Thesis for the Degree of Master of Science

Navigation Method for VTOL Type UAV using a Limit-cycle Navigation Method and Fuzzy Logic Control

Byung-Cheol Min

Department of Electronic and Radio Engineering
Graduate School
Kyung Hee University
Seoul, Korea

August, 2010

Thesis for the Degree of Master of Science

Navigation Method for VTOL Type UAV using a Limit-cycle Navigation Method and Fuzzy Logic Control

Byung-Cheol Min

Department of Electronic and Radio Engineering
Graduate School
Kyung Hee University
Seoul, Korea

August, 2010

Navigation Method for VTOL Type UAV using a Limit-cycle Navigation Method and Fuzzy Logic Control

Byung-Cheol Min

Department of Electronic and Radio Engineering
Graduate School
Kyung Hee University
Seoul, Korea

August, 2010

Navigation Method for VTOL Type UAV using a Limit-cycle Navigation Method and Fuzzy Logic Control

by

Byung-Cheol Min

by

Dr. Dong Han Kim

Submitted to the Department of Electronic
and Radio Engineering and the Faculty of
the Graduate School of Kyung Hee University
in partial fulfillment of the requirements for
Degree of Master of Science

Dissertation Committee : ________________

Chairman ________________

____

____

ACKNOWLEDGEMENTS

First of all, I am extremely grateful to my advisor, Professor Donghan Kim. When I first came to his office to ask for admission to his group, he willingly welcomed me. Since then, he has ceaselessly introduced me to the state of the art technologies related to Robotics, and his support and enthusiasm over the years have made all of the difference. In addition, because of his open mind on my research, I have been able to devise and invent all of my research topics.

I give thanks to my thesis committee members, Professor Jinsang Kim and Professor Sanghoon Hong. Because of their valuable comments and advices on my thesis, it could have more scholarly value.

Thanks to all of the members in the Automatic Control Lab. Professor Chongkug Park has taught me how to diagnose the problem I was facing and approach it for a solution. Dr. Sunho Jo, Dr. Sangwon Kim, Dr. Byoungjoon Mun, Judong Han, Kyunghee Lee, Jungwan Kim, Yunwon Lim, Yongho Kim, Jinsu Jung, Hanguen Kim, Soohyeok Kang and Kyungmin Choi were willing to discuss my research topics and participate in the effort toward my research. Without their devotion to my work, I could not produce significant and compelling results from my research. Especially, I would like to thank my long-time friend and colleague, Chanho Cho. I wish for his great success in the future.

I would like to take this opportunity to thank Professor Hanseok Lee for driving me to the Robotics' real world. Because of him, I was able to fall into a place within the Robotics' field of which I am proud and in which I am currently majoring. In addition, I would like to say thank you to Professor Eric Matson for giving me the opportunity to take the first steps toward obtaining an internationally recognized advanced degree.

Finally, I would like to express warmest thanks to my family, especially my parents. Without their sacrifices and belief in me, I could not accomplish my dream of getting an education. Special thanks to my fiancée, Heesun Choi, for always standing by me and for helping me overcome all problems in the study as well as in daily life. I would like to say here that I really love my family.

CONTENTS

LIST OF FIGURES

LIST OF TABLE

ABSTRACT

Navigation Method for VTOL Type UAV using a Limit-cycle Navigation Method and Fuzzy Logic Control

In this dissertation, a navigation method that plays a pivotal role in operating an unmanned aerial vehicle (UAV) is discussed. The proposed navigation method takes the form of a path planning algorithm combined with a path tracking method. The path planning algorithm generates a path of serial waypoints from a UAV's initial position to its destination, while avoiding static and dynamic obstacles. The path tracking method then enables a UAV to reach the destination by orderly passing through the designated waypoints.

The essence of the path planning algorithm comes from the original limit-cycle navigation method for works in two dimensions. In addition, the ray tracing method is applied to enhance the work in three dimensions. The combined method of ray tracing and limit-cycle is the core of the proposed path planning algorithm. Thus, to cope with dynamic obstacles, we present an algorithm that can successfully generate a path to avoid them. The algorithm is primarily comprised of a collision detection method and standard rules of airplane traffic; when combined, this algorithm eliminates the chance of a collision and allows a UAV to reach the destination effectively in a short period of time.

To control the thrust and attitude of the Vertical Take-off and Landing (VTOL) type UAV, which primarily affects its position, a fuzzy logic control is proposed. The fuzzy logic control is composed of nine fuzzy logic control modules, responsible for position control, velocity control and angle control, respectively; when combined, they adequately control the thrust and attitude of the VTOL type UAV.

As a test model for the proposed navigation method, we introduce a quad-rotor type UAV, a VTOL type UAV, and analyze its dynamics. The proposed navigation method is then verified through a set of simulations. In addition, simulation results are provided to demonstrate the effectiveness and merits of the proposed methods for autonomous path planning and tracking.

CHAPTER 1. INTRODUCTION

1.1 Overview

The unmanned aerial vehicle (UAV) is widely used in many practical applications, e.g., research, surveillance and reconnaissance in specific or dangerous regions that are inaccessible or hard to reach by piloted vehicles that are heavier and larger in size than the UAV. The capabilities and roles of the UAV are evolving into a wide range of applications for both civilian and military use. As a result, their operation areas are being expanded to very complicated environments, such as urban and downtown areas surrounded by buildings. In these environments, UAVs should be equipped with path planning methods for reaching the target in a minimum amount of time and following the settled target precisely, as well as avoiding any fixed obstacles in their path.

A variety of navigation methods for autonomous path planning have been proposed with the purpose of reaching a target in a minimum amount of time. These include vector fields [1] and potential fields [2] [3], which have been widely used as a path planning method. The potential field method adopts the attractive-repulsive features attributed to many of the sensor-actuator transformations of reactive machines. Guiding an aircraft's action with such a plan is accomplished by summing the effect of the individual gradients at the aircraft's location.

There is, however, a major disadvantage in the potential field method that limits their usefulness. The potential field method does not provide an aircraft with information to get out of the local minima. This lack of information can lead to task failure [4].

This work proposes a novel limit-cycle navigation method for the autonomous path planning of the UAVs. The essence of the method comes from the observation that to avoid an obstacle while moving towards a destination, an aircraft may negotiate around the obstacle in a clockwise or counter-clockwise direction. The method combines the merits of reactive approaches with deliberative approaches. More specifically, it is as flexible as a reactive approach and as goal-directed as a deliberative approach. Without the need for self-localization, it continually generates a trajectory that an aircraft follows while navigating in real time, using the limit-cycle characteristics of a 2nd-order nonlinear function. It can overcome the local minima problem and the undesirable influence of any obstacle, both disadvantages of the potential field method [4]. It has also been extended to deal with the situation of multiple obstacles on the route between the aircraft and the target.

The UAV also requires efficient path tracking methods to track the planned path. The studies of path tracking methods include a nonlinear path tracking algorithm [5], vector field path following algorithm [6] and fuzzy logic algorithm [7-10]. A study of a nonlinear UAV path tracking algorithm, similar to the line-of-sight guidance algorithm, was utilized to minimize tracking error by adjusting the position and the attitude of the UAV. In one study, the vector field path following method was adopted to generate straight and orbital paths; path error was reduced by the Lyapunov stability criteria. In another study, path tracking and obstacle avoidance were explored by using a fuzzy logic based approach; the two dimensional based experimental results verified the possibilities of the fuzzy logic algorithm for path tracking method. In Chapter 4, the fuzzy logic control (FLC) method is adopted for path tracking; the experimental results verify the efficiency of the FLC for quad-rotor UAVs in three-dimensional environments.

A quad-rotor type UAV is representative of a Vertical Take-off and Landing (VTOL) type UAV. By using four rotors that can rotate with individually controllable speeds, the mechanical structure of the quad-rotor aircraft becomes simpler than that of the single rotor aircraft [11]. Such advantages result in making the quad-rotor type UAV a simple and light rotor actuator system. In accordance with its exclusive maneuverability, the navigation method will be presented in this work.

1.2 Organization of the Dissertation

The rest of the dissertation is organized as follows. Chapter 2 introduces the theory of the limit cycle and proposes a new path planning algorithm based on the methods of a ray tracing and limit-cycle for path planning. In addition, an algorithm comprised of a collision detection method and standard rules of airplane traffic is presented to cope with the dynamic obstacles. Chapter 3 introduces the advantages of quad-rotor UAVs by comparing it with a single rotor helicopter. The dynamics of the quad-rotor type aircraft are then analyzed, so that the autonomous flight system by the proposed path planning algorithm and path tracking method that will be proposed in Chapter 4 can be operated. Chapter 4 introduces the strategies of the FLC used to minimize the tracking error between the three-dimensional paths and the VTOL type UAV. Finally, the findings and recommendations for future work to improve the capabilities of flight performance are summarized in Chapter 5.

CHAPTER 2. PATH PLANNING ALGORITHM

2.1 Introduction

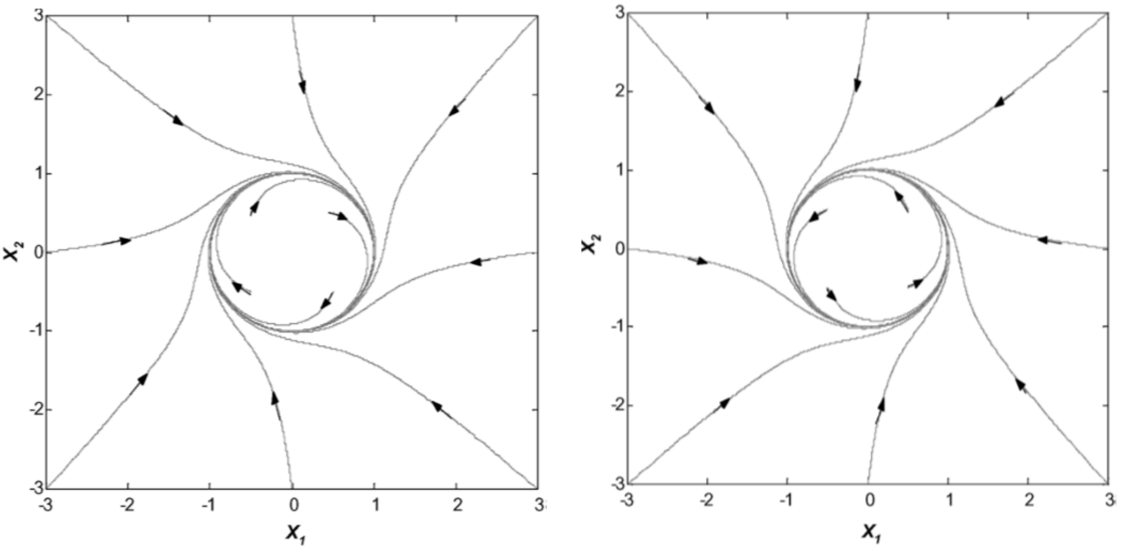
Path planning methods have been studied for decades by researchers who are studying ground vehicles and air vehicles. In the past, people concentrated only in methods for ground vehicles that are used in planar system because technology was not fully developed enough for them to have interests for three dimensions (3D). However, nowadays, vehicles for spatial system have been rapidly developed, and 3D path planning methods became essential factors for them. As a result, a number of researches started to focus on methods and attempted to invent the most efficient methods for aerial vehicles [12- 13].

There are only few cases where researches proposed completely new path planning methods for vehicles with 3D environments [14- 15]. Because path planning methods for two dimensions (2D) have been developed such a long period, researchers believed that extending from planar system to spatial system is more efficient approach for introducing path planning methods for aerial vehicles. On the other hand, various algorithms using the approaches of extending from 2D to 3D have been recently introduced [16- 20]. Like this, in this chapter, the limit-cycle navigation method that is a well defined navigation method in 2D is extended to 3D and is combined with a ray tracing method.

The limit-cycle navigation method introduced in [4] is one of the well-defined methods for 2D path planning. According to the authors, limit-cycle navigation method has notable advantages especially for fast mobile robots. Without any need for self-localization, the robot can continually generate a path that can be easily navigated and followed in real time. Furthermore, this path planning method had overcome the problems of the potential field method: the local minima problem and the confrontation of unexpected obstacles.

2.2 Limit-cycle Navigation Method for Path Planning

2.2.1 Limit Cycle



(a) Clockwise

(b) Counter-clockwise

Figure 2.1 Phase portrait of Limit-cycle.

If the 2nd-order nonlinear system and the Lyapunov function are given as follows,

$$\begin{aligned}\dot{x}_1 &= x_2 + x_1(1 - x_1^2 - x_2^2)\\ \dot{x}_2 &= -x_1 + x_2(1 - x_1^2 - x_2^2)\end{aligned} \tag{2.1}$$

$$V(x) = x_1^2 + x_2^2 \tag{2.2}$$

The derivative of V(x) along the trajectories of the system is given by

$$\begin{aligned}\dot{V}(x) &= 2x_1\dot{x}_1 + 2x_2\dot{x}_2 \\ &= 2x_1x_2 + 2x_1^2(1-x_1^2-x_2^2) - 2x_1x_2 + 2x_2^2(1-x_1^2-x_2^2) \\ &= 2V(x)(1-V(x))\end{aligned} \tag{2.3}$$

The derivative $V(x)$ is positive for $V(x) < 1$ and negative for $V(x) > 1$. Hence, on the level surface of $V(x) = c1$ with $0 < c1 < 1$, all the trajectories will be moving outward, while on the level surface of $V(x) = c2$ with $c2 > 1$, all the trajectories will be moving inward. This shows that the unit circle is a periodic orbit as shown in Figure 2.1 (a). This periodic orbit is called limit cycle. Figure 2.1 (a) shows the phase portrait of Eq. (2.1) with x_1 and x_2 according to the x-axis and y-axis, respectively. These trajectories from all points move toward the unit circle clockwise. On the other hands, the trajectories moving toward the unit circle counter-clockwise can be derived as follows,

$$\begin{aligned}\dot{x}_1 &= -x_2 + x_1(1-x_1^2-x_2^2) \\ \dot{x}_2 &= \quad x_1 + x_2(1-x_1^2-x_2^2)\end{aligned} \tag{2.4}$$

All the trajectories will be then moving inward counter-clockwise, as shown in Figure 2.1 (b).

To utilize this theory as a path planning method, the general form of Eq. (2.1) can be derived by replacing 1 with r:

$$\begin{aligned}\dot{x}_1 &= \quad x_2 + x_1(r - x_1^2 - x_2^2) \\ \dot{x}_2 &= -x_1 + x_2(r - x_1^2 - x_2^2)\end{aligned} \tag{2.5}$$

and the Lyapunov function is

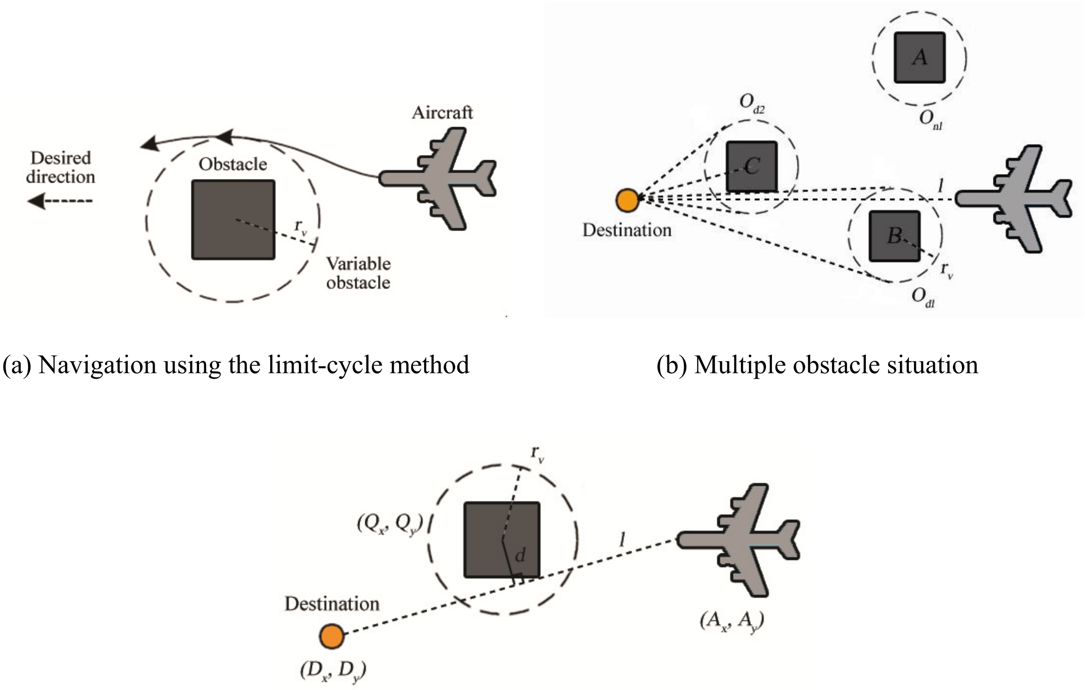
$$V(x) = x_1^2 + x_2^2 \tag{2.6}$$

The derivative of $V(x)$ along the trajectories of the system is given by

$$\begin{aligned}\dot{V}(x) &= 2x_1\dot{x}_1 + 2x_2\dot{x}_2 \\ &= 2x_1x_2 + 2x_1^2(r^2 - x_1^2 - x_2^2) - 2x_1x_2 + 2x_2^2(r^2 - x_1^2 - x_2^2) \\ &= 2V(x)(r^2 - V(x))\end{aligned} \tag{2.7}$$

The derivative $V(x)$ is positive for $V(x) < r^2$ and negative for $V(x) > r^2$.

Consequently, we can adjust the radius and the direction of the limit cycle. The Limit-cycle method constitutes a local navigation plan. This selects an efficient way by which the aircraft can avoid obstacles rather than moving far away from them. These limit-cycle characteristics are applied to the navigation plan.



(a) Navigation using the limit-cycle method

(b) Multiple obstacle situation

(c) Decision of rotational direction

Figure 2.2 The scheme of the limit-cycle navigation method.

2.2.2 Limit-cycle Navigation Method

Figure 2.2 (a) depicts the limit-cycle method which can drive an aircraft towards the desired direction and avoid an obstacle. At this time, the direction, either clockwise or counter-clockwise, should be decided. Figure 2.2 (b) shows a situation in autonomous flight. The rightmost aircraft needs to avoid three obstacles like a building ***A***, ***B*** and ***C*** in moving towards the destination.

Now, the steps to be followed in the process of local navigation by the limit-cycle method are:

1. Make a line l from the aircraft to the destination in a global coordinate Σ_{OXY} as follows:

$$ax+by+c=0 \tag{2.8}$$

2. Consider virtual obstacles as disturbing obstacles, O_{d}'s if the line l crosses them, else, they are considered as non-disturbing obstacles, O_{n}'s.
3. Move towards the destination if there is no O_{d}.
4. In Figure 2 (c), we can calculate the distance d from the center of the obstacle O_{d} to the line l as

$$d=\frac{aQ_x+bQ_y+c}{\sqrt{a^2+b^2}} \tag{2.9}$$

where (Q_x, Q_y), (D_x, D_y) and (A_x, A_y) are the xy-values of the obstacle, the destination and the aircraft, respectively. Eq. (2.1) is extended to fit to the navigation plan by substituting d and r_{v}. If $x_1$and x_2 are matched with x and y in the global coordinate Σ_{OXY}, calculate the desired direction of the aircraft at each position using

$$
\begin{aligned}
\dot{x} &= \frac{d}{|d|} y + x\,(r_v^2 - x^2 - y^2) \\
\dot{y} &= -\frac{d}{|d|} x + y\,(r_v^2 - x^2 - y^2)
\end{aligned}
\tag{2.10}
$$

where x and y are relative values to the obstacle, and r_{v} means a safety margin for collision avoidance. In this equation, if d is positive, the direction of a path is clockwise so that the aircraft avoids the obstacle O_{d} in the direction. On the other hand, if d is negative, the direction of a path is counter-clockwise so that the avoidance takes place in the direction.

5. Generate a path of serial waypoints at regular intervals from the initial position of the aircraft to its destination using Eq. (2.10).

2.3 Extension of Limit-cycle Navigation Method in 3D

Unfortunately, limit-cycle navigation method was used only in planar system in spite of the merits, because it was derived from limit-cycle characteristics of the 2nd-order nonlinear function.

Therefore, in this sub-chapter, a way to utilize the original limit-cycle navigation method in the three-dimensional space is introduced. The 3D path planning procedures is based on the concept of considering the vertical and horizontal plane. Because the horizontal plane, where the UAV's direction towards *xy*-plane includes, is considered in the original limit-cycle navigation method like a ground robot, the vertical plane, where the UAV's direction towards either *xz*-plane or *yz*-plane includes, only needs to be considered. For the horizontal plane, the angle *Ψ,* which is the angle between the current position and the desired position, is defined, depicted in Figure 2.3.

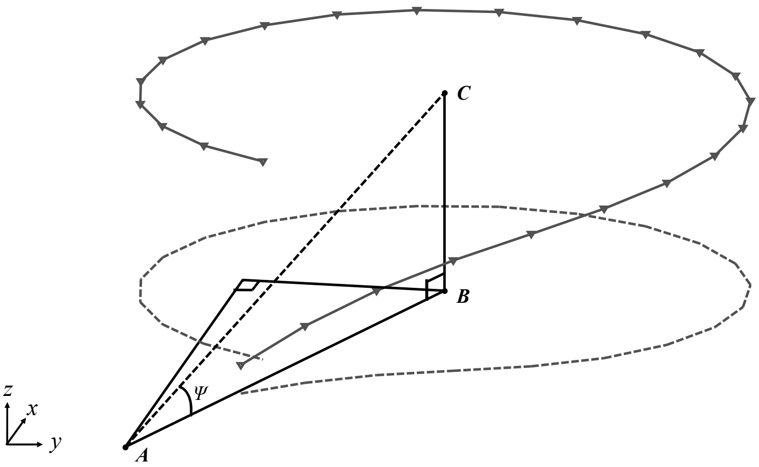


Figure 2.3 The scheme of the extended limit-cycle navigation method for works in three-dimensional space.

In Figure 2.3, ***A***, ***B***, and ***C*** are UAV's current position, desired position in 2D, and desired position in 3D, respectively. The angle *Ψ* is derived by a process by starting from calculating the distances of the lines $\overline{\boldsymbol{AB}}$ and $\overline{\boldsymbol{BC}}$. These distances are able to be easily calculated because the current position, ***A***, and the desired position, ***C***, are given by exact coordinates. The angle, *Ψ,* is then created by using tangent of line $\overline{\boldsymbol{AB}}$ and line $\overline{\boldsymbol{BC}}$.

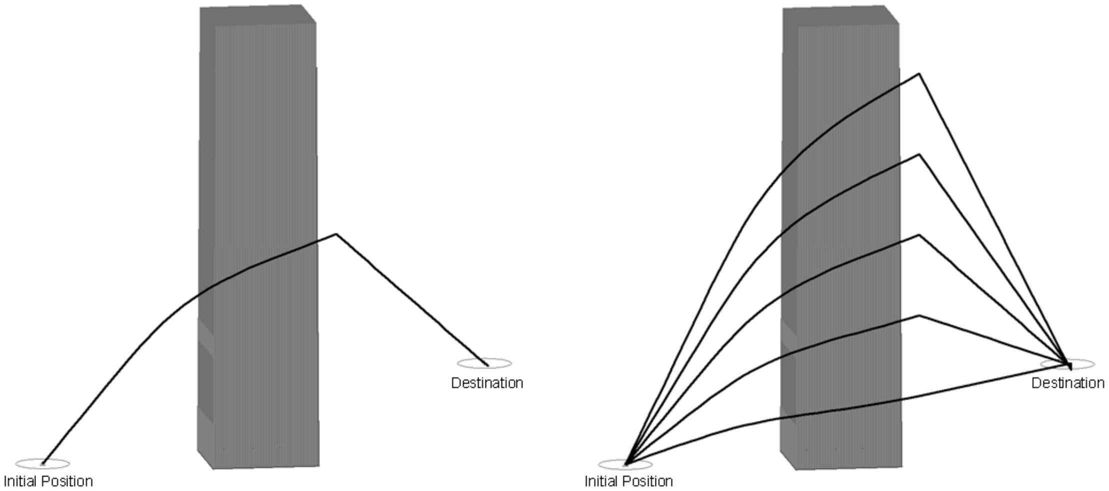
Consequently, when the original limit-cycle navigation method plans a path from *A* to *B* in the *xy*-plane, shown in Figure 2.3 as dash lines, the angle *Ψ* can be calculated. This concept results in extending the original limit-cycle navigation method to 3D and allows a path to be generated in 3D, shown in Figure 2.3 as solid lines.

2.4 Problem Statement

Although the extended limit-cycle navigation method shows a possibility that can be utilized in the three-dimensional space, it also shows several drawbacks.

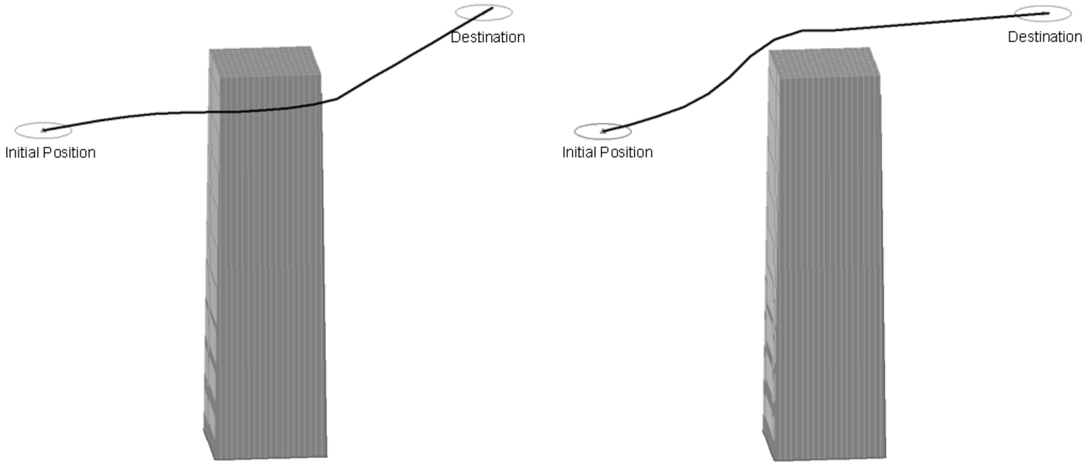
First of all, when a UAV avoid around the obstacle, the method usually draws the path with setting along the center of the obstacle's height. Figure 2.4 (a) is the clear example of the unsuitable path that is generated by using the extended limit cycle navigation method. However, when a UAV does not set the path based on the center of the obstacle, an infinite number of paths can be generated along with *z*-axis as depicted in Figure 2.4 (b). Since the UAV can use only one path to reach the destination, it should select the best way to reach the destination among the possible paths.

Second, the path planned by the extended limit-cycle navigation method generates the path only by avoiding to the side of the obstacle, depicted in Figure 2.5 (a). Depending on the location of the initial position of the UAV and its destination, however, avoiding the path by flying over the obstacle might be the better method than avoiding to the side of the obstacle, as shown in Figure 2.5 (b). Therefore, in next sub-chapter, an enhanced path planning algorithm that can generate the most efficient path will be suggested: a combined method of ray tracing and the extended limit-cycle navigation method.



(a) Unsuitable path (b) Possible paths

Figure 2.4 When the extended limit-cycle navigation method generates a path by avoiding th the side of the obstacle, there are a number of possible paths.



(a) Unsuitable path (b) Suitable path

Figure 2.5 Depending on the location of the UAV's initial position and its destination, avoiding the path by flying over the obstacle might be the better method than avoiding to the side of the obstacle.

2.5 Path Planning Algorithm Based on the Methods of Ray Tracing and the Extended Limit-cycle

In this sub-chapter, the whole calculation and the process of obtaining an enhanced path planning algorithm for avoiding an obstacle in the three-dimensional space is presented. Note that non-holonomic constraints, which can be introduced in fixed wing types of UAV's planning algorithms, can be ignored because this work is about designing the paths for VTOL type UAVs. Therefore, the flow chart of the algorithm is illustrated in Figure 2.6. The enhanced path planning algorithm is based on the methods of ray tracing and extended limit-cycle.

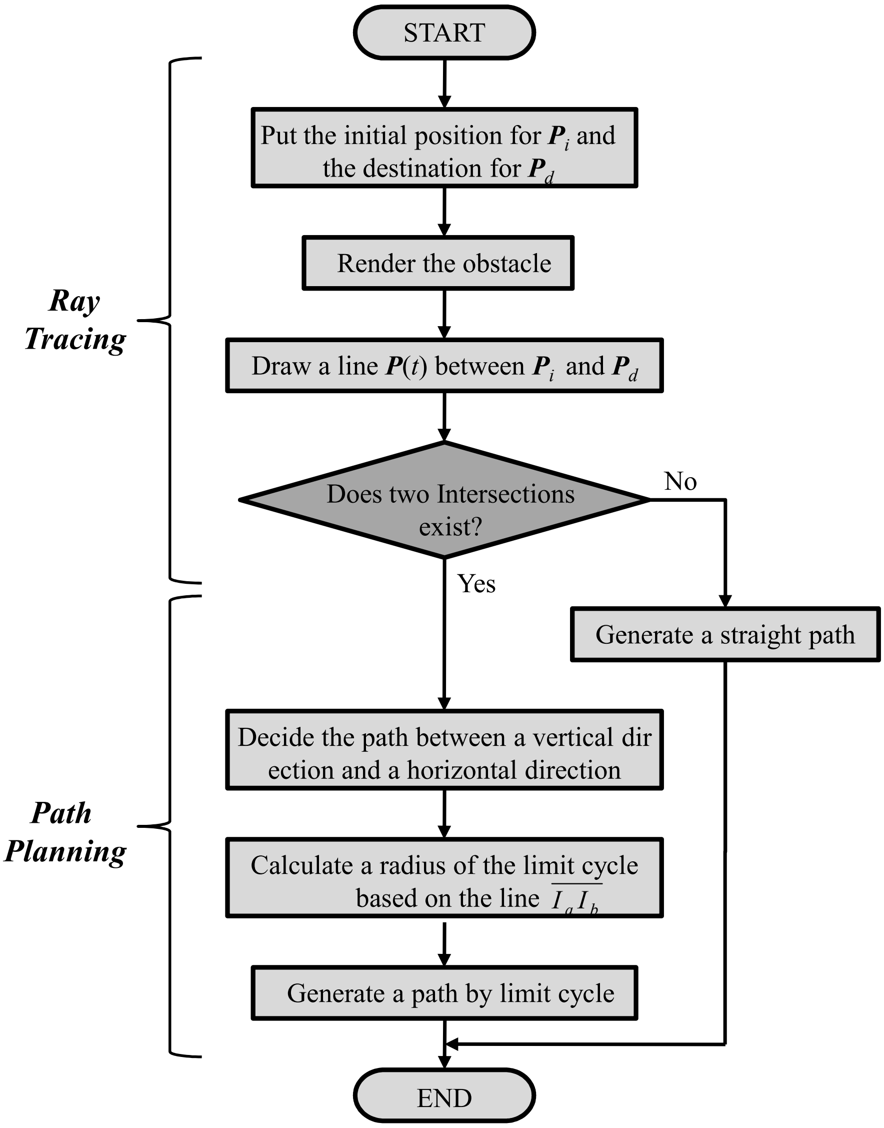

START
Put the initial position for $\boldsymbol{P}_i$ and the destination for $\boldsymbol{P}_d$
Render the obstacle
Draw a line $\boldsymbol{P}(t)$ between $\boldsymbol{P}_i$ and $\boldsymbol{P}_d$
Does two Intersections exist?
No
Yes
Generate a straight path
Decide the path between a vertical dir ection and a horizontal direction
Calculate a radius of the limit cycle based on the line $\overline{I_a I_b}$
Generate a path by limit cycle
END
Ray Tracing
Path Planning

Figure 2.6 The flow chart of the path planning algorithm based on ray tracing and the extended limit-cycle navigation method.

2.5.1 Ray Tracing

Ray tracing is an algorithm that shoots beams of light to determine the accurate location and the size of the objects that can be interacted in the future [21]. This method is primarily used to generate quality images in computer graphics or to calculate the path of waves through a system in physics. In this work, however, the method is used for the aircraft to find the shortest path to reach the destination while avoiding the obstacle.

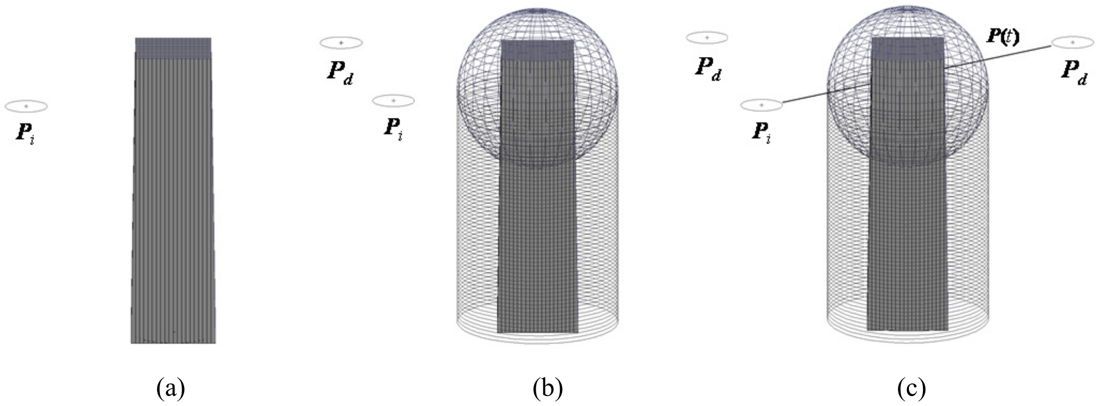


Figure 2.7 The procedure of ray tracing method: (a) Placing two positions $\boldsymbol{P}_i$ and $\boldsymbol{P}_d$ (b) Rendering the obstacle into a sphere and a cylinder (c) Drawing a line $\boldsymbol{P}(t)$ from $\boldsymbol{P}_i$ to $\boldsymbol{P}_d$.

The procedure of ray tracing method is explained in the following steps.

1. Put the initial position for $\boldsymbol{P}_i$ and the destination for $\boldsymbol{P}_d$ as depicted in Figure 2.7 (a). Then, render the obstacle into a sphere or a cylinder depending on its shape as shown in Figure 2.7 (b) because circular shapes are necessary on the obstacles in order to implement limit-cycle [4].

   A sphere with radius r centered at the origin is described as

$$x^2 + y^2 + z^2 = r^2 \tag{2.11}$$

The lateral surface of a cylinder whose radius is r, whose height is h, and whose base is centered on the origin of the xy-plane (Figure 2.8 (b)) is described as

$$x^2 + y^2 = r^2 \tag{2.12}$$
$$0 \leq z \leq h$$

2. Given two positions an initial position $\boldsymbol{P}_i$ and a destination $\boldsymbol{P}_d$ in a global coordinate, we can define a line $\boldsymbol{P}(t)$ shown in Figure 2.7 (c) that passes through these points as

$$\boldsymbol{P}(t) = \boldsymbol{P}_i + t(\boldsymbol{P}_d - \boldsymbol{P}_i) \tag{2.13}$$

where parameter t ranges over all real numbers. This line segment connecting $\boldsymbol{P}_i$ and $\boldsymbol{P}_d$ corresponds to values of t between 0 and 1. When the following equation is solved by each point,

$$\begin{aligned} x &= x_i + t(x_d - x_i) \\ y &= y_i + t(y_d - y_i) \\ z &= z_i + t(z_d - z_i) \end{aligned} \tag{2.14}$$

We now have the problem of determining whether the line $\boldsymbol{P}(t)$ intersects an obstacle or not.

The first method of checking for whether the line $\boldsymbol{P}(t)$ intersects an obstacle whose shape is sphere is followed as:

A sphere with radius r centered at $\boldsymbol{P}_\mathbf{s}$ is described by this equation

$$(x - x_\mathrm{s})^2 + (y - y_\mathrm{s})^2 + (z - z_\mathrm{s})^2 = r^2 \tag{2.15}$$

Substituting the components of the line $\boldsymbol{P}(t)$ in Eq. (2.13) for x, y, and z in Eq. (2.15) gives us

$$(x_\mathrm{i} + t(x_d - x_i))^2 + (y_\mathrm{i} + t(y_d - y_i))^2 + (z_\mathrm{i} + t(z_d - z_i))^2 = r^2 \tag{2.16}$$

Expanding the squares and collection in *t* yields the following quadratic equation.

$$\begin{aligned}
&((x_d - x_i)^2 + (y_d - y_i)^2 + (z_d - z_i)^2)t^2 \\
&+2(x_c(x_d - x_i) + y_c(y_d - y_i) + z_c(z_d - z_i))t \\
&+x_i^2 + y_i^2 + z_i^2 - r^2 = 0
\end{aligned} \tag{2.17}$$

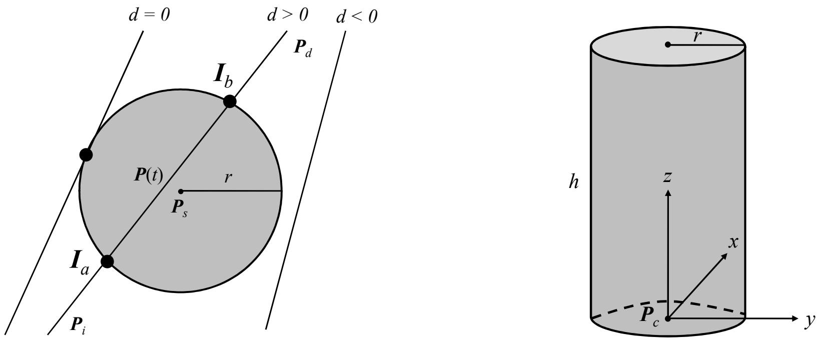
Eq. (2.17) can be expressed as

$$at^2 + bt + c = 0 \tag{2.18}$$

The coefficients *a*, *b*, and *c* in Eq. (2.18) can be expressed as follows.

$$\begin{aligned}
&a = (x_d - x_i)^2 + (y_d - y_i)^2 + (z_d - z_i)^2 \\
&b = 2(x_i(x_d - x_i) + y_i(y_d - y_i) + z_i(z_d - z_i)) \\
&c = x_i^2 + y_i^2 + z_i^2 - r^2
\end{aligned} \tag{2.19}$$

By solving the discriminant $d = b^2 - 4a$, we can know whether the line ***P***(*t*) intersects the sphere or not. As illustrated in Figure 2.8 (a), if $d = 0$, the line ***P***(*t*) hits the sphere tangentially at a point. If $d > 0$, two different intersections, $\boldsymbol{I}_a$ and $\boldsymbol{I}_b$, exist. Also, if $d < 0$, then the line ***P***(*t*) completely misses the sphere.



(a) (b)

Figure 2.8 (a) The discriminant D indicates whether the line P(t) intersects the sphere (b) Object space for a cylinder.

The second method of checking for whether the line $\boldsymbol{P}(t)$ intersects an obstacle whose shape is cylinder is followed as:

A cylinder of radius r centered at $\boldsymbol{P}_{\mathrm{c}}$ is described by this equation

$$(x - x_{\mathrm{c}})^2 + (y - y_{\mathrm{c}})^2 = r^2 \qquad (2.20)$$

$$0 \leq (z - z_{\mathrm{c}}) \leq h$$

As similar processes with a sphere, after substituting the components of the line $\boldsymbol{P}(t)$ in Eq. (2.13) for x and y, expanding the squares and gathering with t yields the following quadratic equation.

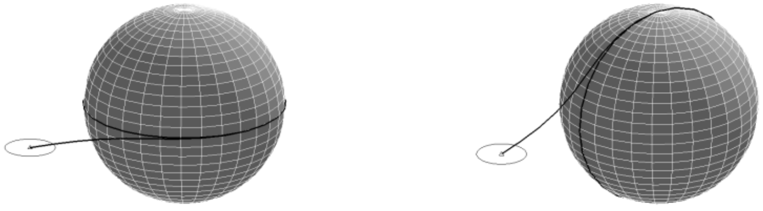
$$((x_d - x_c)^2 + (y_d - y_c)^2)t^2 + 2(x_{\mathrm{c}}(x_d - x_c) + y_{\mathrm{c}}(y_d - y_c))t + x_c{}^2 + y_c{}^2 - r^2 = 0 \qquad (2.21)$$

Like the sphere, the discriminant indicates whether an intersection occurs. Solutions to this equation give the values of t where the line $\boldsymbol{P}(t)$ intersects the cylinder that stretches out infinitely centered on the z-axis. The z-coordinates of the points of intersection must be in the range of $0 \leq (z - z_{\mathrm{c}}) \leq h$ in order to be indicated as the intersecting points.

2.5.2 Path Planning

If there is one or no intersection from the discriminant $D = b^2 - 4ac$, then the path will be straightly generated from the initial position $\boldsymbol{P}_i$ to the destination $\boldsymbol{P}_d$ because there is no need to consider avoiding obstacles. On the other hand, if there are two intersections of the line $\boldsymbol{P}(t)$ and the obstacle, then the shortest path to avoid the obstacle and to reach the destination will be generated with the following process:

1. Since it is obviously true that the shortest path to get the destination while avoiding an obstacle is analogous to the line $\boldsymbol{P}(t)$, first, the path to avoid the obstacle is decided either to a vertical direction or a horizontal direction. The decision depends on the two distances. One is the z-distance between the x-coordinate of the obstacle's center point on the line $\boldsymbol{P}(t)$ and the perpendicular point that intersects the top of the sphere. The other one is the y-distance the x-coordinate of the obstacle's center point on the line $\boldsymbol{P}(t)$ and the perpendicular point that is tangent to the side of the sphere. Therefore, when the two distances are compared, the direction with the shorter distance will be chosen. If a vertical direction is chosen, then limit-cycle will be generated on the xy-plane depicted in Figure 2.9 (a). On the other hand, if a horizontal direction is chosen, and then limit-cycle will be generated on the xz-plane depicted in Figure 2.9 (b).



(a) xy-plane, $z = 0$

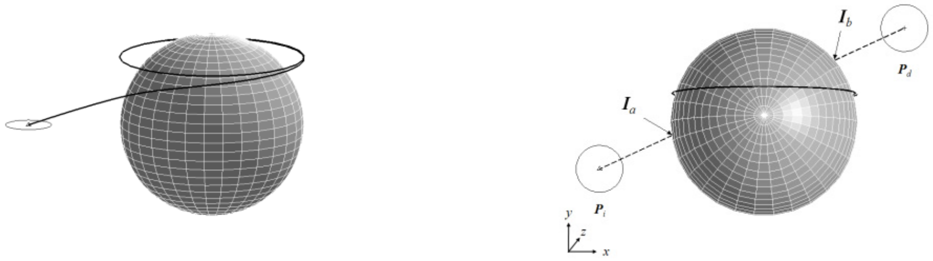
(b) xz-plane, $y = 0$

Figure 2.9 (a) When the horizontal direction is chosen, limit-cycle is generated on the xy-plane. (b) When the vertical direction is chosen, limit-cycle is generated on the xz-plane.

2. After determining the direction to avoid the obstacle, a radius, the essential variable to the limit-cycle navigation method mentioned in [4], is calculated according to the size of the obstacle. However, a major problem is confronted when the UAV navigates off from the center of the obstacle as depicted in Figure 2.10 (a). When the radius is calculated based on the center of the obstacle, the huge gap between the obstacle and the UAV will exist, and this created path will not be the shortest and the most efficient one.

   Therefore, instead of calculating the radius based on the center of the obstacle, when it flies over the obstacle, the radius is recreated based on the line $\overline{\boldsymbol{I}_a\boldsymbol{I}_b}$ in Figure 2.8 (a) considering as it is in *xy*-plane by setting *z*-coordinate as 0. Limit-cycle is then produced on the *xz*-plane by setting *y*-coordinate as 0. When it flies to the side of the obstacle, the radius is recreated based on the line $\overline{\boldsymbol{I}_a\boldsymbol{I}_b}$ considering as it is in *xz*-plane by setting y-coordinate as 0. Limit-cycle is then produced on the *xy*-plane by setting *z*-coordinate as 0.

   Even though it would have been better if the path is drawn the limit-cycle straight along the line, $\overline{\boldsymbol{I}_a\boldsymbol{I}_b}$, the limit-cycle is created parallel to the x-axis with centering on the middle point of $\overline{\boldsymbol{I}_a\boldsymbol{I}_b}$ if it is going over as shown in Figure 2.10 (b) and to the *y*-axis if it is going beside because the limit-cycle can be drawn only in two dimensions. In fact, the hypothetical limit-cycle in the three-dimensional space and the proposed limit-cycle in two dimensions will not have much difference since the radius will be exactly identical by shifting only the direction on the circle. Hence, the final limit-cycle will be drawn based on the radius, the line $\overline{\boldsymbol{I}_a\boldsymbol{I}_b}$, which is the distance between the initial position $\boldsymbol{P}_i$ and the destination $\boldsymbol{P}_d$ that intersects the sphere.



(a) (b)

Figure 2.10 (a) The huge gap occurs if the radius of limit-cycle is calculated based on the center of the obstacle. (b) Only slight difference occurs between the hypothetical limit-cycle in 3D and the proposed limit-cycle in 2D.

2.6 Simulation on Path Planning Algorithm

In this sub-chapter, the new path planning algorithm based on the methods of ray tracing and the extended limit-cycle navigation method is verified through a set of simulation works.

As shown in Figure 2.11, the first simulation was carried out in the case of that the UAV should follow the planned path from the initial position (15, 35, 60) to the destination (90, 55, 70) and avoid the fixed obstacle such as a building in the three-dimensional space. The sphere and the cylinder covering the obstacle designate the safety margin boundary for securing the protection of the UAV. Before finding the shortest path reaching the destination, we first picked out the two paths among possible paths generated by the extended limit-cycle navigation method. Two paths, which were determined by the proposed algorithm that can find the path to get the destination while avoiding an obstacle is analogous to the line $\boldsymbol{P}(t)$, are depicted in Figure 2.11: one is planned in a vertical direction depicted as a solid line, and the other is planned in a horizontal direction depicted as a dashed line. The shortest path was then decided by comparing the two distances. Since the distance of the path in a vertical direction was 82.43m and the distance of the path in a horizontal direction was 89.1m, the first one was determined as the most efficient path to reach the destination.

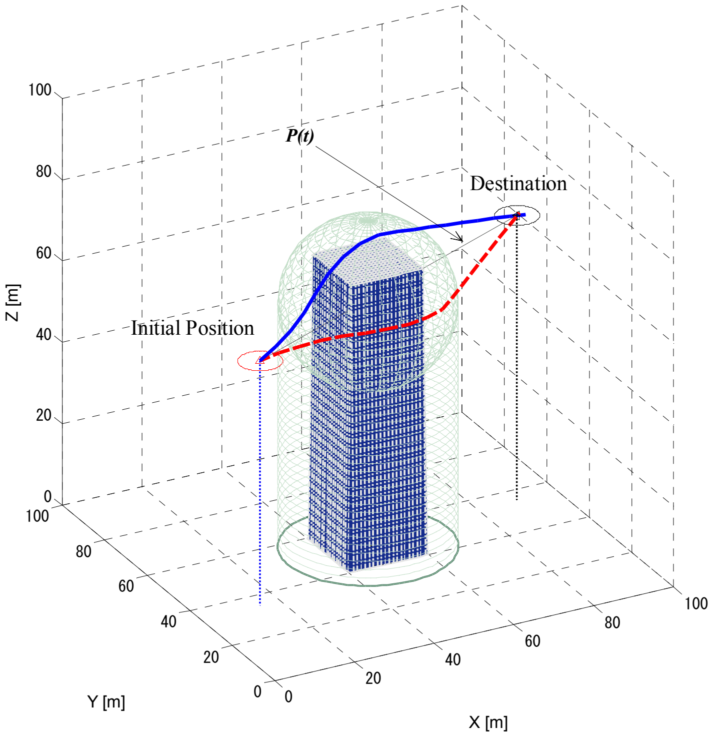
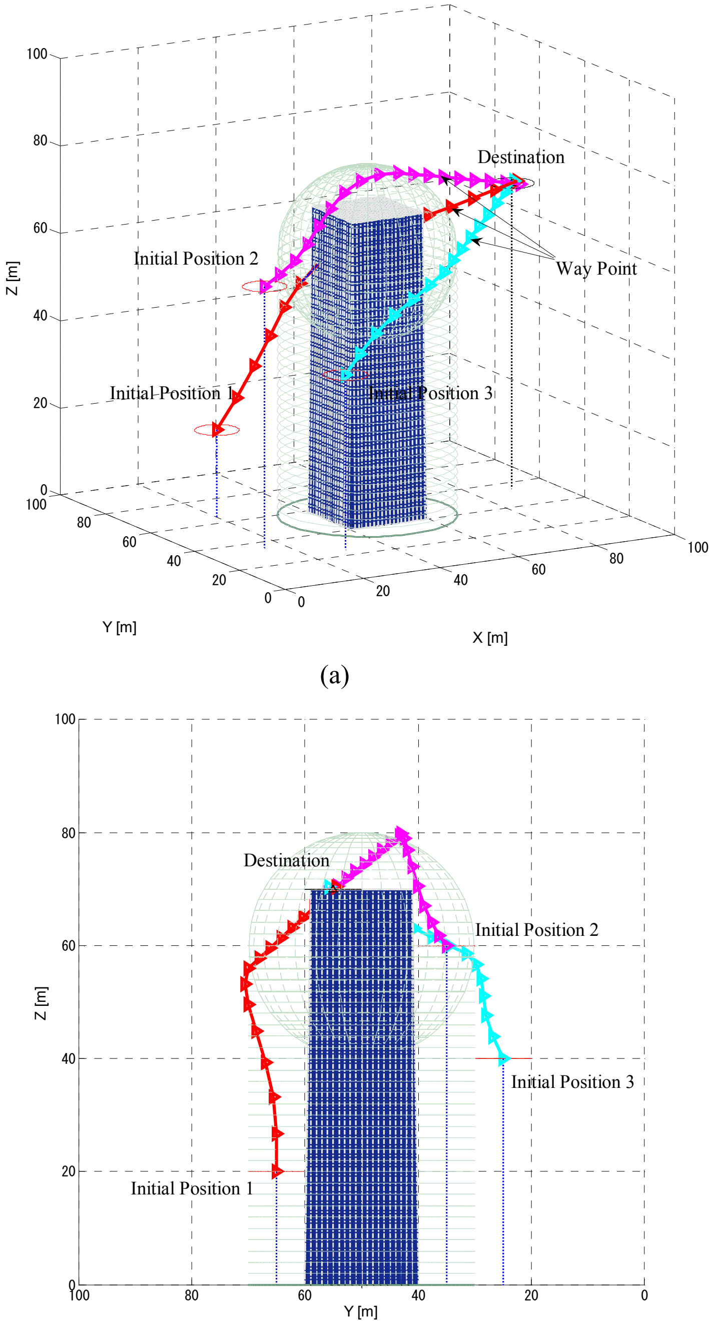


Figure 2.11 Two suitable paths were generated among numerous paths that can be planned by the extended limit-cycle method. The shortest path was then decided by comparing two distances depicted as a solid line.

The second simulation was performed in the cases of that the UAV's initial positions are at three different points: point 1 at (15, 35, 60), point 2 at (30, 25, 40), and the last point 3 at (20, 65, 20) as shown in Figure 2.12 (a). The results of this simulation showed that all of the three paths were successfully generated without any collisions while avoiding the obstacle even though the UAV was started from three different initial positions. Furthermore, as shown in Figure 2.12 (b) depicting the same result as Figure 2.12 (a) but different view from *yz*-plane in detail, three generated paths were verified that no gabs between the sphere and paths occurred. Thus, each of generated paths with limit-cycle showed that they overcame the fact that the limit-cycle can be drawn only in 2 dimensions.



(a)

(b)

Figure 2.12 (a) The three paths from different initial positions are successfully generated. '▷' indicates waypoints from the UAV's initial position to its destination. (b) A view from the side of *yz*-plane. There are no gaps between the sphere and three paths. Thus, each of paths is smooth enough for UAVs to fly because of limit-cycle characteristics.

The next simulation was similarly implemented in the previous case, but a tall rectangular box shaped obstacle was changed to a long horizontal box shaped obstacle depicted in Figure 2.13. First, since the shape of obstacle looks like a capsule, the safety margin boundary was drawn with two spheres and a cylinder. Then, with the similar procedure of applying the proposed path planning algorithm to each of initial positions, three paths were successfully generated depicted as three solid lines.

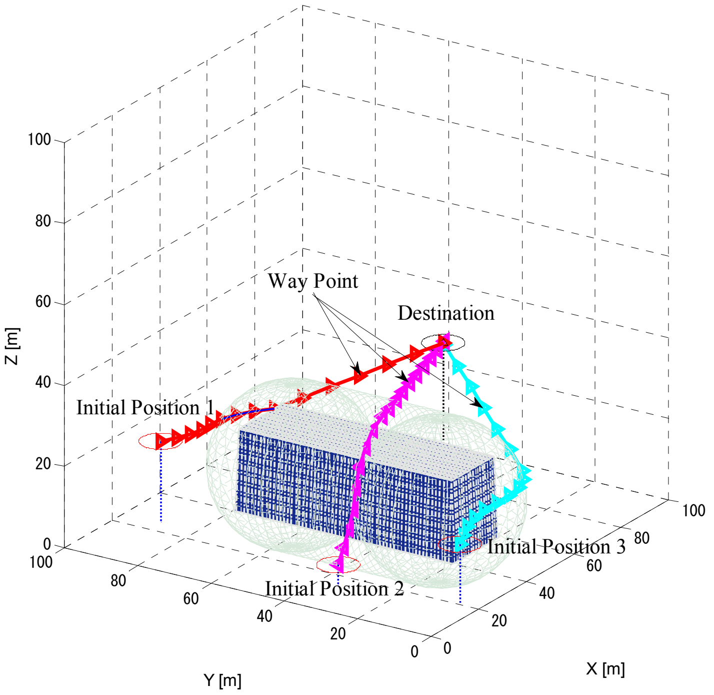


Figure 2.13 To cover a long horizontal box shaped obstacle with the safety margin boundary, two spheres and a cylinder are applied. In this condition, the paths were also successfully generated.

2.7 Path Planning for Dynamic Obstacle Avoidance

We have presented the path planning algorithm to generate a path to avoid static obstacles in the previous sub-chapter. However, it is difficult for a UAV to avoid dynamic obstacles with this algorithm. In this sub-chapter, therefore, we introduce an algorithm that can successfully generate a path to avoid dynamic obstacles. The algorithm is mainly comprised of a collision detection method [21] and standard rules of airplane traffic [22]; when combined, this algorithm reduces the chance of a collision and allows a UAV to reach the goal effectively in a short period of time.

2.7. 1 Collision Detection Method

Figure 2.14 simply draws collision detection process between moving objects such as a UAV and an obstacle. We supposes two objects move from t = 0 to t = 1 at a certain speed in a straight line and cannot cross or overlap.

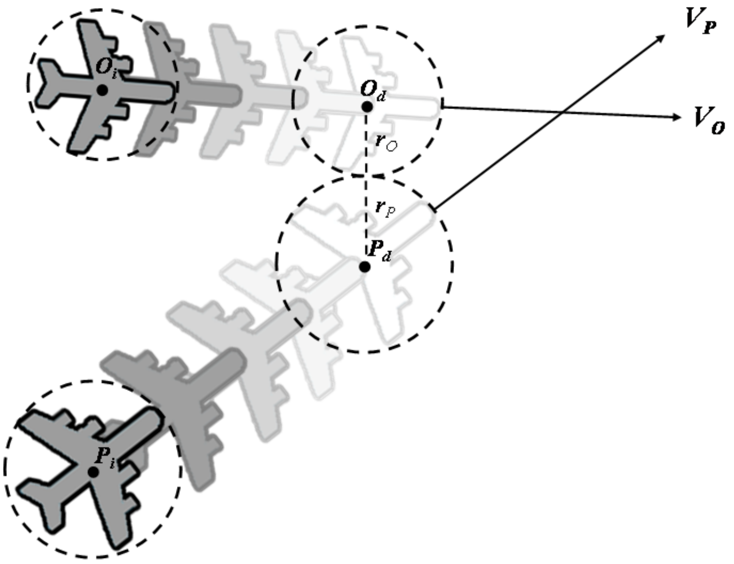


Figure 2.14 Collision detection between two moving objects.

Let's assume that the UAV is enclosed with a sphere of radius r_P and the obstacle is enclosed with a sphere of radius r_O, depicted in Figure 2.14. Then, the initial position and destination of the UAV are $\boldsymbol{P}_i$, $\boldsymbol{P}_d$ and the initial position and destination of the obstacle are $\boldsymbol{O}_i$, $\boldsymbol{O}_d$. In this case $\boldsymbol{V}_P$ and $\boldsymbol{V}_O$, equals to each object's velocity vector, are as follows.

$$\begin{aligned}\boldsymbol{V}_P &= \boldsymbol{P}_i - \boldsymbol{P}_d \\ \boldsymbol{V}_O &= \boldsymbol{O}_i - \boldsymbol{O}_d\end{aligned} \tag{2.22}$$

Thus, the position $\boldsymbol{P}(t)$ of the sphere's center of the UAV and the position $\boldsymbol{O}(t)$ of the sphere's center of the obstacle are as follows.

$$\begin{aligned}\boldsymbol{P}(t) &= \boldsymbol{P}_i - t\boldsymbol{V}_P \\ \boldsymbol{O}(t) &= \boldsymbol{O}_i - t\boldsymbol{V}_O\end{aligned} \tag{2.23}$$

At this point, When the distance d between $\boldsymbol{P}(t)$ and $\boldsymbol{O}(t)$ equals to $r_P + r_O$, then the time t is contact time of moving obstacle and UAV. First of all, square of the distance between $\boldsymbol{P}(t)$ and $\boldsymbol{O}(t)$ is as follows,

$$d^2 = \|\boldsymbol{P}(t) - \boldsymbol{O}(t)\|^2 \tag{2.24}$$

Let we put Eq. (2.23) into Eq. (2.24),

$$d^2 = \|\boldsymbol{P}_i - t\boldsymbol{V}_P - \boldsymbol{O}_i - t\boldsymbol{V}_O\|^2 \tag{2.25}$$

For convenience, we define **A** and **B** as follows.

$$\begin{aligned}\mathbf{A} &= \boldsymbol{P}_i - \boldsymbol{O}_i \\ \mathbf{B} &= \boldsymbol{V}_P - \boldsymbol{V}_O\end{aligned} \tag{2.26}$$

Using Eq. (2.26), Eq. (2.25) can rewrite as follows.

$$\begin{aligned} d^2 &= \|\mathbf{A} + t\mathbf{B}\|^2 \\ &= (\mathbf{B} \cdot \mathbf{B})t^2 + \ 2t(\mathbf{A} \cdot \mathbf{B}) + (\mathbf{A} \cdot \mathbf{A}) \end{aligned} \tag{2.27}$$

This is a quadratic equation in *t*. Writing the quadratic in the form $at^2 + 2bt + c = 0$, with $a = \mathbf{B} \cdot \mathbf{B}$, $b = \mathbf{A} \cdot \mathbf{B}$, and $c = \mathbf{A} \cdot \mathbf{A} - d^2$ gives the solution for *t* as

$$\begin{aligned} t_1 &= \frac{-b+\sqrt{b^2-4ac}}{2a} \\ t_2 &= \frac{-b-\sqrt{b^2-4ac}}{2a} \end{aligned} \tag{2.28}$$

In case the value in square root is negative, the UAV and obstacle are not bumped against each other. In another case $\mathbf{B} \cdot \mathbf{B}$ equals to zero, the moving UAV and obstacle are hauled up or two objects move the same direction and at the same speed, that is, the UAV and obstacle are not bumped against each other either.

At this point, setting $d = r_P + r_O$ gives us the two meeting time t_1 and t_2 when the two spheres are tangent. At the beginning of the paper, however, because we supposed two objects cannot cross and overlap, the collision time t_1 of two objects is only needed to determine when a collision occurs. Thus, if t_1 does not fall in the range [0,1), then no collision occurs during our time interval of interest.

2.7. 2 Standard Rules of Airplane Traffic

In case there are more than two UAVs in the same operation area, standard rules to control their traffic are necessary. In [23], the authors applied “Rules of the air” of the International Civil Aviation Organization (ICAO) annex 2 and “Right of way” of Federal Aviation Regulation (FAR) 91.113 to collision avoidance for UAVs as follows: “If both airplanes approach from opposite sides, they are supposed to give way by turning right away from each other to avoid a collision, and if flying airplanes come into conflicting paths side by side, the left airplane turns right to yield.”

Similarly, we set up the standard rules of airplane traffic as follows:

1. If two UAVs approach from opposite sides (i,e., more than 90 degrees), the UAV is supposed to give way by turning right away from the moving obstacle to avoid a collision. Therefore, a path for the UAV is generated as depicted in Figure 2.15, where $\boldsymbol{P}_i$ and $\boldsymbol{P}_d$ are the initial position and the destination of the UAV and $\boldsymbol{O}_i$ and $\boldsymbol{O}_d$ are the initial position and the destination of the obstacle.

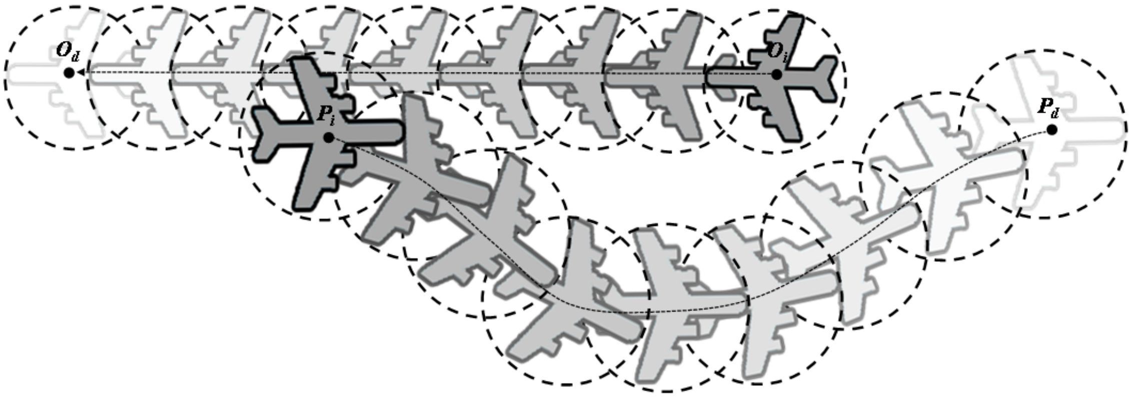


Figure 2.15 Collision avoidance by turing right away from the obtacle when two UAVs appoach form opposite sides.

2. If the dynamic obstacle comes into conflicting the UAV's path within less than 90 degrees, the UAV waits for a while to yield. This process is depicted in Figure 2.16, where $\boldsymbol{P}_i$ and $\boldsymbol{P}_d$ are the initial position and the destination of the UAV and $\boldsymbol{O}_i$ and $\boldsymbol{O}_d$ are

the initial position and the destination of the obstacle.

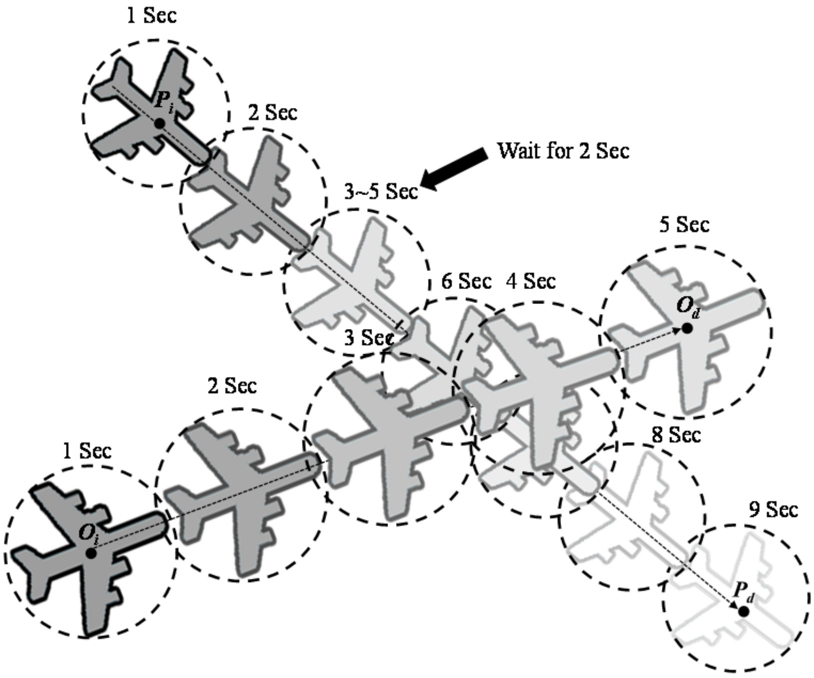


Figure 2.16 Collision avoidance by waiting for a when the dynamic obstacle comes into conflicting the UAV's path within less than 90 degrees.

2.7. 3 Simulation on Dynamic Obstacle Avoidance: the First Standard Rule of Airplane Traffic

In this sub-chapter, the first standard rule of airplane traffic, which comes into effect when an angle between the moving direction of the UAV and the obstacle is bigger than 90 degrees, is discussed with conducting a simulation work. The second standard rule of airplane traffic, which comes into effect when an angle between the moving direction of the UAV and the obstacle is less than 90 degrees, will be discussed after introducing path tracking method because this issue might be solved when considering path tracking rather than when considering path planning.

In the first simulation, the angle between the moving direction of the UAV and the obstacle was set for bigger than 90 degrees. In this case, the UAV is supposed to avoid the moving obstacle by turning right away. The result of this simulation is shown in Figure 2.17. The generated path is depicted with a line from $\boldsymbol{P}_i$ and $\boldsymbol{P}_d$ that are the initial and final positions of the UAV). This path was generated with the path planning algorithm presented in Chapter 2.5 at the predicted position of a collision (42, 52) with the supposition that an obstacle exists at that position. The radius of the virtual obstacle is equals to the sum of radius of the two spheres r_R and r_O depicted in Figure 2. 14. As a result, Figure 2.17 shows that the path for collision avoidance with the dynamic obstacle was successfully generated from the initial position (20, 50) to the destination (52, 50).

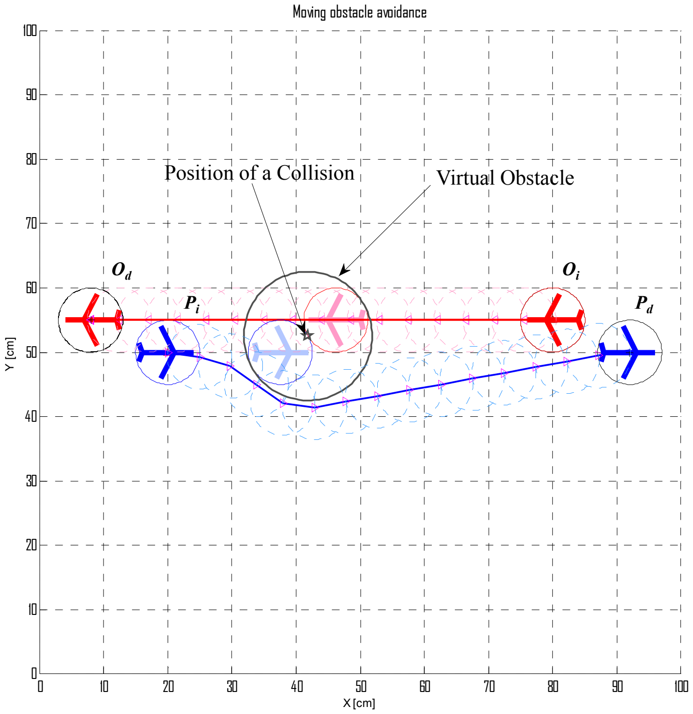


Figure 2.17 The result of dynamic obstacle avoidance in a case where both the UAV and obstacle approach each other from opposite sides.

CHAPTER 3. UAV (UNMANNED AERIAL VEHICLE)

3.1 Introduction

The UAV can be classified into two categories – fixed-wing aircraft and vertical take-off and landing (VTOL) type aircraft. When compared with fixed-wing aircrafts, VTOL type aircrafts (single rotor helicopter, coaxial type helicopter and quad-rotor) have specific characteristics like landing vertically in a limit space. Furthermore, such types are able to hover that make them suitable for applications that may be impossible to be performed using fixed-wing aircrafts [24].

Depending on their shape, VTOL type aircrafts comprise several types of aircrafts. A single rotor aircraft, known as a standard helicopter, is one of the VTOL type aircrafts. This type aircraft is very hazardous to humans and other objects in an indoor environment due to the possibility that exposed rotor blades may collide with obstacles and cause the aircraft to crash. On the other hand, a quad-rotor aircraft is much safer in terms of the rotors, which are smaller and can be enclosed [25].

Accordingly, we primarily focus on analyzing the dynamics of the quad-rotor aircraft and designing controllers for its stable flight.

3.2 Dynamic Modeling of a Quad-rotor Type Aircraft

Deriving mathematical modeling or differential equations is necessary for the control of the quad-rotor position and altitude. However, it is hard for the complicated structure of the quad-rotor type to express its motion with only a simple modeling. In addition, since the quad-rotor type aircraft includes highly nonlinear factors, we need to consider several assumptions in order to get a desired model [26].

1) The body is rigid and symmetrical.

2) The rotors are rigid, i.e. no blade flapping occurs.

3) The difference of gravity by altitude or the spin of the earth is minor.

4) The center of mass and body fixed frame origin coincide.

These assumptions can be formed because of slower speed and lower altitude of the quad-rotor aircraft whose body is rigid having 6 DOF (Degree of Freedom) as compared to a regular aircraft. Under these assumptions, it is possible to describe the fuselage dynamics. Therefore, to mathematically illustrate the fuselage dynamics of the quad-rotor type aircraft, a coordinate system should be defined. The coordinate system can be divided into an earth frame {E} and a body frame {B} as shown in Figure 3.1.

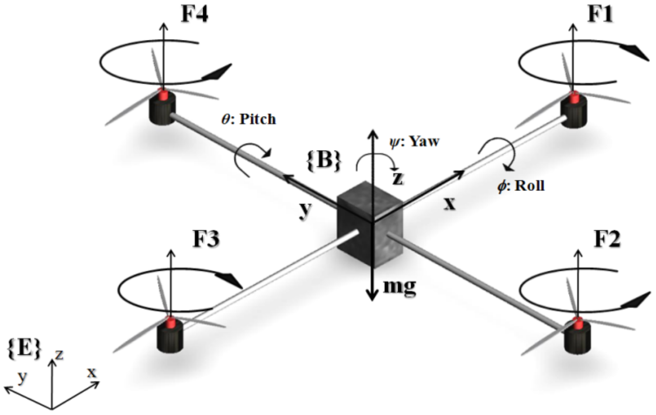


Figure 3.1 The coordinate system with an earth frame {E} and a body frame {B}.

The rotational transformation matrix between the earth frame and the body frame can be obtained based on Euler angles in Figure 3.2 [27].

$$R_{EB} = R_{\psi} R_{\theta} R_{\phi} = \begin{bmatrix} C_{\theta} C_{\psi} & S_{\phi} S_{\theta} C_{\psi} - C_{\phi} S_{\psi} & C_{\phi} S_{\theta} C_{\psi} + S_{\phi} S_{\psi} \\ C_{\theta} S_{\psi} & S_{\phi} S_{\theta} S_{\psi} + C_{\phi} C_{\psi} & C_{\phi} S_{\theta} S_{\psi} - S_{\phi} C_{\psi} \\ -S_{\theta} & S_{\phi} C_{\theta} & C_{\phi} C_{\theta} \end{bmatrix} \tag{3.1}$$

where *C* and *S* indicate the trigonometric cosine and sine functions, respectively.

The transformation of velocities between the earth frame and body frame can be derived from Eq. (3.1).

$$\begin{bmatrix} \upsilon \\ v \\ \omega \end{bmatrix} = R_{EB} \begin{bmatrix} \upsilon_B \\ v_B \\ \omega_B \end{bmatrix} \tag{3.2}$$

Similarly, positions, forces, moments, accelerations and rotational velocities can be transformed based on R_{EB} between coordinate systems. In the body frame, the forces are presented as

$$F_B = \begin{bmatrix} F_{xB} \\ F_{yB} \\ F_{zB} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{3.3}$$

Accordingly, in the earth frame, the forces can be defined as

$$F_E = \begin{bmatrix} F_{xE} \\ F_{yE} \\ F_{zE} \end{bmatrix} = R_{EB}F_B = \begin{bmatrix} C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\phi S_\theta S_\psi - S_\phi C_\psi \\ C_\phi C_\theta \end{bmatrix} (\sum_{i=1}^{4} F_i) \tag{3.4}$$

Therefore, equations of motion in the earth frame are derived by the Newton's laws.

$$m\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} F_{xE} \\ F_{yE} \\ F_{zE} \end{bmatrix} - \begin{bmatrix} K_{fx} & 0 & 0 \\ 0 & K_{fy} & 0 \\ 0 & 0 & K_{fz} \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = \begin{bmatrix} F_{xE} - K_{fx}\dot{x} \\ F_{yE} - K_{fy}\dot{y} \\ F_{zE} - K_{fz}\dot{y} - mg \end{bmatrix} \tag{3.5}$$

where K_{fx}, K_{fy}, and K_{fz} are the drag coefficients, which can be ignored at low speeds.

By extending the Newton's law on rotational dynamics, its equation can be written as [27]

$$\ddot{\phi} = \frac{l(K_l(\omega_4^2 - \omega_2^2))}{I_x} \tag{3.6}$$

$$\ddot{\theta} = \frac{l(K_l(\omega_3^2 - \omega_1^2))}{I_y} \tag{3.7}$$

$$\ddot{\psi} = \frac{K_d(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2)}{I_z} \tag{3.8}$$

where l is the distance from the center of rotors to the quad-rotor aircraft center of mass. K_l and K_d indicate the lift coefficient and drag coefficient respectively and w_i is the angular rotor speed. I_x, I_y, I_z are moments of inertia of the quad-rotor.

As a consequence, the complete dynamic model which governs the quad-rotor aircraft is as follows:

$$\begin{cases} \ddot{x} = \dfrac{(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)u_1 - K_{fx}\dot{x}}{m} \\ \ddot{y} = \dfrac{(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)u_1 - K_{fy}\dot{y}}{m} \\ \ddot{z} = \dfrac{(\cos\phi\cos\theta)u_1 - K_{fz}\dot{z}}{m} - g \\ \ddot{\phi} = \dfrac{l(u_2 - K_l\dot{\phi})}{I_x} \\ \ddot{\theta} = \dfrac{l(u_3 - K_l\dot{\theta})}{I_y} \\ \ddot{\psi} = \dfrac{(u_4 - K_d\dot{\psi})}{I_z} \end{cases} \tag{3.9}$$

where u_i (i=1, 2, 3, 4) are control inputs of the model.

$$\begin{cases} u_1 = F_1 + F_2 + F_3 + F_4 \\ u_2 = F_4 - F_2 \\ u_3 = F_3 - F_1 \\ u_4 = F_1 + F_3 - F_2 - F_4 \end{cases} \tag{3.10}$$

CHAPTER 4. PATH TRACKING METHOD

In this chapter, fuzzy logic control (FLC) for tracking a path that is generated by the path planning algorithm presented in Chapter 2 is introduced. After that, several simulations will be performed, in order to verify performance of a path tracking method based on dynamic modeling of a quad-rotor aircraft presented in Chapter 3.

4.1 Typical Fuzzy Logic Control

Due to the ability to handle nonlinear system, FLC is called as an ideal tool to deal with uncertain situations by representing the input and output relation in the "*if-then*" manner and constructing knowledge base [8].

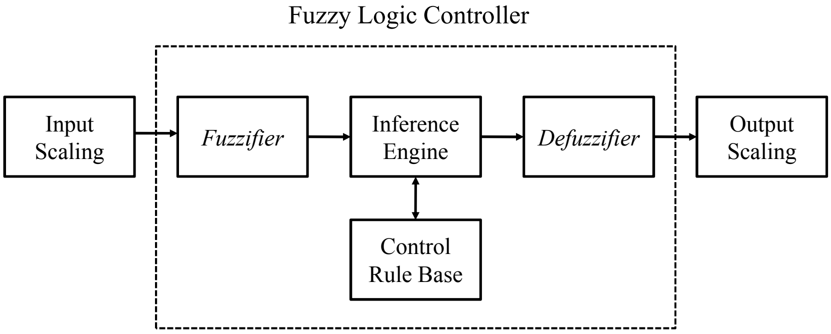


Figure 4.1 Typical Fuzzy Logic Control (FLC).

Figure 4.1 shows a typical FLC that consists of the input and output scaling factors, the *fuzzifier*, *defuzzifier*, inference engine and the rules base. Input and output-scaling factors are essential in terms of designing the rules base without having to consider the physical domain (plant, process) of the input and output signals [28]. *Fuzzifier* is a process of transforming crisp values (numerical values) into grades of membership for linguistic terms of fuzzy sets [29]. The membership function is used to associate a grade to each linguistic term. The next stage is the fuzzy inference system, which is the process of formulating the mapping from a given input to an output using fuzzy logic. The Mandani's max-min inference method is one of the most widely used and proven methodologies [8] [28] [30]. Finally, *defuzzifier* is in a sense the reverse of the *fuzzifier*: it converts all the fuzzy terms created by the rule base of the controller to crisp terms and sends them to the physical system, so as to execute the control of the system.

4.2 Path Tracking Method Scheme

The fundamental path tracking scheme of the VTOL type UAV system in three-dimension coordinate is illustrated in Figure 4.2.

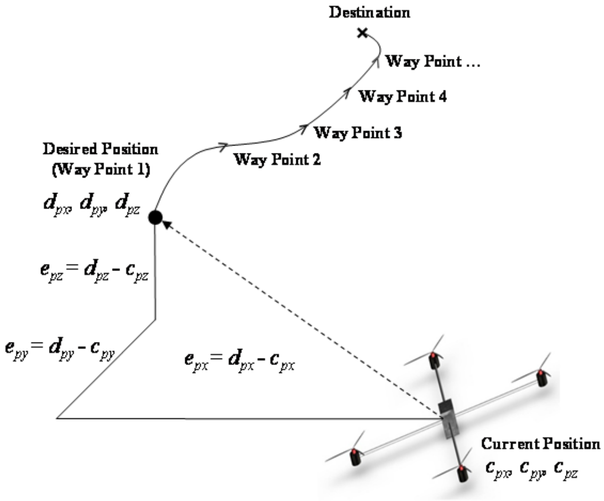


Figure 4.2 The path tracking scheme.

First, waypoints on the path from the UAV's initial position to its destination are generated at regular intervals by the path planning algorithm introduced in Chapter 2. Consequently, if the UAV orderly tracks all of these waypoints, the UAV will be successfully able to reach the destination.

In Figure 4.2, c_{px}, c_{py}, c_{pz} and d_{px}, d_{py}, d_{pz} indicate current position and desired position of the UAV, respectively. e_{px}, e_{py} and e_{pz} are then the horizontal and vertical distance between the current position of the airplane and the desired position, respectively. To move this desired position (way point) effectively in the VTOL type UAV system, the thrust and attitude of the UAV, which mostly affect the its position, must be controlled. Hence, the FLC, which is activated after generating the path, will be in charge of controlling the overall system of the VTOL type UAV so that it can move to any desired position effectively. The FLC, shown in Figure 4.3, is composed of nine fuzzy logic controller modules, module 1 to module 9, which are responsible for position control, velocity control and angle control corresponding to different situations; when combined they may adequately control thrust and attitude of the UAV.

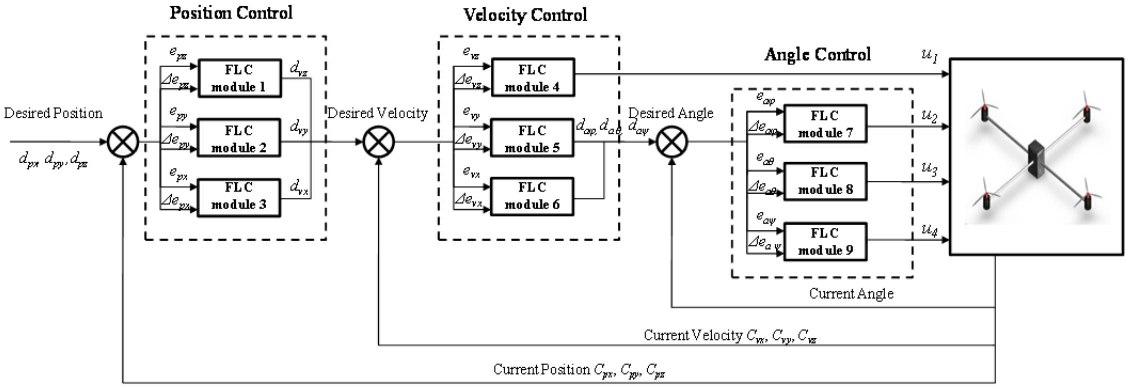


Figure 4.3 The configuration of overall FLC.

The first three fuzzy logic controller modules, module 1 to module 3, for the position control are responsible for *x*, *y* and *z* position, respectively. Each module has two inputs: a) position error $\{e_{px}, e_{py}, e_{pz}\}$, b) position error rate $\{\Delta e_{px}, \Delta e_{py}, \Delta e_{pz}\}$. The position error is the difference between the desired position $\{d_{px}, d_{py}, d_{pz}\}$ and the current position $\{c_{px}, c_{py}, c_{pz}\}$, which is feedback from the airplane. The position error rate can be obtained by

$$\Delta e_{p\{x,y,z\}}(t) = e_{p\{x,y,z\}}(t) - e_{p\{x,y,z\}}(t-1) \tag{4.1}$$

In particular, the z position error rate, Δe_{pz}, indicates whether the VTOL type UAV is approaching the desired z position or if it is going away from it [10]. Outputs are then the desired x, y and z velocity $\{d_{vx,} d_{vy,} d_{vz}\}$, which are also inputs for other fuzzy logic controller modules in charge of the x, y and z velocity control. The second three modules, module 4 to module 6, for the velocity control are composed of the same structure of the first three modules. Each module also has two inputs: a) velocity error $\{e_{vx,} e_{vy,} e_{vz}\}$, b) velocity error rate $\{\Delta e_{vx,} \Delta e_{vy,} \Delta e_{vz}\}$. The velocity error is the difference between the desired velocity and the current velocity $\{c_{vx,} c_{vy,} c_{vz}\}$ of the airplane. Especially, output of the module 4 becomes one of the control inputs of the VTOL type UAV, which is the thrust command, and outputs of module 5 and module 6 are transformed into third three modules, module 7 to module 9, for angle control, which are the desired angle $\{d_{a\varphi,} d_{a\theta,} d_{a\psi}\}$. Finally, the control inputs of the VTOL type UAV, u_i (i=1, 2, 3, 4) denoted in Eq. (3.10), are decided by the outputs of the final three fuzzy logic controller modules.

Crisp values of all inputs are transformed into grades of membership for linguistic terms of fuzzy sets by *fuzzifier*. Linguistic variables have a finite number of linguistic values representing the linguistic variable error $\{e_{px,} e_{py,} e_{pz}\}$, $\{e_{vx,} e_{vy,} e_{vz}\}$, $\{e_{a\varphi,} e_{a\theta,} e_{a\psi}\}$ and error rate $\{\Delta e_{px,} \Delta e_{py,} \Delta e_{pz}\}$, $\{\Delta e_{vx,} \Delta e_{vy,} \Delta e_{vz}\}$, $\{\Delta e_{a\varphi,} \Delta e_{a\theta,} \Delta e_{a\psi}\}$ are: *{Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZO), Positive Small (PS), Positive Medium (PM), Positive Big (PB)}*. Linguistic values that describe the output linguistic variables connecting with the next fuzzy logic controller modules are: *{Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZO), Positive Small (PS), Positive Medium (PM), Positive Big (PB)}*. Their corresponding membership functions are shown in Figure 4.4.

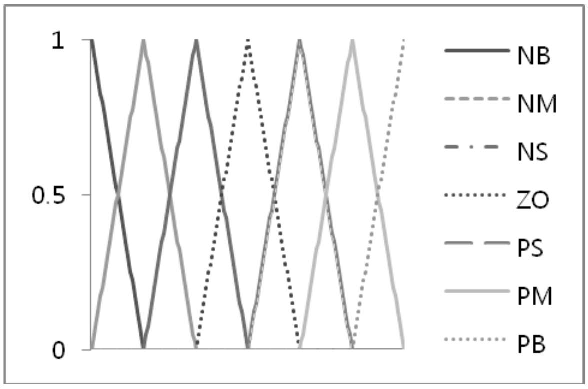


Figure 4.4 Fuzzy memberships for all inputs and outputs.

In order to facilitate computations, triangular membership functions are utilized in this work for all input and output membership functions empirically defined after exhaustive simulation studies [29].

In the FLC, which is composed of nine fuzzy logic controller modules, the proposed fuzzy logic "*if-then*" rule bases performing the inference are given in the following form:

$$R^j\text{: } \textit{IF input } e \textit{ is } A_{1j} \textit{ and... input } \Delta e \textit{ is } A_{nj}$$

$$\textit{THEN output } y \textit{ is } B_j,\ j = 1 \ldots k$$

$$e = \{e_{px}, e_{py}, e_{pz}, e_{vx}, e_{vy}, e_{vz}, e_{a\varphi}, e_{a\theta}, e_{a\psi}\}$$

$$\Delta e = \{\Delta e_{px}, \Delta e_{py}, \Delta e_{pz}, \Delta e_{vx}, \Delta e_{vy}, \Delta e_{vz}, \Delta e_{a\varphi}, \Delta e_{a\theta}, \Delta e_{a\psi}\}$$

$$e, \Delta e, A_{nj}, B_j \qquad \{NB, NM, NS, ZO, PS, PM, PB\}$$

In the fuzzy inference system, which is the process of formulating the mapping from a given input to an output using fuzzy logic, 49 rules are needed to complete this mapping. The rules base for each module is summarized in Table 1, where there are two inputs used in. In addition, the Mandani's max-min inference method is employed because it is one of the most widely used and proven methodologies [8] [28] [30].

Table 4.1 Rules base generated using the function

e/*Δ*e	NB	NM	NS	ZO	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZO
NM	NB	NB	NB	NM	NS	ZO	PS
NS	NB	NB	NM	NS	ZO	PS	PM
ZO	NB	NM	NS	ZO	PS	PM	PB
PS	NM	NS	ZO	PS	PM	PB	PB
PM	NS	ZO	PS	PM	PB	PB	PB
PB	ZO	PS	PM	PB	PB	PB	PB

Finally, *defuzzifier* is performed using the center of gravity method [28]. It performs *defuzzification* by finding the center of the area encompassed by all the rules, and is mathematically described by

$$u[e,\Delta e] = \frac{\sum_{i=1}^{n} u_i \mu U(u_i)}{\sum_{i=1}^{n} \mu U(u_i)} \tag{4.2}$$

where $u[e, \Delta e]$ refers to the *defuzzified* overall control output, while u_i refers to the output variable and μU represents the aggregated membership function.

4.3 Simulation on Path Tracking Method

In this sub-chapter, the simulation was performed to verify the fuzzy logic control method, which is responsible for the control of path tracking of the VTOL type UAV. In this simulation, the quad-rotor UAV, which was introduced and analyzed in Chapter 3, was used as a test model in the case of which there are only two given positions: an initial position and desired position. To reach the desired position from the initial position, the quad-rotor UAV was controlled by the proposed FLC.

First, as the results shown in Figure 4.5 where '+' indicates traces of path tracking by the quad-rotor UAV while ascending, the UAV effectively reach the desired position without any overshoots. In addition, Figure 4.6 shows the each velocity ratio of *x*-*y*-*z* axes per time axis based, where '◆' is the velocity ratio of *x*-axis, '+' is the velocity ratio of *y*-axis, and '*' is the velocity of *z*-axis. The variation of velocity '*' should be faster than those of '◆', '+', because the distance of *z*-axis between the initial position and desired position is longer than those of the *x*-axis and *y*-axis. Figure 4.7 represents the each position of *x*-*y*-*z* axes per time axis based. As the experimental results show, the positions of each axis converge at the same time in order for the UAV to reach the desired position.

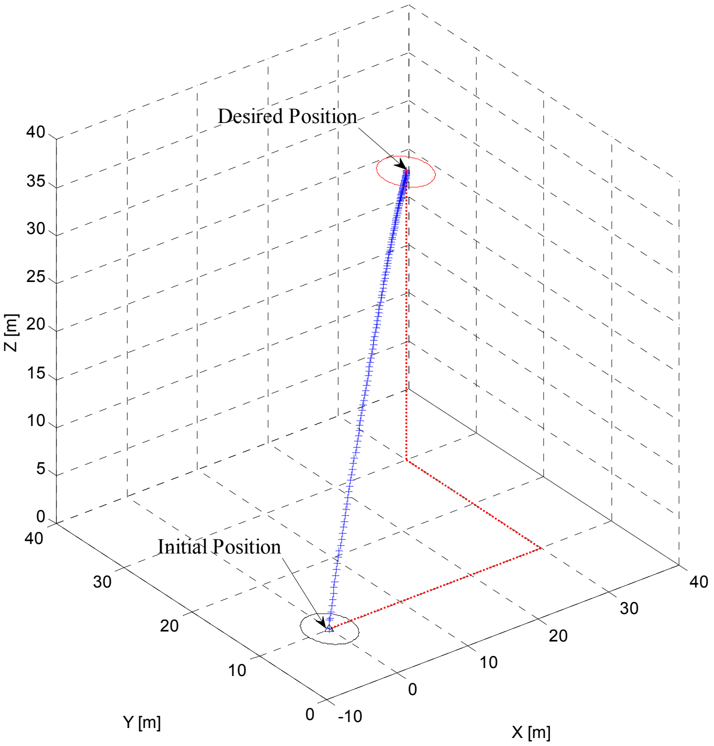


Figure 4.5 The traces of path tracking in the case of which there are only two given positions while ascending using FLC.

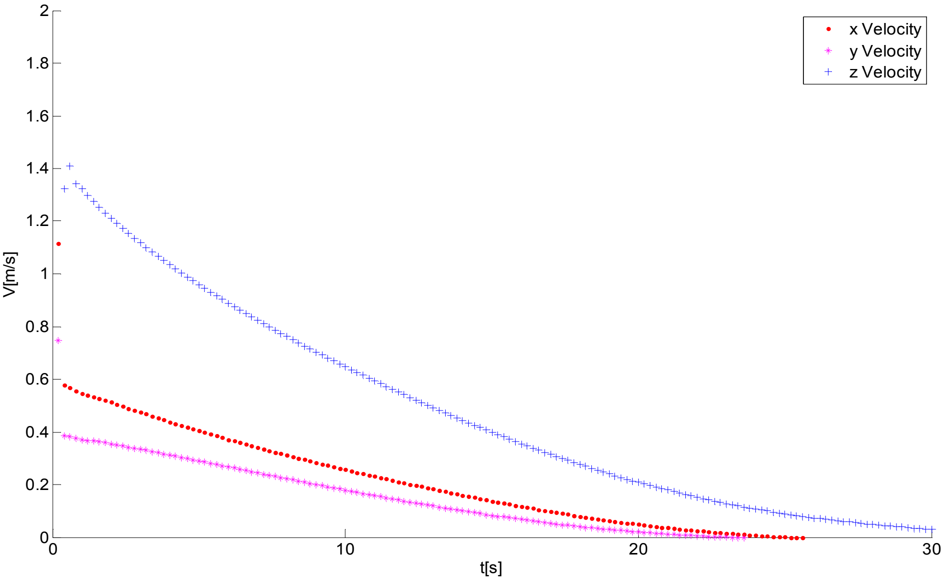


Figure 4.6 The each velocity ratio of *x*-*y*-*z* axes per time axis based while the quad-rotor UAV is ascending.

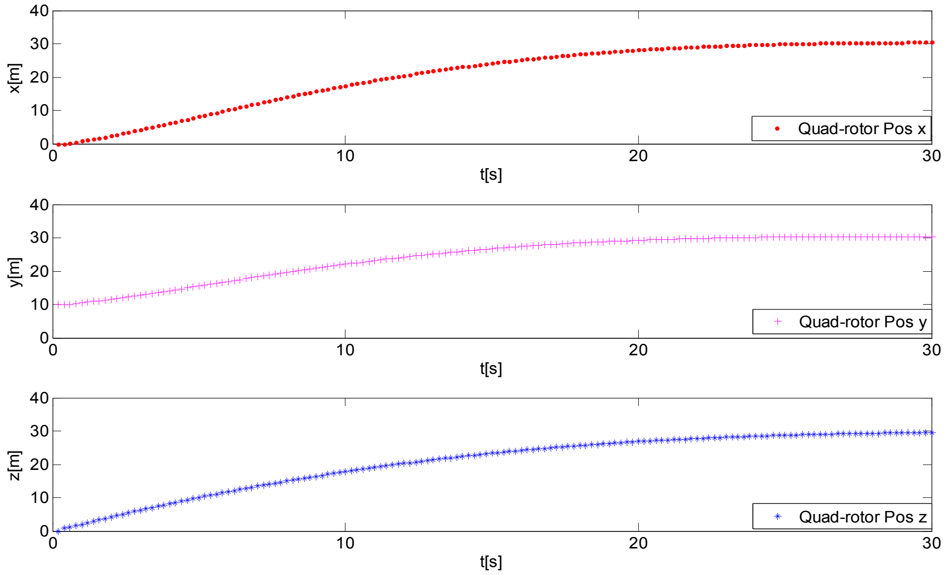


Figure 4.7 The each position of *x*-*y*-*z* axes per time axis based while the quad-rotor UAV is ascending.

Second, Figure 4.8 represents the simulation results of path tracking using the fuzzy logic method while the quad-rotor UAV is descending. As the evident results shown in Figure 4.8, where traces generate a smooth curve, the experimental results show the effective path tracking solutions. Figure 4.9 shows the velocity ratio of each *x*-*y*-*z* axes per time axis based while the UAV is descending. Compared to the Figure 4.6, the each variation of velocity is opposite to that of velocity in Figure 4.6. In addition, the each position of *x*-*y*-*z* axes per time based in Figure 4.10 verifies that the UAV reaches safely at the desired position at the same time.

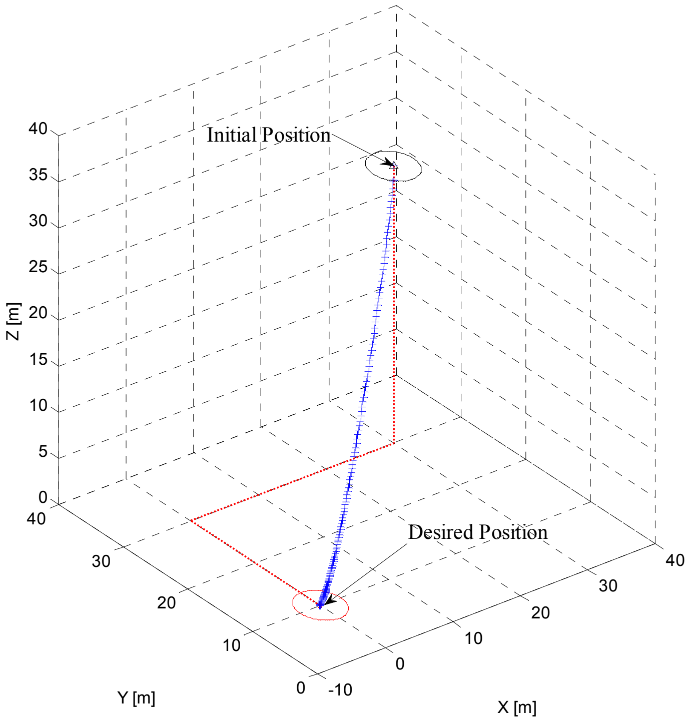


Figure 4.8 The traces of path tracking by the quad-rotor UAV while descending using FLC.

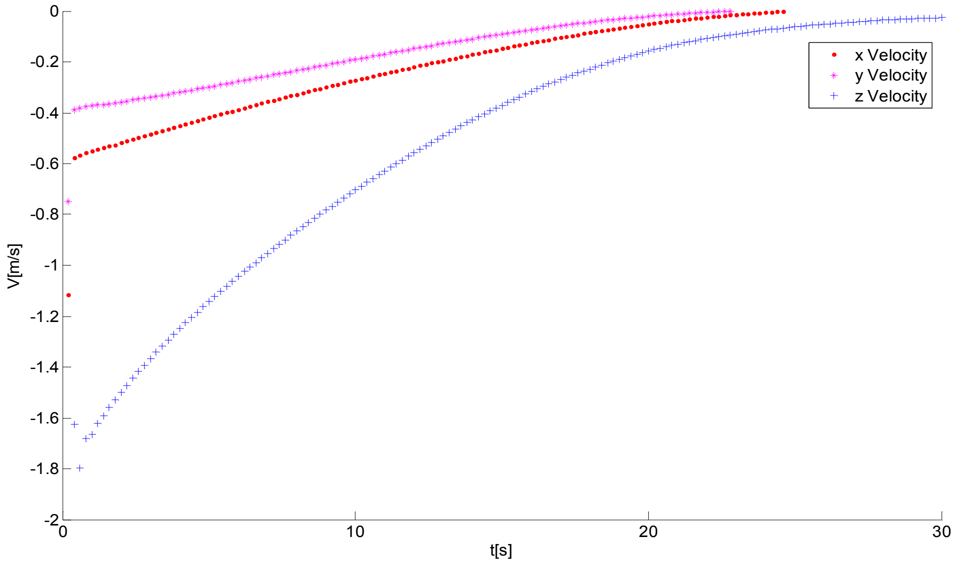


Figure 4.9 The each velocity ratio of *x*-*y*-*z* axes per time axis based while the quad-rotor UAV is descending.

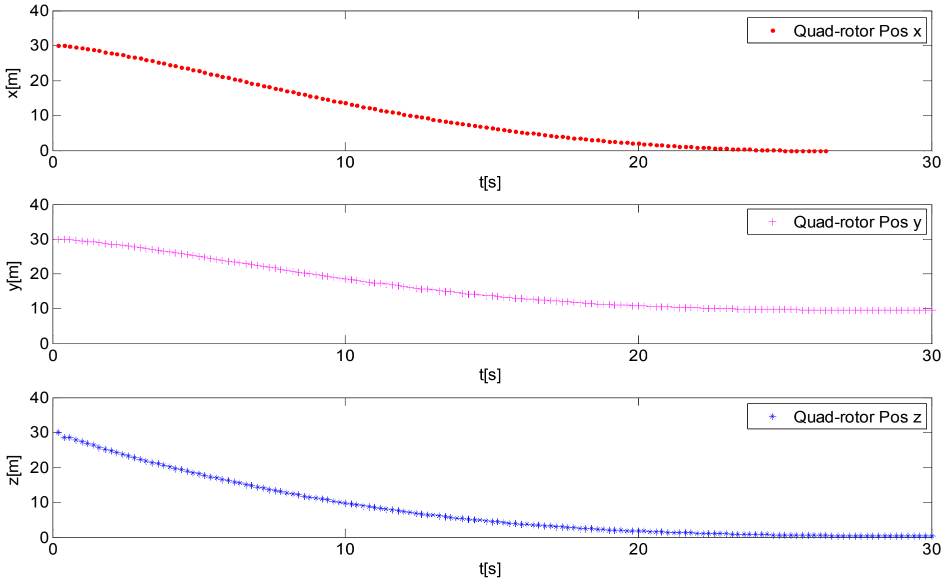


Figure 4.10 The each position of *x*-*y*-*z* axes per time axis based while the quad-rotor UAV is descending.

4.4 Simulation on Path Planning Algorithm Combined with Path Tracking Method

A simulation on the proposed navigation method comprised of a path planning algorithm and tracking method was conducted, and the results have been depicted in Figure 4.11. The conditions of this simulation derived from the previous simulations, which showed effectiveness results of the path planning algorithm depicted in Figure 2.12. In Figure 4.11, '▷' indicates waypoints from the UAV's initial position to its destination generated by the proposed path planning algorithm, and the solid lines passing through them mean the traces of path tracking by the quad-rotor UAV using the fuzzy logic method. This experimental results show that the quad-rotor UAV was successfully able to reach the destination by orderly passing through the designated waypoints wherever it started at different initial positions 1, 2 and 3.

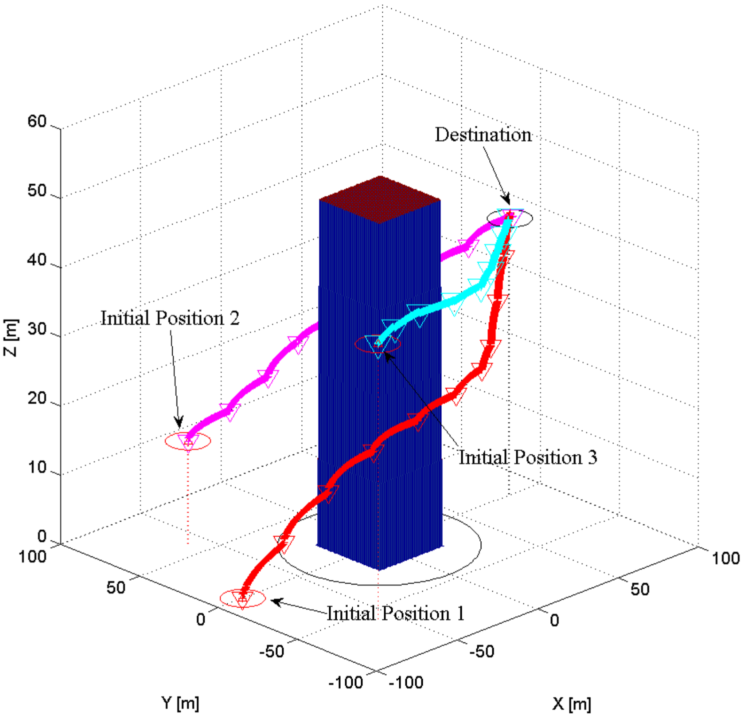


Figure 4.11 The result of a simulation on the proposed navigation method: There are three different initial positions and the same destination. '▷' indicates waypoints from the UAV's initial position to its destination.

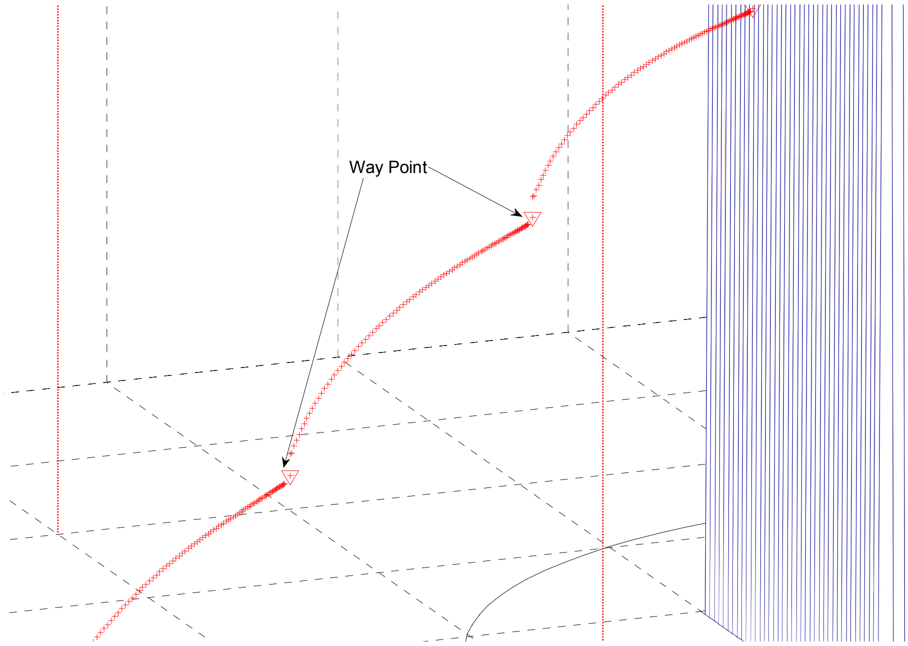


Figure 4.12 An enlarged portion of the problem of path tracking in Figure 4.11.

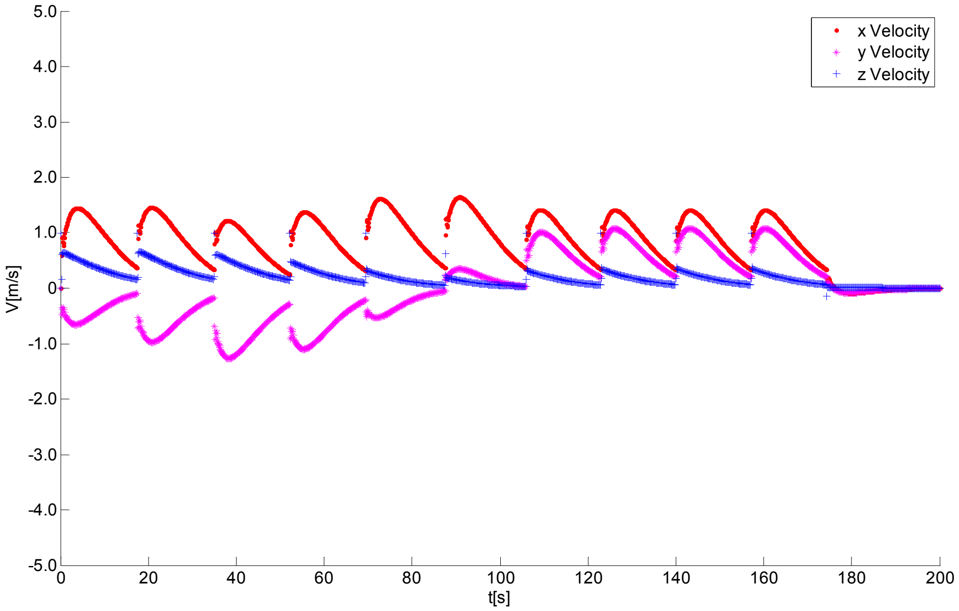


Figure 4.13 The each velocity ratio of *x*-*y*-*z* axes per time axis based while the quad-rotor UAV is flying toward the destination. There is a problem that the UAV repeats stop-and-go at regular intervals.

However, as we can see in the Figure 4.12, if the UAV tracks each of the designated waypoints generated by the proposed path planning algorithm, it throttles down as approaching the waypoints, depicted in Figure 4.13. In this case, when operating the UAV in the real environment, it repeats stop-and-go until reaching the destination, which might eventually result in the inefficient and slow flight. Therefore, through a following procedure such problems are solved.

1. Spheres centered at the waypoints are drawn in the form of $x_i^2 \ + y_i^2 + z_i^2 = r^2$, where x, y, and z represent the position of the waypoint (desired position), and i means the order of waypoints, and r means a radius of sphere, depicted in Figure 4.14.
2. In order for the UAV to reach the destination, it must pass the spheres with radius r before approaching each of the designated waypoints. Therefore, when the UAV passes the i_{th} sphere, the current desired position (x_i, y_i, z_i) is changed to the following desired position $(x_{i+1}, y_{i+1}, z_{i+1})$.

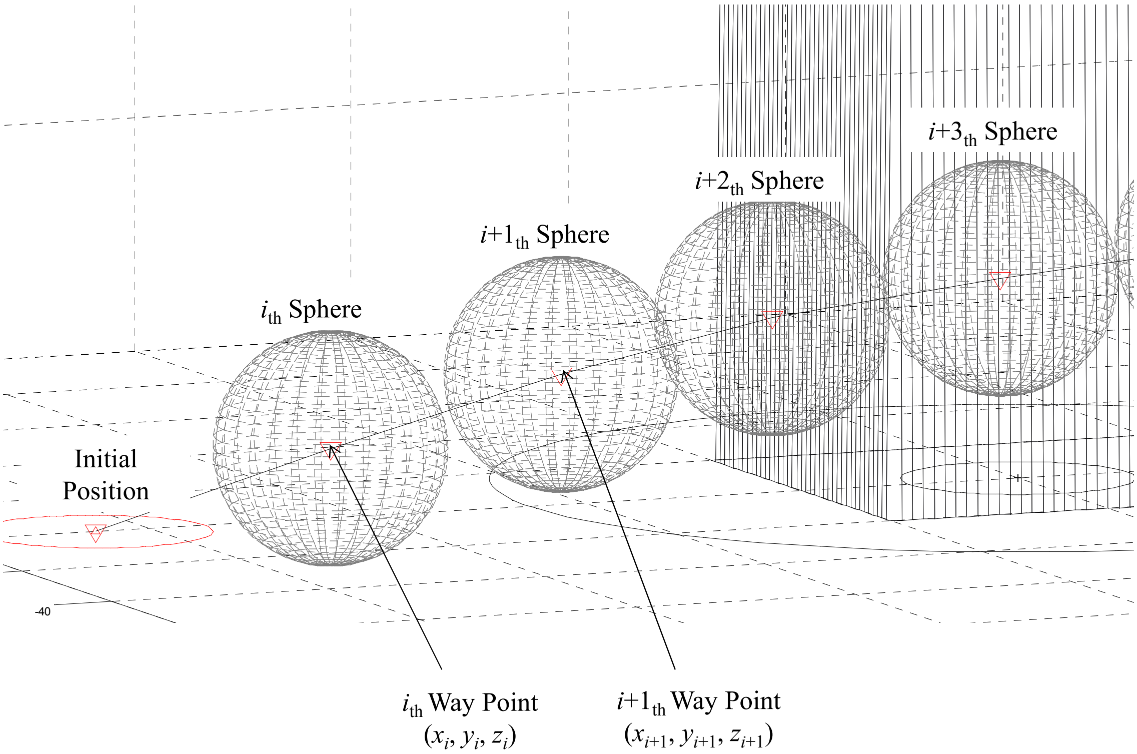

i+3$_{\text{th}}$ Sphere
i+2$_{\text{th}}$ Sphere
i+1$_{\text{th}}$ Sphere
i_{th} Sphere
Initial
Position
-40
i_{th} Way Point
(x_i, y_i, z_i)
i+1$_{\text{th}}$ Way Point
$(x_{i+1}, y_{i+1}, z_{i+1})$

Figure 4.14 Spheres centered at the waypoints to solve a problem that the UAV repeats stop-and go around the waypoints.

Figure 4.15, 4.16, and 4.17 show results of the simulation that was conducted in the same condition as before. As Figure 4.17 shows, the UAV was able to pass the waypoints by maintaining a relatively steady velocity, compared to Figure 4.13. In Figure 4.16, even though the UAV could not pass the designated waypoints more accurately than the last one because it moved toward the following desired position $(x_{i+1}, y_{i+1}, z_{i+1})$ before reaching the current desired position (x_i, y_i, z_i), the UAV was successfully able to reach the destination, as shown in Figure 4.15.

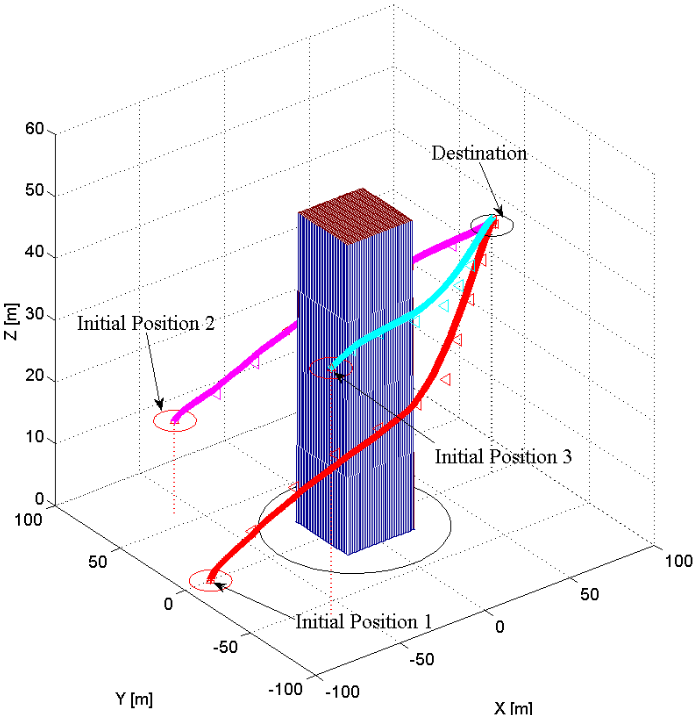


Figure 4.15 The result of a simulation on the new path tracking solving the problem that the UAV repeats stop-and go around the waypoints.

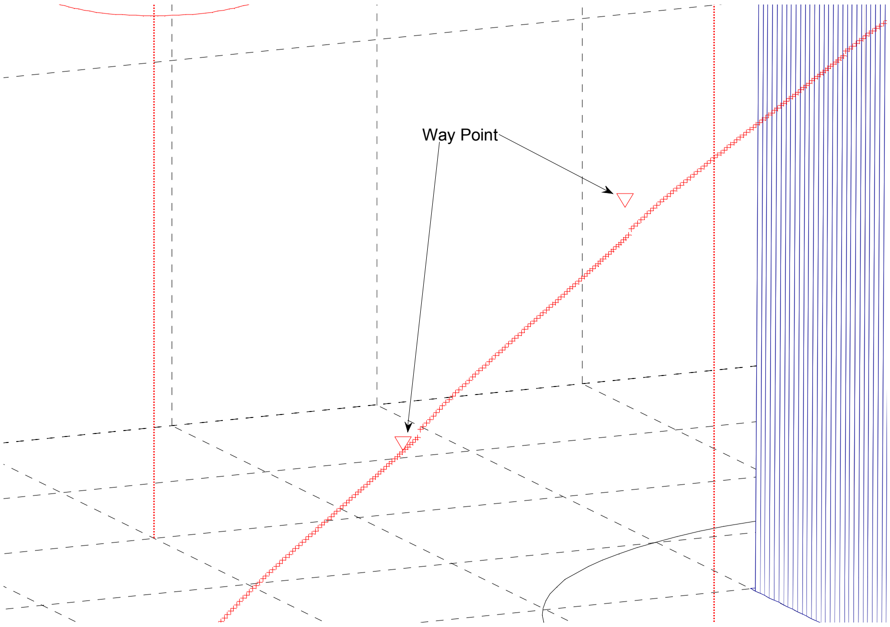


Figure 4.16 An enlarged portion of the simulation result.

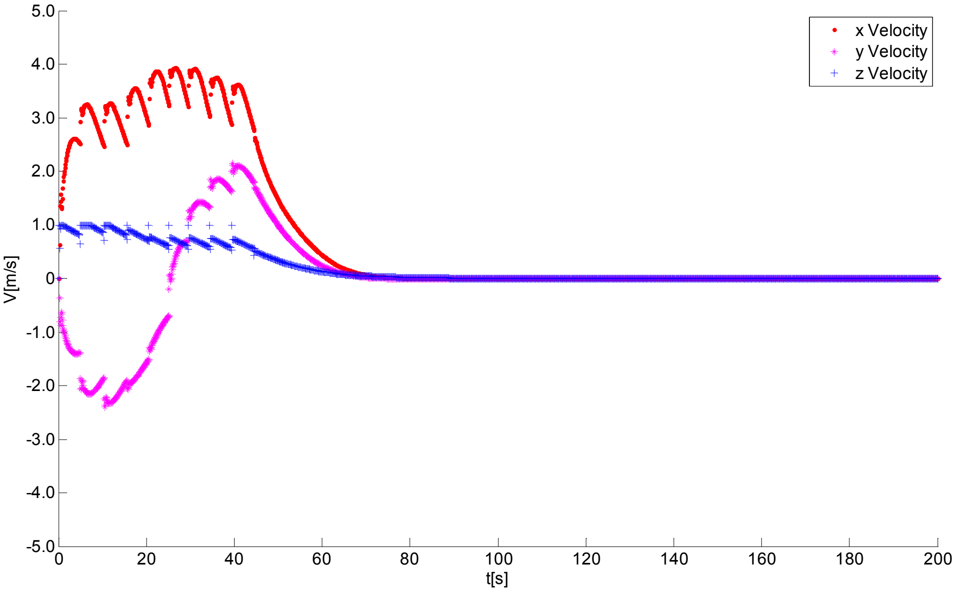


Figure 4.17 The each velocity ratio of *x*-*y*-*z* axes per time axis based while the quad-rotor UAV is flying toward the destination.

4.5 Simulation on Dynamic Obstacle Avoidance: the Second Standard Rule of Airplane Traffic

In this sub-chapter, a simulation mentioned in Chapter 2.7.3 is conducted to prove the second standard rule of airplane traffic. Note that this rule comes into effect when an angle between the moving direction of the UAV and the obstacle is less than 90 degrees.

From Figures 4.18 to Figure 4.21 show the results of the simulation. In this simulation, the angle between the moving direction of the UAV and the obstacle was set for less than 90 degrees. Therefore, the UAV was supposed to wait for a while until the obstacle passes, in order to avoid collisions with it. As a result, the UAV waited at the predicted position of a collision (43, 58, 61) for about 20 sec to avoid collisions with the obstacle, as depicted in Figure 4.20 and Figure 4.21 (area A). On the other hand, the dynamic obstacle is moving at a constant velocity, as depicted in Figure 4.19. Consequently, the UAV could successfully reach the destination (80, 30, 50) from the initial position (20, 80, 60).

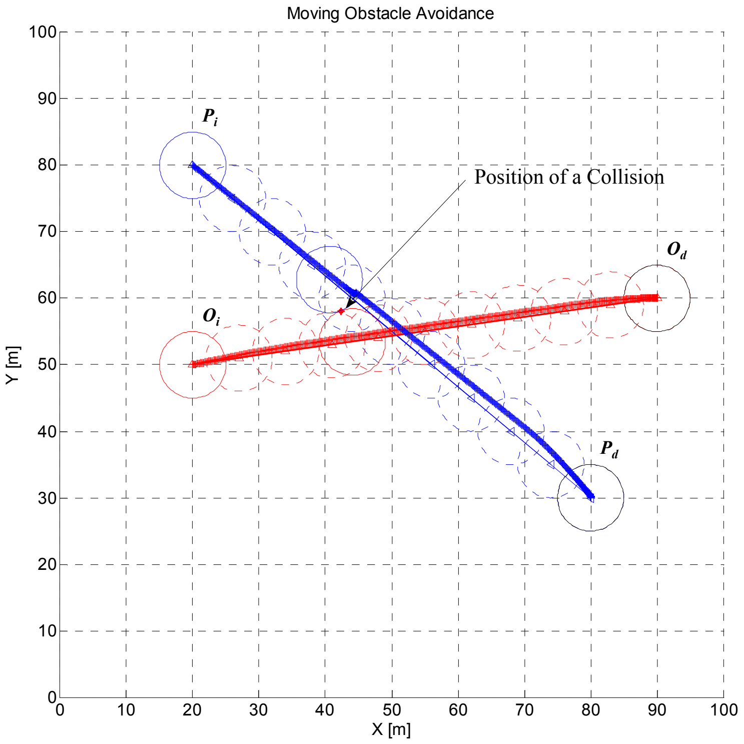


Figure 4.18 The result of dynamic obstacle avoidance in a case where the obtacle comes into a conflicting the UAV's path within less than 90 degree.

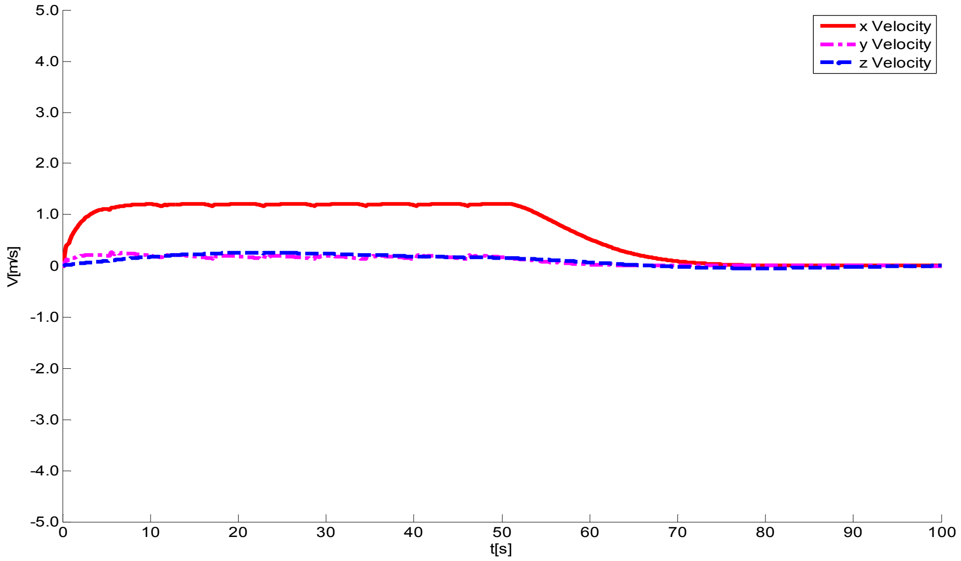


Figure 4.19 The each velocity ratio of *x*-*y*-*z* axes per time axis based while the dynamic obstacle is flying toward the destination.

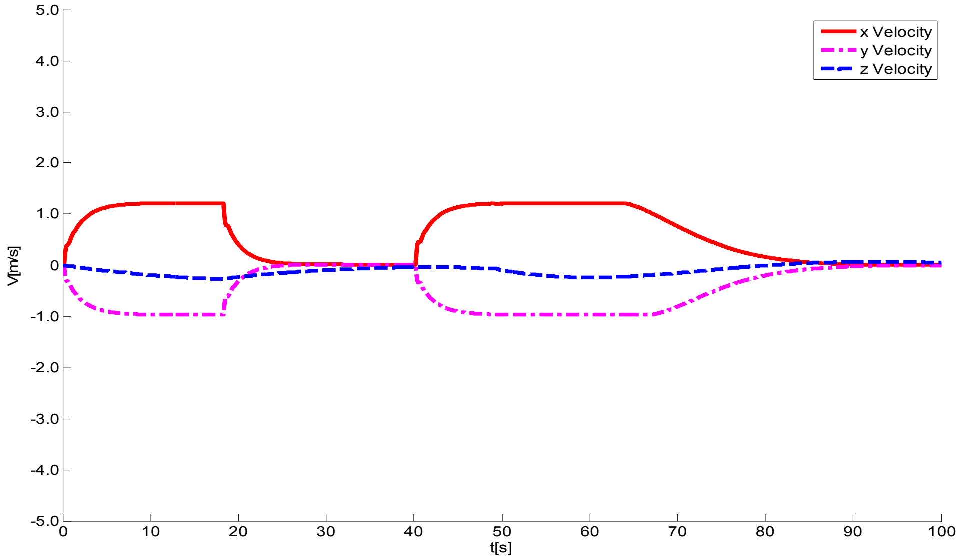


Figure 4.20 The each velocity ratio of *x*-*y*-*z* axes per time axis based while the UAV is flying toward the destination.

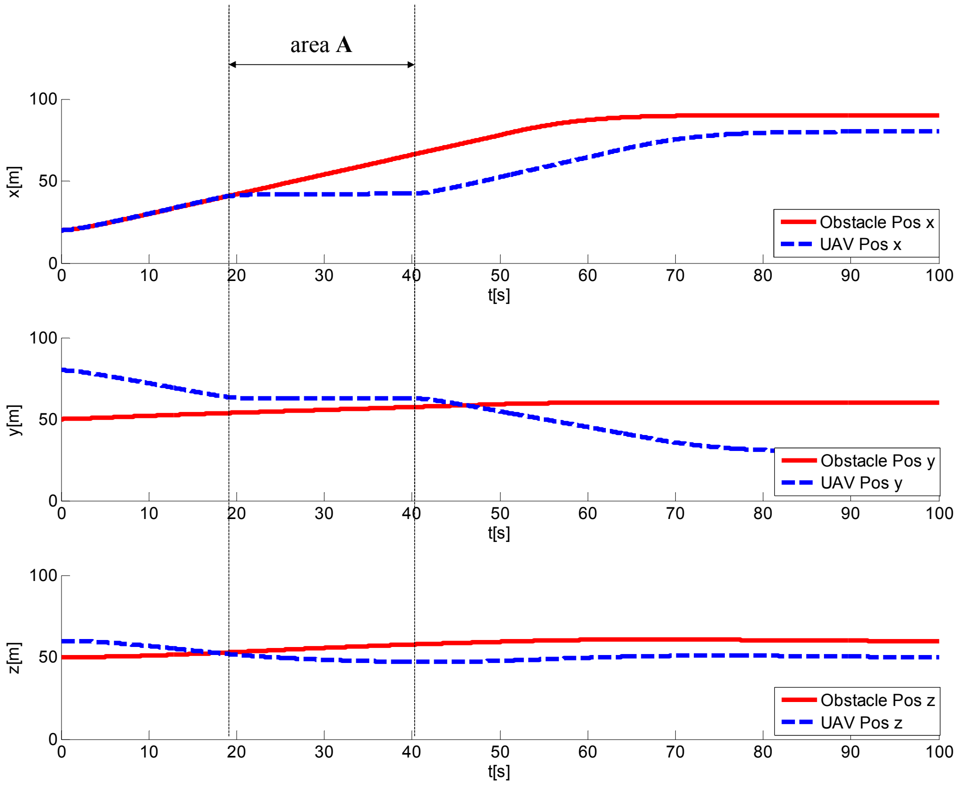


Figure 4.21 The each position of *x*-*y*-*z* axes per time axis based while the UAV and the dynamic obstacle are flying toward their own destinations. The UAV waited from around 20 sec to 40 sec to avoid collisions with the obstacle.

CHAPTER 5. CONCLUSIONS

First of all, we have presented the path planning algorithm for a UAV by introducing the expansion of the original limit-cycle navigation method in three-dimensional spaces. However, regardless of the UAV's initial and final positions, the path was generated based on the center of the obstacle and always planned along the side of the obstacle, which could lead on for a UAV to follow the inefficient path. Therefore, we have established the new path planning algorithm based on the methods of ray tracing and the extended limit-cycle navigation method. First of all, to minimize the time consumption, the path is decided whether it should be generated over the obstacle or side of the obstacle by finding out the most suitable straight line that passes through both the initial position and the destination. Moreover, when a UAV is avoiding little off from the center of the obstacle, a gap existed between the obstacle and the UAV, and this space was removed by accurately calculating the radius of the rendered sphere. As the results of the simulation show, we have validated that the proposed path planning algorithm could generate the most efficient and suitable path for the UAV.

In addition, to cope with dynamic obstacles, we have presented the algorithm that generates a path to avoid collisions with dynamic obstacles. The algorithm was mainly comprised of a collision detection method and standard rules of airplane traffic. As a result, the path was successfully generated from the initial position to the destination.

Next, in order for the UAV to converge quickly on the desired position (way point) generated by the proposed path planning algorithm, we have presented the fuzzy logic control method for path tracking. The FLC was composed of nine fuzzy logic controller modules, which are responsible for position control, velocity control and angle control corresponding to different situations. Thus, to solve the problem that the UAV repeats stop-and go around the waypoints, virtual spheres were centered at the waypoints. We then let the UAV fly to the following sphere when it was passing the sphere that is centered at the current desired position. Consequently, the simulation results showed that the proposed the fuzzy logic control method for path tracking validated how to track the path and serial waypoints at a constant velocity, in a short time.

For future studies, in terms of path planning, instead of rendering only rectangular box shaped obstacles, obstacles with irregular shapes such as ovals, stars and polygons, should be considered. Furthermore, the effect on the interval of waypoints will be studied. In terms of path tracking, the novel FLC method, able to cope with disturbance factors such as wind or turbulence, will be designed. In addition, the superior convergence ability of the proposed fuzzy logic method used to track the path will be verified using real experiments.

REFERENCES

[1] DR Nelson, DB Barber, TW McLain, RW Beard, “Vector Field Path Following for small unmanned air vehicles”, *Proceedings of the American Control Conference*, 2006.

[2] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots”, *In Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 500–505, 1985.

[3] K. Sigurd, J. P. How, “UAV trajectory design using total field collision avoidance”, *In Proceedings of the AIAA Guidance, Navigation and Control Conference*, August 2003.

[4] D-H. Kim, J-H. Kim, “A real-time limit-cycle navigation method for fast mobile robots and its application to robot soccer,” *Robotics and Autonomous Systems*, vol.42, pp. 17-30, 2003.

[5] G. Ambrosino, M. Ariola, U. Ciniglio, F. Corraro, A. Pironti and M. Virgilio, “Algorithms for 3D UAV Path Generation and Tracking”, *In Proc. of the IEEE International Conference on Decision & Control*, pp.5275-5280, DEC. 2006.

[6] Derek R. Nelson, D. Blake Barber, Timothy W. McLain, and Randal W. Beard, “Vector Field Path Following for Miniature Air Vehicles”, *IEEE Trans. on Robotics*, vol. 23, pp. 519-529, 2007.

[7] Tao Dong, X. H. Liao, R. Zhang, Zhao Sun, and Y. D. Song, “Path Tracking and Obstacle Avoidance of UAVs – Fuzzy Logic Approach”, *In proc. of the IEEE International Conference on Fuzzy Systems*, pp. 43-48, 2005.

[8] Dongging Shi, Selekwa, M. F, Collins, E. G., Jr, and Moore, C.A, “Fuzzy Behavior Navigation for an Unmanned Helicopter in Unknown Environments”, *IEEE international conference on Systems, Man and Cybernetics*, vol.4, pp.3897-3902, 2005.

[9] AI-Younes. Y, and Jarrah. M. A, “Attitude Stabilization of Quadrotor UAV using Backstepping Fuzzy Logic & Backstepping Least-Mean-Square controllers”, *In Proc. of 5th ISMA international symposium on Mechatronics and Its Applications*, pp.1-11, 2008.

[10] Doitsidis, L, Valavanis, K. P, Tsourveloudis, N. C, and Kontitsis. M, “A Framework for Fuzzy Logic based UAV Navigation and Control”, *In Proc. of IEEE international conference on Robotics and Automation*, vol.4, pp.4041-4046, 2004.

[11] Kimon P.Valavanis, “Advances in Unmanned Aerial Vehicles”, 1st Ed., *Springer Ltd.*, 2007.

[12] J Tisdale, Z Kim, and JK Hedrick, "Autonomous UAV Path Planning and Estimation," *IEEE Robotics & Automation Magazine*, pp.35-42, June 2009.

[13] Yoonsoo Kim, Da-Wei Gu, and Ian Postlethwaite, "Real-time path planning with limited information for autonomous unmanned air vehicles," *Automatica (Journal of IFAC)*, vol.44, issue 3, pp. 696-712, March 2008.

[14] Prazenica, R., Kurdila, A.,Sharpley, R., and Evers, J., “Vision-based geometry estimation and receding horizon path planning for UAVs operating in urban environments,” *American Control Conference*, June 2006.

[15] Sinopoli, B.,Micheli, M., Donato, G., and Koo, T.J., “Vision based navigation for an unmanned aerial vehicle,” *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2, pp. 1757-1764, 2001.

[16] Jerzy Z. Sasiadek and Iganacy Duleba, “3D Local Trajectory Planner for UAV,” *Journal of Intelligent and Robotic Systems*, vol. 29, pp. 191-210, Oct. 2000.

[17] Myung Hwangbo, Kuffner, J., and Kanade, T., “Efficient Two-phase 3D Motion Planning for Small Fixed-wing UAVs,” *Robotics and Automation, 2007 IEEE International Conference on*, pp. 1035-1041, April 2007

[18] Ambrosino, G., Ariola, M., Ciniglio, U., Corraro, F., Pironti, A., and Virgilio, M., “Algorithms for 3D UAV Path Generation and Tracking,” *Decision and Control, 2006 45th IEEE Conference on*, pp. 5275-5280, Dec. 2006.

[19] IK Nikolos, N Tsourveloudis, and KP Valavanis, “Evolutionary Algorithm Based Off-Line Path Planner for UAV Navigation,” *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, vol. 33(6), pp. 898-912, Dec. 2003.

[20] D Shi, MF Selekwa, EG Collins Jr, and CA Moore, “Fuzzy Behavior Navigation for an Unmanned Helicopter in Unknown Environments,” *Systems, Man and Cybernetics, 2005 IEEE International Conference on*, vol. 4, pp. 3897-3902, Oct. 2005.

[21] Eric Lengyel, *Mathematics for 3D Game Programming & Computer Graphics*, CHARLES RIVER MEDIA, 2004.

[22] ICAO Annex 2, Rules of the Air, "Avoidance of Collisions", November 2005.

[23] D-S Jang, S-J Cho, M-J Tahk, H-J Koo and J-S Kim, “Fuzzy based Collision Avoidance against Multiple Threats for Unmanned Aerial Vehicles,” *Proceedings of the Aircraft Symposium*, Vol. 43, 2005.

[24] Bouabdallah, S., Murrieri, P. and Siegwart, R., "Design and control of an indoor micro quadrotor," *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 5, pp. 4393–4398, 2004.

[25] Kimon P.Valavanis, Advances in Unmanned Aerial Vehicles, 1st Ed., *Springer Ltd*., 2007

[26] H. Bouadi, M. Bouchoucha and M. Tadjine, “Sliding Mode Control based on Backstepping Approach for an UAV Type-Quadrotor,*” International Journal of Applied Mathematics and Computer Sciences*, vol. 4, no. 1, pp. 12-17, 2008.

[27] Pedro Castillo, Rogello Lozano and Alejandro E Dzul, Modelling and Control of Mini-Flying Machines, 1st Ed., *Springer London Ltd*., 2005.

[28] L. Reznik, “Fuzzy Controllers”, *Oxford U. K.: Newnes*, 1997.

[29] Soon Kiat Yee, and Milanovic, J. V, “Fuzzy Logic Controller for Decentralized Stabilization of Multimachine Power Systems”, *IEEE Trans. On Fuzzy Systems*, vol.16, pp.971-981, 2008.

[30] Moon-Su Lee, Myung-Jin Jung, and Jong-Hwan Kim, “Evolutionary Programming-based Fuzzy Logic Path Planner and Follower for Mobilie Robots”, *In Proc. of the 2000 congress on Evolutionary Computation*, vol.1, pp.139-144. 2000.