=0}^{N-1} \frac{h_k}{6} \left\{ L(x_k, u_k) + 4L(y_k, v_k) + L(x_{k+1}, u_{k+1}) \right\}, \quad k = 0, ..., N-1 \tag{6.3.6}$$

Where,

$$M(x_N, t_N) = 0 \tag{6.3.7}$$

$$L(u_k) = \frac{1}{2}\sum_{k=0}^{N-1} \vec{u}_k^2 \tag{6.3.8}$$

$$L_c(u_k) = \frac{1}{2}\sum_{k=0}^{N-1} \vec{v}_k^2 \tag{6.3.9}$$

$$v_k = \frac{(u_k + u_{k+1})}{2} \tag{6.3.10}$$

then, the cost function equation can be re-written as:

$$J \cong \sum_{k=0}^{N-1} \frac{h_k}{6}\left\{L(u_k) + 4L_c(v_k) + L(u_{k+1})\right\} \tag{6.3.11}$$

The inequality constraints used in simulation can be found in Table 6.5.1 below.

Table 6.3.1 Inequality constraints

Inequality constraints		
$[-25,-25,-5]m/s$	$\leq [\dot{u},\dot{v},\dot{w}] \leq$	$[25,25,5]m/s$
$[-20,-20,-20]°/s$	$\leq [\dot{p},\dot{q},\dot{r}] \leq$	$[20,20,20]°/s$
$[-200,-200,-10]m$	$\leq [\dot{x}_i,\dot{y}_i,\dot{z}_i] \leq$	$[200,200,10]m$
$[-30,-30,-30]°$	$\leq [\dot{\phi},\dot{\theta},\dot{\psi}] \leq$	$[30°,30°,30°]$
$-30°$	$\leq [\delta_\epsilon] \leq$	$30°$
$-30°$	$\leq [\delta_r] \leq$	$30°$
$[0,0,0]N$	$\leq [T_{dp}, T_{\mathrm{t}}, T_{\mathrm{ds}}] \leq$	$[10E5,10E5,10E5]N$

CHAPTER 7:
SIMULATION RESULTS

7.1 Results Analysis

This section presents the simulation results that illustrate the effectiveness of the direct collocation method using different discretization methods [19]. The trajectory optimization problem was numerically solved by a standard solver for NLP like the Sequential Quadratic Programming method in Matlab using fmincon [15]. The Matlab codes were divided into three main scripts. The first script was the main script that calls the other Matlab scripts. Lower and upper bounds as well as initial guess variables are defined in the main script. The second Matlab script contains the cost function equation and the controls variable calculations. And the last Matlab script contains all the inequality and equality constraints. A complete flow chart of the simulation is described as shown Figure 7.1.1.

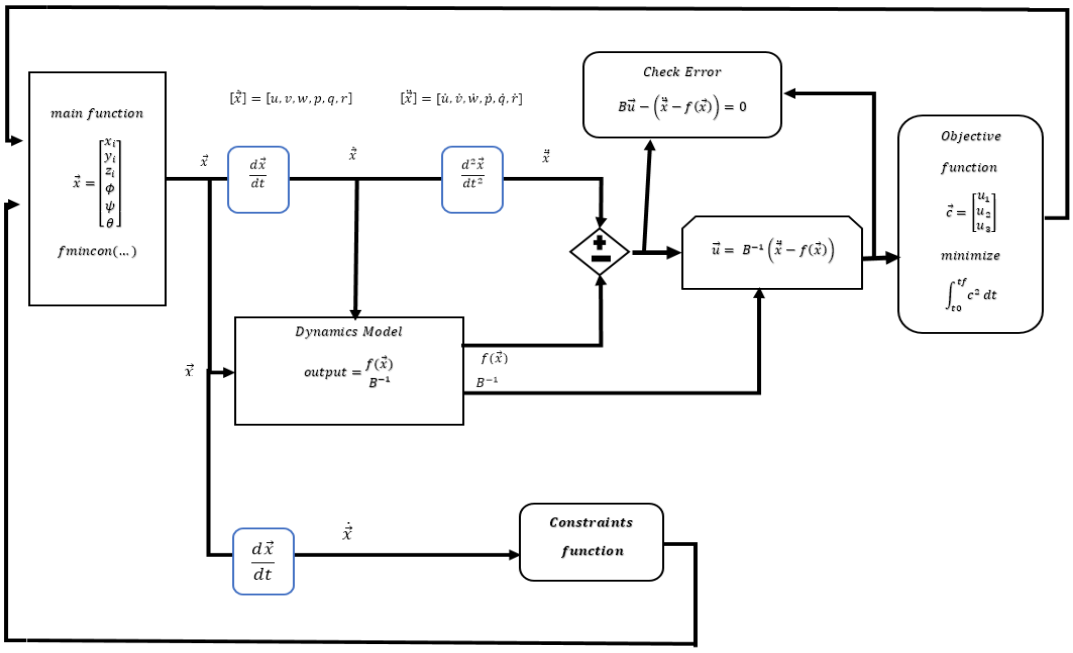


Figure 7.1.1 Simulation flow chart

As shown in Figure 7.1.1, positions and Euler angles are the guessing state variables that are used to find the other six state variables. The positions and the Euler angles are differentiated using a differentiation matrix.

The geometry parameters of the airship are listed in Table 7.1.1 and they can be obtained from ref [8].

Table 7.1.1 Airship model parameters

Parameter	*Value*	*Unit*	*Parameter*	*Value*	*Unit*
m	55749.7	kg	ρ	0.072157	$\mathrm{kg/m^3}$
V	736311	m^3	I_{xx}	$5x10^7$	$\mathrm{kg \cdot m^2}$
L	250	m	I_{yy}	$2.9x10^8$	$\mathrm{kg \cdot m^2}$
D	75	m	I_{zz}	$2.9x10^8$	$\mathrm{kg \cdot m^2}$
Ref Area	$8.15x10^3$	m^2	I_{xz}	$-6x10^4$	$\mathrm{kg \cdot m^2}$
$\boldsymbol{z_c}$	15	m			

Table 7.1.2 Simulation conditions

States variable	*Initial value*	*Final value*
$[u,\ v, w]$	$[0,20,0]m/s$	$[0,20,0]m/s$
$[p, q, r]$	$[0°, 0°, 0°]$	$[0°, 0°, 0°]$
$[x_i, y_i, z_i]$	$[100,0,0]m$	$[100,0,0]m$
$[\phi, \theta, \psi]$	$[0°, 0°, 0°]$	$[0°, 0°, 0°]$

Table 7.1.2 contains the initial and final values for the simulation. Equation 7.1.1, represent the periodic equation of the reference trajectory.

$$\begin{aligned} x_i &= 100\cos(\xi) \\ y_i &= 100\sin(\xi) \\ z_i &= 0 \end{aligned} \tag{7.1.1}$$

Where,

$$\xi = \frac{2\pi(i-1)}{N} \qquad i = 1,\ldots,N+1 \tag{7.1.2}$$

During the simulation, the airship optimal control problem is discretized over multiple number of nodes. Matlab fmincon optimization algorithm is used to compute the optimal trajectories for various sets of initial conditions [19]. The optimized results are plotted against the initial guess variables. In Figure 7.1.2 to Figure 7.1.5, the simulation result for 40 nodes.

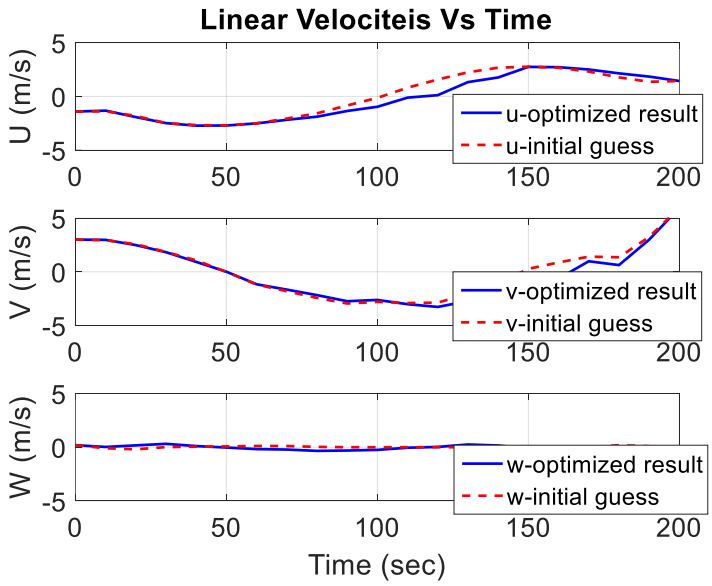


Figure 7.1.2 Linear velocities

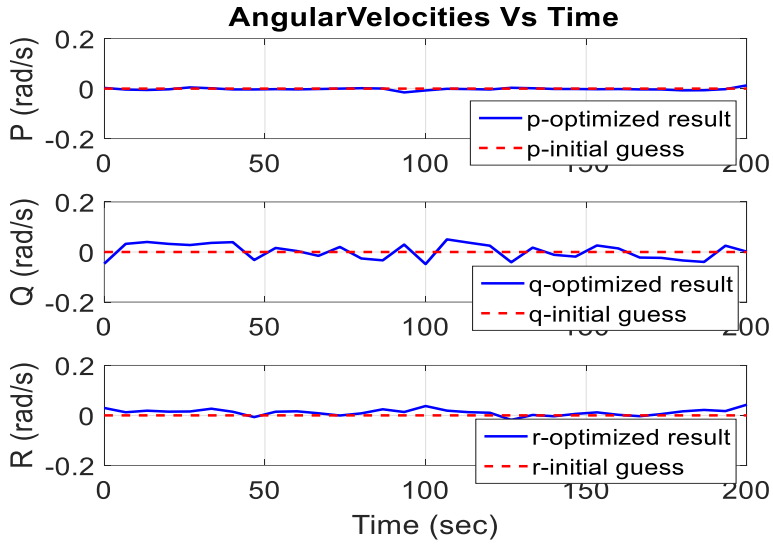


Figure 7.1.3 Angular velocities

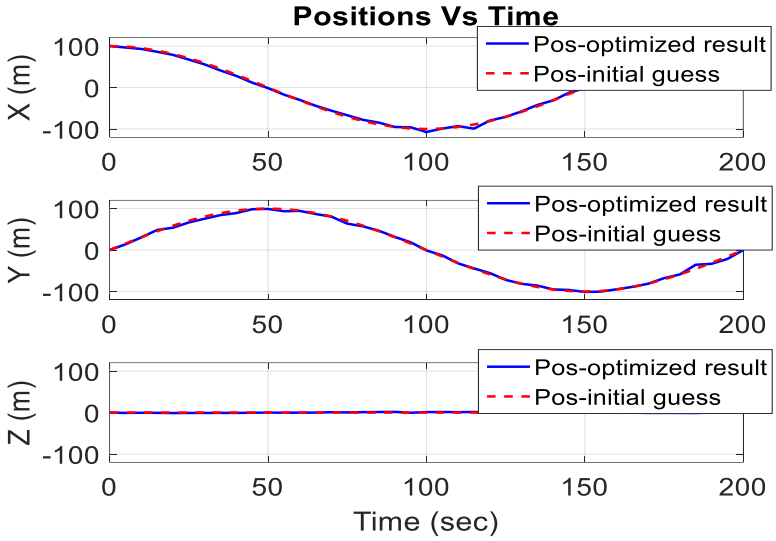


Figure 7.1.4 Positions

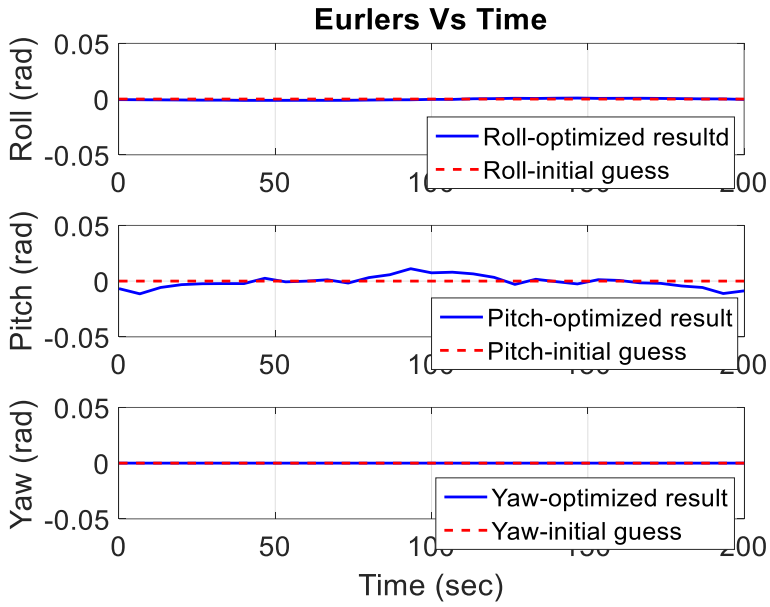
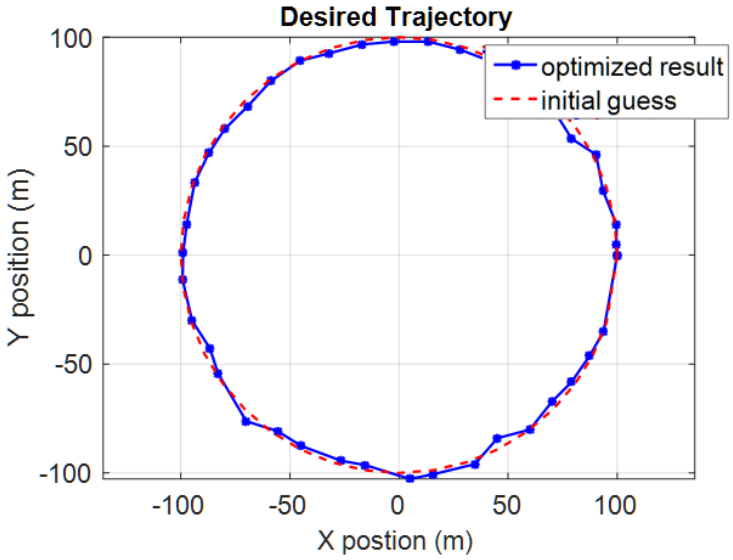


Figure 7.1.5 Euler angles



0..

Figure 7.1.6 Airship trajectory planning

In Figure 7.1.2, the linear velocity in the x direction has some small spikes and this is due to the numerical differentiation matrix used to approximate the derivative of the discrete state variables. From Figure 7.1.5 it can be seen that pitching angle is very small and this is due to the

fact that the airship is following a level circular path. Figure 7.1.6 the airship tracks very well the reference periodic motion of the initial guess.

Table 7.1.3 Performance index history

	Hermite-Simpson method		***Trapezoid Method***	
Nodes	*Performance Index*	*CPU Time(s)*	*Performance Index*	*CPU Time(s)*
15	1.081753e+03	9.247897	3.555666e+04	13.206917
20	2.751440e+02	21.022873	1.653565e+03	38.022367
30	2.66094e+02	34.305048	4.718256e+02	56.151667
40	1.380984e+02	69.575394	2.533452e+02	102.444070
50	1.197780e+02	80.940285	2.093050e+02	138.101372
60	5.236397e+01	97.157609		

The simulation performance index and CPU time are collected and presented in Table 7.1.3. Both discretization methods converge to a solution for various node numbers [16], [17], [18]. For trapezoid method, performance index decreases as the node number increases while CPU time keeps increasing. That is expected because as the node number increases, the step size become smaller, therefore, it takes longer for the simulation to converge to a solution. For Hermite-Simpson discretization method, simulation displayed the same characteristic as the Trapezoid method. However, there is big difference in CPU time between these two methods. For each node number, it takes the simulation longer to converge when using Trapezoid method than Hermite-Simpson method. It is very evident from Table 7.1.3 that Hermite-Simpson discretization method gives better approximated solution to the problem. That was expected because Hermite-Simpson is a high order discretization method than Trapezoid. From node

number 60 and higher, the trajectory optimization problem converges with a reasonable solution for the Hermite-Simpson method while fails for the Trapezoid method. CPU time for both methods indicate that the simulation is very slow as was expected because direct collocation method is very slow and difficult to converge to a solution if not well constrained and tuned.

CHAPTER 8: CONCLUSION

In this paper, direct collocation method is used with different discretization methods to solve a trajectory optimization problem. The algorithm converts the optimal control problem into nonlinear programming problem. the cost function is subjected to the dynamic and path constraints as well some inequality constraints. Different node numbers are used during the simulation and computed results are presented in the paper. This work shows that it is feasible to apply direct collocation method to solve airship optimal trajectory problem using different discretization methods. Both discretization methods are used to obtain solutions for the problem, however, Hermite-Simpson discretization method gives a better approximated solution as expected. A higher order discretization method can be used to better approximate the solution of the airship optimal control problem and that will be study in the near future.

LIST OF REFERENCES

[1] Y. Li, M. Nahon, and I. Sharf, "Airship Dynamics Modeling: A Literature Review," *Progress in Aerospace Sciences, Elsevier,* vol. 47, no. 3, pp. 217-239, 2010.

[2] Avenant, G., "Autonomous Flight Control System for an Airship," Stellenbosch University, Stellenbosch, South Africa, 2010.

[3] Nakpiam, J., "Control of Airship Motion in the Presence of Wind," Texas University, Arlington, TX, 2007.

[4] Williams, P., "A Gauss-Lobatto Quadrature Method for Solving Optimal Control Problems," *Australian and New Zealand Industrial and Applied Mathematics Journal,* Vol. 47, 2006, pp. 101–115.

[5] S. Hima and Y. Bestaoui, "Motion Generation on Trim Trajectories for an Autonomous Anderactuated Airship," In *4th International Airship Convention and Exhibition*, Cambridge England, July 2002.

[6] Jon N. Ostler, W. Jerry Bowman, Deryl O. Snyder, and Timothy W. McLain, "Performance Flight Testing of Small, Electric Powered Unmanned Aerial Vehicles,". *International Journal of Micro Air Vehicles,* Volume 1, Number 3, 2009.

[7] Kornienko, A., "System Identification Approach for Determining Flight Dynamical Characteristics of an Airship from Flight Data," Ph.D. thesis, University of Stuttgart, 2006.

[8] J. B. Mueller and Y. J. Zhao, "Development of an Aerodynamic Model and Control Law Design for a High-Altitude Airship," In *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit,* number AIAA-6479, Chicago, IL, 2004. AIAA.

[9] A.J.J. Lemmers and A.P.L.A. Marsma, “A Basic Flight Simulation Tool for Rigid Airship,” *Report, National Aerospace Laboratory NLR,* Aug 2000.

[10] H. Lamb, “The Inertia-Coefficients of an Ellipsoid Moving in Fluid,” *Reports and Memoranda,* no. 623, pp. 128-129, 1918.

[11] Cook, M. V., “The Linearized Small Perturbation Equations of Motion for an Airship,” *Cranfield report,* UK, (1990).

[12] Acanfora, M., “New Approach and Results on the Stability and Control of Airship,” PhD thesis, Universita’Deglistudi di Napoli,2011.

[13] De Marco, A., L. Duke, E., Berndt, J. S., “A General Solution to the Aircraft Trim Problem,” *American Institute of Aeronautics and Astronautics* AIAA-2007-6703

[14] Betts, J. T., “Practical Methods for Optimal Control Using Nonlinear Programming,” *Industrial and Applied Mathematics*.

[15] Paul Williams, “Direct Approaches for Solving Optimal Control Problems in Matlab,” *RMIT University,* Melbourne, Australia

[16] J.T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of Guidance, Control, and Dynamics,* 21(2):193-207, 1998.

[17] D. Hull, “Conversion of Optimal Control Problems into Parameter Optimization Problems,” *Journal of Guidance, Control, and Dynamics*, 20(1):57-60, 1997.

[18] A.L. Herman and B.A. Conway, “Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules,” *Journal of Guidance, Control, and Dynamics,* 19:592-599, 1996.

[19] Yueming Hu. "Optimal Path Planning for Autonomous Airship Based on Clonal Selection and Direct Collocation Algorithm," *2008 IEEE International Conference on Networking Sensing and Control*, 04/2008.

[20] Albert L. Herman* and Bruce A. Conway, “Direct Optimization Using Collocation Based on High-Order Gauss-Lobatto Quadrature Rules,” *Journal of Guidance, Control, and Dynamics Vol. 19. No. 3*, May-June 1996.