AUTONOMOUS HOPPING ROTOCHUTE

A Thesis
Presented to
The Academic Faculty

by

Derya Aksaray

In Partial Fulfillment
of the Requirements for the Degree
MASTER OF SCIENCE in the
School of AEROSPACE ENGINEERING

Georgia Institute of Technology
May 2011

AUTONOMOUS HOPPING ROTOCHUTE

Approved by:

Dr. Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Mark Costello, Co-Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Stephane Dufresne
Aerospace Systems Design Laboratory
Georgia Institute of Technology

Date Approved: March 28, 2011

Dedicated to My Beloved Family…

ACKNOWLEDGEMENTS

I would first like to express my deepest sincere gratitude to my advisor, Dr. Dimitri Mavris, and my co-advisor, Dr. Mark Costello, both of whom provided me their invaluable knowledge and experience. I am very grateful to be a student of Dr. Mavris, who was always able to point me in the right direction. His patience and understanding are greatly appreciated. I also feel very privileged to have worked with Dr. Costello, who created an excellent environment for my research.

I am very thankful to Dr. Stephane Dufresne for his continuous support and helpful feedback throughout the development of this work. I am also very grateful to the help of Jane Chisholm. She not only revised my thesis but also taught me many things regarding the progress of the writing of my thesis. I thank the students in "Costello's Research Group," including Jessica Newman, Kyle French, and Sam Zarovy. I also would like to thank my friends at Georgia Tech who provided valuable interaction. I appreciate the feedback provided by my friends, Jonathan Sands and Andrew Turner, from ASDL.

In particular, I would like to thank Ahmet Yasin Yazıcıoğlu, who provided fruitful discussions and continuous support from the beginning to the final stages of this work.

Finally, I would like to express my deepest gratitude to my mother and my father for their constant love, patience, and encouragement; my brother and my sister-in-law for their love and support. Although we are very far from each other, I always feel their steadfast encouragement to continue on my path. I couldn't have succeeded without them. Thank you.

Derya Aksaray

Georgia Institute of Technology

March, 2011

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

x, y, z	Components of mass center position vector with respect to the inertial frame
u, v, w	Components of mass center velocity vector with respect to the body frame
p, q ,r	Components of mass center angular velocity with respect to the body frame
φ, θ, ψ	Euler roll, pitch and yaw angles of Hopping Rotochute
T_{IB}	Transformation matrix from body reference plane to inertial reference frame
T_{BI}	Transformation matrix from inertial reference plane to body reference frame
t_{app}	Pulse width, the time duration of the rotor system
$\Delta x, \Delta y$	Components of a desired vector in x and y coordinate system
η, υ, ω	Random values expressing the uncertainties
σ	Standard deviation
ε	Hop error
$\bar{\varepsilon}$	Simulation error
h	Half width

UAV	Unmanned Aerial Vehicle
FLC	Fuzzy Logic Control
MPC	Model Predictive Control
LMPC	Linear Model Predictive Control
NMPC	Nonlinear Model Predictive Control
IM	Internal mass

SUMMARY

The Hopping Rotochute is a promising micro vehicle with the capability of exploring rough and complex terrains with minimum energy consumption. While it is able to fly over obstacles via thrust produced by its coaxial rotor, its physical architecture, inspired from a “Weebles Wooble,” provides re-orientation wherever it hits the ground. Therefore, this aerial and ground vehicle represents a potential hybrid vehicle capable of reconnaissance and surveillance missions in complex environments.

The most recent version of the Hopping Rotochute is manually controlled to follow a trajectory. The control commands, listed in a file prior to the particular mission, are executed exactly as defined, like a “batch job,” regardless of the uncertain external events. This control scheme is likely to cause great deviations from the route. Consequently, the vehicle may finish the mission very far away from the desired end point. However, if a vehicle is capable of receiving the control commands during a mission, “interactive processing” can be realized and efficient path tracking would be achieved. Hence, the development of the Hopping Rotochute that follows a trajectory autonomously reveals the foundation of this thesis.

Two control approaches inspired the proposed methodology for developing an autonomous trajectory-following algorithm. The first approach is rule-based control that enables decision making through conditional statements. In this thesis, rule-based control is used to select a target point for a particular hop based on the existence of an obstacle and/or wind in the environment. The second approach is model predictive control employed to predict future outputs from *hop performance models*. In other words, this technique approaches the problem by providing intelligence pertaining to how a

particular hop will end up before being attempted. Hence, the optimum control commands are selected based on the predicted performance of a particular hop.

This research demonstrates that the autonomous Hopping Rotochute can be realized by rule-based control embedded with some performance models. In the assumption of known boundaries such as wall and ceiling information, this study has two aims: (1) to avoid obstacles by creating a smaller operational volume inside the real boundaries so that the vehicle is restricted from exiting the operational volume and no violation occurs within the real boundaries; (2) to estimate the wind by previous hops to select the next hopping point with respect to the estimated wind information. Based on the developed methodology, simulations are conducted for four different scenarios in the existence of obstacles and/or wind, and the results of the simulations are analyzed. Finally, based on the statistics of simulation results, the effectiveness of the proposed methodology is discussed.

CHAPTER 1

INTRODUCTION

Today, the satellites and the unmanned air vehicles (UAVs) like Global Hawk [1] are playing an important role for acquiring macro level intelligence from high altitudes. However, they are not very efficient at urban and indoor environments in order to obtain micro level intelligence. In contrast to the satellites and the UAVs, the micro vehicles have the potential to provide the local reconnaissance for an individual soldier in the field. Thus, they are proposed to keep personnel out of harms way by providing novel situational awareness, which was emphasized in the Defense Science Board's 1996 Summer Study on "Tactics and Technologies for 21st Century Warfighting" [3]. Therefore, due to their capability of providing local information, the micro vehicles are potentially more capable concepts for the urban and indoor environments.

Recently, many micro vehicle studies are present in the literature. There are the fixed wing micro vehicles, the rotary wing micro vehicles, the flapping wing micro vehicles, micro ground vehicles…etc. Among them, there is especially one with a very unique design called the Hopping Rotochute. This hybrid micro vehicle is capable of flying over the obstacles, and sustains its upright position on the ground. Hence, it poses an efficient concept for the missions in complex terrains such as inside of caves and damaged buildings.

Like the other flying micro vehicles, the Hopping Rotochute is also susceptible to the external events during its operation. It tends to deviate from its route when it encounters wind or any uncertainty from the environment, which leads to hop to an undesired position. Unfortunately, hopping to the undesired position induces more crucial problems since the most recent Hopping Rotochute operates with some control commands planned prior to mission without considering uncertainty. From a mission point of view, this situation can lead to failure due to diverging from the route at each

hop. Hence, the Hopping Rotochute should be developed in terms of autonomous trajectory-following in which each control command for a particular hop will be calculated right before the hop is conducted.

This chapter starts with an overview of various micro vehicle concepts existing in the literature. Then the essential problems of these miniature vehicles are discussed in terms of guidance and control. Finally, the research objectives are introduced, and the outline of the thesis is presented including the short summaries of each chapter.

1.1 Micro Vehicle Concepts

The world of micro vehicle concepts contains diverse types of configurations under varying stages of design, development and testing [4]. This section will discuss specific characteristics of fixed wing, rotary wing, flapping wing and ground vehicles. Finally, a new concept called hybrid vehicle will be introduced.

Figure 1: Micro vehicle families [2]

Recently, the most common types of micro vehicles are the fixed wing, rotary wing, flapping wing and ground vehicles. Figure 1 from reference [2] illustrates some examples from each family. The configuration selection for a specific mission, either civilian or military, is based on the capabilities of the vehicles. Therefore, the following

paragraphs will briefly explain the properties of each configuration. Before starting to the explanations of each configuration, Table 1 from reference [6] is introduced. This table displays a qualitative comparison among various concepts. In this qualitative mapping, while scale of 1 indicates a bad performance, scale of 3 represents a good performance. For instance, when a micro helicopter and a micro airplane are compared with respect to stationary flight, a micro helicopter exhibits a better performance than a micro airplane. On the other hand, if the same vehicles are compared with respect to payload/volume, then a micro airplane displays a better performance than a micro helicopter. Consequently, the aim of Table 1 is to present a qualitative comparison of various vehicles with respect to different criterion. Qualitative scores are added up at the end, and the total scores are displayed in the last row of Table 1.

Table 1: Qualitative comparison of miniaturization of various concepts [6]

FLYING PRINCIPLES COMPARISON FOCUSED ON ABILITY TO MINIATURIZATION. (1=BAD, 3=GOOD)

	Airplane	Helicopter	Bird	Autogiro	Blimp
Power cost	2	1	1	2	3
Control cost	2	1	1	2	3
Payload/volume	3	2	2	2	1
Maneuverability	2	3	3	2	1
DOF	1	3	3	2	1
Stationary flight	1	3	2	1	3
Low speed fly	1	3	2	2	3
Vulnerability	2	2	3	2	2
VTOL	1	3	2	1	3
Endurance	2	1	2	1	3
Miniaturization	2	3	3	2	1
Indoor usage	1	3	2	1	2
Total	20	28	26	20	26

The fixed wing micro vehicles tend to have large payload capabilities observed from “payload/volume” criteria of Table 1 with the scale of 3. Their mission ranges vary from 500 m to 1 km while their endurance is at least 10 minutes [5]. Moreover, they are more capable to operate in gusty weathers when they are compared with the other flying micro vehicles. Thus, they have wide application areas. However, they are generally incapable of operating in narrow halls. In addition, beside some particular kinds, they are

usually not efficient for low speed flight or hovering, which is represented as the scale of 1 in “low speed fly” and “stationary flight” criterion of Table 1.

The rotary wing micro vehicles can hover and fly at low speeds. Compared with the fixed wing micro vehicles, they have less payload and endurance capabilities. However, they still exhibit various benefits displayed in Table 1. Based on the qualitative comparison of criterion shown in this table, a micro helicopter presents the best performance with a total score of 28. (Recall that this score depends on listed criterion with equal weightings. For instance, if endurance or power cost has more priority in a mission, then a micro helicopter will most likely display a worse total score since it has the score of “1” for both of the criterion.)

The flapping wing micro vehicles are biologically inspired, which increases their stealth due to their more natural look. Like the rotary wing micro vehicles, they have the capability of vertical, stationary and low speed flight. Nonetheless, they are not as efficient as the fixed wing micro vehicles in terms of range and endurance.

Unlike the micro aerial vehicles, the micro ground vehicles have longer endurances and larger payload capacities. They are used for wider range of missions such as surveillance, inspection, manipulation, etc. Nevertheless, when they encounter a large obstacle or a hole, they may get trapped.

As seen, each family of the micro vehicles has particular advantages and disadvantages. In order to increase the capabilities of micro vehicle concepts, another family is introduced as the “hybrid micro vehicles”. In this family, the concepts exhibit the combination of one or more physical phenomena, implying the improved capabilities as well as some weaknesses due to the combination of two or more vehicle concepts.

One of the popular hybrid vehicles is a “hopper”. The design of a “hopper” concept emerges from the disability of ground vehicles on rough surfaces, which dictate to operate flying vehicles for better mobility [7]. In this manner, flying capable vehicles can fly over the obstacles without interacting with the obstacles. Note that the

continuously flying vehicles are very sensitive to sudden gust, wind or other atmospheric events due to being airborne through the mission. Moreover, being airborne throughout the mission causes more energy consumption. Hence, in a complex environment, one would prefer to operate a vehicle that can fly, be less exposed to atmospheric disturbances, and consume less energy. Therefore, the hopping vehicles are proposed as more applicable vehicles for rough landscapes due to their flying and ground capabilities in consideration of less energy consumption. The literature presents various operational environments for these vehicles. For instance, there are hopper vehicle designs for exploring Martian surface, which is assumed as a rough surface. The hopper concept has been particularly preferred in this environment to fly over the obstacles when required [7]. Furthermore, the Hopping Rotochute is proposed as another type of hybrid vehicle, which is capable to explore rough and complex terrains with minimum energy consumption [8].

1.2 Guidance and Control Challenges Facing Micro Vehicles

There are several unique challenges that a micro vehicle can encounter during its guidance and control. Particularly, flying capable micro vehicles pose more challenges than the other micro vehicle types due to the operation at low Reynolds numbers, implying a very high sensitivity to the flight conditions and atmospheric disturbances. Especially the atmospheric disturbances cause sudden increase in the drag and decrease in the lift. Hence, flying capable micro vehicles are extremely susceptible to trajectory deviations and controllability loss as emphasized in reference [9]. From a more general perspective, some of the operational challenges have been listed in reference [10] as below:

1) Flight in lowest altitudes, near or between obstacles.
2) Operation out of line of sight.
3) Operation in gusts of wind.

4) Safe control for urban operations and indoor missions even in the case of collision with walls.
5) High precision navigation for urban operations and indoor missions.
6) Obstacle detection and avoidance.

In the presence of these challenges, “trajectory deviation” is not desirable since it leads to failure in the mission even with some damage in the vehicle. For instance, when a micro vehicle operates in an environment with narrow halls, it is not acceptable to have trajectory deviations since it can hit the walls. Unfortunately, the trajectory deviation is very common in the presence of uncertainty. Hence, its effect should be minimized during the mission, which can only be achieved by an adaptive control technique implying the decision of control commands during the mission. In this way, if a trajectory deviation exists, the next control command attempts to correct it.

If a vehicle performs the commands given prior to a mission, the trajectory deviation becomes more significant. In this situation, the vehicle can not determine whether the control command is convenient for that specific position. It just applies the given command. This becomes critical if the deviation from the route is high because this situation leads to diverge from the desired trajectory due to the induced effects of unsuccessfully selected control commands. Consequently, the trajectory deviation is a critical issue for a micro vehicle. It is particularly more critical, if a vehicle performs pre-planned commands throughout the mission.

1.3 Research Objectives

The previous sections summarized various micro vehicle configurations and the challenges that they can encounter during a mission. Moreover “trajectory deviation” has been emphasized as a critical problem especially for a vehicle controlled by pre-planned commands.

The most recent version of Hopping Rotochute, introduced as a promising hybrid micro vehicle for complex environments, performs the control commands designed prior to mission. Therefore, the trajectory deviation becomes inevitable for this vehicle in the presence of uncertainty. In order to minimize the trajectory deviation, this thesis proposes to develop "Autonomous Hopping Rotochute", which can decide on the control commands during the mission. Thus, the objective of this thesis is as following:

To develop a trajectory following algorithm that allows the Hopping Rotochute to follow any pre-planned trajectory autonomously.

Furthermore, this thesis aims to experiment the trajectory-following algorithm in various simulation environments including obstacles and/or gusts of wind.

1.4 Thesis Outline

The organization of the thesis is presented as a schematic diagram in Figure 2 in which the triangle represents how the fundamentals of the thesis are narrowed down. The brief descriptions of each chapter are presented as following.

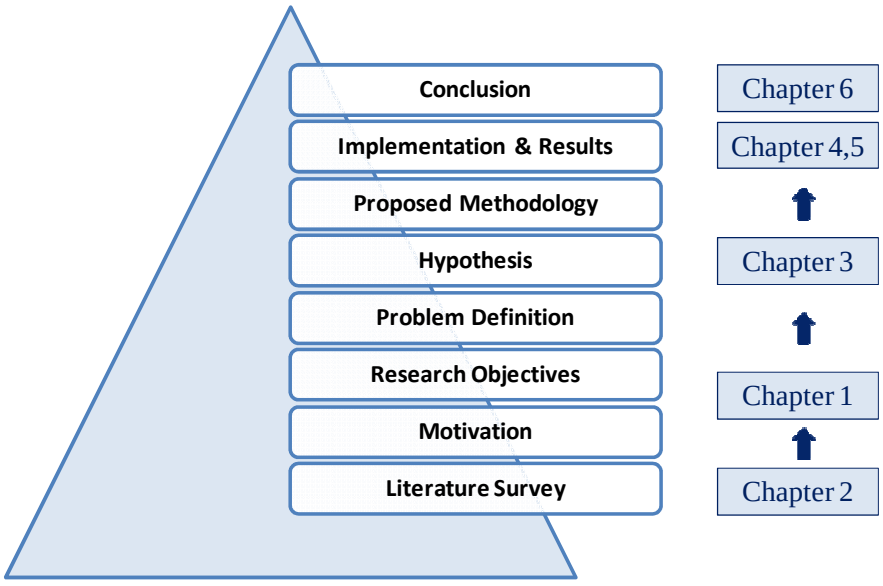

Figure 2: Thesis organization

Chapter 2 – Background: This chapter describes the past work and the relevant control methodologies in the literature. It presents not only the details of a Hopping Rotochute but also a review for online and offline control techniques.

Chapter 3 – Methodology: This chapter constructs the research questions and hypotheses along with the methodology developed to answer the research questions.

Chapter 4 – Implementation of the Proposed Methodology: This chapter focuses on the details of *hop performance models*. First the goodness of model fits is mentioned. Subsequently, the trends of the models are discussed. Finally, some example simulations and statistical results of the proposed methodology are presented.

Chapter 5 – Testing the Trajectory Following Algorithm: In the presence of gust and/or obstacles, the performance of the trajectory-following algorithm varies. Therefore, this chapter presents the simulation results of the algorithm in various scenarios involving gust and/or obstacles in the environment.

Chapter 6 – Conclusion and Future Work: This chapter reviews the research questions and hypotheses, and derives conclusion from the presented results leading suggestions for future research.

CHAPTER 2

BACKGROUND

The aim of this thesis is to develop a trajectory-following algorithm that allows a Hopping Rotochute to follow any pre-planned trajectory autonomously. To achieve this goal, first of all, this study must analyze the dynamics and the current control technique of Hopping Rotochute, and examines several control methodologies that enable autonomous trajectory-following. Therefore, this chapter aims to introduce the Hopping Rotochute and some control techniques applicable to autonomous trajectory-following.

The first section presents the Hopping Rotochute vehicle with its associated physical specifications and addresses past work related to the dynamics and the most recent control technique of the vehicle. The second section introduces control techniques applicable to the development of the trajectory-following algorithm. Among the presented control techniques, rule-based control and model predictive control are particularly useful for developing the autonomous control algorithm of the Hopping Rotochute.

2.1 Hopping Rotochute

The Hopping Rotochute is a hybrid micro vehicle consisting of a coaxial rotor surrounded by a spherical cage. The vehicle includes a small electrical motor driving the rotor system and rotates an internal mass for directional control. Operation of the vehicle is sustained by thrust produced by the rotor system. According to the direction of the internal mass, the thrust vector is inclined, and the vehicle flies at an inclination. Since the motor is not powered during the entire mission, the vehicle first begins to ascend, and then it descends. In other words, it exhibits a projectile motion. One of the important properties of the Hopping Rotochute is its physical architecture, which enables the

vehicle to re-orientate itself with the same position wherever it impacts to the ground. Figure 3 displays the current prototype of the vehicle.

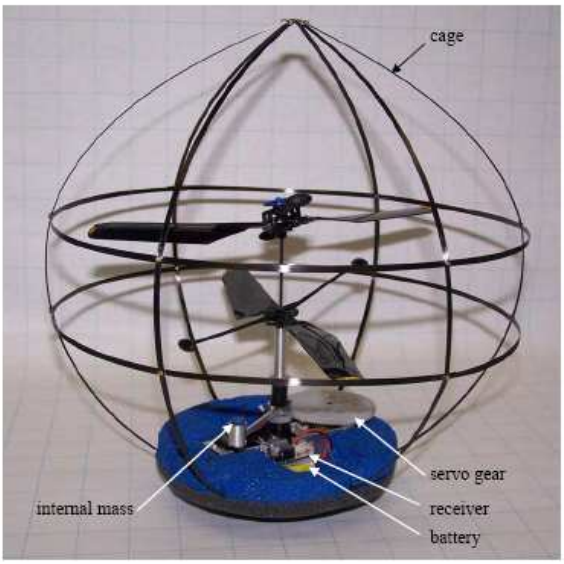

Figure 3: The Hopping Rotochute [8]

2.1.1 System Specifications

Some physical specifications of recent Hopping Rotochute are illustrated in Table 2.

Table 2: Hopping Rotochute specifications

Body Specifications	
Overall Height (cm)	25.4
Max Horizontal Cage Diameter (cm)	24.8
Rotor Blade Specifications	
Radius of the Rotor Blades (cm)	10.2
Aerodynamic Mean Chord (cm)	2
Rotor Blade Pitch (deg)	30
Maximum Rotor Speed (RPM)	4000
Mass Specifications	
Body Mass (g)	64.4
Internal Mass (g)	6
Battery Mass - 300 mAh (g)	19.8
Total Mass (g)	90.2

Further explanations of other components include [11]:

- The internal mass is operated by a micro servo allowing it to rotate ±180 deg around the interior perimeter of the body.
- The spherical cage not only locates and protects the components inside the vehicle but also provides the vehicle to sustain upright position on the ground.
- The foam cushion damps the motion when the vehicle impacts to the ground in order to prevent the electronic damage.

The properties in Table 2 will be used in acquiring the simulation results in subsequent sections.

2.1.2 Equations of Motion

The dynamic model of Hopping Rotochute has been developed with a six degree of freedom model consisting of the inertial positions of the total mass center and three Euler orientation angles [8]. Figure 4 shows the corresponding free body diagram of a Hopping Rotochute with the relevant reference frames including the rotor reference frame, the internal mass reference frame, the body reference frame, and the inertial reference frame represented respectively by the subscript letters as R, P, B and I.

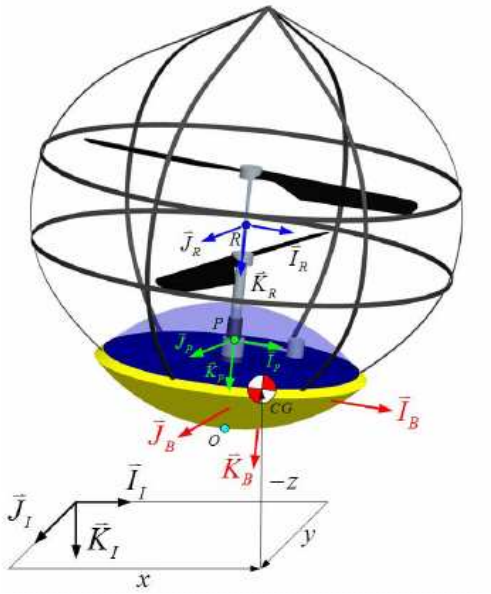

Figure 4: Free body diagram of a Hopping Rotochute [8]

The kinematic equations of motion are described in equations (2.1) and (2.2) as in reference [8].

$$\begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{Bmatrix} = [T_{IB}] \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{2.1}$$

$$\begin{Bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{Bmatrix} = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \begin{Bmatrix} p \\ q \\ r \end{Bmatrix} \tag{2.2}$$

In equations (2.1) and (2.2); x, y and z are the spatial coordinates in inertial reference frame; u, v and w are the velocity components in the body reference frame; φ, θ and ψ are Euler angles; p, q and r are the rotational velocity components in the body reference frame. Furthermore, the trigonometric functions are represented with shorthand notation such as: $\cos(\alpha)=c_\alpha$, $\sin(\alpha)=s_\alpha$, $\tan(\alpha)=t_\alpha$. The transformation matrix from the body reference frame to the inertial reference frame is represented by T_{IB} shown in equation (2.3) as:

$$T_{IB} = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix} \tag{2.3}$$

Preceding equations are employed in existing dynamic simulation code of a Hopping Rotochute. The simulations conducted by this code aim to imitate the real experiments of the vehicle. The main inputs of the code include detailed vehicle properties such as the geometry of each component or the thrust mapping of the rotor system as well as the environment properties and the control commands. The goal of the simulations is to observe the motion of Hopping Rotochute for various input conditions. Figure 5 summarizes the simulation code in terms of input and output.

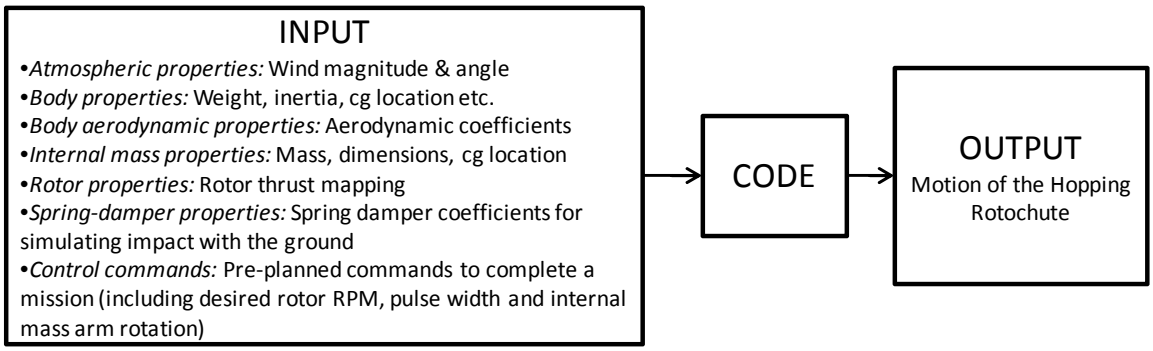

Figure 5: Summary of existing simulation code

2.1.3 Dynamic Model Validation

The preceding section explained that the simulation code has been developed for representing the motion of a Hopping Rotochute. In order to use this code with confidence, the validation results are presented in past work as in reference [8]. Briefly, the dynamic models employed in the simulation code have been validated by the motion capture system, which is able to record an example hop. The validation has been done by comparing the results of the simulation with respect to the records from the motion capture system.

Examples of the validation charts are displayed in Figure 6 through 9. In these charts, the experimental data from the motion capture system is represented as a solid line, and the theoretical data from the computer simulation is represented as a dashed line. For instance, Figure 6 illustrates a 3D plot of a particular hop conducted experimentally and theoretically. As it can be observed, the motions of the vehicle represented by solid and dashed lines exhibit a similar trend.

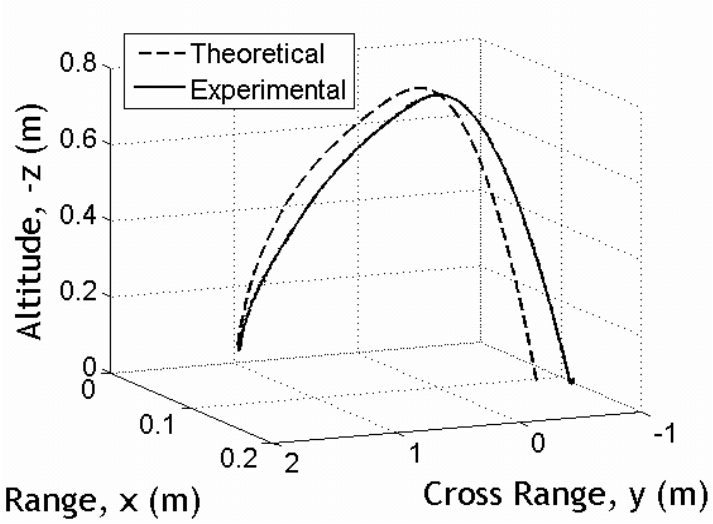

Figure 6: Flight test altitude vs. range vs. cross range [8]

In order to quantitatively compare the theoretical and experimental data of a particular hop, Figure 7 through 9 display the spatial coordinates of the vehicle with respect to time for the same hop. Thus, Figure 7, Figure 8 and Figure 9 show theoretical and experimental displacements in the x, the y, and the z axes, respectively. In Figure 7 and Figure 8, maximum distances between the solid and dashed line are respectively 0.02 m and 0.15 m at 2 sec.

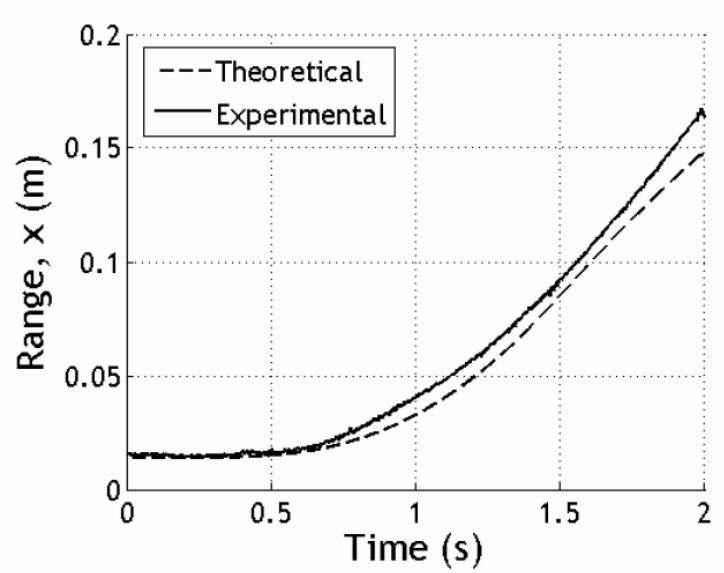

Figure 7: Flight test range vs. time [8]

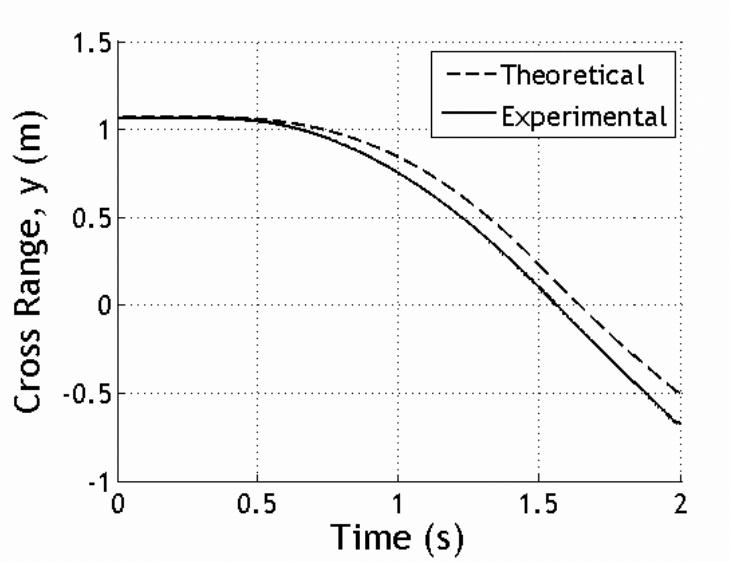

Figure 8: Flight test cross range vs. time [8]

In Figure 9, the theoretical and experimental data mostly coincide, which indicates that the simulation model is consistent with the experimental data. Reference [8] mentioned that the vehicle reached a maximum altitude of 0.77 m at 1.6 sec. while achieving a total range of 1.73 m associated with the experiment and 1.57 m for the simulation model.

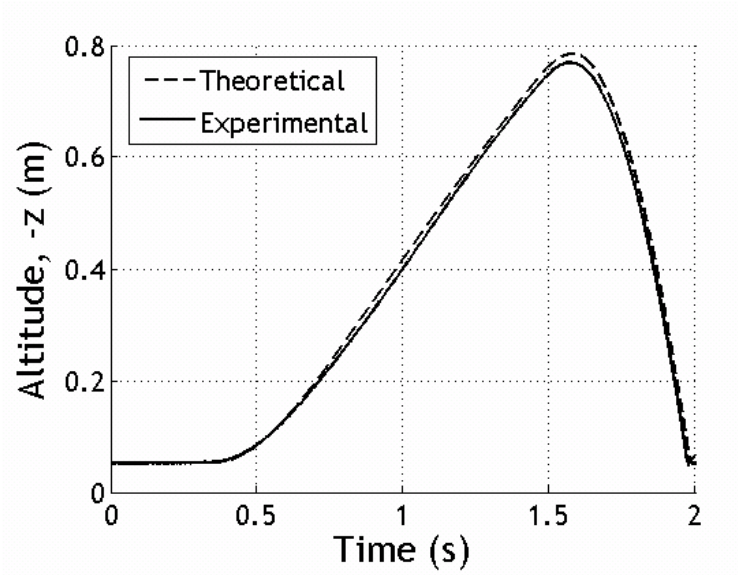

Figure 9: Flight test altitude vs. time [8]

Validation charts show that computer simulation is accurate when the results are compared with the experimental data. Hence, computer simulation can be used confidently to conduct further research about the Hopping Rotochute.

2.1.4 Control Technique for Path Tracking

The most recent version of Hopping Rotochute tracks a path by performing particular commands that are given prior to the mission. These commands include the following:

- RPM, which indicates the desired rotational speed of the rotor.
- Pulse width, which indicates the operation time of the rotor system.
- Angular position of internal mass (IM) arm, which indicates the inclination of the vehicle or the hop direction.

These commands are written down in a file, and the Hopping Rotochute is expected to perform all commands sequentially. Note that because the recent Hopping Rotochute implements pre-planned commands given prior to mission, given commands cannot be modified during the mission.

2.2 Control Techniques for Trajectory-Following

Vehicles can track a path using two control techniques. One of them is offline control, and the other one is online control. The fundamental difference between these two techniques is in their approach to producing control commands. Therefore, as online control is more applicable to autonomous trajectory following, this section starts with an overview of both online and offline control, but then provides further details about online control. Finally, rule-based control and model predictive control techniques are reviewed since these methodologies have inspired the development of this algorithm.

In offline control, commands, which are given prior to the mission, are performed exactly as given regardless of any external event. This process can be referred to as a

“batch job”, in which all commands are saved to a file and executed in a sequence. The most recent version of Hopping Rotochute employs offline path tracking, which is satisfied by pre-planned control commands of RPM, pulse width, and internal mass arm position. This technique is practical if there is no uncertainty; however, real world applications consist of many uncertainties driven by either a system or the environment. Hence, offline control in the presence of uncertainty may cause divergence from a route since control commands may require correction during a mission.

In online control, commands are produced during a mission, and an individual command is performed whenever it is obtained. In other words, this can be called “interactive processing”, in which a particular control command is immediately processed. For efficient trajectory-following, the Hopping Rotochute must be able to implement online control for path tracking. In this manner, it will be able to modify the control commands when required to prevent deviations from the route.

Recently, many studies have examined online path tracking for an unmanned vehicle. The most well known correspond to linear and nonlinear feedback controllers. These traditional techniques require accurate dynamic models, and their designs are too complex, especially for MIMO (Multi Input Multi Output) systems with unequal number of inputs and outputs. Furthermore, they are not considered robust against uncertainty as mentioned in reference [19]. Other techniques for online path tracking employ conditional inference. The advantage of employing conditional inference is that these techniques do not require accurate mathematical models. They achieve path tracking by defined rules, and they are efficient in generic decision making.

From a general point of view, path tracking requires vehicle and environmental information. In some cases with strong nonlinearities, obtaining accurate mathematical models for either the vehicle or its interaction with the environment may become challenging leading to the need for conditional inference, which eliminates the need for

mathematical models. Hence, the literature presents some studies that use conditional inference as rule-based control.

The previous section of this thesis depicted an accurate dynamic model for a Hopping Rotochute. However, the interaction of the environment with the Hopping Rotochute is uncertain. The implementation of traditional control techniques in the existence of vehicle's mathematical model and environmental uncertainty may cause deviations in the path tracking. Hence, this situation induces the use of conditional rules in the control algorithm of a Hopping Rotochute.

The major disadvantage of rule-based control is its inefficiency in generalization, suggesting a control algorithm governs situations in which the rules exist. However, it is still applicable to many nonlinear problems for decision making. (More details are given in subsequent sections.)

Decision making based on conditional inference is the major property of rule-based control, which can be a practical method for online path tracking. The important question arises as to what information is used in conditional statements. A conventional approach is to utilize sensor information in conditional statements. However, the most recent version of the Hopping Rotochute does not employ any onboard sensors. Therefore, embedding some prediction models in the algorithm is a goal of this research. In this manner, conditional statements are aimed at involving a desired condition and its predicted control inputs. For instance, a conditional statement could be "IF x is desired target point, THEN v is required control input." Thus, this idea will lead to creation of models that will be used to predict control inputs for desired hopping performance. Use of these prediction models resembles the popular control technique of model predictive control.

Due to the existence of uncertainty pertaining to environment-vehicle interaction, rule-based control, embedded with some model information (similar to model predictive

control), will be employed in the control algorithm. Hence, the following sections briefly explain the main ideas of these techniques.

2.2.1 Rule-Based Control

Rule-based control is a control technique applicable to systems lacking mathematical models for decision making. Based on previous knowledge or experience, some conditional statements are created, and the control algorithm employs these conditional statements in order to make inferences about a situation. In the literature, one of the popular rule-based control techniques is called fuzzy logic controller (FLC), which employs fuzzy inference. Fuzzy logic control is based on set theory, in which the boundaries of the sets are fuzzy. The aim is to make decision through conditional statements based on fuzzy sets. For instance, Figure 10, which illustrates a fuzzy set example, indicates a pure occurrence of an output as "1" and the non-occurrence of an output as "0". As it is seen from the figure, the left-hand side of the *x* interval corresponds to the pure occurrence of output 1 while the right-hand side of the *x* interval (in the absence of output 3 occurrence) corresponds to the pure occurrence of output 2. However, interval *x* represents a fuzzy condition in which the occurrence of both outputs 1 and 2 are observed. Hence, when the input is inside interval *x*, a fuzzy inference is implemented for outputs 1 and 2, such as "output 1 is observed *fairly* while output 2 is observed *slightly*," or vice versa. Note that the same approach can be induced by outputs 2 and 3.

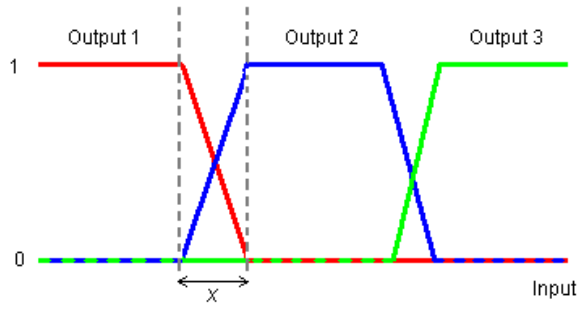

Figure 10: Fuzzy set example

In addition to applying pure FLC, researchers can use many variations of FLC to enhance decision making so that it is more generalizable. Note that decision making with conditional statements as in pure FLC cannot be generalized since only existing rules can be applied. Therefore, the authors of reference [19] combine fuzzy logic with neural networks (FNN). In this study, they claim that while path tracking is achieved by classical fuzzy rules, neural networks can provide systems capable of learning, and nonlinear expressing. Hence, the combination of these techniques increases the decision making capability of a vehicle in order to cope with unknown environments. Nonetheless, it is challenging to construct a neural network structure. Another reference, [20], presents a derivative of fuzzy logic, namely adaptive fuzzy control. This technique has been implemented in a wheeled ground vehicle that includes a sonar sensor. The main idea is that the vehicle creates a line between the start and the target points. When sensor information indicates an obstacle at the vicinity of the line, the method modifies the line in order to prevent possible impact, as shown in Figure 11. In this figure, *f* is a function of the robot-obstacle distance with respect to sensor information. Based on the *f* value, the line pointing x_d, the target point in the absence of the obstacle, is modified to the line pointing x_r, the pseudo-target point in the presence of the obstacle. In this manner, until the vehicle approaches to x_d, the route is modified as described above.

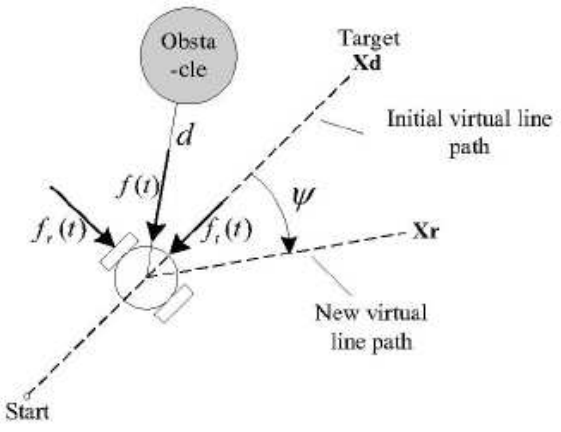

Figure 11: Adaptive fuzzy logic control [20]

Moreover, the method that has been described employs adaptive control for minimizing tracking errors. In order to visualize the benefits of adaptive control, Figure 12 illustrates a comparison between adaptive and non-adaptive path tracking. The figure shows that when the tracking error is minimized, adaptive path tracking is more likely to converge to the desired path than non-adaptive path tracking.

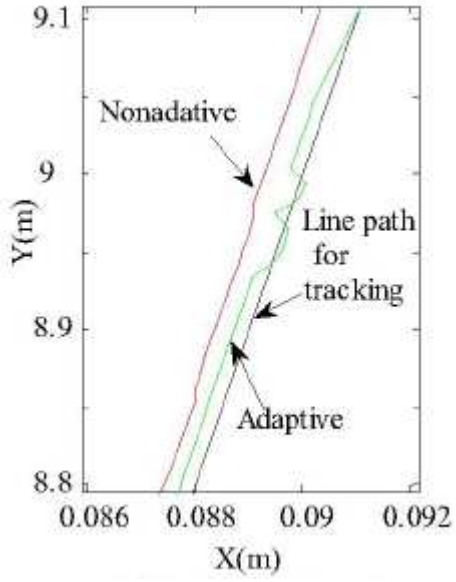

Figure 12: Adaptive fuzzy logic path tracking [20]

This thesis may not necessitate the creation of fuzzy sets, as in FLC. For instance, the existence of wind will be estimated by a threshold value determined from statistical results. An error that is smaller than the threshold will indicate the absence of wind while an error that is equal to or larger than the threshold will indicate the presence of wind. In this manner, since the decision-making technique resembles a switch mechanism, the boundaries of the sets are not fuzzy. Thus, Figure 13 presents a non-fuzzy set in which a condition represented as an intersection of two or more sets is not observed. (Note that Figure 13 consists of two sets, namely “Wind” and “No wind” sets.) (In addition, the existence of wind can also be represented as a fuzzy set when the error is close to the threshold. However, because of the lack of knowledge and practicality, this study will

employ non-fuzzy sets, as shown in Figure 13. More details will be explained in the following chapters.)

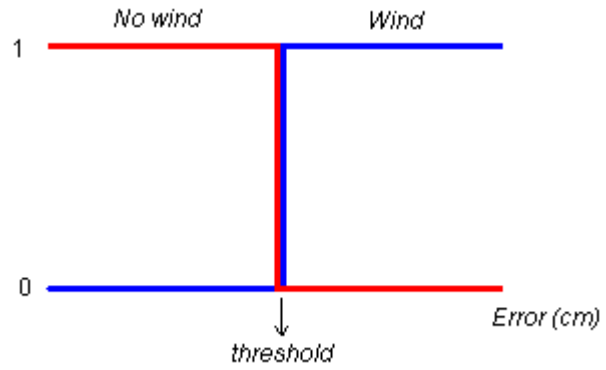

Figure 13: Example to a non-fuzzy set

Hence, the algorithm developed in this thesis involves non-fuzzy sets with some thresholds, implying that it is not possible to observe two or more conditions at the same time. Thus, pure rule-based control without fuzzy inference is employed in the algorithm on the basis of comparing some values with respect to some thresholds. Therefore, conditional statements represented by a set of linguistic descriptions such as the followings are created.

IF track condition 1 AND avoid condition 1, THEN command 1.

IF track condition 2 AND avoid condition 2, THEN command 2.

.

.

IF track condition n AND avoid condition m, THEN command nm.

We can observe rule-based control at a wide variety of application areas, listed as follows:

Automated Highway Systems: Automatic steering, braking and throttle control of vehicles, and automatic train operation systems [12].

Autonomous Vehicles: Ground and underwater vehicles.

Manufacturing Systems: Scheduling and deposition process control.

Robotics: Position control and path planning.

Rule-based control is a practical technique when mathematical relations cannot be modeled in terms of input-output, and strong nonlinearity is present in the system. Instead of employing analytical equations, rule-based control provides decision making by conditional statements. Recall that the major drawback is that creating the algorithm requires many rules, which need a lot of prior information.

2.2.2 Model Predictive Control

In recent years, model predictive control (MPC) has become one of the most popular process control methodologies. It differs from other methodologies such as optimal control, adaptive control, or robust control by employing a prediction in the controller. Prediction is a very important concept in decision making. For instance, Rossiter claims that “prediction is invaluable for avoiding unforeseen disasters” [13]. He gives an example about a human who crosses the road, stating that it is not sufficient to cross a road if there are no cars between the person and the other side. It is also important to predict whether there are cars, some distance away, that will cross in front of the person soon. Based on these conditions, the person decides whether he can cross. In addition, while one is crossing the road, he continues to update the predictions. If it is required, the trajectory may change based on the updated predictions. Hence, this example implies that the prediction information plays an important role in controlling a process.

In MPC, a model is used to generate system predictions. This model is not required to be a physical model. As long as the results give enough accuracy, the simplest model, which is called “fit for purpose”, can be used [13]. However, it is important to note that the precision of control always depends on the precision of the model.

Since the late seventies, many variations of MPC have been developed by researchers who have focused on linear model predictive control (LMPC) and nonlinear model predictive control (NMPC). A survey in reference [14] includes more details about variations of MPC. Moreover, Bemporad [15] has developed an adaptive model predictive control for the path tracking of a fixed wing unmanned aerial vehicle. He achieved more accurate path tracking by employing a variable control horizon that depicts the future. Based on the tracking conditions, long- or short-term prediction is obtained.

While investigating the main structure of the various MPC algorithms, one can observe that the common elements of the algorithms do not change. Hence, the derivatives of MPC are obtained by employing different options for these elements. Camocha and Bordons state the major elements of MPC in reference [16] as follows:

- Prediction Model: This model represents the dynamic behavior of the process. The aim of the model is to calculate the predicted outputs for future instances.
- Objective Function: This is the cost function that will be used in obtaining the control law.
- Obtaining the control law: This algorithm produces the control commands by minimizing the cost function, defined in the former step.

In this research, MPC prompted the use of prediction models in the control algorithm. Hence, *hop performance models* are created for control command calculation.

In this manner, as in MPC, a control based on model prediction is employed in the proposed methodology.

2.3 Summary

Chapter 2 introduced the Hopping Rotochute and several control techniques for path tracking, specifically, rule-based control and model predictive control. The Hopping Rotochute is a hybrid micro vehicle capable of exploring rough and complex terrains with minimum energy consumption. Flying and ground characteristics of a Hopping Rotochute provide the unique dynamics of the vehicle. Previous work presents the validated studies pertaining to the dynamic behavior of the vehicle. One of the weak properties of the Hopping Rotochute is that it uses offline control for path tracking.

For efficient path tracking, online control techniques are convenient because they allow control command corrections during the mission. One of the most practical online control techniques in the absence of accurate models is rule-based control. This control technique employs some conditional rules for decision making. An example rule is "IF a condition exists, THEN a consequence occurs." Hence, for determining the control command, a differential equation representing the input/output behavior is not required.

A process control technique that employs a prediction model in the control algorithm is model predictive control. This method prompted the use of *hop performance models* for the prediction of the most likely control commands in achieving a desired hop. Thus, the models create rules for deciding efficient control commands. An example rule is "IF x is the target, THEN u is the required control command to achieve x." With the models, the algorithm can calculate the required control command, u, to achieve a desired hop, x. Consequently, the models help to predict the control command of a hop candidate.

CHAPTER 3

METHODOLOGY

The foundation of this research is based on the Hopping Rotochute, which lacks autonomous trajectory-following. Chapter 2 stated that the current prototype of the Hopping Rotochute executes control commands uploaded prior to a particular mission. This control scheme is most likely to cause trajectory deviations in the existence of uncertainty. In order to achieve efficient path tracking, a methodology is proposed for the development of an autonomous Hopping Rotochute. The aim of this chapter is to create the research questions, to construct hypotheses, and to propose a methodology. Hence, the first section describes the research questions and the hypotheses while the second section presents the proposed methodology.

3.1 Research Questions and Hypothesis

Information in the literature survey was used to determine observations that would be made in this study. The observations were used to create a set of research questions, and the answers to the research questions were used to construct hypothesis pertaining to the methodology for autonomous trajectory-following. The literature revealed three observations. The first observation pertains to the motion of the Hopping Rotochute, which can be represented as the integration of consecutive discrete events. Here, the “discrete event” corresponds to an individual hop controlled by pre-programmed commands. The only connection between the hops is that the beginning of a particular hop can be expressed as the end of the previous hop. The second pertains to uncertainties encountered by a Hopping Rotochute. Although all sources of uncertainty are unknown, some of them can be stated as the following:

(1) Position knowledge: The Hopping Rotochute may not know its exact position.
(2) Mechanical uncertainties: The Hopping Rotochute may not rotate its components precisely due to the inefficiency of gear or other mechanical systems.
(3) Environmental conditions such as temperature, humidity or day/night experiments: The Hopping Rotochute may not reflect the same behavior in all environmental conditions. For instance, mechanical parts may not operate consistently at different temperatures.
(4) Atmospheric events: The Hopping Rotochute may not know the exact instance of exposure to gusts of wind that occur suddenly or not at all.

Hence, it is important to assess uncertainty in order to represent the real world effects.

Finally, the third observation pertains to the operation environment of a Hopping Rotochute, which is designed to perform indoor missions. In such missions, it is possible to encounter narrow halls with limited ceilings. Thus, the Hopping Rotochute needs to follow a trajectory without hitting the boundaries. Moreover, these environments may include openings such as windows or doors that could allow gusts of wind inside the room. Hence, path tracking becomes a challenge for a Hopping Rotochute in the existence of wind, which causes deviations from its route based on its direction and magnitude. As a review, Figure 14 outlines the observations and the corresponding research questions.

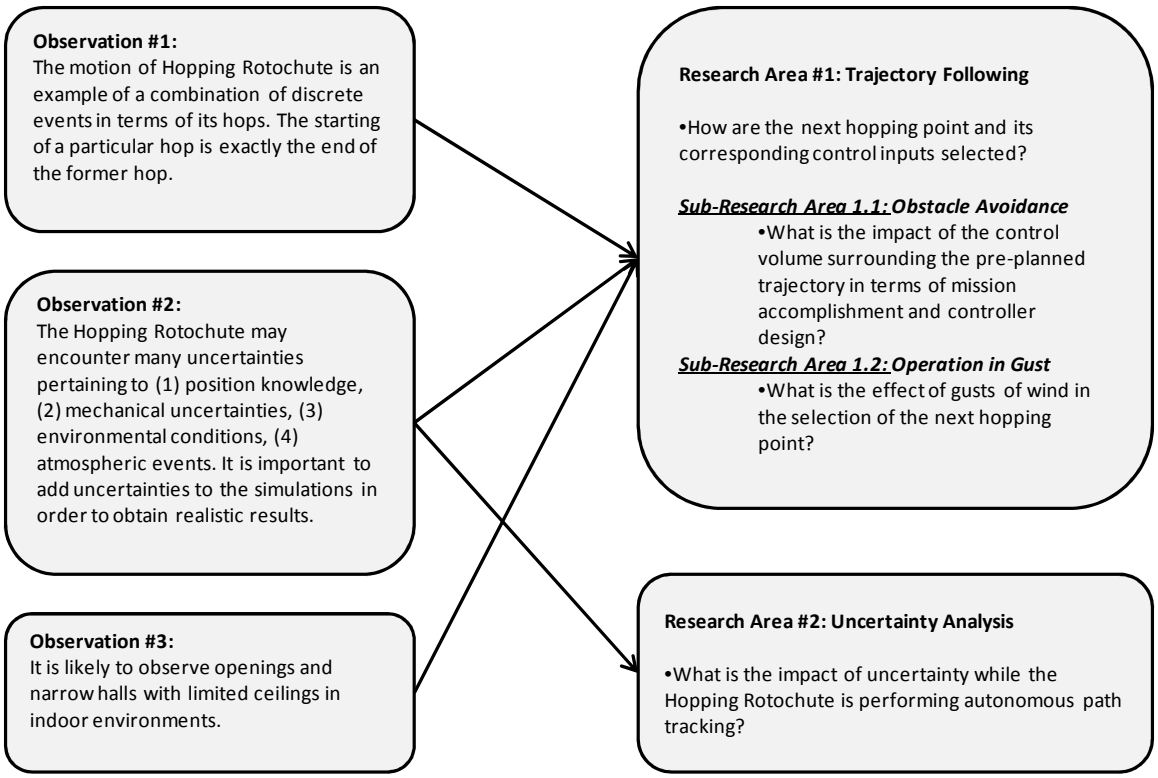

Figure 14: Observations and research questions

After introducing the observations mentioned in the previous section, the thesis determines the research areas. The first pertains to the trajectory following, which focuses on the selection of the next hopping point at various conditions. The second research area involves an uncertainty analysis that investigates the impact of uncertainty while a Hopping Rotochute is operating. The following sections will describe the research questions and their prospective answers (the hypotheses). The first research question relates to the trajectory following. Hence, it questions the selection technique of next hopping point. In this manner, the first research question is as follows:

> **Research Question 1:** How are the next hopping point and its corresponding control inputs selected?

Based on the first observation, which mentioned a combination of discrete events, a discrete control can be applied to a Hopping Rotochute, suggesting that each hop can be controlled individually. The second observation reveals uncertainty pertaining to the

atmospheric events in the environment. Traditional control techniques such as linear or nonlinear feedback controllers do not allow this uncertainty to be employed in the algorithm. Thus, conditional inferences are implemented in the controller design, leading to conditional statements for the control input decision.

The control input decision requires the calculation of control commands for a particular hop. This calculation can be provided as long as the hop performance is known. Since the calculated control command will correspond to a future hop, hop performance needs to be predicted before it is attempted. Therefore, a regression model is created from the results of previously conducted experiments. In this manner, the experiments provide data, related to hopping performance under environmental concerns. These data provide the *hopping performance models* in this thesis. The aim of these models is to predict the required control commands for a desired hop. Thus, conditional inferences as in rule-based control, and prediction models as in model predictive control, are employed in the algorithm of the next hopping point selection. Hence, hypothesis 1 states the following:

> **Hypothesis 1:** Assuming a known trajectory and boundary conditions, the next hopping point can be selected at the end of a particular hop by conditional statements in which the *hopping performance models* are embedded for hop predictions.

Hypothesis 1 can be proven by analyzing the error that is the distance between the target point and actual landing point at the end of each hop. If the error is reasonable, it will imply that the algorithm presents good control command calculation to achieve a desired hop. In addition, the model fits of *hopping performance models* address the accuracy of the hop prediction. In order to achieve an accurate hop prediction, implying the knowledge of at which position a particular hop will end, the model fits need to be good.

Note that a good model fit corresponds to a regressed model capable of reflecting the actual hop performance.

Based on the third observation, the Hopping Rotochute is likely to observe narrow halls with ceiling limitations in its operational environment. In these environments, accomplishing a mission by staying certain distance away from the known boundary conditions becomes a major goal. In order to achieve this goal, this thesis proposes to define a 3D volume, like a notional tunnel with less height and less width than the actual room, around the pre-planned trajectory. Hence, the next research question is as follows:

> **Research Question 1.1:** What is the impact of the control volume surrounding the pre-planned trajectory in terms of mission accomplishment and controller design?

In the absence of uncertainty, it is theoretically impossible for the Hopping Rotochute to impact the boundaries if a notional volume is defined inside the actual room, and any control inputs are computed with respect to this volume. In this manner, hypothesis 2 states the following:

> **Hypothesis 1.1:** If a volume surrounding the pre-planned trajectory is defined inside the given boundary conditions such as the side walls and the ceiling, the Hopping Rotochute does not interact with the boundaries unless it exits this volume.

Hypothesis 1.1 can be proven by repeated simulations in the presence of uncertainty to establish whether any violation exists with the boundaries.

Although the Hopping Rotochute is expected to operate inside buildings, it is likely to observe gusts of wind in the environment from openings such as doors or windows. Hence, the next research question states the following:

Research Question 1.2: What is the effect of gusts of wind in the selection of the next hopping point?

In a gusty environment, if the control algorithm selects the next hopping point regardless of wind, the Hopping Rotochute will start to diverge from the route based on the direction and the magnitude of the wind. In order to prevent this situation, if the vehicle aims to hop to a specific point *beyond* the trajectory, the wind drift can allow it to land *on* the trajectory. Hence, before the vehicle conducts a particular hop, an estimation of wind drift is required. As a result, hypothesis 1.2 proposes the following:

Hypothesis 1.2: The effect of gusts of wind will be used to estimate the wind drift, which will determine the selection of the next hopping point beyond the trajectory.

Hypothesis 1.2 can be proven by repeated simulations in order to establish the average deviation from the route in the presence of wind. If the various windy simulations present reasonable deviations from the desired trajectory, it implies that the wind drift is estimated correctly, and autonomous trajectory-following is accomplished in a gusty environment. Moreover, because the graph of the internal mass (IM) arm position involves the target IM and the actual hop angular positions, this graph proves that the Hopping Rotochute is hopping beyond the trajectory and landing close to the trajectory. In the existence of wind, these angular positions are expected to be significantly different

since the target IM position extends linearly beyond the trajectory and the actual hop angular position extends linearly towards the trajectory.

Real world applications involve many uncertainties from unknown sources. Therefore, modeling these uncertainties is not an easy process. However, counting them in the simulations is crucial to the analysis of whether the controller is working well under the effect of uncertainty. Therefore, the final research question is as follows:

> **Research Question 2:** What is the impact of uncertainty while the Hopping Rotochute is performing autonomous path tracking?

Since each uncertainty cannot be modeled, they are simplified in two main categories, namely mechanical/instrumental and environmental uncertainties. In this thesis, they will be called “noise” and “bias”, respectively. These uncertainties are assumed to mainly affect positioning, RPM, and pulse width. Hence, hypothesis 2 states the following:

> **Hypothesis 2:** Accuracy of the trajectory-following algorithm can be assessed from uncertainty analysis in which the statistical performance of the algorithm is examined along with the confidence interval study.

Hypothesis 2 can be proven by investigating the simulation results under the effect of uncertainty for repeatable accuracy, whose existence will indicate an acceptable error with less variability. Hence, the repeated accuracy can indicate the accuracy of the trajectory-following algorithm.

3.2 Proposed Methodology

The proposed methodology for autonomous trajectory-following combines the rule-based control with some prediction models. The weaknesses of the traditional control techniques (such as their complex design and not being robust under uncertainty) impose to apply rule-based control to this problem, and the need of hop performance prediction leads to create some regression models to estimate the required control commands. Hence, the proposed methodology consists of conditional statements in which the control commands are calculated from the regression models. Figure 15 presents the general view of the methodology steps. Briefly, step 1 corresponds to the creation of the regression models; step 2 calculates the target point for a particular hop; step 3 employs the calculation of control commands, and step 4 conducts an uncertainty analysis.

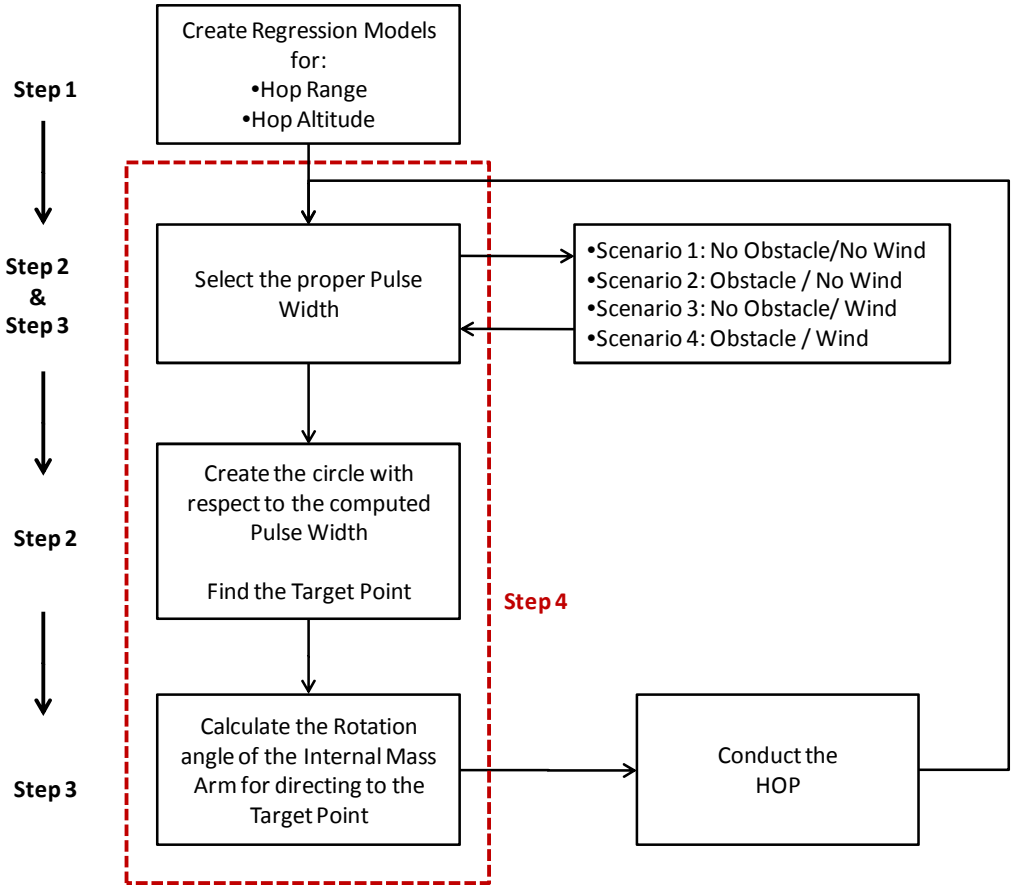

Figure 15: The steps of proposed methodology

3.2.1 Step 1: Create the Model

The main reason to create the regression models is to establish a relationship between the control commands and the hop performance parameters referring as the maximum altitude and displacement. From the fundamentals of physics, the displacement and altitude are directly related with the mass, the applied force, and the force duration. In the Hopping Rotochute case, the distance travelled in horizontal plane and the maximum altitude gained are functions of a Hopping Rotochute's weight, the thrust produced by the rotor system, and the duration of the rotor system operation.

In the assumption of a specific Hopping Rotochute is assigned to a particular mission, the weight becomes a constant parameter for the displacement and altitude. Recall that a Hopping Rotochute is not fuel-powered. Moreover, it is not dropping any payload throughout the mission. Thus, a weight loss is not observed during its operation.

Since the Hopping Rotochute is powered with a rotor system, the thrust produced by the rotor can be expressed as a function of revolution per minute (RPM). Hence, RPM of the rotor system becomes the first key parameter influencing the hopping performance. The second key parameter is the duration of the rotor system operation, referred as *pulse width* (t_{app}) of a Hopping Rotochute. While the pulse width is increasing, the distance travelled is also increased since the rotor system is producing thrust for a longer amount of time. Hence, the regression models of hopping distance and maximum hopping height will be created with respect to RPM and pulse width of the rotor system. The details of the models will be presented in Chapter 4.

3.2.2 Step 2: Calculate the Target Point

The fundamental step for autonomous trajectory-following is the computation of the target hopping point. Therefore, step 2 introduces a practical method for selecting the next hopping point by combining the information obtained in step 1. Note that step 1 is the section in which the regression models are created for hopping distance and height.

Hopping distance is specifically emphasized in step 2 since the distance traveled in the horizontal plane is counted in the calculation of the target point, demonstrated in Figure 16. In this figure, the center of the circle represents the position of the Hopping Rotochute; the circle represents the vehicle range for a specific hop; and the curved line represents the desired trajectory with start and end points.

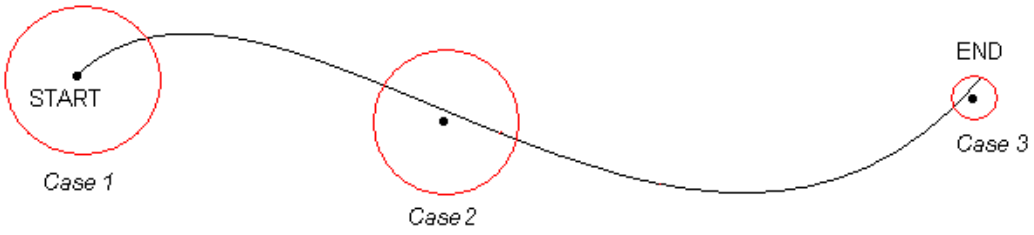

Figure 16: Trajectory-following algorithm

The trajectory-following algorithm calculates the point at which the circle and the trajectory intersect. In cases such as case 2 in Figure 16, in which multiple intersection points are present, the algorithm selects the point closest to the final point. Furthermore, the algorithm is capable of determining whether it needs to shrink the circle in order to hop smaller distances. An example of this situation is case 3 in Figure 16. Finally, the last scenario represents the condition in which the circle and the trajectory do not intersect. In this situation, the algorithm selects the point on the circle that is closest to the trajectory. Thus, the vehicle attempts to approach the nearest point on the trajectory.

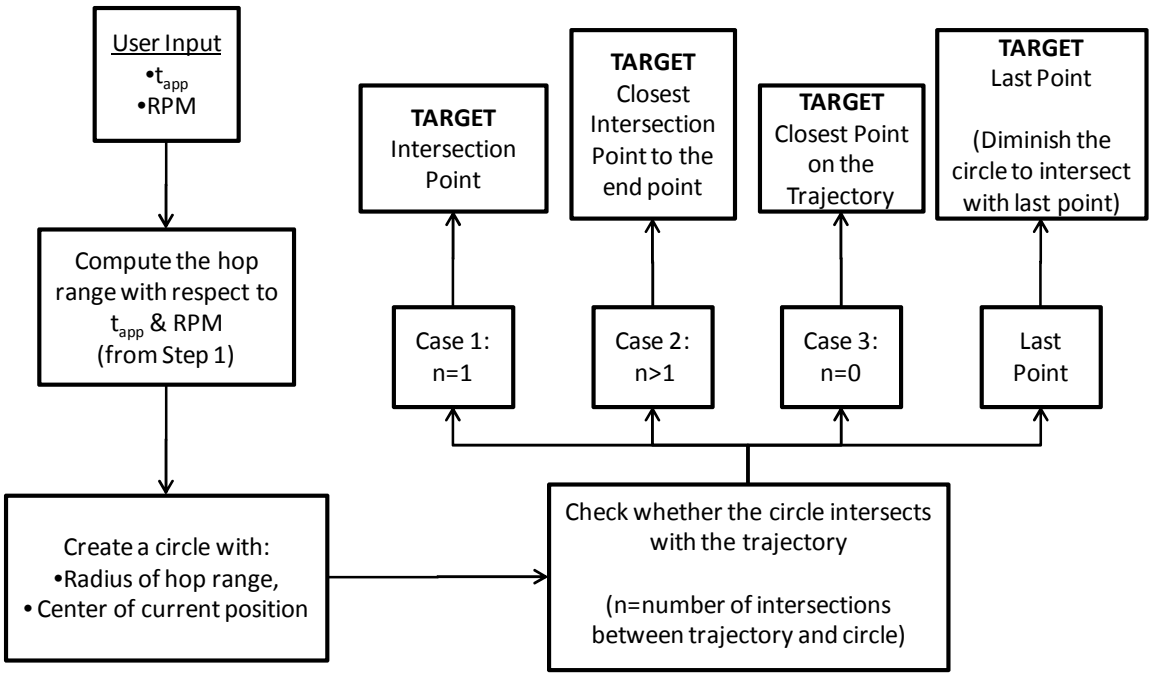

Figure 17: Algorithm for the selection of the target point

Based on the demonstration explained in Figure 16, a rule-based control algorithm is created as in Figure 17. The algorithm starts with the initial pulse width and RPM information. From the regression models created in step 1, the corresponding hop range is computed, and the computed value becomes the radius of the circle introduced in Figure 16. Finally, the intersection points of the circle and the trajectory are investigated to apply the suitable case rule in the algorithm.

3.2.3 Step 3: Decide the Optimum Control Commands

The preceding sections presented the existence of a pre-planned trajectory, the *hop performance models* created in step 1, and the algorithm developed in step 2 for the selection of the next hopping point. After obtaining the information introduced in the former sections, the proposed methodology continues with step 3, in which a decision about the optimum control commands is made.

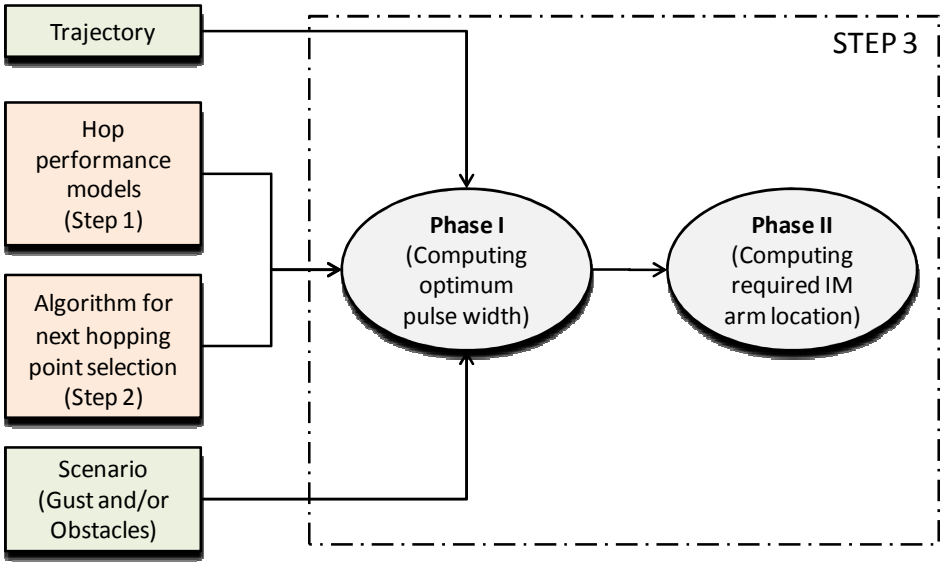

Figure 18: Schematic diagram of step 3

As in Figure 18, step 3 consists of two phases in which the control commands are computed. In the former sections, RPM and pulse width were introduced as the key parameters of the hopping distance and height. Assuming the efficiency of the rotor system, varying pulse width to change the hop performance is preferred against varying RPM throughout the mission. Hence, RPM becomes a constant parameter determined by the user for efficient operation, and pulse width becomes the first control command pertaining to hopping performance.

The second control command, the IM arm location, pertains to the directional control. Because the directional control of the Hopping Rotochute is sustained by the IM arm, which inclines towards its position, the position of the IM arm needs to be located accurately for hopping to the desired point. Thus, in step 3, control commands are calculated; phase I of step 3 computes the optimum pulse width and phase II of step 3 computes the optimum IM arm location.

Phase I

The aim of this section is to compute the first control command, which is the pulse width, (t_{app}), for desired hopping performance. As mentioned, the user inputs the desired RPM. The critical point in the calculation of the pulse width arises in the selection of the next hopping point. As anticipated, the next hopping point is not unique for a mission. For a particular Hopping Rotochute, the next hopping point differs based on the environment. For instance, the next hopping point is selected by avoiding impact with the ceiling or side walls; or the next hopping point is not selected on the trajectory in the presence of wind. All of these variations in the environment result in the selection of different hopping points leading to different pulse widths.

Indoor environments are most likely to contain obstacles and/or gust. Although the Hopping Rotochute knows the locations of the obstacles, it does not know the magnitude and direction of the wind. Hence, the subsequent sections will discuss the details of the algorithm for calculating the pulse width in various scenarios.

Scenario 1 – No Obstacle/ No Wind

This baseline scenario assumes no obstacle and no wind in the environment. In this case, the user is required to give the pulse width as an input. Once the pulse width is known, the algorithm is able to calculate how far the vehicle can hop. The target point is computed by the algorithm as shown in Figure 17. Pulse width is not calculated until the last point since it is a user input. Only in the vicinity of the last point; a convenient pulse width is computed in order to finalize the mission.

Scenario 2 – Obstacle/ No Wind

This scenario assumes narrow halls with a ceiling limitation. In Scenario 2, the vehicle aims to propagate on the trajectory without hitting the side walls or the ceiling. As mentioned before, the trajectory information along with the boundaries is given as

input prior to the mission. Trajectory information includes all discrete points forming the trajectory, and the boundary information involves all discrete points forming the side walls and the ceiling. Hence, the given data enable the creation of a 3D volume surrounding the trajectory.

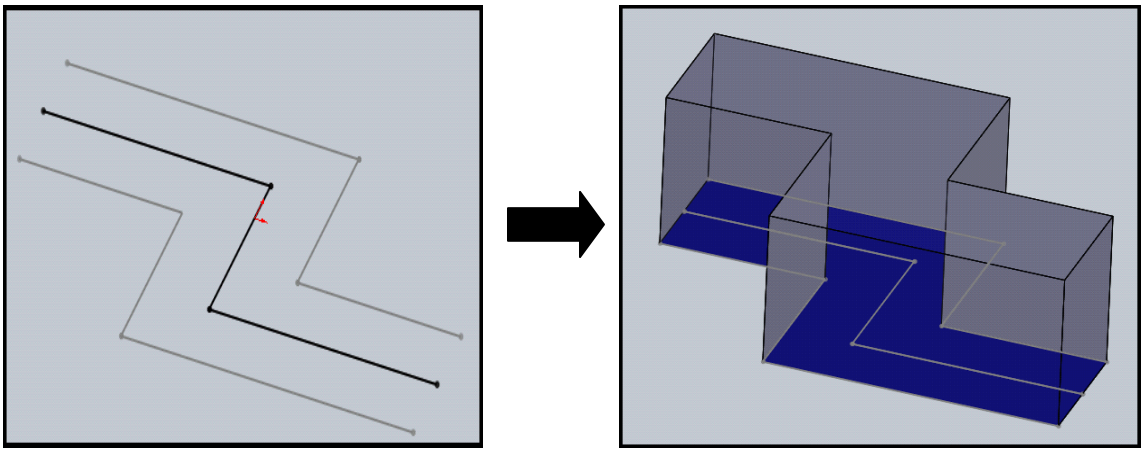

Figure 19: Pre-planned trajectory

Figure 20: 3D volume created from a trajectory

In order to increase the accuracy of the trajectory-following, narrower volumes, from which the Hopping Rotochute is restricted from leaving, can be generated. For instance, Figure 19 displays the notional walls created by assuming a clearance from the actual trajectory, and Figure 20 illustrates the notional volume introduced as the region accessible to the vehicle.

Based on creating a notional volume surrounding the trajectory, this thesis proposes a rule-based control algorithm for calculating the pulse width in accordance with obstacle avoidance. The algorithm, a flow chart, is presented in Figure 21.

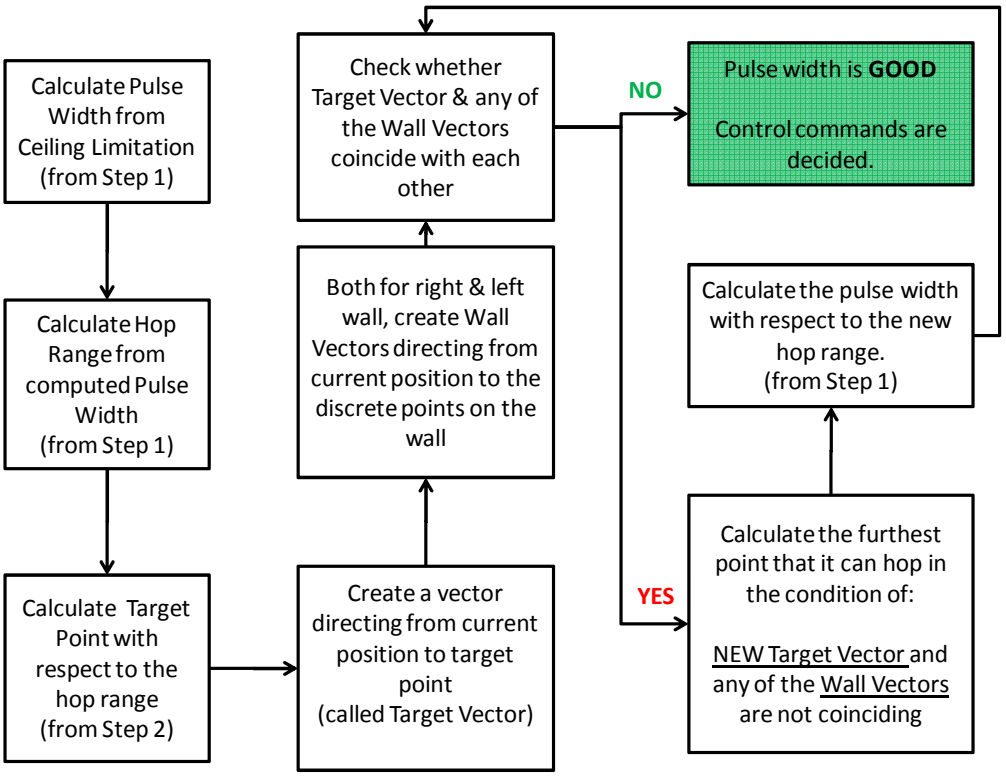

Figure 21: Obstacle avoidance algorithm

The obstacle avoidance algorithm starts with the calculation of the pulse width with respect to the ceiling limitation determining the maximum hopping height, which is used in the *hop altitude model* to calculate the corresponding pulse width. This pulse width is used in the *hop distance model* to compute the hop range of a particular hop. Then the target point as described in step 2 is calculated. When the target point is determined, a vector called a *target vector*, extending linearly from the current position of the Hopping Rotochute to the target point, is created. On the other hand, *wall vectors*, extending linearly from the current position of the Hopping Rotochute to the known discrete points of the boundaries, are created. The algorithm checks whether the *target vector* coincides with any of the *wall vectors*. In the absence of coincidence, the calculated pulse width at the beginning of the algorithm is good to hop. However, in the presence of coincidence, impact is expected. In order to avoid this situation, the algorithm decides to hop smaller distances, and iteratively sets the target point to a closer point until

it finds a collision-free path, indicating no coincidence between the *target vector* and any *wall vectors*. Finally, based on the new target point, the required pulse width is calculated.

Scenario 3 – No obstacle/ Wind

In this scenario, an opening creates a gusty environment inside the room, and a segment of the trajectory is subjected to the gusts of wind as shown in Figure 22. The figure depicts a curved line that indicates the trajectory and the arrows that represent a uniform wind generated from a particular opening. When the vehicle enters the windy zone, it may not end a specific hop at the target point due to the wind drift. This situation demands a gust-tolerant trajectory-following.

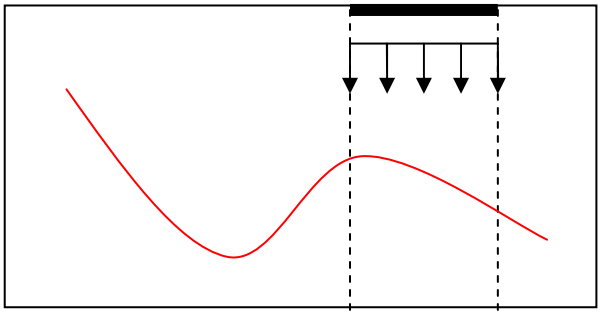

Figure 22: Trajectory subjected to wind

The development of the algorithm assumes the following:

1. *The magnitude and direction of the wind is not known*. An estimation technique is used to obtain this information. In this process, a term is defined as error (ε), representing the distance between the target point and the actual landing point. The estimation technique uses the error of the previous hop to interpret the direction and magnitude of the wind.

2. *Maximum limit of the wind magnitude is known*. The velocity of the wind in indoor environments is in a range between 0 m/s and 4 m/s [18]. Some trial experiments were conducted to analyze the durability of a Hopping Rotochute in

the presence of wind. It was observed that if the head wind is larger than 1m/s, the Hopping Rotochute is unable to approach its final point. Hence, this thesis assumes that the maximum magnitude of the wind velocity is 1 m/s.

3. *The wind velocity is assumed to be uniform*. Inside the windy zone, the wind velocity does not vary in a spatial direction. This assumption eliminates the complex analysis of wind.

This thesis proposes a rule-based algorithm for gust-tolerant trajectory-following. The schematic diagram of the algorithm is presented in Figure 23. At the end of each hop, the algorithm starts with the calculation of error (ε), representing the distance between the desired end point and actual landing point.

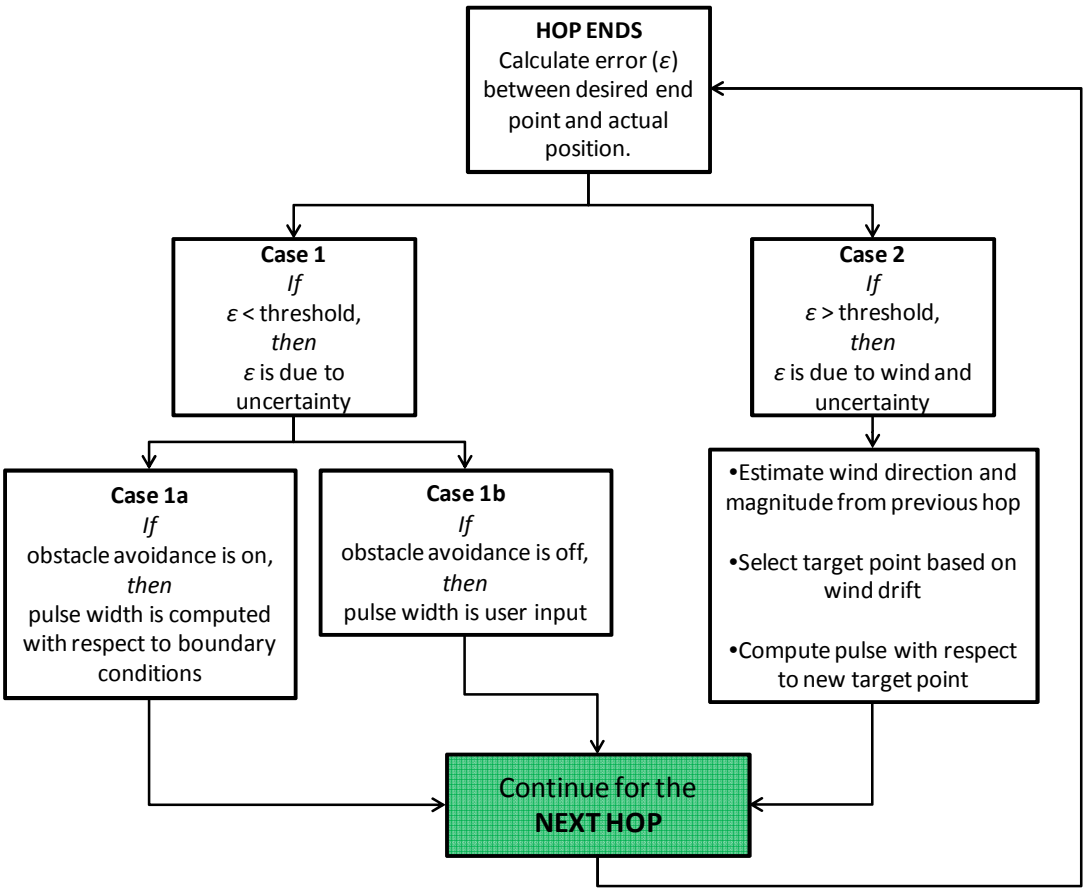

Figure 23: Gust-tolerant algorithm

Real-world applications cannot obtain exactly zero ε due to uncertainty. Thus, ε will always be a nonzero number at the end of a particular hop. Moreover, when the vehicle is subjected to gusts of wind, ε is expected to be larger based on the wind magnitude. Here, a question arises at which point ε begins to indicate the presence of wind. In order to answer this question, another model, whose development is similar to that of the *hop performance models* explained in step 1, is created. The goal of this model is to observe ε with respect to the variations of wind and hop altitude. (More details about this model will be given in Chapter 4.) Particularly, the observations obtained from this model allow the determination of a threshold that is the maximum possible value of ε in the absence of wind. Hence, two cases referring to the presence and the absence of wind are created and illustrated in Figure 23.

If ε is smaller than the threshold value, as in case 1, this implies that the vehicle is not subjected to wind. In this condition, the algorithm checks whether obstacle avoidance is on. If obstacle avoidance takes place, then the pulse width is computed by taking into account the boundary conditions. If no obstacle avoidance occurs, the pulse width is the user input. On the other hand, if ε is larger than the threshold value, as in case 2, this implies that the vehicle is subjected to wind. In this condition, the algorithm estimates the wind direction and magnitude. Subsequently, it selects the next target point by considering the wind drift, and calculates the required pulse width to hop to the selected target point. Finally, the next hop is conducted using the calculated pulse width, and the same procedure is applied at the end of each hop.

Scenario 4 – Obstacle/ Wind

This scenario is a combination of scenarios 2 and 3. Hence, it contains all of the assumptions and logical rules pertaining to the trajectory-following algorithm. Even though the environment consists of variable ceiling heights, the algorithm assumes the minimum ceiling height, since unpredicted wind can cause deviations from the route. The

disadvantage of this assumption leads to longer mission completion time, particularly for situations in which the hop altitude is much smaller than the local ceiling height. Nonetheless, obstacle avoidance along with the lowest possible trajectory deviations is achieved throughout the mission.

Phase II

At the beginning of this section, the control commands were introduced as the pulse width and the rotation angle of the internal mass (IM) arm. In Phase I, the selection criteria of the pulse width were described along with the target point of a particular hop. At the end of Phase I, the target point information was forwarded to Phase II. The goal of Phase II is to calculate the angular position of the IM arm so that the Hopping Rotochute hops to the desired target point.

Chapter 2 introduced that the directional control of a Hopping Rotochute is established by rotating the IM arm aligned in the direction of the target point. In other words, the IM vector, extending linearly from the center of the vehicle along the internal mass arm, must coincide with the *target vector*, extending linearly from the center of the vehicle towards the target point, as shown in Figure 24. Once the IM vector aligns with the *target vector*, the Hopping Rotochute hops toward the target point.

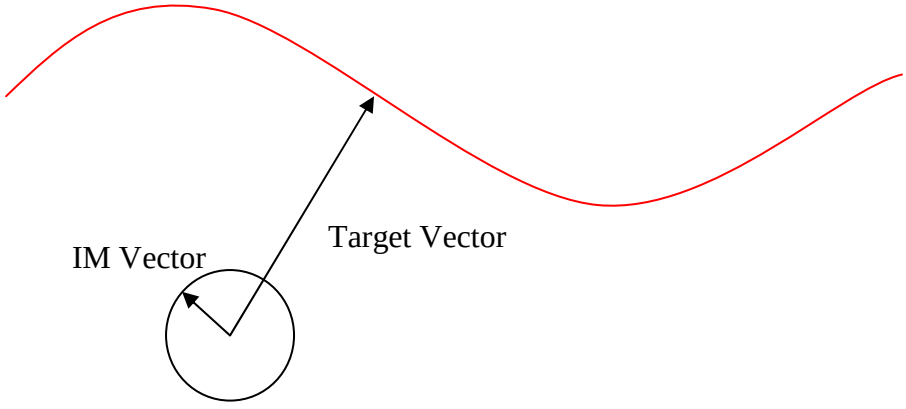

Figure 24: Directional control

The critical issue in the calculation of the IM arm position pertains to implementing the relevant reference frames. The *target vector* displayed in Figure 24 is defined in the inertial reference frame [17] in which the equations (3.1) and (3.2) correspond to the x- and y-components of the vector.

$$\Delta x_I = x_{T\arg et} - x_{Current} \tag{3.1}$$

$$\Delta y_I = y_{T\arg et} - y_{Current} \tag{3.2}$$

In order to relate this vector with the IM vector, it should be transformed to the body frame reference system [17]. The matrix T_{BI} represents the transformation matrix from inertial frame to the body frame, assuming a two axis system. In equation (3.3), ψ is the yaw angle of the Hopping Rotochute.

$$T_{BI} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \tag{3.3}$$

Consequently, the *target vector* is defined in the body reference frame as follows:

$$\begin{Bmatrix} \Delta x_B \\ \Delta y_B \end{Bmatrix} = T_{BI} \begin{Bmatrix} \Delta x_I \\ \Delta y_I \end{Bmatrix} \tag{3.4}$$

Finally, the angular position of the transformed vector is calculated as follows:

$$\theta_{t\arg et} = \tan^{-1}\left(\frac{\Delta y_B}{\Delta x_B}\right) \tag{3.5}$$

Hence, θ_{target} becomes the target angular position for the IM arm, concluding the calculation of the second control command pertaining to the directional control.

3.2.4 Step 4: Assess the Uncertainty

The algorithm described in preceding sections does not employ any uncertainty; in other words each step assumes perfect knowledge pertaining to the vehicle and environment. This assumption does not represent the reality. The uncertainty is generally driven from the environment and/or the sub-systems of the vehicle. This thesis assumes the uncertainty included in the followings:

1. The position data: The sensors may not give accurate measurements.
2. RPM of the rotor system: The mechanical components may cause fluctuation from the desired RPM.
3. Pulse width of the rotor system: The mechanical components may not operate for the desired duration.

Previous sections defined the noise and bias terms as mechanical/instrumental and environmental uncertainty, respectively. The noise terms emulate the uncertainty driven per hop, and the bias terms emulate the uncertainty driven per simulation. The lack of knowledge about the noise/bias characteristics requires the use of random numbers with a triangular distribution, which is used widely to model unknown variables when minimum, maximum and most likely values are known. Consequently, the position (x,y), RPM, and the pulse width parameters with uncertainty are described in equations (3.6) through (3.9). In these equations, η, υ, ω are the random values modeled by triangular distribution to represent the uncertainty. The repeated simulations will be carried out with the defined uncertainty and the statistical results will be obtained by Monte Carlo simulations.

$$x + \eta_{noise} + \eta_{bias} \tag{3.6}$$

$$y + \eta_{noise} + \eta_{bias} \tag{3.7}$$

$$RPM + \upsilon_{noise} + \upsilon_{bias} \tag{3.8}$$

$$t_{app} + \omega \tag{3.9}$$

3.3 Summary

The most recent version of Hopping Rotochute lacks autonomous trajectory-following. This research gap establishes the foundation of this thesis. Due to the dynamic behavior of a Hopping Rotochute and the presence of uncertainty driven by the vehicle and/or the environment, this thesis proposes a methodology that combines a rule-based algorithm with some prediction models. The methodology starts by creating the regression models of hop distance and hop altitude. Based on the user input and scenarios, the optimum pulse width is calculated. Subsequently, the target point is decided and the second control command, which is the angular position of the internal mass arm, is computed. Finally, the hop is conducted.

CHAPTER 4

IMPLEMENTATION OF THE PROPOSED METHODOLOGY

This chapter details the implementation of the proposed methodology described in Chapter 3. The focus of this chapter is on the regression models mentioned in steps 1 and 3 of the proposed methodology. Moreover, this chapter involves some example simulations and the statistical results of various scenarios.

4.1 Regression Models

Chapter 3 introduced the proposed methodology consisting of regression models allowing the methodology to approach the problem from a model predictive control perspective. This thesis creates three models that are explained in the following sections.

4.1.1 Hop Distance Model

The *hop distance model* provides the information of how far the vehicle can hop with a specific pulse width and RPM. Therefore, the model requires the generation of some data points that correspond to a single hop's range associated with its control inputs. In the generation of the data points, the selection of RPM and pulse width range is crucial. As an initial implementation, the previous work [8] has been reviewed and a range of 3000 to 4000 is selected for the RPM, and 0 to 1 second is selected for pulse width. Based on the range selections, the data points are generated and the maximum hop altitude of each data point is observed. The author of reference [8] has emphasized that the optimum hops for the maximum total range are the ones with smaller altitudes. He proves his claim by conducting some trade studies. For instance, Figure 25 illustrates the summary of a particular study in which the effect of altitude on the total range is investigated. Figure 25 shows that the maximum total range can be obtained by smaller hops compared with larger hops. An altitude of 2 m is particularly the optimum altitude for obtaining the maximum total range. Moreover, the internal mass (m_{IM}) has been

varied in the trade studies, and the same trends have been observed as shown in Figure 25.

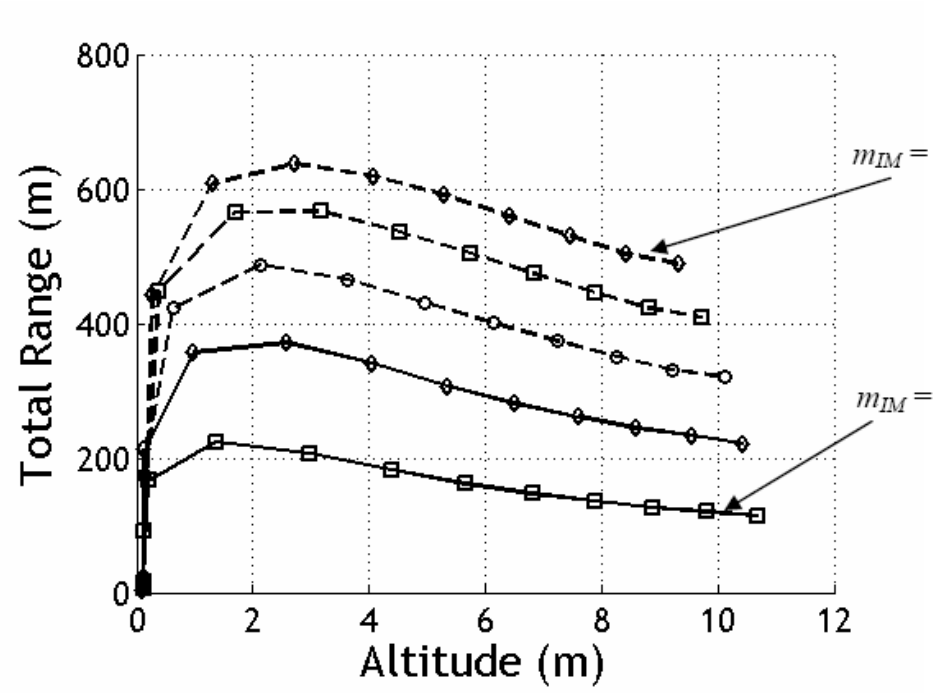

Figure 25: Total range vs. maximum altitude [8]

Based on the results of the previous work, this thesis assumes the maximum hop altitude as 2 m that is a reasonable value in an indoor environment. Thus, the maximum hop altitude of 2 m is accepted as a limit altitude in this thesis. As mentioned in the former section, the generated data are obtained by varying RPM from 3000 to 4000, and the pulse width from 0 to 1 second. These data are filtered with respect to the limit altitude of 2 m. Hence, the regression models are created from the filtered points having RPM interval from 3000 to 3500, and the pulse width interval from 0 to 0.6 second. Finally, Figure 26 presents the 3D trend of a 2nd order polynomial model for the hop distance, and Table 3 displays the properties of the model.

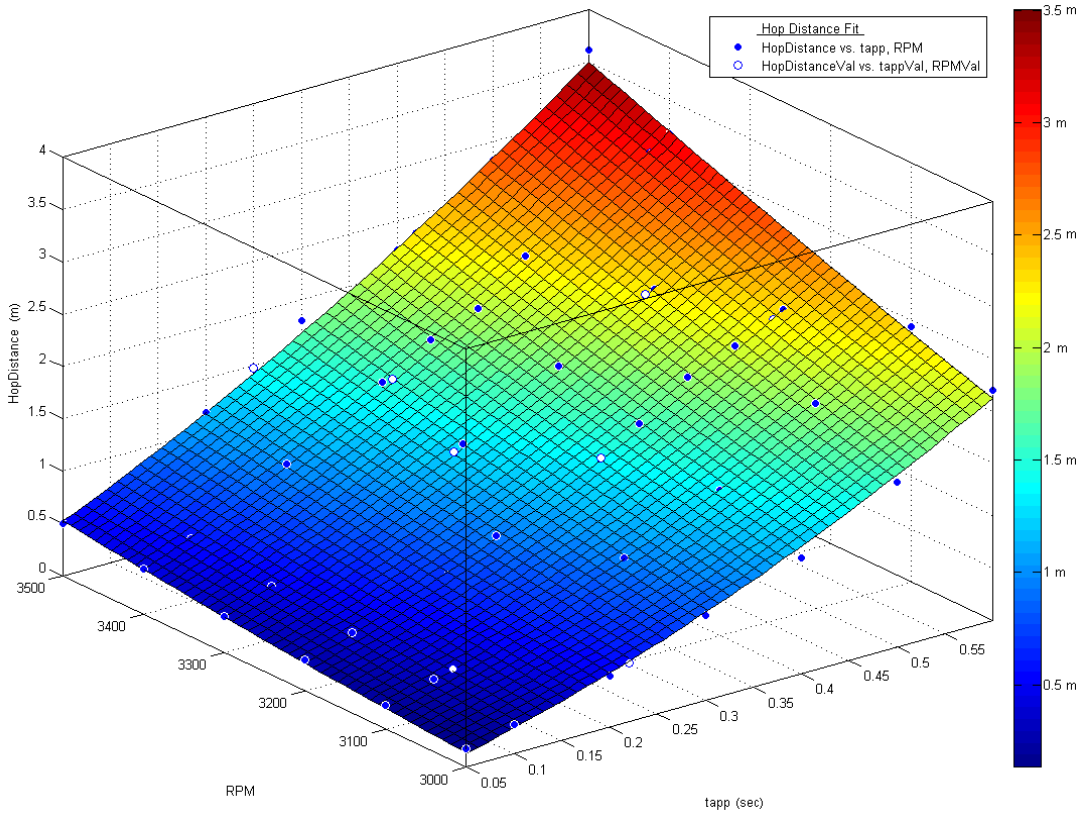

Figure 26: The fit of the hop distance model

Table 3: Hop distance model properties

Total Number of Model Points	52
Total Number of Validation Points	10
Range of RPM	3000 – 3500
Range of Pulse Width	0.05 – 0.6
R^2	0.9940

In order to analyze the goodness of the hop distance model fit, some random points are selected for the validation case. The regression model is created by 52 points, and 10 random points are generated for the model validation. Figure 27 and Figure 28 illustrate the residuals of the model and the validation data points with respect to the input parameters. The results show that the residuals are in the range between 13 and -28 cm, which correspond to a relative error with respect to the hop distance as 6.7% and -10.7%, respectively.

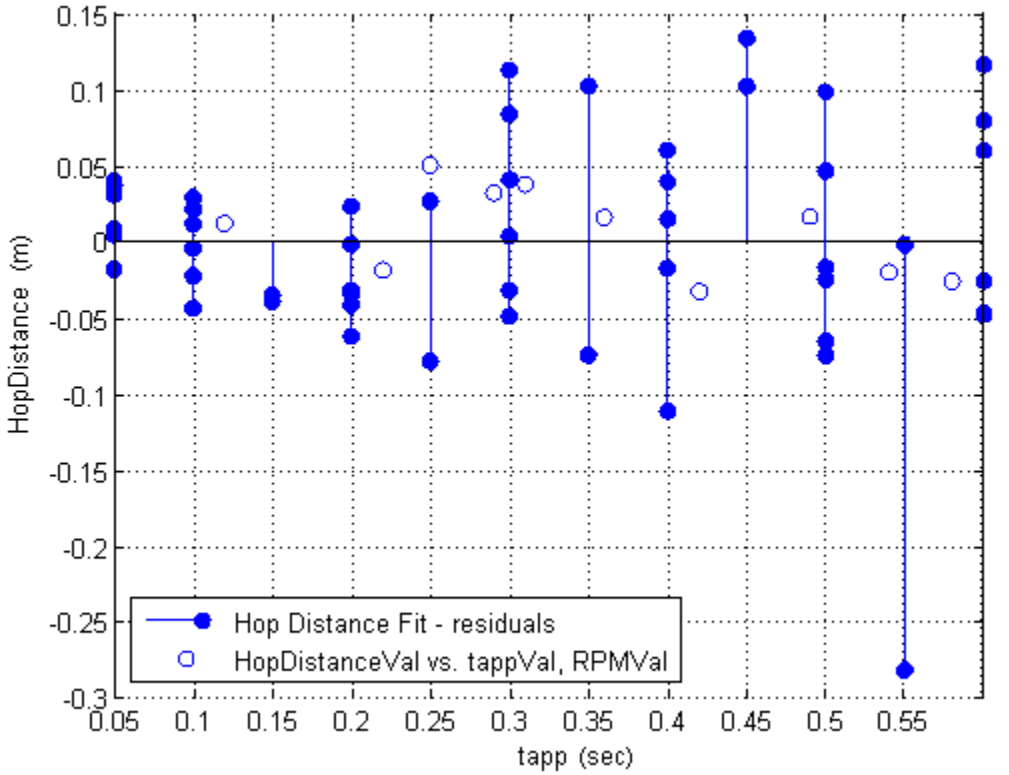

Figure 27: Residual plot of hop distance model vs. pulse width

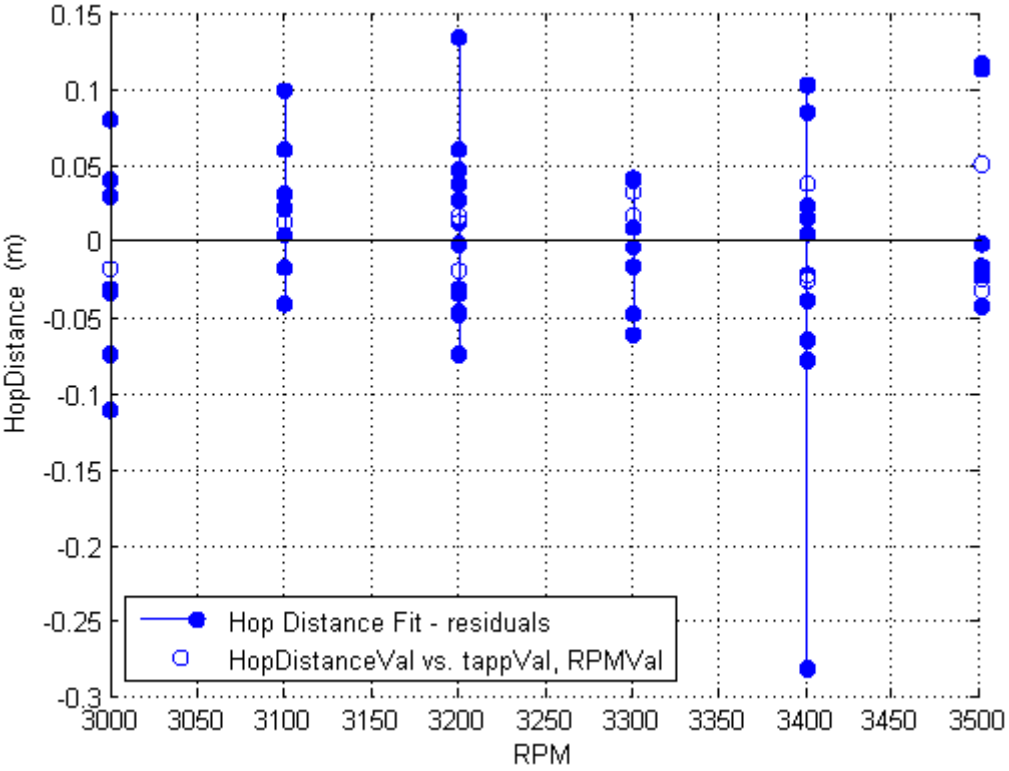

Figure 28: Residual plot of hop distance model vs. RPM

The previous figures indicate that only one point corresponds to the maximum residual (approximately -30 cm) when RPM is 3400, and the pulse width is 0.55 seconds. The residuals associated with the remaining points range from 15 cm to -10 cm implying a relative error with respect to the hop distance between 10% and -10%, respectively. Hence, the model fit is good. Moreover, one can analyze the characteristics of the hop distance with respect to the control parameters by looking at the contour plot displayed in Figure 29. The hop distance is linearly increasing with respect to the increase in pulse width and/or RPM, seen from the similar widths and linear edges of the contours in Figure 29. This is a physically anticipated result since the increase in RPM or pulse width result in more thrust in the rotor system. Thus, the vehicle travels more distance per hop with higher thrust values.

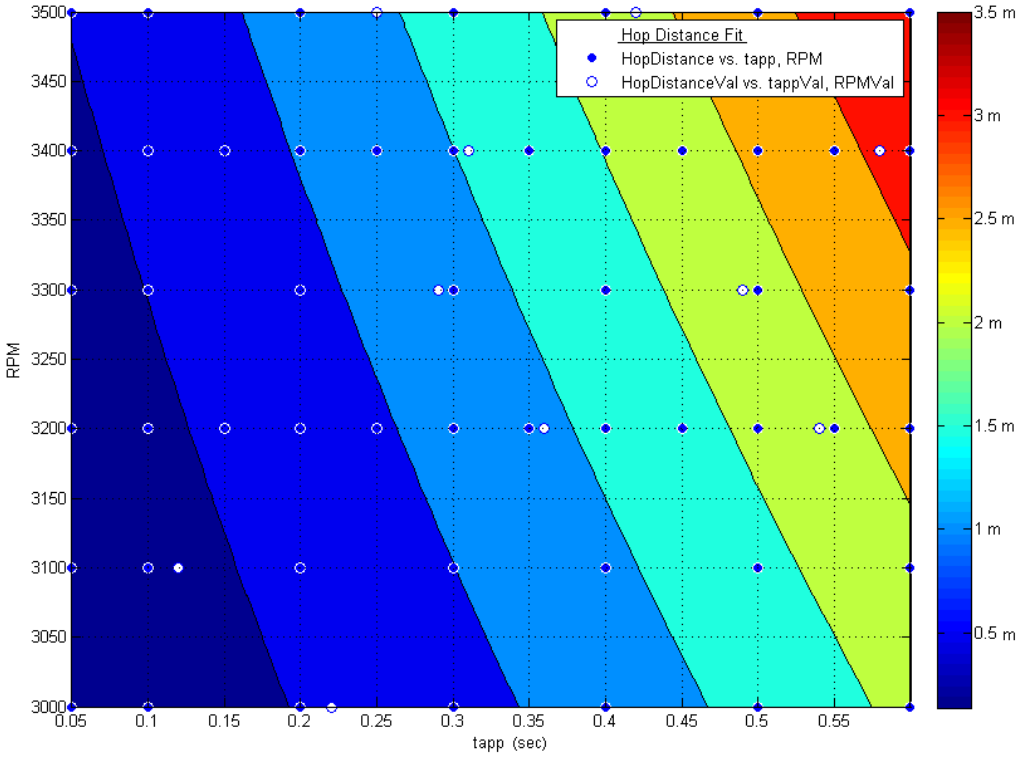

Figure 29: Contour plot of the hop distance model

Furthermore, another interpretation of Figure 29 pertains to the sensitivity of the hop distance to the control commands. In this figure, the hop distance tends to vary more when the pulse width is changed with constant RPM since it is likely to observe different contours, implying more change in the hop distance. On the other hand, when the pulse width is fixed and RPM is varied, the number of different contours observed is less than the previous case. Hence, the hop distance is more sensitive to the pulse width than RPM.

4.1.2 Maximum Hop Altitude Model

The second important parameter of the hop performance is the maximum hop altitude. The *hop altitude model* presented in Figure 30 answers the question of how high the vehicle can hop for particular control commands. Like the *hop distance model*, the same procedure is implemented to decide on the ranges of RPM and the pulse width. For this analysis the same data points are used to create the model. The only difference is using the hop altitudes and its corresponding control inputs instead of the hop distance. Hence, Figure 30 displays the 3D trend of a 2^{nd} order polynomial fitting for the maximum hop altitude, and Table 4 displays the properties of the model.

Table 4: Maximum hop altitude model properties

Total Number of Model Points	52
Total Number of Validation Points	10
Range of RPM	3000 – 3500
Range of Pulse Width	0.05 – 0.6
R^2	0.9996

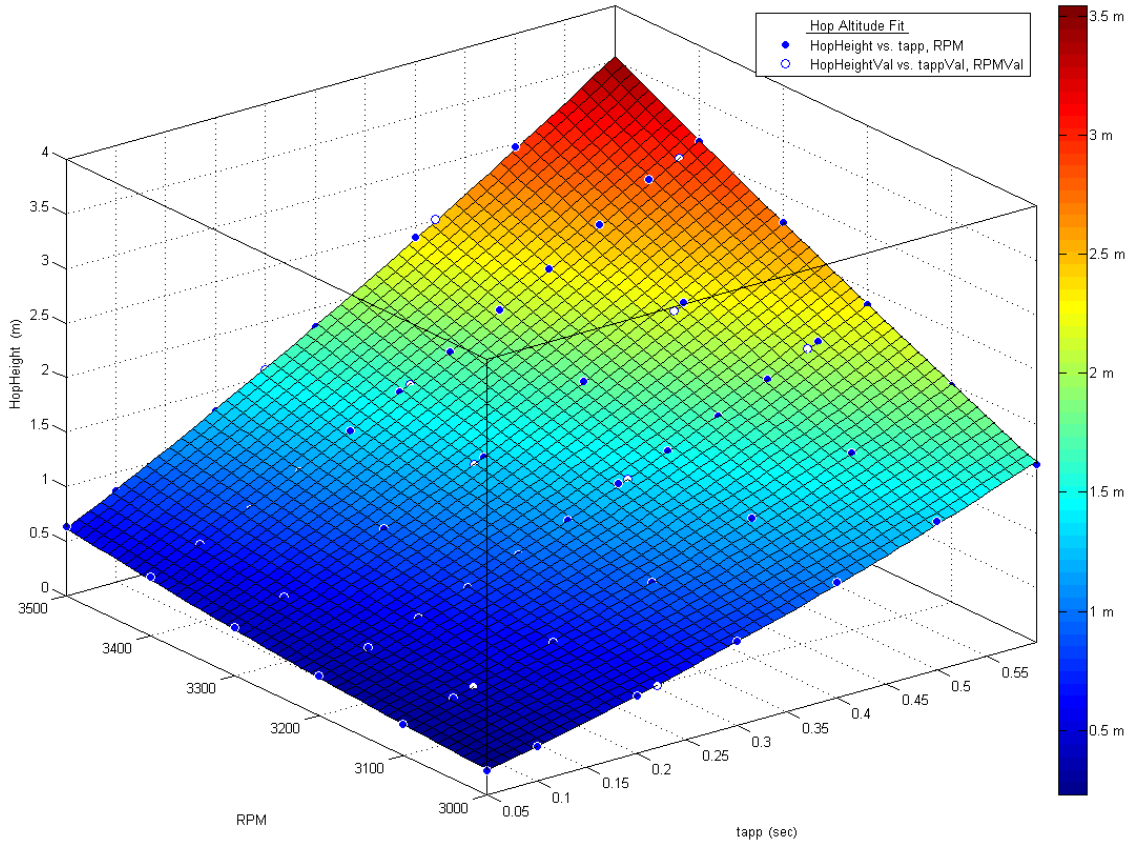

Figure 30: The fit of the maximum hop altitude model

Like the *hop distance model*, 52 points are used for the model regression and 10 random points are used for the model validation. In order to analyze the goodness of model fit, the residuals of the model and validation data points with respect to input parameters are plotted in Figure 31 and Figure 32. The figures show that the model fit is substantially good since the residuals are in the range between 4.6 cm and -4.5 cm, which correspond to a relative error with respect to maximum hop altitude as 1.6% and -1.3%, respectively.

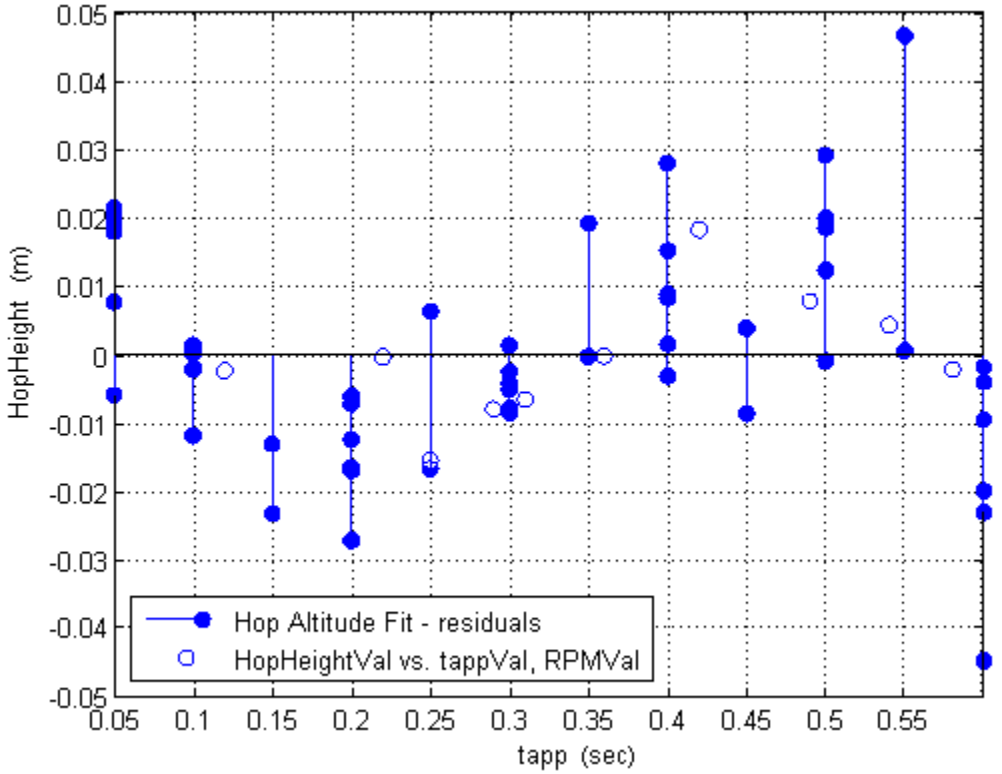

Figure 31: Residual plot of maximum hop altitude model vs. pulse width

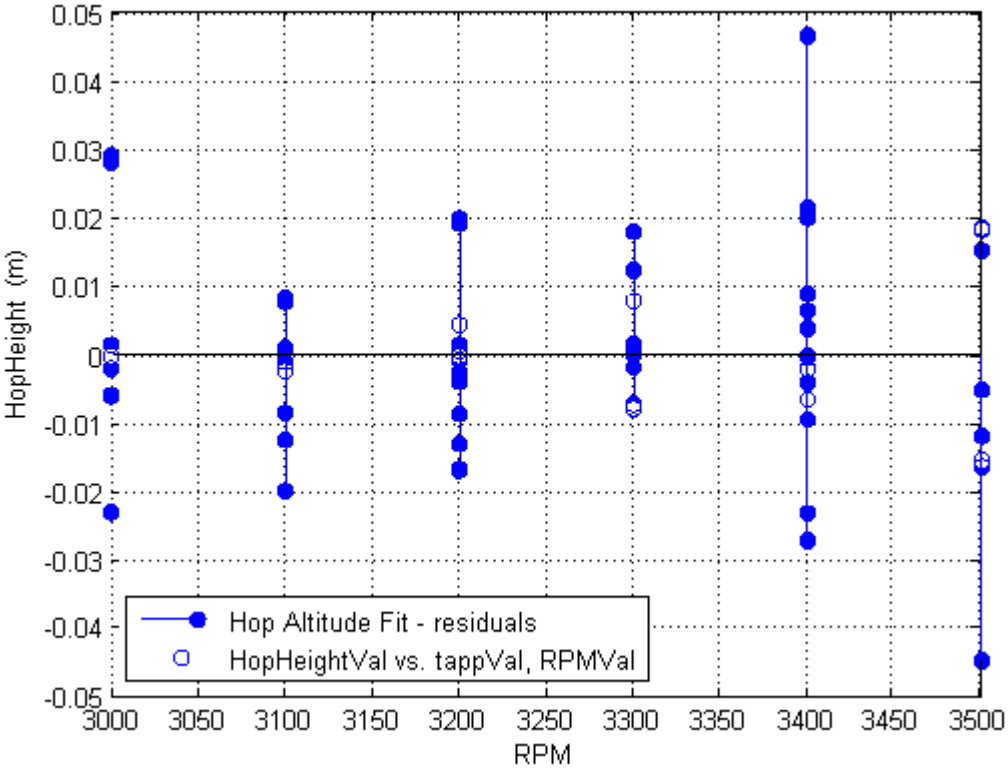

Figure 32: Residual plot of maximum hop altitude model vs. RPM

Furthermore, the characteristics of the hop altitude are analyzed with respect to the control commands, namely RPM and the pulse width. When the pulse width and RPM are increased, the hop altitude increases approximately following a linear trend. This can be seen from Figure 33 in which the contour areas indicate no significant change.

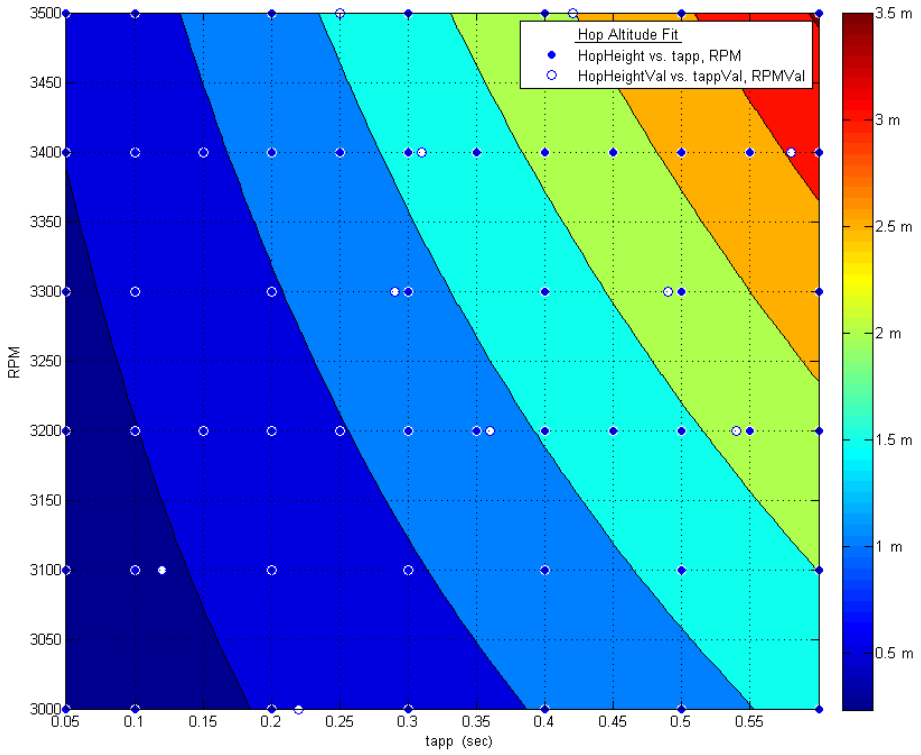

Figure 33: Contour plot of the maximum hop altitude model

Moreover, the hop altitude is more sensitive to the pulse width than RPM. In Figure 33, if the pulse width is varied when RPM is fixed, various contours are observed. On the contrary, if RPM is varied when the pulse width is fixed, less numbers of contours are observed than the previous case. Hence, the change in pulse width results in more significant changes in hop altitude.

4.1.3 Wind & Hop Altitude Effect Model

The two models discussed in sections 4.1.1 and 4.1.2 provide information about how far and how high a Hopping Rotochute can travel per hop on horizontal and vertical

planes. Note that these models assume no wind. The aim of the *wind & hop altitude effect model* is to provide information about the error with respect to the hop altitude and the wind velocity. In other words, this model analyzes how much the wind velocity and the hop altitude are affecting the error of a single hop. Here, it is important to define the term error (ε), which is the distance between the target point and the actual landing point, as depicted in equation 4.1:

$$\varepsilon = \sqrt{(x_{target} - x_{actual-landing})^2 + (y_{target} - y_{actual-landing})^2} \tag{4.1}$$

As mentioned before, one can expect to observe a non-zero error at the end of each hop due to uncertainty. If the Hopping Rotochute is subjected to wind, then the error becomes larger with respect to the strength of the wind. Here, the critical information pertains to the value of error (and the errors larger than that value) ensuring the presence of wind. Hence, the goal of the *wind & hop altitude effect model* is to interpret a threshold that can differentiate whether the Hopping Rotochute is subjected to wind. This goal enables the development of a gust-tolerant trajectory-following algorithm.

A regression model is created for the error (ε) with respect to the hop altitude and the wind velocity. In order to visualize the model in three dimensions, the wind directions of 0 and 180 degrees are taken into account. The experiments for data generation are conducted in a horizontally pre-planned trajectory. The wind directions of 0 and 180 degrees represent the tail and head winds respectively, and they imply the least and the most challenging missions. Hence, the consequences of wind directions other than 0 and 180 degrees are assumed to be included inside the error data. Furthermore, some experiments with variable wind magnitudes were conducted in a simulation environment. In cases in which the head wind magnitude was around 1m/s, the simulation code must be broken since a Hopping Rotochute could not approach the final point. Hence, the data

generation assumes a wind magnitude between -0.5 and 0.5 m/s. Wind magnitudes larger than 0.5 m/s and smaller than 1 m/s could be used; however, the following sections will emphasize that this model specifically investigates a threshold value indicating the presence of wind. Therefore, the selected range of a wind magnitude of -0.5 to 0.5 m/s is convenient for the threshold investigation.

Figure 34 displays the 3rd order polynomial fitting for the error with respect to the wind velocity and maximum hop altitude, and Table 5 illustrates the properties of the model. A total of 40 simulations were carried out to generate the data points, and the wind magnitude was selected randomly in each simulation inside the range of -0.5 to 0.5 m/s.

Table 5: Properties of the error model

Total Number of Model Points	198
Range of Wind Magnitude (m/s)	-0.5 – 0.5
R^2	0.9677

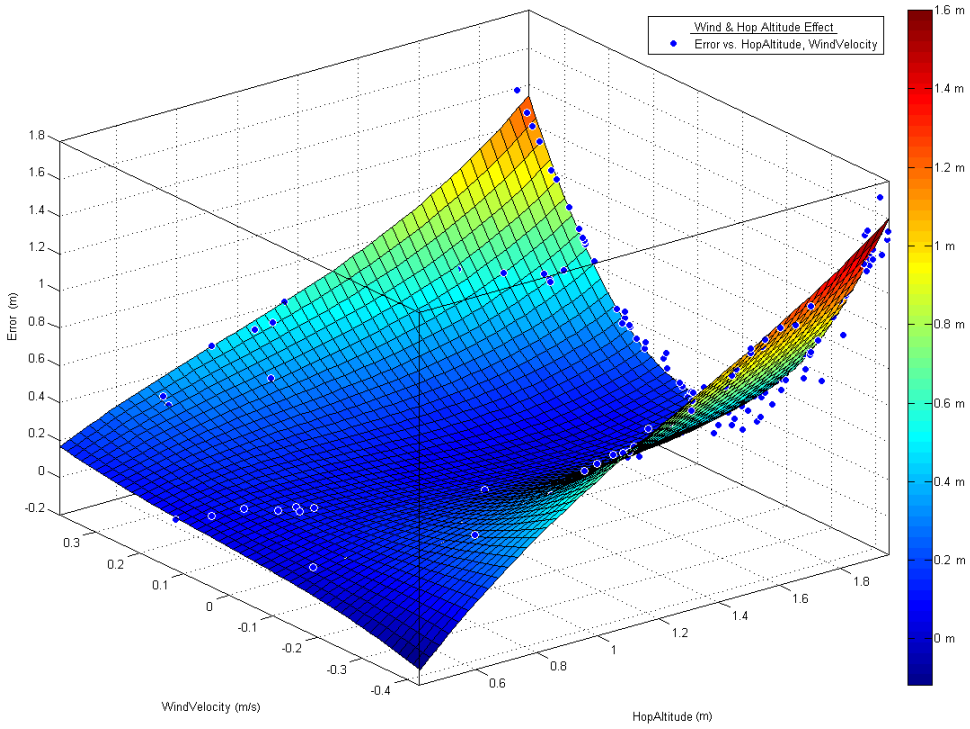

Figure 34: The fit of the wind and hop altitude effect model

The trends of Figure 34 are shown in more detail in the next three figures. Figure 35 shows the error with respect to the wind velocity. A parabolic trend is observed between the error and the wind velocity. As anticipated, larger wind magnitudes lead to more error. Hence, the smallest error magnitude is observed when the wind velocity is zero. As shown in Figure 35, the error in the absence of wind, implying that the error is due to pure uncertainty, is at most 18 cm. To indicate the presence of wind, this thesis selects a stricter, more conservative threshold value of 14 cm instead of 18 cm. In other words, the selection of this threshold value will ensure that whenever the error is larger than 14 cm, the vehicle is exposed to wind. Figure 36 pertains to the error with respect to the hop altitude. As the figure shows, high altitude hops result in substantially more error, indicating that the Hopping Rotochute stays in the air and is exposed to wind for a longer amount of time. Figure 37 pertains to the contour plot of the *wind & hop altitude model*. This figure shows that high hop altitudes and large wind velocities lead to maximum error, seen at the top and bottom right of Figure 37.

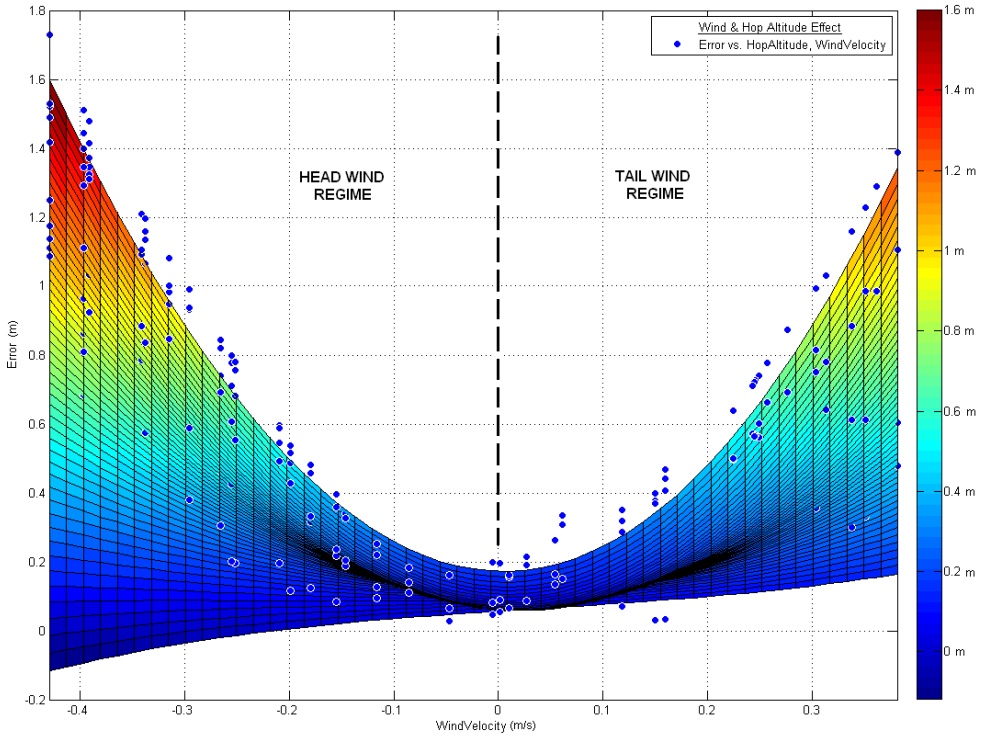

Figure 35: Position error vs. wind velocity

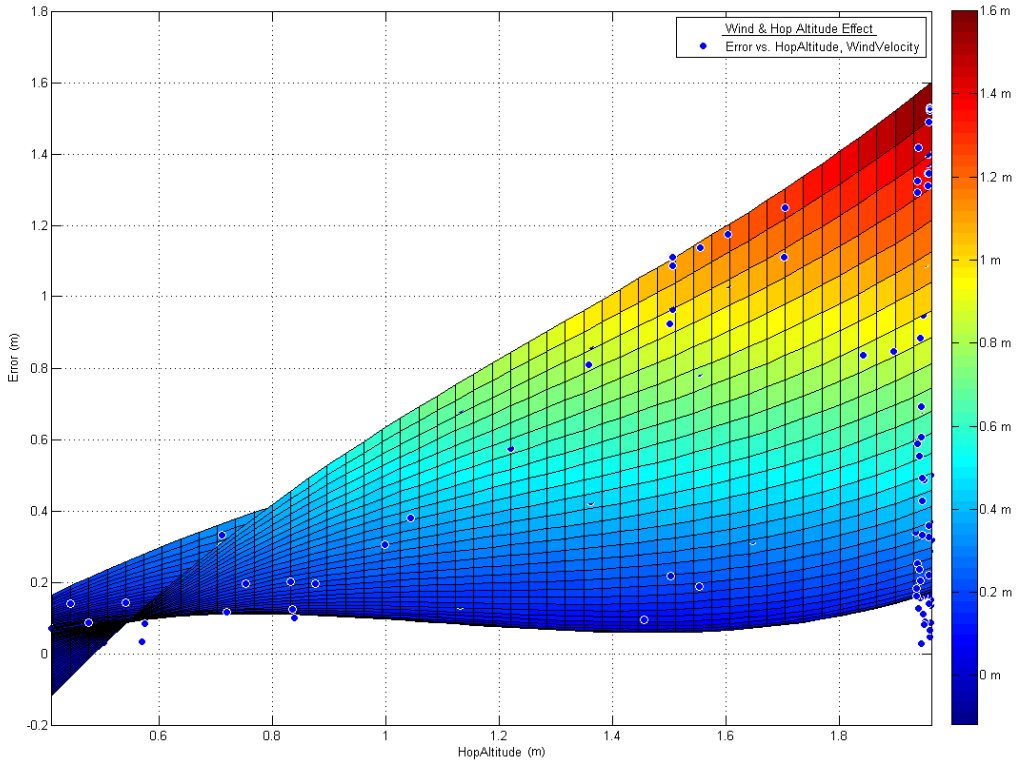

Figure 36: Position error vs. maximum hop altitude

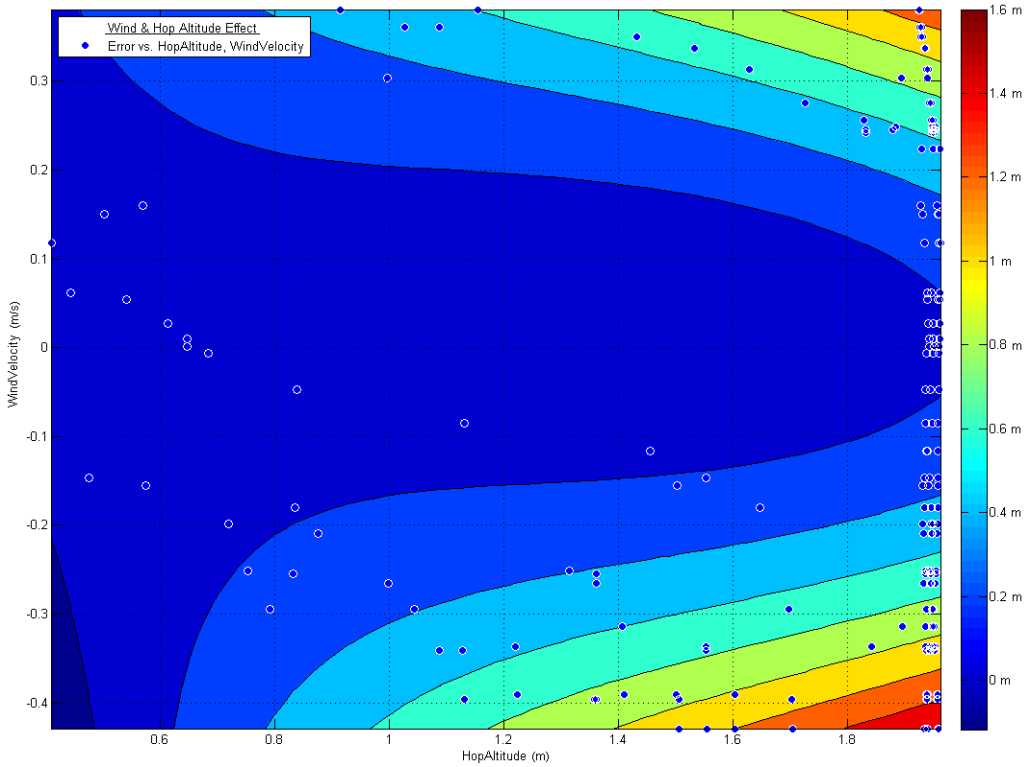

Figure 37: Contour plot of the wind & hop altitude effect model

An interesting result occurs in Figure 37 when the hop altitude is reduced for a particular wind velocity. In cases involving large wind velocities, smaller errors can be sustained by only decreasing the altitude of the hops. Thus, hopping small distances implies small errors. This tradeoff represents a significant compromise for the vehicle in terms of accuracy and the mission completion time, and can be rephrased as follows:

If a Hopping Rotochute is traveling in a gusty environment, an accurate trajectory-following is satisfied by hopping small distances, lengthening the mission completion time.

The *wind & hop altitude effect model* provides two major pieces of information, (1) the threshold value of *14 cm*, which is able to differentiate whether the vehicle is subjected to wind; and (2) the occurrence of smaller errors, which can be maintained by smaller hops in a gusty environment. The threshold and smaller errors construct the basis of the development of the gust-tolerant trajectory-following algorithm.

4.2 Example Simulations

The methodology discussed in this thesis proposes an autonomous trajectory-following algorithm for a Hopping Rotochute. The details of the methodology have been described until now. This section will present example simulations in which the proposed methodology has been implemented. The aim of this section is to test the proposed methodology. If the results of the test simulations are reasonable, then more challenging scenarios will be employed. Thus, a route is created for this section in which the pure algorithm as described in step 2 of Chapter 3, and the obstacle avoidance algorithm are tested. Note that the algorithm with obstacle avoidance takes into account the boundaries surrounding the trajectory, and aims to follow the trajectory inside the boundaries. On the

other hand, the pure algorithm only aims to find the most optimum target point with maximum range.

First, the results with the pure algorithm are presented as shown in Figure 38. In this figure, the bold solid line pertains to the trajectory of the vehicle and the thin solid line represents the desired trajectory. In addition, the boundaries are shown by the dotted lines surrounding the desired trajectory. Note that the boundaries shown in Figure 38 may not necessarily be a physical boundary. The aim of showing them is to observe whether the Hopping Rotochute tends to violate any of the boundaries. As it is seen from Figure 38, the Hopping Rotochute is mostly travelling out of the boundaries. Hence, the pure algorithm is not very successful in the presence of boundaries even though the Hopping Rotochute completes the mission at the desired end point.

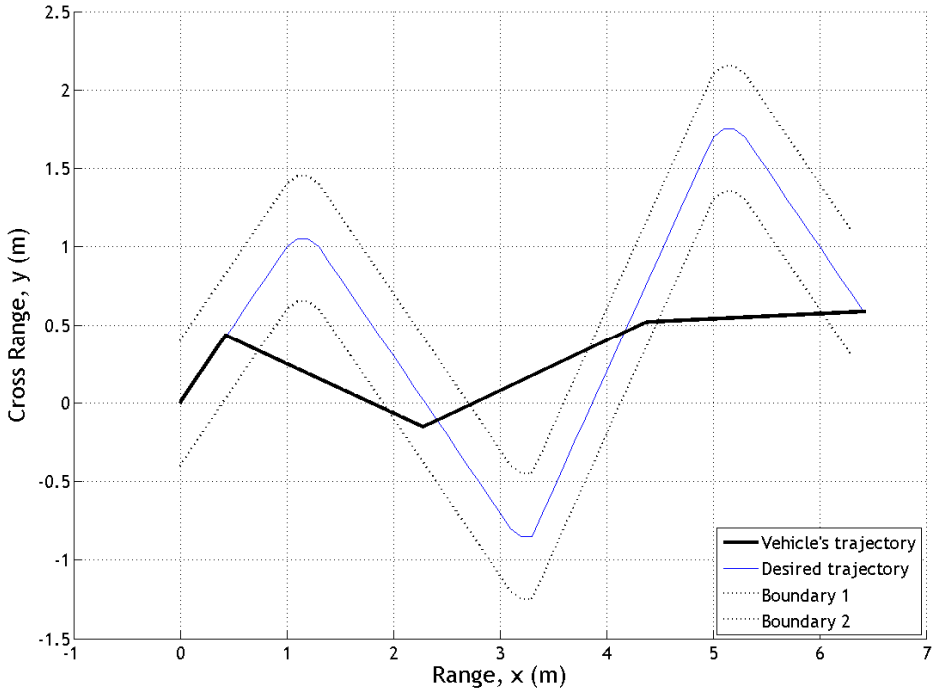

Figure 38: Trajectory-following with the pure algorithm

Figure 39 illustrates the angular position of the IM arm and the actual hop angular position. Note that the actual hop angular position is calculated with respect to the actual landing and starting point of a particular hop in the following equation:

$$\tan^{-1}\left(\frac{y_{landing} - y_{start}}{x_{landing} - x_{starts}}\right) \tag{4.2}$$

In Figure 39, the solid line represents the desired angular position of the IM arm, and the dashed line represents the actual hop angular position. These terms explain that the IM arm is positioned to a desired angle; however, the vehicle is not exactly traveling at the desired angle due to uncertainty, which causes some fluctuations. At the beginning of each simulation, the IM arm is initialized to 0 degrees. Note that 0 degrees of the IM arm position implies traveling on the x-axis in Figure 38. For the first hop, the algorithm computes the desired position of the IM arm as 45 degrees, seen from Figure 39.

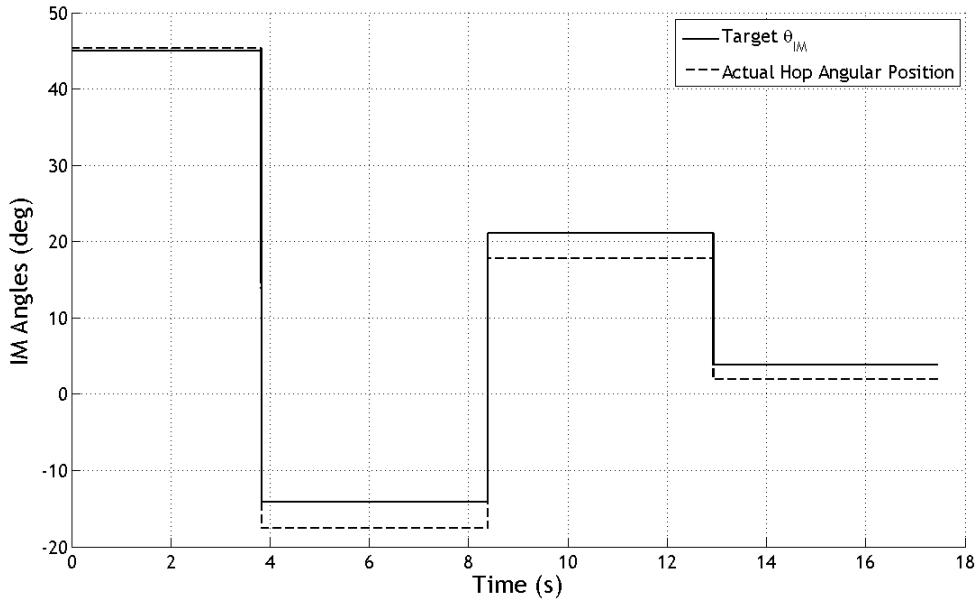

Figure 39: IM arm position with the pure algorithm

For the second hop, the proposed methodology computes the target IM arm position as -15 degrees as shown in Figure 39. Note that the IM arm position in this figure provides information about which angle the IM arm needs to be positioned. In

other words, it is not the required rotation of the IM arm. For instance, the IM arm needs to be rotated -60 degrees in order to switch its position from 45 to -15 degrees.

As it can be seen from the second hop (second discrete part of the solid line in Figure 39), the actual hop angular position is attained around -18 degrees due to uncertainty. Figure 40 presents additional information about the altitudes of the hops. In this particular simulation, the Hopping Rotochute completes its mission with four hops. The first is smaller than the others because the algorithm checks whether the vehicle is exposed to wind. The small hop specifically prevents large deviations at the beginning of the mission.

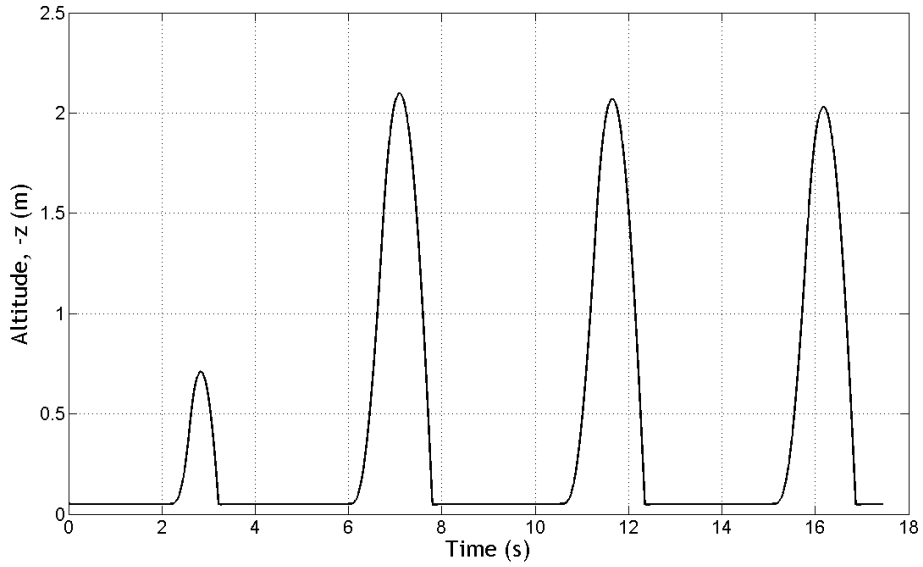

Figure 40: Altitude plot with the pure algorithm

Secondly, the same route is implemented in the algorithm with obstacle avoidance, as in Figure 41. In this figure, the bold solid line pertains to the trajectory of the vehicle, and the thin solid line pertains to the desired trajectory. Likewise, the dotted lines represent the boundaries. As the figure shows, the algorithm takes into account the boundaries in the selection of the next hopping point. Hence, the vehicle stays inside the

boundaries. Compared with Figure 38, Figure 41 presents a more accurate trajectory-following.

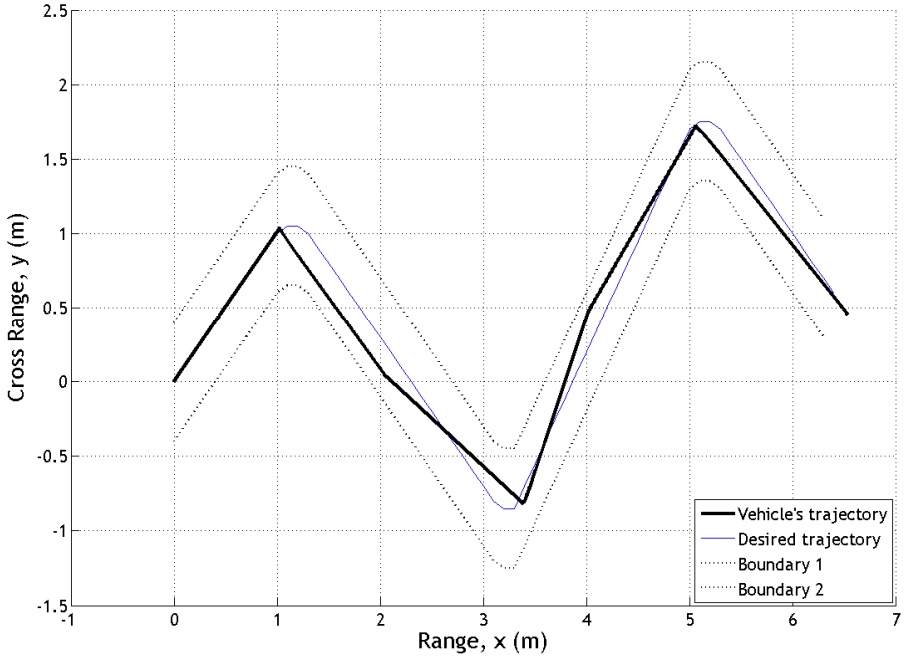

Figure 41: Trajectory-following with the algorithm that includes obstacle avoidance

Figure 42 shows the angular position of the IM arm represented by a solid line and the actual hop angular position represented by a dashed line. Once again, in these positions small fluctuations are observed. Moreover, Figure 43 illustrates the altitude plot of this simulation, in which the Hopping Rotochute completes its mission with eight hops. (Recall that the first simulation ended the mission after four hops on the same route). The first hop in Figure 43 is a small hop. If wind is present, the small hop creates a small deviation at the beginning of the mission, which is preferable because it prevents divergence from the route. Furthermore, the last hop is at low altitude because only a small distance remains for the last hop. The rest of the hops aim to go as far as possible under the assumption of staying inside the boundaries.

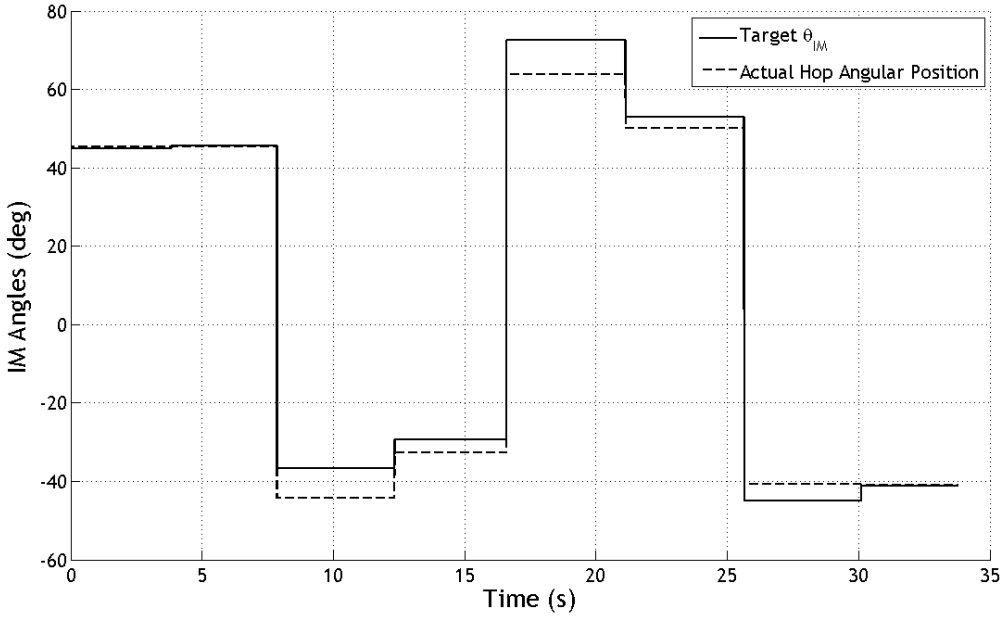

Figure 42: IM arm position with the algorithm that includes obstacle avoidance

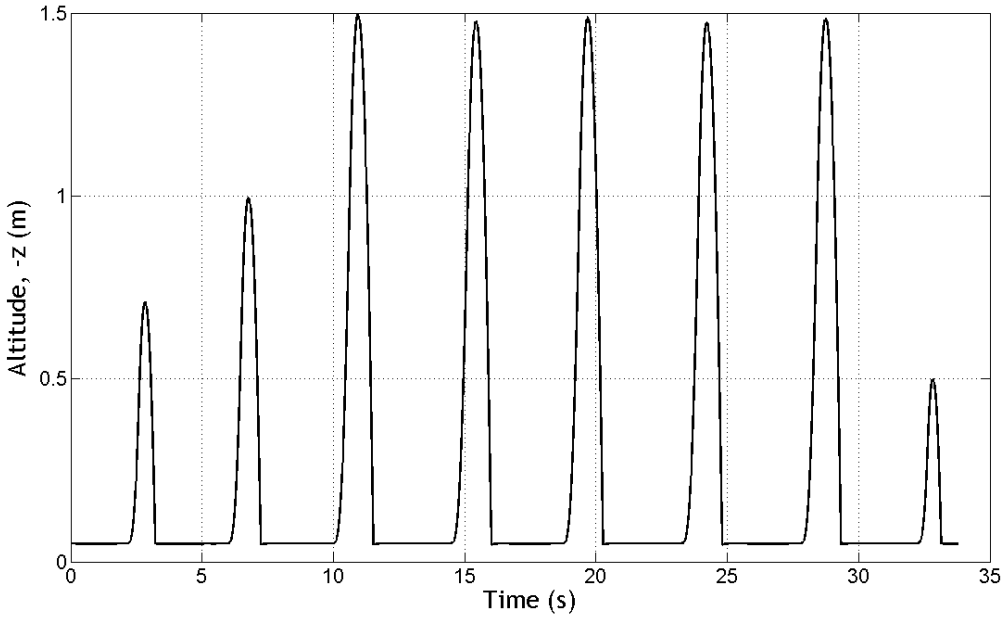

Figure 43: Altitude plot with the algorithm that includes obstacle avoidance

The results of the example simulations show that the trajectory-following algorithm advances a Hopping Rotochute to track any pre-planned route autonomously, implying the absence of human interaction for control command calculation throughout the mission. However, a tradeoff between the accuracy of the trajectory-following and the mission completion time occurs.

As the accuracy of the trajectory-following increases, the number of hops, or, the mission completion time, also increases.

4.2.1 Statistical Results

The performance of the trajectory-following algorithm is evaluated via repeated simulations in the existence of uncertainty. This section presents the simulation results and depicts the statistical performance of the developed algorithm. The uncertainty is assumed to be present in the position data, RPM, and the pulse width since the rotor system and positioning sensor are some of the major contributors of uncertainty. However, it is difficult to assess all sources of uncertainty. Therefore, they are modeled as independently identically distributed (i.i.d) random variables with triangular distribution. Triangular distribution is used in random variable modeling when only minimum, maximum, and most likely values are known. In the simulations, uncertainty is modeled as shown in Table 6. For a particular system, the distribution parameters proposed in Table 6 can be modified based on the availability of further information on sensors and actuators.

Table 6: Uncertainty parameters

UNCERTAINTY PARAMETER	UNIT	MINIMUM	MOST LIKELY	MAXIMUM
x_{noise}	m	-0.1	0	0.1
x_{bias}	m	-0.1	0	0.1
y_{noise}	m	-0.1	0	0.1
y_{bias}	m	-0.1	0	0.1
RPM_{noise}	-	-25	0	25
RPM_{bias}	-	-25	0	25
t_{app_noise}	sec	-0.05	0	0.05

Figure 44 shows the distributions of the generated noise and bias terms of position data. For each term, minimum possible value is -0.1 m, the most likely value is 0 m, and the maximum possible value is 0.1 m. It is intended to conduct the first experiments with VICON motion capture system, which will emulate an onboard position sensor by recording motion of the vehicle in real time. This system's accuracy is claimed to be at as low as 0.001 m [21]. Therefore, the most likely value is chosen around 0 m while the maximum and minimum possible values are chosen 100 times larger than the accuracy of the motion capture system in order to include extreme situations. This thesis assumes "100" as a convenient number since -0.1 m to 0.1 m (-10 cm to 10 cm) is a reasonable interval for position error.

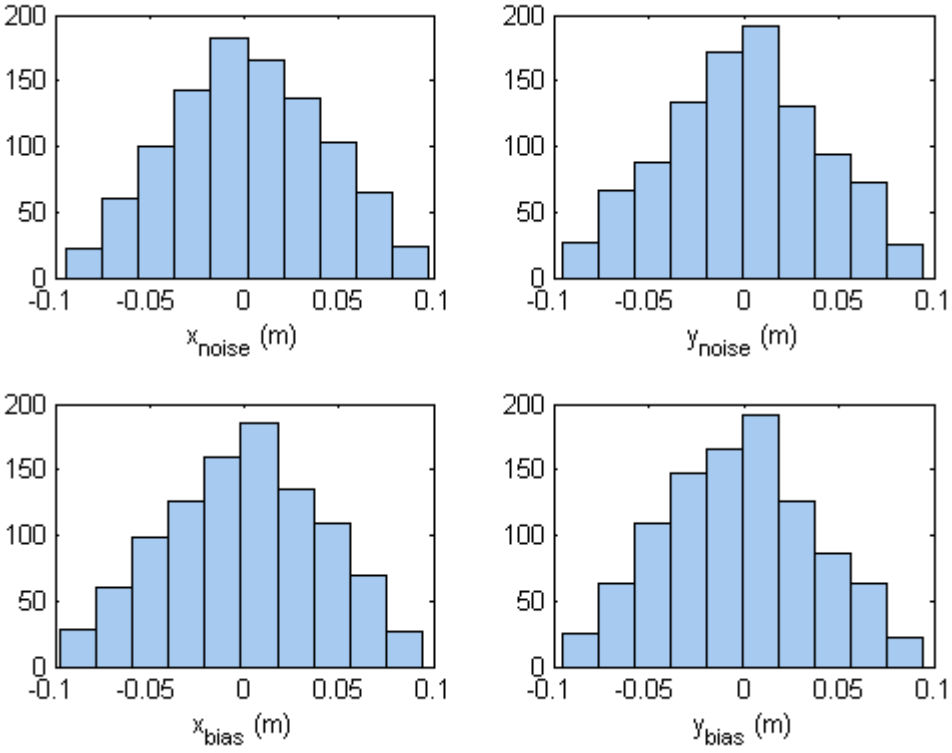

Figure 44: Noise and bias distributions of position data

In addition, the noise terms shown in Figure 44 characterizes the uncertainty that vary at each hop whereas the bias terms change for each simulation, and remain constant throughout a particular simulation. In this scheme, the instrumental or mechanical uncertainty caused by sensors or rotating components respectively are embedded in the noise terms while the environmental conditions such as temperature, humidity, day/night are included in the bias terms.

Similarly, Figure 45 illustrates the distributions of generated RPM noise, RPM bias, and pulse width noise. For the RPM terms, the minimum possible value is -25, the most likely value is 0, and the maximum possible value is 25. This implies a total RPM uncertainty in the range between -50 and 50, in which the summation of bias and noise terms is taken into account. It is mentioned that the maximum RPM of the most recent

Hopping Rotochute is 4000. This thesis assumes the total uncertainty approximately 1% of maximum RPM, which represents a reasonable value.

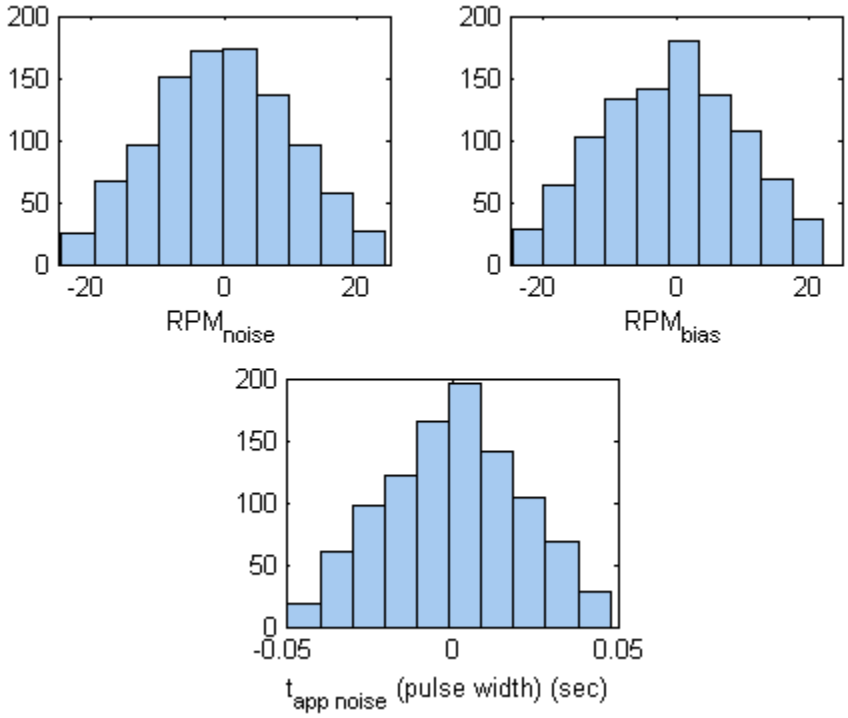

Figure 45: Uncertainty distributions of RPM and pulse width

For pulse width noise, the minimum possible value is -0.05 sec, the most likely value is 0 sec, and the maximum possible value is 0.05 sec. Note that Figure 29 and Figure 33 depict a maximum hop distance and hop altitude less than 50 cm with 0.05 sec of pulse width. Moreover, the pulse width values less than 0.05 sec result in unsuccessful hops since they provide insufficient lift for a Hopping Rotochute. Hence, the values less than 0.05 sec are assumed as possible uncertainty for a particular pulse width. Lastly, the pulse width does not have a bias term since it is assumed independent of environmental conditions.

Repeated simulations are performed to demonstrate how successful the vehicle hops to the target point in the existence of uncertainty. An evaluation criterion is created as “simulation error” to estimate the statistical performance of the algorithm. For a

particular hop, the error (ε) was defined as the distance between the target point and the actual landing position. For a single simulation with n hops, the simulation error ($\bar{\varepsilon}$) is computed as the average of errors (ε) as shown in equation (4.3).

$$\bar{\varepsilon} = \frac{1}{n}\sum_{i=1}^{n} \varepsilon_i \tag{4.3}$$

As each error term is a random variable, the simulation error is also a random variable, and each simulation error is considered to be a sample from the actual simulation error distribution. The performance of the algorithm is evaluated based on the expected value of the simulation error ($E[\bar{\varepsilon}]$). An unbiased point estimator for $E[\bar{\varepsilon}]$ is the average of the simulation errors for m simulations.

$$\hat{E}[\bar{\varepsilon}] = \frac{1}{m}\sum_{i=1}^{m} \bar{\varepsilon}_i \tag{4.4}$$

Since the performance evaluation is based on the presented estimator, the number of simulations should be chosen properly to achieve a desired confidence interval, which is taken to be 95% in this thesis. This confidence interval for m simulations is defined with a t distribution centered at the point estimate value, and it is given as following:

$$\hat{E}[\bar{\varepsilon}] \pm h \tag{4.5}$$

$$h = t_{0.95,m-1}\frac{\sigma_s(\bar{\varepsilon})}{\sqrt{m}} \tag{4.6}$$

In equations (4.5) and (4.6), h is the half width, $t_{0.95,m-1}$ corresponds to t critical value, and σ_s is the sample standard deviation of $\bar{\varepsilon}$. A reliable estimation can be achieved by keeping h as small as possible. Note that $t_{0.95,m-1}$ is a monotonically decreasing function with respect to the number of simulation. Thus, equation (4.6) implies that an increase in m leads to a decrease in h. Moreover, as m goes to infinity, h goes to zero, as shown in equation (4.7).

$$\lim_{m\to\infty} h = \lim_{m\to\infty} t_{0.95,m-1}\frac{\sigma_s(\bar{\varepsilon})}{\sqrt{m}} = 0 \tag{4.7}$$

However, a compromise occurs between the accuracy and the computational expenses. In order to decide the optimum number of simulations, some test studies are conducted for the confidence interval. Figure 46 displays the relationship between the number of simulations and *t* critical values. While the number of simulations is increasing, first *t* critical value is decreasing dramatically. When the number of simulation is greater than 20, the reduction in *t* critical value is becoming less significant. This can be seen from the zoomed section shown in Figure 46. Consequently, when the standard deviation is assumed constant, the half width for the simulations $m \geq 20$ is mostly dependent to the square root of the simulation number.

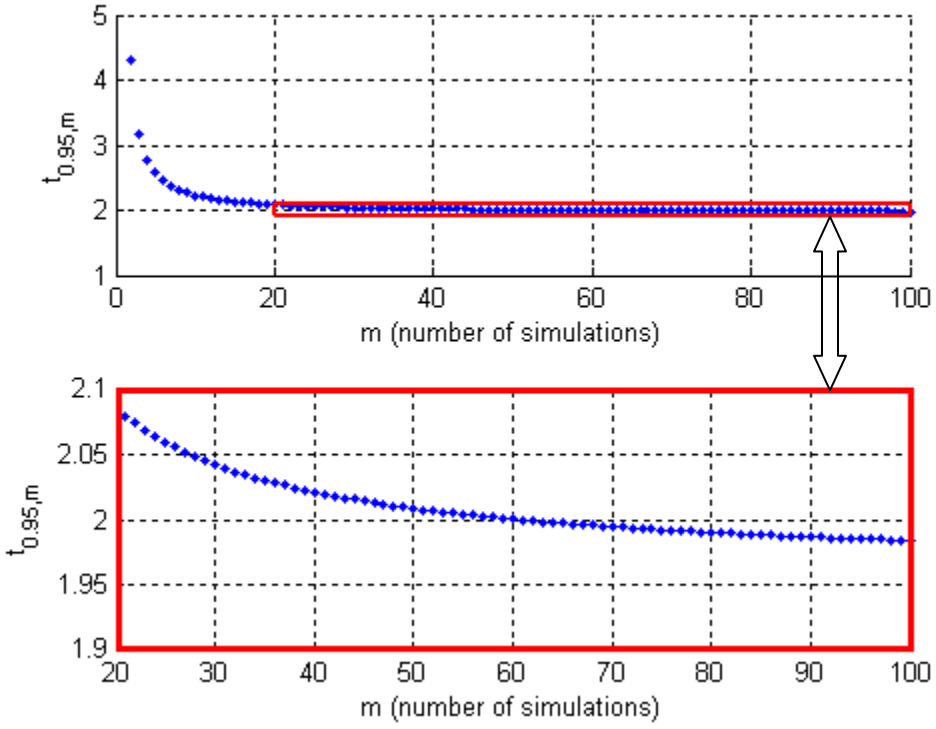

Figure 46: *t* critical values vs. number of simulations

In order to achieve a desired half width, first, an initial half width calculation is conducted by 20, 40 and 60 simulations for various environments such as no wind/no obstacle, only wind, and only obstacle. The aim of using various environments is to find out the optimum number of simulations for robust evaluation of the algorithm. The

results of no wind/no obstacle simulations are presented in Table 7 in which the mean error, the standard deviation, *t* critical value, and the half width are listed for each sample simulations. The mean error is observed approximately 17 cm for each sample simulation; however, the half widths are quite different from each other. The half widths are computed as approximately 3 cm, 1.5 cm, and 1 cm for 20, 40, and 60 simulations, respectively. Based on the results, the half width of 60 simulations seems quite reasonable since 1 cm is a physically acceptable error. The half width may have been improved with more number of simulations; however, the improvement will be on the basis of millimeters. Due to the consideration of computational expenses, the calculations are stopped at 60 simulations.

Table 7: Confidence interval of sample simulations without wind and obstacle

	Unit	20 Simulations	40 Simulations	60 Simulations
Mean	(m)	0.1778	0.1702	0.1647
Std. Dev.	(m)	0.0587	0.0454	0.0412
$t_{0.95}$		2.093	2.023	2.001
h	(m)	0.0275	0.0145	0.0106

The second case analysis of the half width includes obstacle in the environment. The results of this case are depicted in Table 8 in which the mean error, the standard deviation, *t* critical value, and the half width are illustrated for each sample simulation. The results show that a sample with 20 simulations is not good enough for predicting the mean error. While the samples with more simulations have an error mean trend around 17 cm, the sample with 20 simulations correspond to an error about 15 cm. Moreover, the half widths calculated from the samples with 20, 40 and 60 simulations are 1.6 cm, 1.4 cm and 1.2 cm, respectively. Like the previous case, a sample with 60 simulations presents the best estimation of mean error with a smaller confidence interval.

Table 8: Confidence interval of sample simulations with obstacle

	Unit	20 Simulations	40 Simulations	60 Simulations
Mean	(m)	0.1521	0.1709	0.1762
Std. Dev.	(m)	0.0337	0.0425	0.0456
$t_{0.95}$		2.093	2.023	2.001
h	(m)	0.0158	0.0136	0.0118

Finally, the last case for computing the half width pertains to the simulations in a gusty environment. While the trajectory is subjected to gust with unknown direction and magnitude, the samples with 20, 40, and 60 simulations are carried out. The results are presented in Table 9 in which the mean error, the standard deviation, *t* critical value, and the half width are illustrated for each sample simulation. The results show that the mean error in a windy environment is around 22 cm. The half widths for 20, 40 and 60 simulations are computed as 3 cm, 2 cm, and 1.5 cm, respectively. Like the previous cases, the best result is obtained from a sample with 60 simulations in which the mean error of 22 cm is estimated in the range between 20.5 cm and 23.5 cm.

Table 9: Confidence interval of sample simulations with wind

	Unit	20 Simulations	40 Simulations	60 Simulations
Mean	(m)	0.2369	0.2081	0.2161
Std. Dev.	(m)	0.0684	0.0699	0.0605
$t_{0.95}$		2.093	2.023	2.001
h	(m)	0.0320	0.0224	0.0156

Consequently, the optimum number of simulations is selected at 60 implying a desired half width less than 1.5 cm in any environment mentioned above. More simulations will result in smaller confidence intervals; however, it is also important to take into account the computational expenses. For instance, a particular simulation consisting of 10 hops with 1 m hop altitude and no uncertainty runs for approximately 15 minutes. Based on uncertainty, the environments including wind and/or obstacles, and the total distance of desired path, more hops to accomplish a particular mission are expected.

In other words, if the number of simulations is more than 60, it is likely to observe computer execution times on the day basis. Furthermore, after 60 simulations, the half width is expected to become smaller than 1 cm. While this is a numerically significant improvement, it is not practical to increase the computation expenses for obtaining better accuracy on the millimeter basis. Hence, the samples with 60 simulations are good at evaluating the performance of the trajectory-following algorithm with reasonable confidence intervals in various scenarios.

Table 10: Summary of the statistical studies

	MEAN ERROR	STANDARD DEVIATION
No Wind/ No Obstacle	16.4 ± 1.0 cm	4 cm
With Obstacle	17.6 ± 1.1 cm	5 cm
With Wind	21.6 ± 1.5 cm	6 cm

The proposed methodology shows repeatable accuracy for a Hopping Rotochute to track a path autonomously. The mean errors, which represent the average deviation from the route in each environment, are presented in Table 10. The maximum deviation is observed in simulations including wind. However, the Hopping Rotochute is still successful to follow the trajectory with an average deviation of 21 cm over a mission range of 6.5 m.

4.3 Summary

The proposed methodology was developed by conditional statements based on the performance models. The results show that the regression models created for the prediction of a particular hop have good model fittings. Thus, the vehicle has the capability to predict the control commands of a desired hop. Moreover, the same models help to understand some major points stated as the follows:

1. *Obstacle Avoidance:* From the trajectory data, the minimum ceiling value is selected to become the height limitation of the mission. Local ceiling values could have been considered; however, the gust is not predictable during the mission. Hence, smaller hops have less risk to be affected by the unpredictable gust. As a result, the minimum ceiling limitation determines how big a hop can be for that specific mission.

2. *Gusty Environment:* Based on the results of Figure 35 and Figure 36, *14 cm* is a threshold that can ensure whether the vehicle is exposed to wind in the previous hop. Consequently, the algorithm checks the error at the end of each hop. If the error is larger than *14 cm*, then the algorithm deems that a Hopping Rotochute would be exposed to wind in the next hop. Hence, the next hopping point is selected by considering the wind drift, computed from the previous hop.

CHAPTER 5

TESTING THE TRAJECTORY-FOLLOWING ALGORITHM

In Chapter 4, some example simulations were presented in order to show the applicability of the proposed methodology. This chapter particularly focuses on more challenging missions to understand the success of the algorithm. The most likely environment is an indoor mission containing the obstacles and the gusts of wind. Hence, this chapter specifically takes into account four scenarios including the obstacles and/or gust. The developed algorithm will be tested in each of the scenarios, and the statistical results will be presented to infer about the success of the algorithm. In this manner, the results of this chapter attempt to answer particularly the research questions 1.1 and 1.2.

5.1 Scenario 1: No Obstacle/No Wind

This basic scenario assumes no obstacle and no wind in the environment. A closed large room that has a length of 8.5 m and a width of 4 m is used, and the trajectory is located as shown in Figure 47. The performance of the Hopping Rotochute will be investigated and summarized at the end of this scenario.

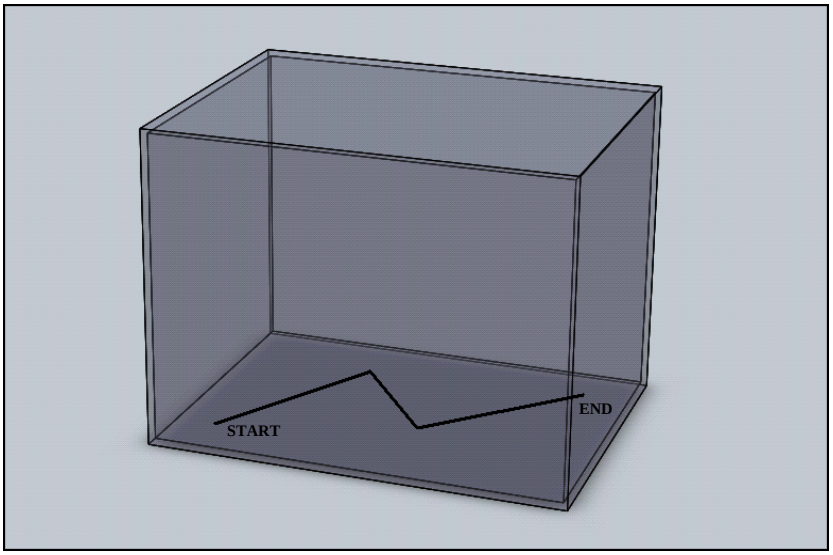

Figure 47: Environment of scenario 1

As mentioned before, 60 simulations are conducted for this scenario and one, including a total of 6 hops, is presented in Figure 48 that represents the desired trajectory as the solid line and the Hopping Rotochute's trajectory as the dashed line. The figure shows that these lines mostly coincide with each other, indicating that the vehicle travels on the trajectory most of the time. Note that the trajectory-following algorithm in this scenario assumes a fixed hop range determined by the user (Step 3 of chapter 3 depicted the details). Hence, when the vehicle comes to the coordinates of (1.5, 0.65), the algorithm decides to hop (3.75, 0.45) instead of (2, 0.8) due to the user preferences. At the end, the Hopping Rotochute accomplishes the mission by approaching close enough to the end point (7, 1.15).

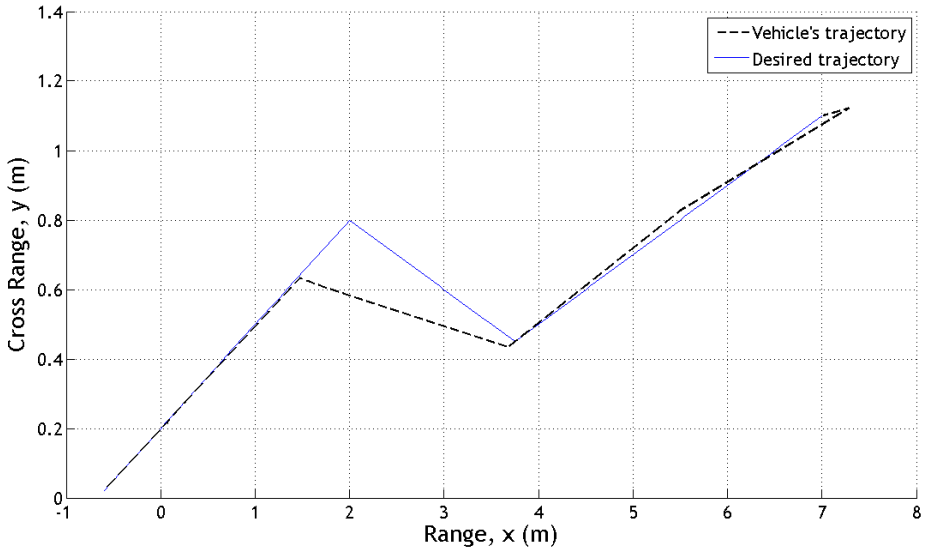

Figure 48: Trajectory of a Hopping Rotochute at scenario 1

One can observe that the vehicle is not exposed to wind, interpreted from Figure 49. This figure shows the target IM position, represented by a solid line, and the actual hop angular position, represented by a dashed line. Note that the target IM position corresponds to the desired angular alignment of the IM arm, calculated with respect to desired target point; whereas the actual hop angular position pertains to the achieved angular displacement, calculated with respect to actual landing position at the end of a

particular hop. In environments with no wind, the target IM position and the actual hop angular position should be close enough. In this manner, Figure 49 implies that the vehicle has not been exposed to wind during its mission since the solid and the dashed lines are very close to each other. Note that small deviations seen in Figure 49 should be due to uncertainty.

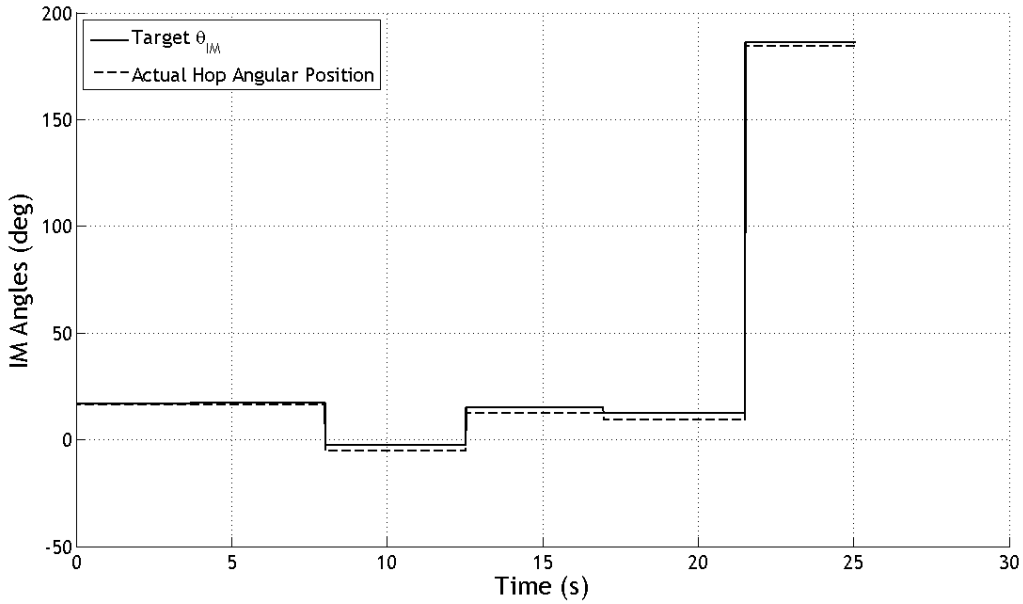

Figure 49: Angular position of the IM arm at scenario 1

In order to visualize the mission in 3D, Figure 50 is presented. As it is seen, the Hopping Rotochute finalizes this specific mission with 6 hops that are represented by the dashed line. At the end of 5^{th} hop, the algorithm realizes that it has gone away from the final point. Therefore, 6^{th} hop is a small hop with an opposite direction in order to approach to the final point. Moreover, Figure 51 illustrates the altitude-time graph of this particular mission. As it is seen, each peak represents an individual hop. There are two small hops observed during the mission. The first one is small due to conceive the environment whether a local wind is present. The last hop is small because of being close enough, meaning smaller than 0.2 m, to the final point.

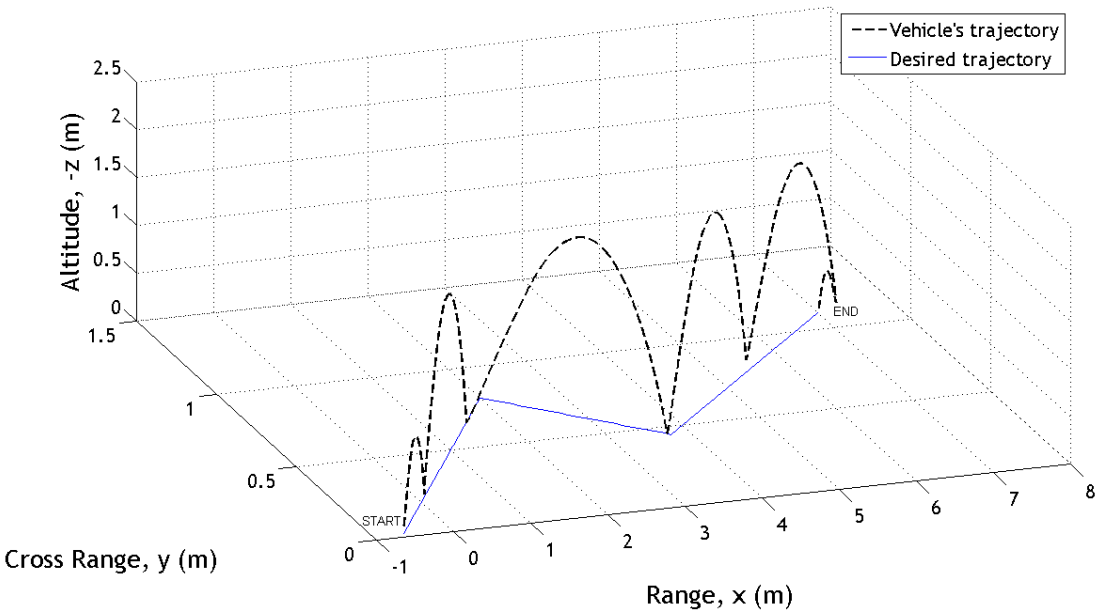

Figure 50: 3D Trajectory of a Hopping Rotochute at scenario 1

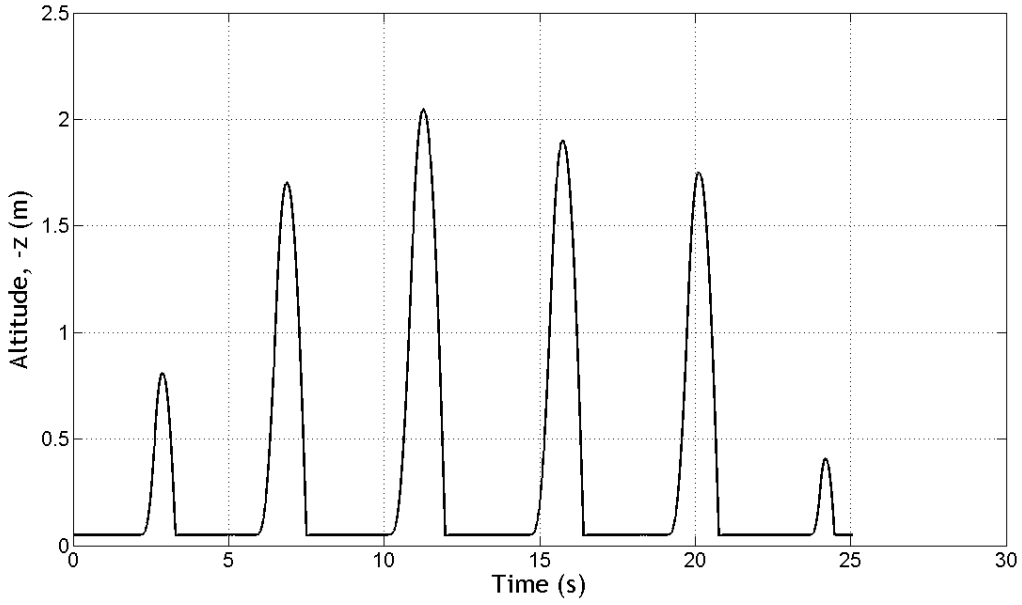

Figure 51: Altitude plot at scenario 1

In order to analyze the average deviation from the route for this scenario, 60 simulations were conducted in the existence of uncertainty and average errors were calculated. Note that ε (error) was defined as the distance between the target calculated by the developed algorithm and the actual landing point at the end of a particular hop. On the other hand, $\bar{\varepsilon}$ (*average error*) was defined as the mean error for a particular simulation (each error is summed up and divided by the number of hops). As a result, Figure 52 illustrates the *average error* distribution of scenario 1 based on 60 simulations in the presence of uncertainty. The maximum *average error* is 0.38 m, and the minimum *average error* is around 0.05 m. Thus, the mean of *average errors* is 0.18 m with a standard deviation of 0.07. Furthermore, a confidence interval study is conducted, and the results show that the half width is 0.02 m indicating the estimated *average error*, 0.18 m, can vary between 0.16m and 0.20 m. Hence, scenario 1 can be reliably represented with an *average error* of 0.18 m. Although this scenario assumes no wind and no obstacle, the results indicate a high *average error* (0.18 m). The hops performed in this mission are high altitude hops because of no obstacle limitation. Hence, the hops with high altitudes have a tendency to result in more deviation from the route.

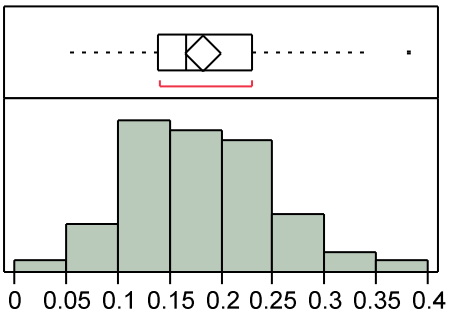

Figure 52: Error distribution of scenario 1

5.2 Scenario 2: Obstacle/No Wind

Scenario 2 is one of the challenging missions for a Hopping Rotochute. It contains two rooms connected by a low ceiling hall. The first room presented in Figure 53 has dimensions of 3 m of length, 2.6 m of width, and 3 m of height. The second room has dimensions of 3 m of length, 2.4 m of width, and 2 m of height. Finally, the connection hall has 1.8 m of length, 0.8 m of width, and 1.5 m of height. The aim for this scenario is to follow the trajectory without hitting the side walls and the ceiling.

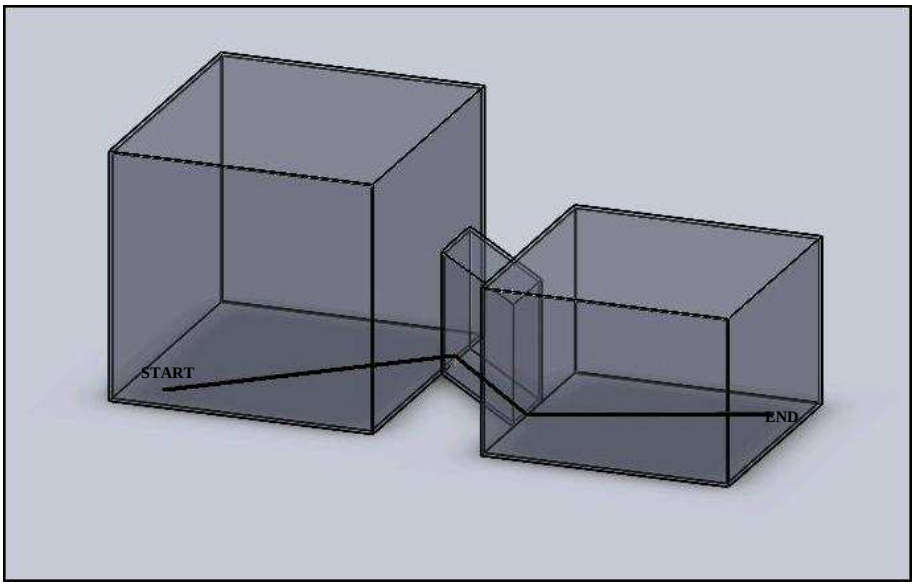

Figure 53: Environment of scenario 2

From the trajectory and boundary information, the smallest width is observed as 0.8 m. Thus, the trajectory-following algorithm decides to create a notional volume by considering a clearance of 0.4 m from right and left of the trajectory. Moreover, 1.5 m is observed as the smallest altitude determining the ceiling limitation of this scenario. The notional boundaries surrounding the trajectory are created as seen in Figure 54. Like the previous scenario, Figure 54 shows the solid line as the desired trajectory, the dotted lines as the notional boundaries, and the bold solid line as the trajectory of the vehicle. The results of the simulation show that the Hopping Rotochute follows the path accurately

without violating the boundaries, interpreted from the closeness of vehicle's trajectory (bold solid line) and desired trajectory (thin solid line).

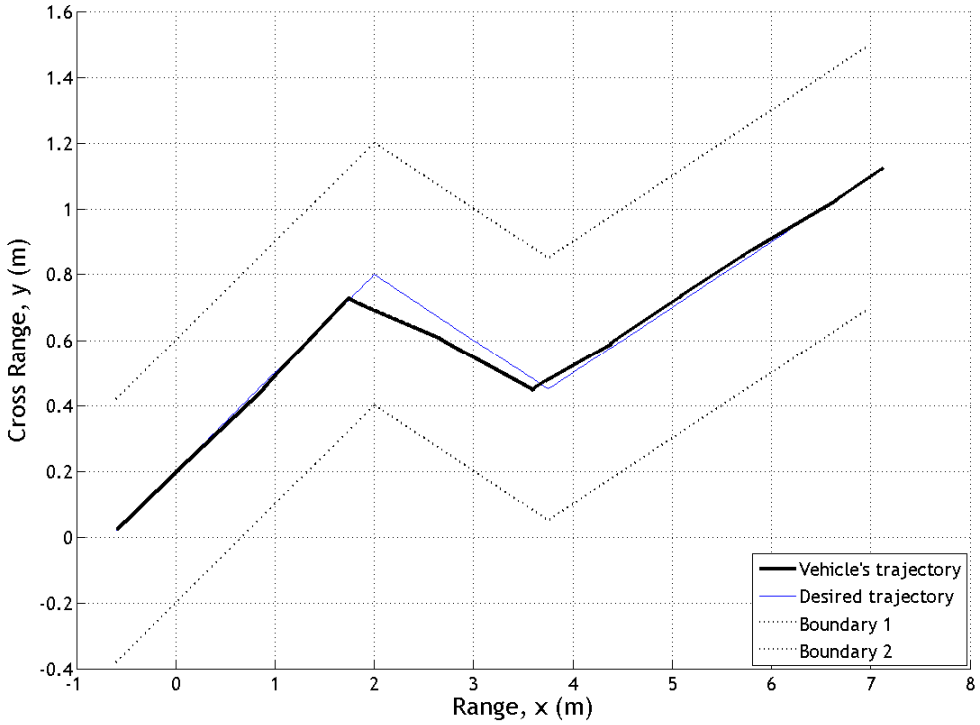

Figure 54: Trajectory of a Hopping Rotochute at scenario 2

As mentioned in the beginning of the scenario, this environment does not include any wind. Hence, the target IM position and the actual hop angular position are expected to be close enough. Figure 55 illustrates the angular position diagram showing significant differences between the target IM position, represented by solid line, and the actual hop angular position, represented by dashed line. However, Figure 55 indicates that this difference is less than 10 degrees at each hop. Hence, the fluctuation should be due to uncertainty. Furthermore, Figure 56 displays the visualization of this scenario in 3D. As it is seen, the Hopping Rotochute is inside the boundaries and finalizes the mission by 10 hops shown as the bold solid line in the figure.

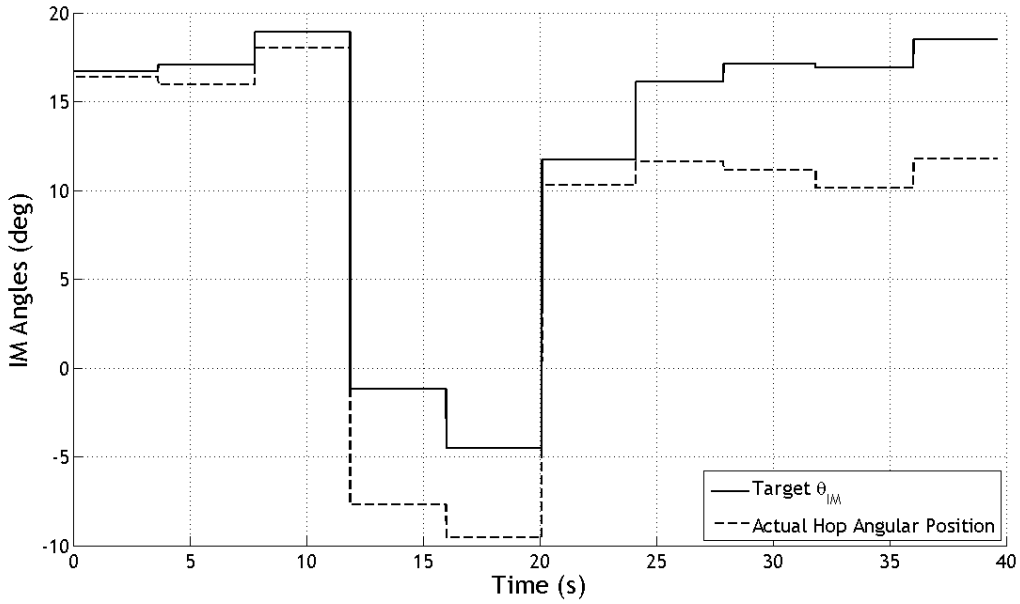

Figure 55: Angular position of the IM arm at scenario 2

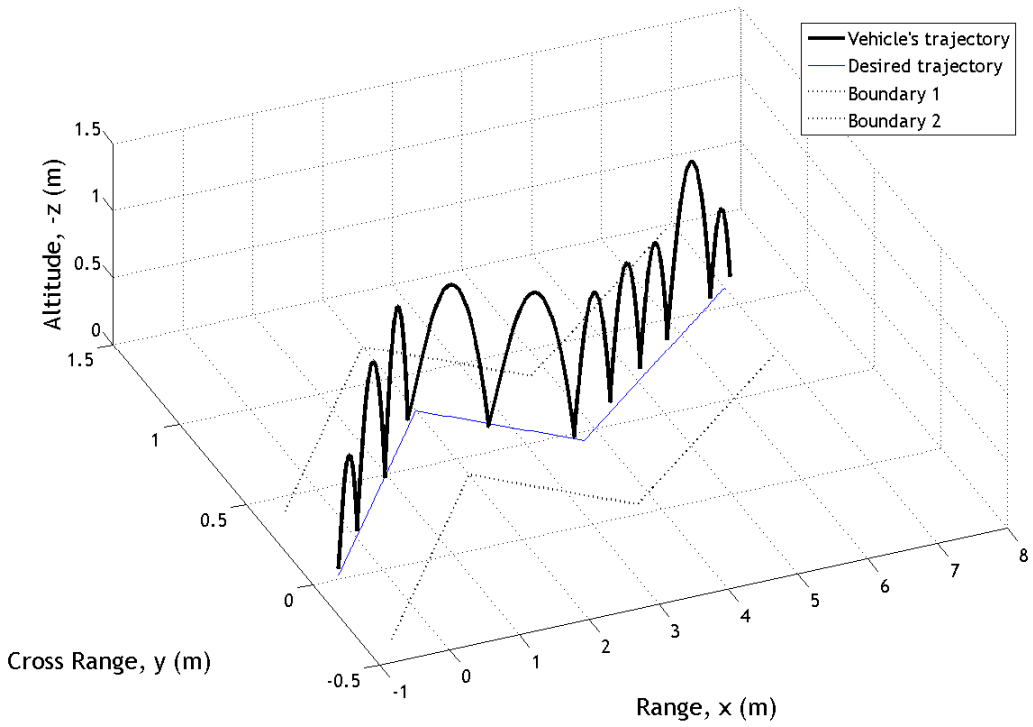

Figure 56: 3D Trajectory of a Hopping Rotochute at scenario 2

Previous graphs show that the Hopping Rotochute does not impact the side walls. In order to see whether any violation exists for the ceiling, the altitude plot is presented in Figure 57. As the figure shows, the maximum altitude obtained in this particular mission is 1.2 m corresponding to the 9th hop. Since the ceiling limitation 1.5 m, no ceiling impact is observed.

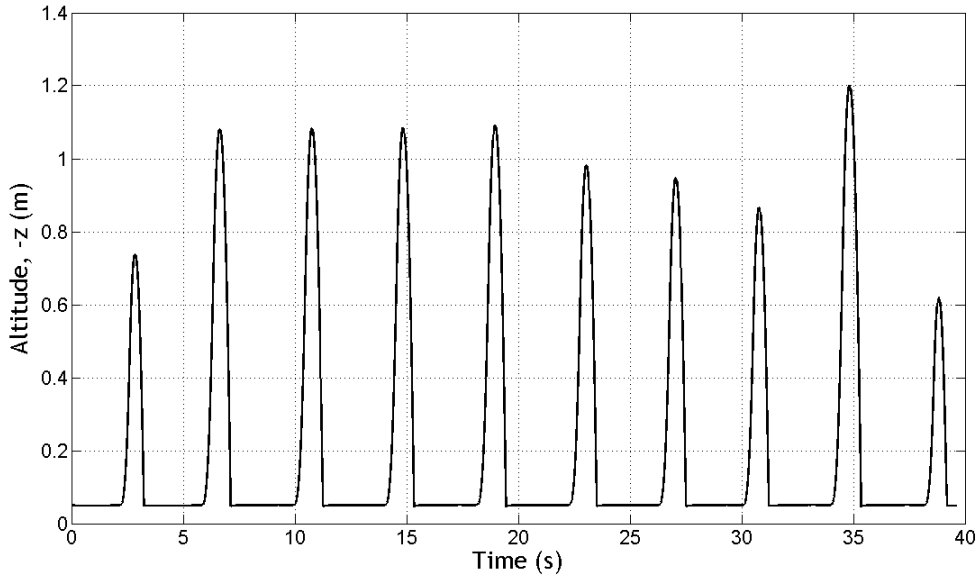

Figure 57: Altitude plot at scenario 2

Finally, in order to analyze the average deviation from the route in scenario 2, the *average error* was calculated for 60 simulations employing uncertainty. The *average error* distribution of 60 simulations is plotted in Figure 58. The maximum *average error* is around 0.17 m, and the minimum *average error* is around 0.05 m. The mean *average error* is 0.11 m with a standard deviation of 0.03. Moreover, the confidence interval study shows that the half width is 0.01 m implying a reliable estimation. Hence, the estimated *average error* (0.11 m) of scenario 2 can vary between 0.10 m and 0.12 m. If the estimated *average errors* are compared with each other, the results show that the accuracy of scenario 2 is better than the scenario 1. This proves that small altitude hops result in less error.

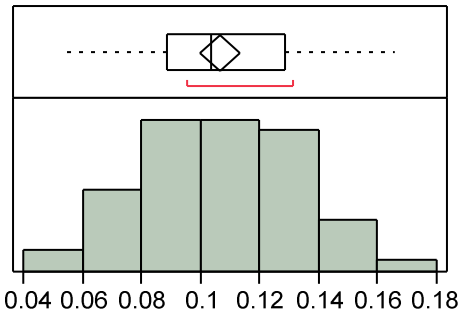

Figure 58: Error distribution of scenario 2

5.3 Scenario 3: No Obstacle/Wind

This scenario contains a single room with an opening that creates gusts of wind inside the room. The room of the representative environment, illustrated in Figure 59, has the same dimensions as that in scenario 1. The aim of this scenario is to analyze the performance of a Hopping Rotochute while it is subjected to unpredictable wind on an unknown segment of the trajectory. The aim of analyzing this scenario is specifically to answer the research question 1.2.

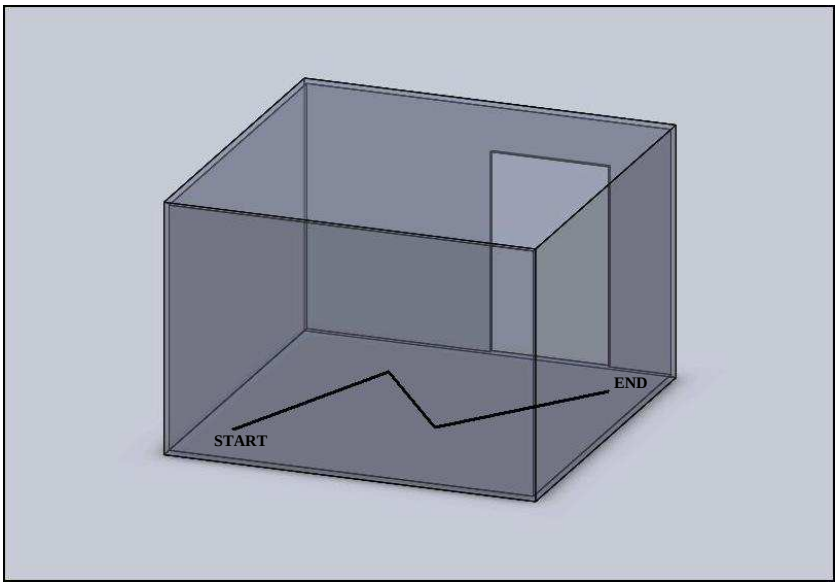

Figure 59: Environment of scenario 3

An example mission in this scenario is presented in Figure 60. Like the graphs of the previous scenarios, the trajectory of the vehicle is presented as a dashed line, and the desired trajectory is presented as a solid line. As seen, the deviation from the route increases dramatically beyond 4 m on the x-axis due to the existence of an opening that creates gusts of wind inside the room approximately with a magnitude of 1m/s in –y direction. (Recall that the magnitude and the direction of wind is random in each simulation). In these situations, this thesis proposes estimating the wind from the previous hop and implementing the wind estimation in the selection of the next hopping point. In this scheme, the vehicle starts to converge to the desired trajectory after 6 m on the x-axis. Finally, the mission is finalized when the Hopping Rotochute has reached the final point within a distance of less than 0.15 m, illustrated in Figure 60.

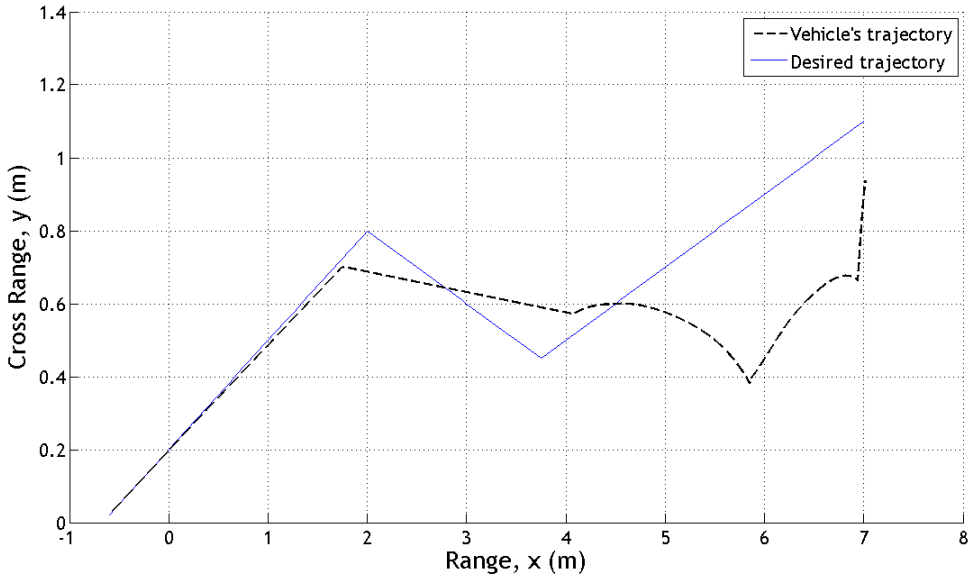

Figure 60: Trajectory of a Hopping Rotochute at scenario 3

The effect of wind can be analyzed in more detail from the IM and actual hop position graph, displayed in Figure 61. In this figure, the solid line represents the target IM position, and the dashed line represents the actual hop angular position. Due to the presence of wind, these lines should exhibit a drastic difference, and specifically at about

13 seconds, as shown in Figure 61. When the algorithm realizes that the vehicle is exposed to wind, it selects the target point *beyond* the trajectory instead of *on* the trajectory. For instance, in Figure 61 between 17 and 22 seconds, the vehicle selects the IM position of the next hopping point of 45 degrees; however, the achieved angular displacement is 15 degrees, suggesting that the strong component of wind is on the –y direction (wind blowing top to bottom in Figure 60), and even though the control command positions the IM arm at 45 degrees, the Hopping Rotochute stops at 15 degrees due to the wind drift. (Recall that the magnitude of wind is properly selected such as between 0 and 1 m/s, implying that it is possible to re-orientate the IM arm to converge to the trajectory when the vehicle is exposed to wind.)

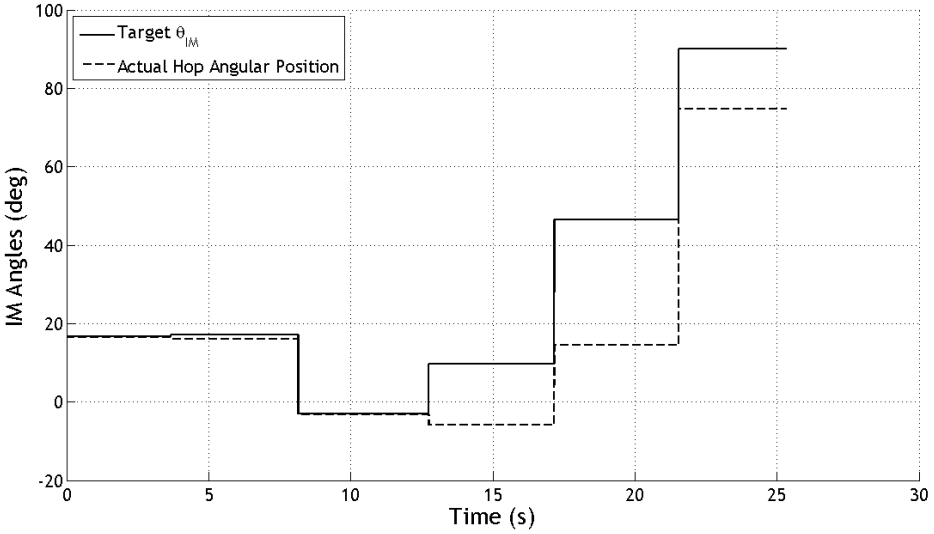

Figure 61: Angular position of the IM arm at scenario 3

In order to visualize the mission of scenario 3 in 3D, Figure 62, in which the dashed line represents the hops of the vehicle and the solid line represents the desired trajectory, is presented. As mentioned, the vehicle diverges from the route after 4 m of the x-axis. However, it attempts to converge to the trajectory by using the wind estimation in the next hopping point selection. In this way, two more hops are conducted after 6 m of the x-axis, and the mission is finalized when the Hopping Rotochute reaches

the final point within an acceptable range of 20 cm. This particular simulation in scenario 3 ends after six hops, which can be seen from the peaks in Figure 63. Since there is no ceiling limitation, the hop altitudes are obtained beyond 2 m.

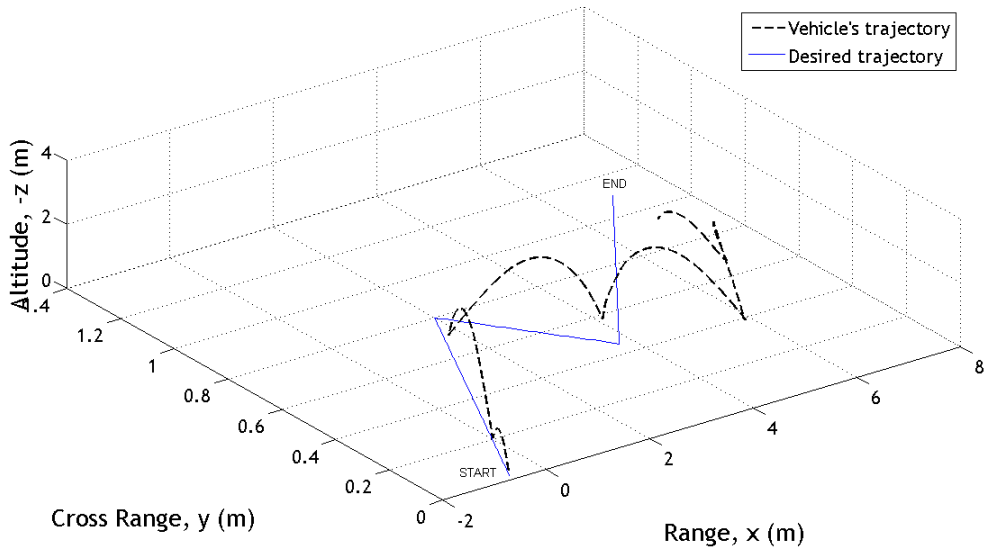

Figure 62: 3D Trajectory of a Hopping Rotochute at scenario 3

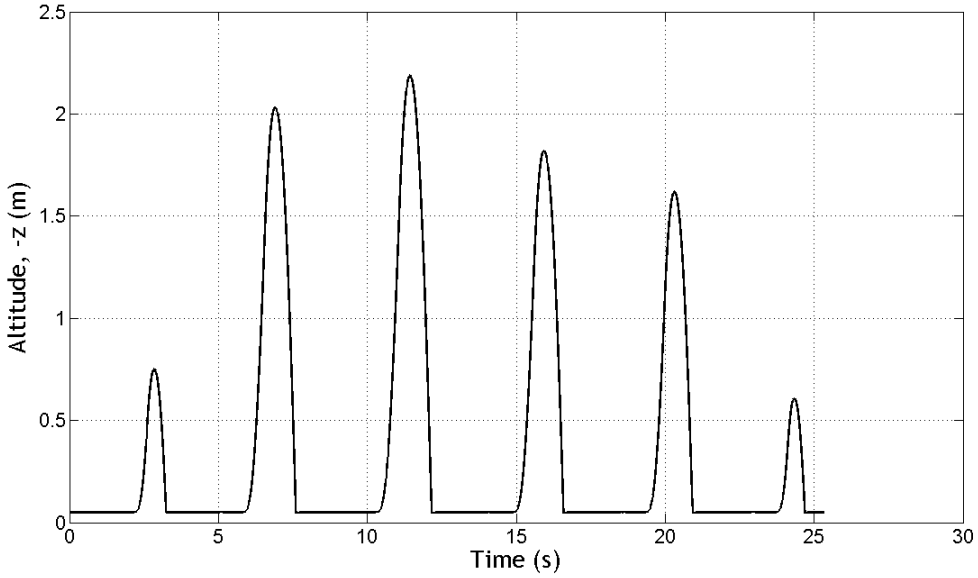

Figure 63: Altitude plot at scenario 3

In order to evaluate the average deviation from the route, the same *average error* calculation is done as described in the former sections. Figure 64 illustrates the *average error* distribution of scenario 3 based on 60 simulations that include uncertainty. The maximum *average error* is 0.42 m, and the minimum *average error* is 0.11 m. Thus, the mean *average error* is around 0.21 m with a standard deviation of 0.07. The confidence interval studies computed the half width as 0.02 m, which yields a reliable estimated *average error* of 0.21 m for scenario 3. If these results are compared with those of scenario 1, one will realize that wind leads to higher deviations from the route. (Recall that the magnitude and direction of wind are uncertain.)

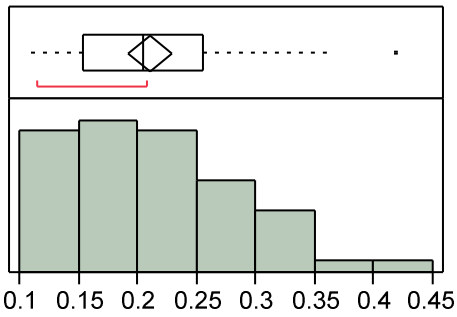

Figure 64: Error distribution of scenario 3

5.4 Scenario 4: Obstacle/ Wind

Of the studied scenarios, the most challenging is scenario 4. This scenario is the combination of scenarios 2 and 3 since the environment contains some obstacles as well as an opening that creates unpredictable gusts of wind. The environment of scenario 4 is illustrated in Figure 65 in which the dimensions of room are the same as in scenario 2, and the opening is present in the second room.

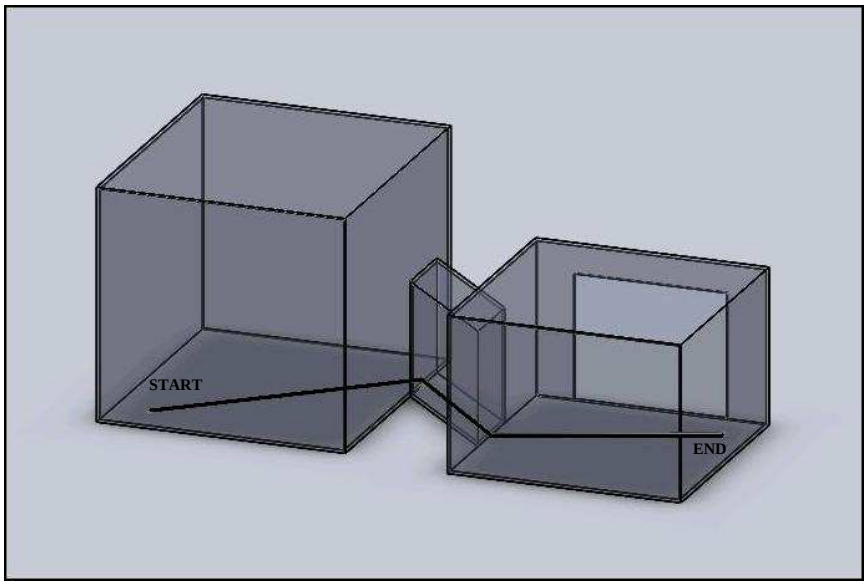

Figure 65: Environment of scenario 4

Like the other scenarios, 60 simulations have been conducted for scenario 4. One is shown in Figure 66. In this figure, the bold solid line corresponds to the trajectory of the vehicle, the thin solid line corresponds to the desired trajectory, and the dotted lines surrounding the trajectory correspond to the boundaries. The figure shows that the Hopping Rotochute deviates from the route in the third segment of the trajectory, which corresponds to the second room. The first segment of the trajectory represents the first room and the second the connection hall. The deviation from the route in the third segment implies that the vehicle is subjected to wind in the second room. Even though wind is present, the Hopping Rotochute successfully completes the mission by reaching the final point with an acceptable range of 20 cm.

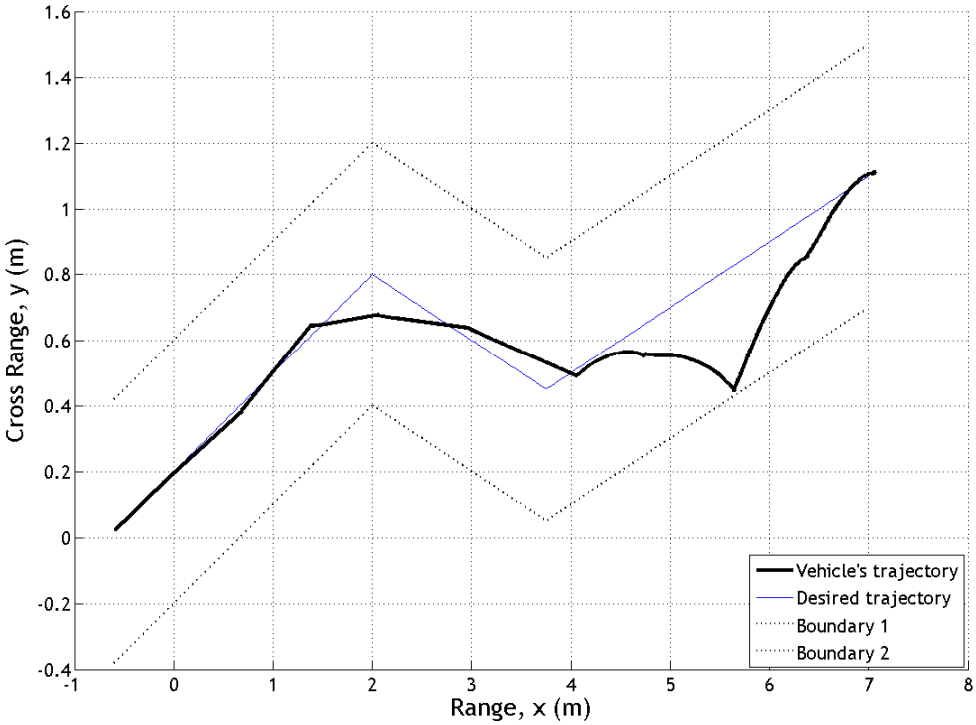

Figure 66: Trajectory of a Hopping Rotochute at scenario 4

As stated, in windy zones, the algorithm selects the next hopping points *beyond* the trajectory, observed from Figure 67, which shows a significant difference between the target IM arm position and the actual hop angular position. In the figure, the solid line represents the target IM position, and the dashed line represents the actual hop angular position. While the Hopping Rotochute is not subjected to wind, which corresponds to the hops at the first and second segments of the route, the target IM and the actual hop angular positions are close to each other. In other words, the algorithm selects a point on the trajectory (the IM is aligned at that position,) and the vehicle hops to the desired point with acceptable accuracy. Hence, small angle deviations occur until the vehicle is exposed to wind. Whenever the Hopping Rotochute is subjected to wind, the vehicle deviates from the desired hopping position due to the wind drift. This deviation is observed from the significant difference, around 20 degrees, between the target IM and

the actual hop positions shown in Figure 67. In these situations, the wind drift is estimated by the algorithm, and the next hopping point is selected *beyond* the trajectory in order to finalize the hop *on* the trajectory. Hence, Figure 66 shows that the Hopping Rotochute attempts to converge to the route after being exposed to the wind in the second room (or the third segment of the trajectory).

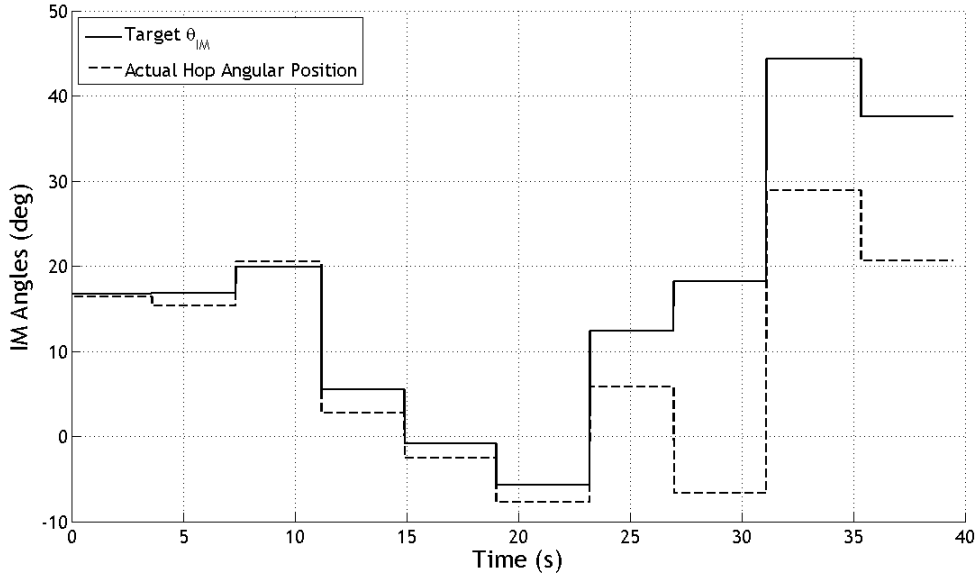

Figure 67: Angular position of the IM arm at scenario 4

In order to illustrate the mission in 3D, Figure 68 represents the vehicle trajectory by the bold solid line, the desired trajectory by the thin solid line, and the abstract boundaries surrounding the trajectory by the dotted lines. In order to realize accurate path tracking, the Hopping Rotochute should be inside the boundaries. Figure 68 shows that even though the Hopping Rotochute closely approached one of the boundaries when it first encountered wind, it stayed inside the boundaries and followed the desired trajectory.

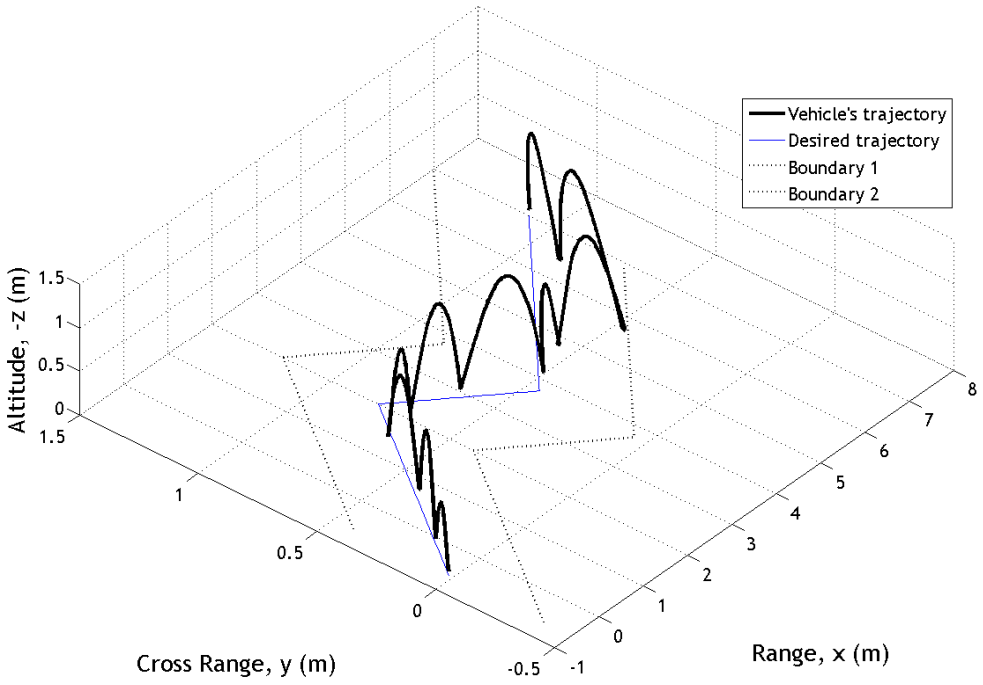

Figure 68: 3D Trajectory of a Hopping Rotochute at scenario 4

The representative mission of scenario 4 was accomplished by ten hops that correspond to the peaks in Figure 69. Each peak, shows a change in altitude by time, has a maximum altitude of less than 1.4 m. Recall that the lowest altitude in scenario 4 was 1.5 m. Hence, the Hopping Rotochute did not violate any ceiling limitation.

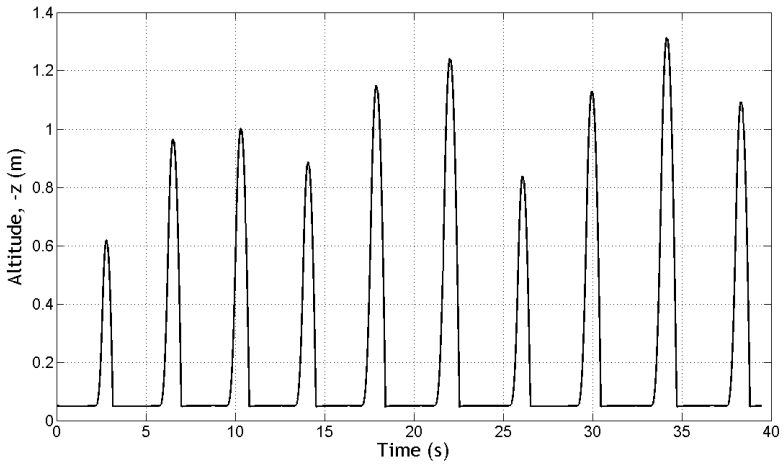

Figure 69: Altitude plot at scenario 4

Finally, the error distribution of scenario 4 is displayed in Figure 70. Like the other scenarios, 60 simulations with uncertainty were carried out in order to obtain the *average error* of each simulation. Once again, the wind magnitude and direction are uncertain. Based on these assumptions, the results show a mean *average error* of 0.124 m with a standard deviation of 0.03. The maximum *average error* is 0.21 m, and the minimum *average error* is around 0.07 m. Furthermore, the confidence interval study shows that the estimated *average error*, 0.12 m, can vary between 0.11 and 0.13 m. Hence, the estimated *average error* represents a reliable result for scenario 4.

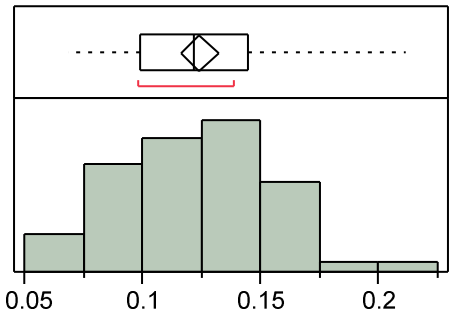

Figure 70: Error distribution of scenario 4

5.5 Summary

This chapter presented four scenarios that include wind and/or obstacles. The results of each scenario in which the simulations are performed 60 times with uncertainty are illustrated Table 11. The following presents results of the study:

1. The missions with high altitude hops can be accomplished with fewer hops; however, high altitude hops create more deviations throughout the mission, proven by the data in Table 11. The missions without obstacle avoidance result in fewer hops in contrast to higher mean errors and standard deviations, indicating a tradeoff between the mission completion time (total number of hops) and the accuracy of the trajectory-following.

2. The trajectory-following can be satisfied in gusty environments by wind estimation. When a particular hop ends with errors larger than 0.14 m, wind is estimated with respect to the previous hop, and the next hopping point is computed based on wind drift. In this manner, the next hopping point is selected *beyond* the trajectory, and the actual hop is expected to end *on* the trajectory. The results show that this approach leads to a successful gust-tolerant algorithm since the mean errors in scenarios 3 and 4, seen in Table 11, are closely in an acceptable range of 0.20 m.

Table 11: Summary of the scenario results

	AVG TOTAL # OF HOPS	AVERAGE MAX ALTITUDE	OBSTACLE AVOIDANCE	SUBJECTED TO WIND	AVERAGE MEAN ERROR	STANDARD DEVIATION
Scenario 1	5	2.1 m	NO	NO	0.18 m	0.07
Scenario 2	10	1.2 m	YES	NO	0.11 m	0.03
Scenario 3	5	2.2 m	NO	YES	0.21 m	0.07
Scenario 4	11	1.3 m	YES	YES	0.12 m	0.03

CHAPTER 6

CONCLUSION AND FUTURE WORK

The goal of this research was to develop an autonomous trajectory-following algorithm for a Hopping Rotochute. Past studies showed that the control commands of the most recent Hopping Rotochute were given prior to a mission. This control scheme is unsuccessful since real world applications involve uncertainty driven by the environment or the subsystems of a vehicle. Consequently, the control commands given prior to a mission may not satisfy a desired path tracking. Therefore, an algorithm is required to make decisions throughout a mission while a Hopping Rotochute is propagating on its trajectory. As a result, the objective of this research, defined in Chapter 1, was the following:

To develop a trajectory-following algorithm that allows the Hopping Rotochute to follow any pre-planned trajectory autonomously.

This research objective determined the scope of this work and entailed a literature survey that refined the scope. Chapter 2 presented the specifications of the Hopping Rotochute and some control techniques applicable to trajectory-following. Subsequently, Chapter 3 presented hypotheses formulated to answer the research questions. Chapter 4 and 5 explained the proposed methodology and implemented it in various scenarios to analyze the effectiveness of the methodology. Finally, Chapter 6 presents a summary of the work, revisiting the hypotheses first and then recommending future research.

6.1 Hypothesis Review

This thesis involves two major research questions, one of which includes two more related sub-questions. Thus, four research questions and four hypotheses were defined in Chapter 3. Then chapters 4 and 5 attempted to answer these questions and prove the hypothesis. The following reviews the research questions and hypotheses.

Research Question 1: How are the next hopping point and its corresponding control inputs selected?

Hypothesis 1

Assuming a known trajectory and boundary conditions, the next hopping point can be selected at the end of a particular hop by conditional statements in which the hopping performance models are embedded for hop predictions.

An online control technique is required for autonomous trajectory-following in order to achieve accurate path tracking. Due to the inefficiency of traditional control techniques such as feedback controllers incapable of handling uncertainty, the selection of the control commands is achieved by rule-based control. In a rule-based control algorithm, conditional statements are required, so this thesis creates conditional statements by embedding *hop performance* models that enable the algorithm to predict the control commands of a desired hop. The properties of regressed *hop performance* models show that they have good model fits implying accurate predictions within the assumed model boundaries. In this manner, the selected control commands based on the predictions are convenient. Thus, this hypothesis allows a Hopping Rotochute to achieve accurate path tracking by successfully selecting the next hopping point.

Research Question 1.1: What is the impact of the control volume surrounding the pre-planned trajectory in terms of mission accomplishment and the controller design?

Hypothesis 1.1

If a volume surrounding the pre-planned trajectory is defined inside the given boundary conditions such as the side walls and the ceiling, the Hopping Rotochute does not interact with the boundaries unless it exits this volume.

Creating a notional volume inside the real boundaries theoretically enables a Hopping Rotochute to avoid obstacles unless it exits this region. The simulation results showed that the Hopping Rotochute never interacted with the real boundaries if hypothesis 1.1 is implemented. Hence, obstacle avoidance is satisfied by creating a notional volume surrounding the trajectory in consideration of the dimensions of real boundaries.

Research Question 1.2: What is the effect of gusts of wind in the selection of the next hopping point?

Hypothesis 1.2

The effect of gusts of wind will be used to estimate the wind drift, which will determine the selection of the next hopping point beyond the trajectory.

Estimating wind and employing this estimation in the next hopping point are the major goals of the gust-tolerant trajectory-following algorithm. As mentioned, the presence of wind is decided if the distance between the target point and actual landing point is significant. Thus, since a hopping point selected *on* the trajectory cannot be achieved by a Hopping Rotochute in the presence of wind, the next hopping point is selected *beyond* the trajectory by taking into account the estimated wind computed from the previous hop. The simulation results showed that the implementation of hypothesis 1.2 achieves the development of a gust-tolerant trajectory-following for a Hopping Rotochute within an acceptable range of error.

Research Question 2: What is the impact of uncertainty while the Hopping Rotochute is performing autonomous path tracking?

Hypothesis 2

Accuracy of the trajectory-following algorithm can be assessed from uncertainty analysis in which the statistical performance of the algorithm is examined along with the confidence interval study.

Statistical performance of the proposed methodology is achieved by uncertainty analysis, whose results pertain to the accuracy of the trajectory-following algorithm. This thesis assumes that uncertainty, created as random values with triangular distributions, is present in position data, RPM, and pulse width. In order to determine the sufficient number of simulations to yield conclusions, a confidence interval study is conducted and its results showed that 60 simulations are sufficient to estimate the mean *average error* with less variability. Hence, this thesis assumed that the estimated mean *average error* was indicative of the accuracy of the algorithm.

6.2 Future Research

Although the trajectory-following algorithm proposed in this thesis showed success in the computer simulations, it should be implemented in the real Hopping Rotochute in order to prove its effectiveness. The methodology presented in this work assumes that the position data are known by the vehicle at the end of each hop. Unfortunately, the current prototype of the Hopping Rotochute does not employ any position sensors. As a first step recommendation, experiments could be conducted with a VICON motion capture system, a computer combined with the motion capture system, and a transmitter for sending the computed commands. In this manner, the VICON system would become the position sensor of the Hopping Rotochute, the computer combined with the motion capture system would determine the control commands with

respect to the algorithm, and the transmitter would send the control commands to the vehicle.

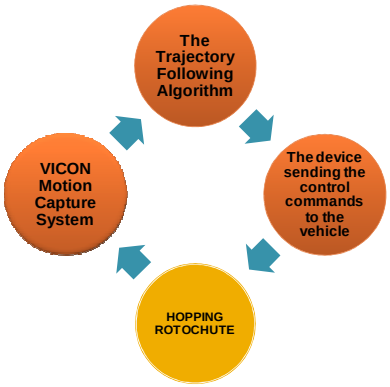

Figure 71: Recommended future experiment

In addition to the future work pertaining to the experiments, a couple of recommendations are presented for the development of the trajectory-following algorithm. As emphasized in the methodology, a discrete control technique has been employed, which implies the implementation of control command at the end of each hop. In order to achieve more accurate trajectory-following, the orientation of the IM arm can be changed during the flight. In that way, the control command calculation would not be discretized with respect to the end of hops. A continuous, feedback loop can be employed for the re-orientation of the IM arm. Moreover, the error calculated at the end of each hop can be used for a robust analysis by creating a neural network that can use the errors to improve the accuracy of the trajectory-following.

REFERENCES

[1] Twomey, T., Perry, J. and Logan, M., "Global Hawk for Broad Area Maritime Surveillance", AIAA 3rd "Unmanned Unlimited" Technical Conference, Chicago, IL, September 2004, AIAA 2004-6377

[2] ASDL MAST (Micro Autonomous Systems Technologies) Grand Challenge Team Project, Georgia Institute of Technology, Atlanta - GA, Spring 2009

[3] McMichael, M.J. and Francis, M.S., "Micro Air Vehicles – Toward a New Dimension in Flight", http://www.casde.iitb.ac.in/IMSL/MAVDOCS/www.darpa.mil/tto/MAV/mav_auvsi.html, 1997

[4] Costello, M., "Challenges Facing Micro Air Vehicle Flight Dynamics and Controls Engineers", 46th AIAA Aerospace Sciences and Meeting Exhibit, Reno, Nevada, January 2008, AIAA 2008-521

[5] Pralio, B., Guglieri, G. and Quagliotti, F., "Design and Performance Analysis of a Micro Aerial Vehicle Concept", 2nd AIAA "Unmanned Unlimited" Systems, Technologies and Operations, San Diego, CA, September 2003, AIAA 2003-6546

[6] Bouabdallah, S., Murrieri, P. and Siegwart, R, "Design and Control of an Indoor Micro Quadrotor", Proc. of The International Conference on Robotics and Automation (ICRA), 2004, http://asl.epfl.ch/aslInternalWeb/ASL/publications/uploadedFiles/325.pdf

[7] Yu, D.R., Lv, X.W., Bao, W. and Yao, Z.L., "Preliminary Design Analysis of a Hopper Vehicle for Mars Mission", Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, vol.224, 2009

[8] Beyer, E.W., "Design, Testing, and Performance of a Hybrid Micro Vehicle – The Hopping Rotochute", PhD Thesis, Georgia Institute of Technology, Atlanta, GA August 2009

[9] Campbell, C.C. and Maciejowski, J.M., "Control and Guidance of a Highly Flexible Micro Air Vehicle using Model Predictive Control", AIAA Guidance, Navigation and Control Conference, Chicago, IL, August 2009, AIAA 2009-5874

[10] Arning, R.K. and Sasses, S., “Flight Control of Micro Aerial Vehicles”, AIAA Guidance, Navigation and Control Conference, Providence, Rhode Island, August 2004, AIAA 2004-4911

[11] Beyer, E. and Costello, M., “Performance of a Hopping Rotochute”, International Journal of Micro Air Vehicles, vol 1,no 2, June 2009

[12] Yasunobu, S. and Miyamoto, S., “Automatic Train Operation System by Predictive Fuzzy Control”, Industrial Applications of Fuzzy Control, North-Holland, 1/18, 1985

[13] Rossiter, J.A., “Model Based Predictive Control : A Practical Approach”, CRC Press, ISBN 0-203-58568-2, 2005

[14] Qin, S.J. and Badgwell, T.A., “A Survey of Industrial Model Predictive Control Technologies”, Science Direct, Control Engineering Practice, August 2002

[15] Bemporad, A., “Model Predictive Control Design: New Trends and Tools”, 45th IEEE Conference on Decision and Control, San Diego, CA, December 2006

[16] Camacho, E.F. and Bordons, C., “Model Predictive Control”, Springer Publications, ISBN 3-540-76241-8, 1998

[17] Etkin, B., “Dynamics of Atmospheric Flight”, Dover Publications, ISBN 0-471-24620-4, 1972

[18] Baldwin, P.E.J. and Maynard, A.D., “A Survey of Wind Speeds in Indoor Workplaces”, The Annals of Occupational Hygiene, vol 42, no 5, July 1998

[19] Jiang, D.Z. and Min, W.Z., “Mobile Robot Path Tracking in Unknown Dynamic Environment”, Robotics, Automation and Mechatronics, 2008 IEEE Conference, September 2008

[20] Li, B. and Zhang, C., “Adaptive Fuzzy Control for Mobile Robot Obstacle Avoidance based on Virtual Line Path Tracking”, IEEE International Conference of Robotics and Biomimetics, December 2006

[21] VICON. “Vicon Products.” VICON Motion Systems Limited. http://www.vicon.com/products/cameras.html (accessed July 2010)