# A REACTIONARY OBSTACLE AVOIDANCE ALGORITHM FOR AUTONOMOUS VEHICLES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

GİZEM YÜCEL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
AEROSPACE ENGINEERING

JUNE 2012

Approval of the thesis:

A REACTIONARY OBSTACLE AVOIDANCE ALGORITHM FOR AUTONOMOUS VEHICLES

submitted by **GIZEM YÜCEL** in partial fulfillment of the requirements for the degree of **Master of Science in Aerospace Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _______________
Dean, **Graduate School of Natural and Applied Sciences**

Prof. Dr. Ozan Tekinalp _______________
Head of Department, **Aerospace Engineering**

Assist. Prof. Dr. İlkay Yavrucuk _______________
Supervisor, **Department of Aerospace Engineering, METU**

**Examining Committee Members:**

Prof. Dr. Ozan Tekinalp _______________
Aerospace Engineering Department, METU

Assist. Prof. Dr. İlkay Yavrucuk _______________
Aerospace Engineering Department, METU

Prof. Dr. Kemal Leblebicioğlu _______________
Electrical and Electronics Engineering Department, METU

Assist. Prof. Dr. Ali Türker Kutay _______________
Aerospace Engineering Department, METU

Assoc. Prof. Dr. Tayfun Çimen _______________
Engineering Development Directorate,
ROKETSAN Missile Industries Inc.

**Date:** _______________

**I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.**

Name, Surname: Gizem Yücel

Signature:

# ABSTRACT

A REACTIONARY OBSTACLE AVOIDANCE ALGORITHM FOR AUTONOMOUS VEHICLES

Yücel, Gizem

M.Sc., Department of Aerospace Engineering

Supervisor: Assist. Prof Dr. İlkay Yavrucuk

June 2012, 58 Pages

This thesis focuses on the development of guidance algorithms in order to avoid a prescribed obstacle primarily using the Collision Cone Method (CCM). The Collision Cone Method is a geometric approach to obstacle avoidance, which forms an avoidance zone around the obstacles for the vehicle to pass the obstacle around this zone. The method is reactive as it helps to avoid the pop-up obstacles as well as the known obstacles and local as it passes the obstacles and continue to the prescribed trajectory. The algorithm is first developed for a 2D (planar) avoidance in 3D environment and then extended for 3D scenarios. The algorithm is formed for the optimized CCM as well. The avoidance zone radius and velocity are optimized using constraint optimization, Lagrange multipliers with Karush-Kuhn-Tucker conditions and direct experimentation.

Keywords: autonomy, obstacle avoidance, collision cone method, constrained optimization, Karush-Kuhn-Tucker conditions

# ÖZ

OTONOM ARAÇLAR İÇİN BİR REAKSİYONEL ENGELDEN KAÇIŞ ALGORİTMASI

Yücel, Gizem
Yüksek Lisans, Havacılık ve Uzay Mühendisliği Bölümü
Tez Yöneticisi: Yrd. Doç. Dr. İlkay Yavrucuk

Haziran 2012, 58 Sayfa

Bu tez öncelikli olarak çarpışma konisi metodunu kullanarak yeri belli bir cisime çarpışmayı önleyecek bir güdüm algoritmasının geliştirilmesini konu almaktadır. Çarpışma konisi metodu geometriyi kullanan, yani engellerin çevresinde, aracın, engelin etrafından dolanıp geçmesini sağlayan bir önleme küresi oluşturan bir engelden açış algoritmasıdır. Metot, önceden tanımlanmış engellerin yanı sıra uçuş sırasında ortaya çıkan engeller için de kullanılabilir reaksiyonel ve sadece önüne çıkan engelleri aşıp, önceden tanımlanan yoluna devam eden yerel bir yöntemdir. Geliştirilen algoritma öncelikle 2 boyutlu senaryolar oluşturulmuş, daha sonra 3 boyutlu senaryolar için de geliştirilmiştir. Önleme küresi yarıçapı ve hız değerlerini optimize eden bir algoritma da oluşturulmuştur. Optimizasyon, kısıtlı optimizasyon yöntemleri olan Karush-Kuhn-Tucker durumları ile Lagrange çarpanları metodu ve deneysel metot kullanılarak yapılmıştır.

Anahtar Kelimeler: otonomi, engelden kaçış, çarpışma konisi metodu, kısıtlı optimizasyon, Karush-Kuhn-Tucker durumları

## ACKNOWLEDGEMENTS

I would like to express my appreciation to my supervisor Dr. Ilkay Yavrucuk for his guidance and suggestions throughout the research and thesis study.

I would like to thank my leader in ROKETSAN, Emel Mahmutyazıcıoğlu for her understanding and encouragement.

I want to express my gratitude to Yağmur Atay for her help on the formatting of this thesis and as well for her support and friendship. Moreover, Derya Ünsal, Büşra Kul and Güliz Bulut deserve thanks with their motivation, encouragement and as well, by sharing the thesis period experience. I want to thank my dearest friends, Başak Yorulmaz, Seyfullah Aktaşoğlu, Aybüge Çekinmez, Oğuz Yüksel and Mert Gümüş for sharing valuable times, their support and motivation.

Last, but definitely not the least, I would like to express my deepest thanks to my parents, Pervin-Mehmet Yücel and my sister Simge for their endless love, support, trust, motivation, understanding and patience throughout my life, especially during this thesis study. Without their love and encouragement, it would not be possible for me to finish this thesis.

to my family…

# TABLE OF CONTENTS

# LIST OF FIGURES

FIGURES

## LIST OF TABLES

TABLES

# LIST OF ABBREVIATIONS

| | | |
|---|---|---|
| **CCM** | **:** | Collision Cone Method |
| **CDR** | **:** | Conflict Detection and Resolution |
| **CPA** | **:** | Closest Point Approach |
| **dd** | **:** | Detection Distance |
| **DOF** | **:** | Degree Of Freedom |
| **GPS** | : | Global Positioning System |
| **MEG** | **:** | Minimum Effort Guidance |
| **MPC** | **:** | Model Predictive Control |
| **NED** | **:** | North-East-Down |
| **PNG** | **:** | Proportional Navigation Guidance |
| **RPV** | **:** | Remotely Piloted Vehicles |
| **SQP** | **:** | Sequential Quadratic Programming |
| **UAV** | **:** | Unmanned Aerial Vehicle |
| **VFH** | **:** | Vector Field Histogram |

# LIST OF SYMBOLS

$\vec{a}$ : Acceleration Vector

$a_{lat}$ : Lateral acceleration of the vehicle in body frame

$a_x$ : Horizontal acceleration of the vehicle in NED frame

$a_y$ : Lateral acceleration of the vehicle in NED frame

$a_z$ : Vertical acceleration of the vehicle in NED frame

**ds** : Arc Length Element

**dθ** : rate of change of the turn angle

$\ell$ : length of the intersection line

$\vec{R}$ : Position Vector

$r_{obstacle}$ : position of the obstacle

$R_{sphere}$ : radius of the obstacle avoidance zone

$R_2$ : radius of the auxiliary sphere to avoid the obstacle in a smoother manner

**s** : Length of a Curve

$\hat{T}$ : Unit Tangent Vector

$u$ : Horizontal velocity of the vehicle in NED frame

$v$ : Lateral velocity of the vehicle in NED frame

$\vec{V}$ : Velocity Vector

**V** : Total Velocity of the vehicle

$\overrightarrow{V\_b}$ : Velocity Vector in body frame of the vehicle

$w$ : Vertical velocity of the vehicle in NED frame

$x$ : Horizontal position of the vehicle in NED frame

$y$ : Lateral position of the vehicle in NED frame

$z$ : Vertical position of the vehicle in NED frame

**α** :velocity vector direction change angle

**γ** : Elevation Angle

**κ** : Curvature of a Curve

**λ** : Lagrange Multipliers

**μ** : Horizontal Angle Change on a Sphere

**ρ** : Radius of Curvature

**φ** : Vertical Angle Change on a Sphere

**χ** : Heading Angle

$\omega_{NED}$ : angular velocity of the vehicle on a sphere

# CHAPTER 1

# INTRODUCTION

Unmanned aerial vehicles (UAV) are becoming more important with the increasing need for reconnaissance, surveillance and delivery tasks in military and civilian areas [1]. One key area of research is autonomy, in particular making decisions without human interference. The goal is to make machines smart and act more human like.

There are various levels of autonomy[1]. Autonomy comes in a wide range from RPVs (Remotely Piloted Vehicles) to fully autonomous UAVs. The most basic level of autonomy is the radio-controlled systems. Radio-controlled systems are both commanded and controlled by the operator for the reason that they have no feedback mechanisms. Therefore the first improvement is to introduce feedback. As a result of the need for the vehicle to get the information about the environment, sensors (camera, seeker, inertial measurement unit, etc.) are integrated on the vehicles. Finally, the system becomes a set of software algorithms with advanced hardware.

Autonomous systems can be achieved with help of sensors, autopilot, navigation and guidance algorithms on them. They both sense the environment and make decisions on the control of the vehicle. In other words, autonomy brings in vehicle tactical intelligence, allows them to have reactive changes to new situations and static and dynamic obstacles on their way without a need for a human operator.

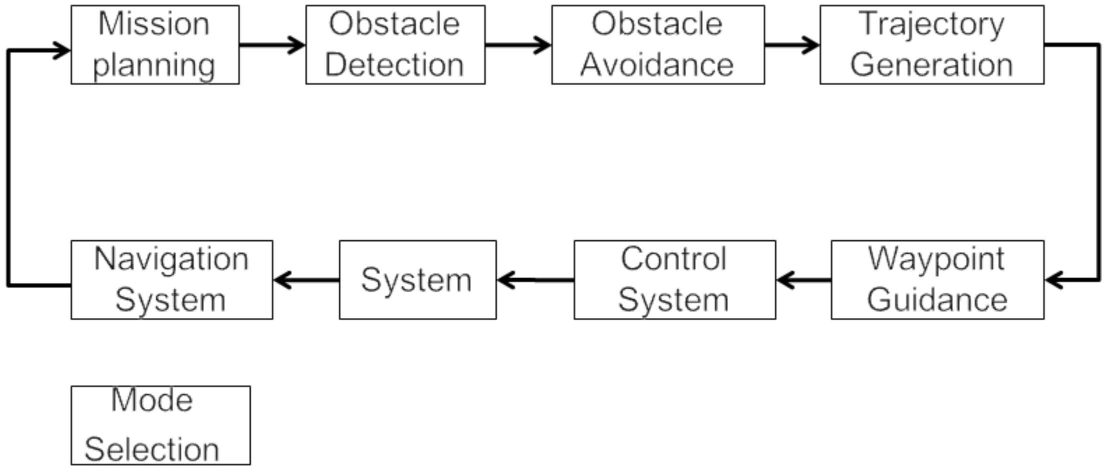


**Figure 1 –** Flowchart Example of an Autonomous System

Some related technology can be listed as; sensor fusion, communication, path planning, mission planning, mode selection, obstacle avoidance, limit detection and avoidance, trajectory generation, task allocation and scheduling, cooperative tactics.
An example of a flowchart in an autonomous system is shown in Figure 1. Mission planning of the vehicle is performed with the information from the sensors based on a preselected activity. For example, while the vehicle is on its mission, there could be an obstacle on its trajectory. The camera or seeker detects the obstacle and the guidance algorithm guides the vehicle to avoid these obstacles and if necessary, the trajectory is replanned. The avoidance and guidance algorithms are controlled as well, mostly by the help of an autopilot and these data are fed back into the system to make the vehicle continue its mission.
The focus of this work is to develop a guidance algorithm for obstacle avoidance.

## 1.1 Thesis Outline

The remaining part of the thesis is organized as follows:

- In Chapter 2, literature survey carried out about the obstacle avoidance methods are given.
- In Chapter 3, problem formulation in the simulation model and the trajectory generation is described.

- In Chapter 4, the CCM with constant velocity, the method selected for the algorithm in this thesis, is explained and the simulation results are given.
- In Chapter 5, 2D CCM is extended to 3D algorithm, the details of the algorithm are given and the results of the simulation are shown.
- In Chapter 6, optimization applied on the CCM is explained with comparison and simulation results.
- In Chapter 7, with conclusion and future work is stated.

# CHAPTER 2

# LITERATURE SURVEY

Navigation features, which help unmanned vehicles to achieve their autonomous missions, are global path planning and local obstacle/collision avoidance. Global path planning deals with the path the vehicle takes from the start of the mission to the goal whereas, local obstacle avoidance algorithms considers only the obstacles on the path. Hybrid algorithms, in which both of these methods are integrated, are another alternative used for the navigation and guidance of unmanned vehicles.

## 2.1 Global Path Planning Algorithms

There are various types of global path planning algorithms, such as graph search algorithms, rapidly exploring random tree, potential field method and minimum effort guidance. In [2], these methods are summarized and compared.

Graph search algorithms can be formed by many techniques such as A*[3], best-first greedy [4], visibility graph method [5]. In these methods, the environment the vehicle travels is divided into grids. The kinematics of the vehicle is taken into account to define the size of the grids. Avoiding the known obstacles, nodes for the vehicle trajectory are defined. The algorithm finds the optimal path solution using these nodes and grids by using the mathematical curves, for instance splines, Dubin paths, clothoids as shown in Figure 2. At every step of the vehicle motion, alternative nodes and paths for the vehicle trajectory are calculated and stored as a database for the algorithm. This makes the solution rather complicated with the need of huge data storage. At last, these algorithms are highly dependent on the resolution of the graphs and require heading change.

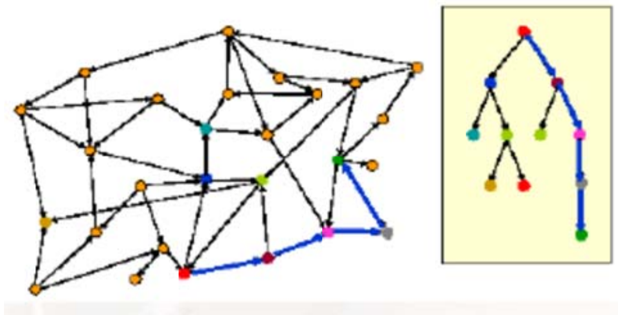

**Figure 2 -** Representation of a Graph Search Algorithm [6]

Rapidly exploring random tree [7], [8] is a path planning method that creates random nodes in order to reach the aim point. Taking into account the random nodes and another parameter, which is mainly considered as the distance between the nodes, other new nodes to go ahead the random nodes are generated while checking for the obstacles and feasibility of the path to the goal, which is seen in Figure 3. The efficiency of this algorithm is the usage of quad trees instead of uniform grids. This reduces the need of memory and provides efficient environment information. The path found by RRT method may not be optimal and in general a graph search algorithm is used to optimize the distance of the path. In complex environments, this method has the risk of not finding the trajectory to the target point.

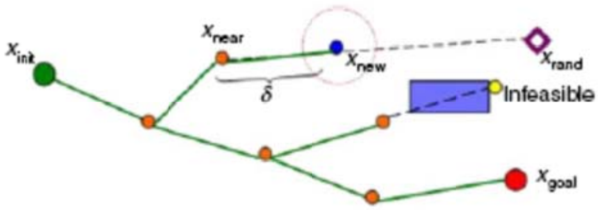


**Figure 3 -** RRT Algorithm Representation [2]

One of the mostly used path planning and obstacle avoidance method is the potential field method [9]. Forces acting on the waypoints, aim point and the obstacles direct the vehicle. Waypoints and aim point have an attraction field around, whereas obstacles have the repelling forces on them as shown in Figure 4. The resultant forces form the trajectory by directing the vehicle, therefore the forces acting highly depends on the angle to the aim point/obstacle/waypoint. The distance from the target point/obstacle/waypoint is important in terms of the effect of the force field (e.g. the vehicle should be close enough to the aim point/obstacle/waypoint). This method has some disadvantages as well; passing through narrow ways is challenging for this algorithm and the vehicle may stay immovable between obstacles/waypoints and cannot complete the mission.

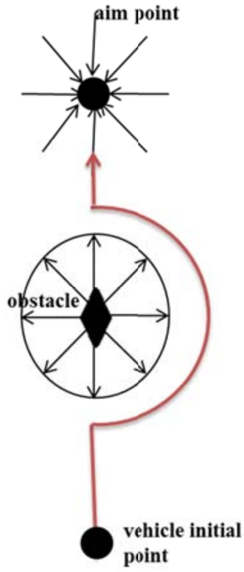


**Figure 4 –** Representation of Potential Field Method

Minimum effort guidance strategy is a path planning method which is the modification of the widely used missile guidance method, Proportional Navigation Guidance (PNG) [10]. PNG is used to guide the missile to the target by keeping the line of sight rate of the missile constant. This method can be used in path planning as well. However, at every time step, the algorithm should run in order to go ahead the next point to follow on its path while avoiding the obstacles. This means a huge computational load. As an alternative, another method, minimum effort guidance (MEG) is introduced [11]. MEG directs the vehicles on its trajectory with only one algorithm. Besides, in this algorithm, a collision cone approach is used. Collision cone approach [12],[13] uses the Euclidean geometry by forming a sphere around the obstacle and the tangent lines from the vehicle to the sphere create the obstacle/collision avoidance zone.

## 2.2 Local Obstacle Avoidance Algorithms

Local obstacle avoidance algorithms are classified as model predictive control approach, potential field method, vision-based neural network approach, conflict detection and resolution. In [2], these methods are summarized and compared.

Model predictive control (MPC) [14] is another control method for nonlinear dynamic systems. This approach optimizes the path using a receding horizon method in order to take into account the changes in the dynamic environment. Although it has many advantages like allowing evasive maneuvers, following the trajectory with a good performance and stability, it has the disadvantage of its computational load and need for a large data storage.

Vision-based neural network approach [15] is another method for obstacle avoidance. It has a planning manner with two branches; one of the branches is for path planning and the second is for trajectory tracking and obstacle avoidance. This method is generally used with the visibility graphs. The obstacles can be detected on the visibility graphs and the neurons are put on the vertices of the obstacle in order to create an avoidance zone. The neurons are joined with straight lines which create the shortest path to reach the goal. Generation of the trajectory by this method is, dependent on the activity of the neurons on the edges of the obstacle and the aim point as well. Tracking of this trajectory by the second branch is achieved using the model predictive control, in other words, for every step of the trajectory, an optimization is carried out. Vision-based neural network approach should get the environment information from trustworthy vision based methods and for the reason that it uses the MPC algorithm, it has a computational load.

Conflict detection and resolution (CDR) [16] is a method, which is commonly used for air traffic control, especially to avoid the collision of multiple air vehicles. This is a geometric approach and forms a sphere around the obstacle or moving vehicle. This method has many different applications to the obstacle/collision avoidance criterion. In [17], collision criteria are studied using Euclidean geometry; then MEG is applied as a guidance algorithm. In [18], the time constant values of the air vehicle for the control of the heading and pitch angle commands are used and a method, which is called "Vector Sharing Resolution", is used to study the avoidance maneuver on the miss distance line of the conflicting vehicles. In [19], the line of sight vector between the conflicting air vehicles are used in the mixed geometric collision cone algorithm with the heading angle, elevation angle and speed change separately and combined. In [20], the collision cone algorithm is used with the minimum distance separation between the vehicles and control strategies are involved as well. In [12], collision cone algorithm is implemented using the heading/elevation change of the vehicles.

## 2.3 Classical Obstacle Avoidance Algorithms

Some common classical obstacle avoidance methods are as well briefly explained here: Certainty grid method (vector field histogram (VFH), VFH+, VFH*), elastic band method and dynamic window approach.

Certainty grid methods [21] are classical and older compared to the other methods studied in this literature survey. These methods use a 2-D environment model with grids; each has a certainty value of having obstacles and strongly dependent on the sensor data and cell size. Vector Field Histogram [22] (VFH) method is a version of certainty grids. However, as opposed to 2-D Cartesian, polar histogram, which takes the heading angle and the probability as reference. All the directions are checked to find obstacles and eliminating the areas with obstacles or making the obstacles bigger in order to form a trajectory while avoiding the obstacles. After this, the heading of the vehicle is determined. $VFH^+$ and VFH* are the improvements of VFH. $VFH^+$ [23]considers the moving trajectories and the type of the trajectory can be arcs rather than straight lines as well. VFH*[24] concerns the steps ahead and forming branches on it. The trajectory is defined by optimization of the paths on the branches.

The Elastic Band Concept [25] is another obstacle avoidance method. In this method, the path is planned and bubbles (which mean enough distance to obstacles) are used on the trajectory to avoid obstacles. The bubbles are used all along the trajectory-start to end.

The Dynamic Window Approach [26] is the first method to take into account the vehicle kinematics as addition to the direction of motion. This method forms a velocity space with translational (V) and rotational (ω) vehicle velocities. The velocity space is then reduced to dynamic window according to the reachable velocities in the next time step. The dynamic window is a rectangle which has the velocity at that moment as the center and the vertices are the reachable velocities based on the acceleration limits of the vehicle. The motion direction is decided by using a cost function, which has heading, velocity and the distance to the closest obstacle as constraints.

## 2.4 Summary and Objective

Path planning and obstacle/collision avoidance methods are mainly classified as heuristic and optimal methods. In this literature survey, mostly heuristic methods are considered. Heuristic methods are the experience-based methods, which in general have light and fast algorithms and they are newly evolving methods. Among the heuristic methods, conflict detection and resolution method is selected as the main subject of this thesis, for the reason that it uses geometric methods, which are easy to implement, improve and apply the real time systems. This method is a reactionary method, which computes the motion in the next time step using the current motion data. It needs the priorities and the limits of the vehicle it is implemented.

# CHAPTER 3

# PROBLEM FORMULATION

## 3.1 Problem Formulation

A 3 Degree of Freedom (DOF) point mass model is used in the algorithm. $\vec{R}$, $\vec{V}$, $\vec{a}$ are the vehicle's position, velocity and acceleration vectors, respectively and $\chi$ is the heading angle; $\gamma$ is elevation angle of the vehicle. The kinematics equations are shown in Equation 3.1-3.5. In the algorithm, NED (North-East-Down) Coordinate System is used as a reference frame as shown in Figure 5.

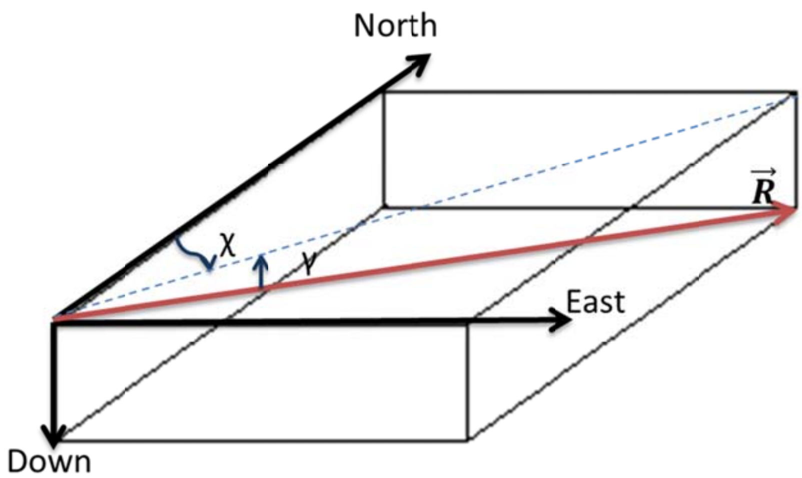


**Figure 5** - Reference Coordinate System

$$\vec{R} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{3.1}$$

$$\dot{\vec{R}} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \vec{V} \tag{3.2}$$

$$\dot{\vec{V}} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \vec{a} \tag{3.3}$$

$$\gamma = \sin^{-1} \frac{w}{|\vec{v}|} \tag{3.4}$$

$$\chi = \tan^{-1} \frac{v}{u} \tag{3.5}$$

The vehicle has a constant speed. However, lateral and vertical accelerations are commanded.

Lateral acceleration written in the vehicle body axis while the vehicle is carrying out curvature maneuver is found from Equation 3.6

$$a_{lat} = \frac{|\vec{V}|^2}{R\_sphere} \tag{3.6}$$

The algorithm is developed for a 3D environment. Initially, the motion is executed in 2D, in other words, either in horizontal or vertical planes; therefore the vehicle carries out 2D obstacle avoidance, in addition the avoidance algorithm is extended to 3D. The obstacles are stationary and their positions are known a priori. The sensors are assumed to be perfect. The algorithm is developed in the MATLAB/Simulink environment.

## 3.2 Trajectory Generation

The trajectory of the vehicle is generated by the initial velocity of the vehicle with the heading and elevation angles. Velocity in the body frame of the vehicle is transformed into the North East Down (NED) frame. Then, the velocity is integrated with respect to time in order to get the position. The trajectory of the vehicle is calculated as shown in Equations 3.7-3.9.

$$\hat{C}\ (B, NED) = \begin{bmatrix} \cos\gamma cos\chi & -sin\chi & sin\gamma cos\chi \\ \cos\gamma sin\chi & cos\chi & sin\chi sin\gamma \\ -sin\gamma & 0 & cos\gamma \end{bmatrix} \tag{3.7}$$

$$\vec{V} = \hat{C}\ (B, NED) * \overrightarrow{V\_b} \tag{3.8}$$

$$\vec{R} = \int \vec{V}\ dt \tag{3.9}$$

# CHAPTER 4

# COLLISION CONE METHOD WITH CONSTANT VELOCITY

The Collision Cone Method (CCM) is a heuristic, reactive, local collision/obstacle avoidance method. It is widely known as a conflict detection and resolution method or method of closest point approach (CPA).

The CCM is performed in two stages with a "Sense-and-Avoid" concept. Conflict detection is the first stage, which consists of the detection of the obstacle and conflict resolution is the second stage, in which the obstacle avoidance is carried out. Conflict detection is achieved by means of sensors such as electro-optic/infrared/radar/laser cameras or seekers, etc. The vehicle will have a minimum allowable distance criterion in order to avoid the obstacle/collision. In the CCM, a sphere with the radius of minimum avoidance distance is formed. In the conflict resolution stage, collision avoidance is carried out around a collision sphere. The problem is shown in Figure 6.

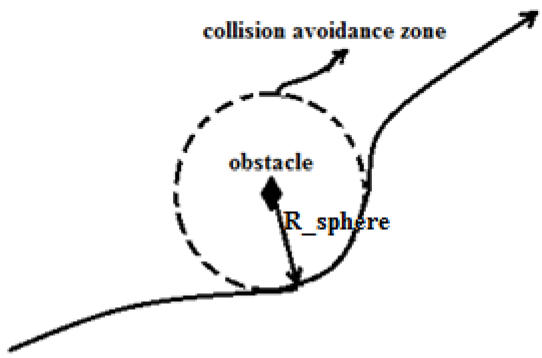


**Figure 6 -** Collision Cone Method

In this thesis, the conflict detection part is simply modeled with a detection distance criterion. As the obstacle is on the path, when the distance between the obstacle and the vehicle is equal to the detection distance (dd) of the vehicle sensor, the detection algorithm works and the collision cone algorithm is initiated.

As it is expressed in Part 3.2, the trajectory is introduced in the simulation with the velocity vector; in other words, the key element in the algorithm is the velocity vector. Avoidance is carried out by the change of the direction of the velocity vector.

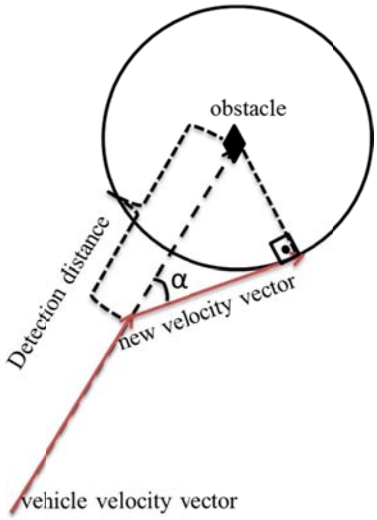


**Figure 7 -** Obstacle Avoidance Algorithm

In Figure 7, the avoidance algorithm is shown. When the obstacle is detected, the velocity vector changes its direction to the closest point to its trajectory on the sphere. The closest point is the end of the tangent line from the velocity vector to the sphere. The change of the direction angle is found from Equation 4.1.

$$\alpha = \sin^{-1} \frac{R_{sphere}}{Detection\ Distance} \tag{4.1}$$

The method, which changes the direction of the vehicle with the angle α, causes an instantaneous and sharp turn. This may result in a failure in the vehicle due to the high loads while turning. The vehicle had better turn in its load limits, with proper acceleration, as a result in a smoother manner. Consequently, curvature maneuver is set to the avoidance algorithm as in Figure 8.

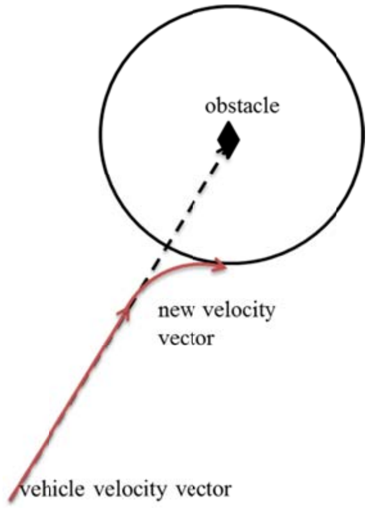


**Figure 8 –** Obstacle Avoidance Maneuver

### 4.1 Curvature and Radius of Curvature

In this maneuver shown in Figure 9, the velocity vector proceeds as a tangent line along the curve, which is a part of another sphere.

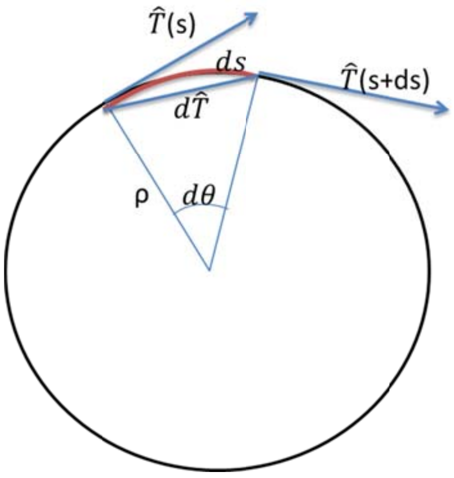


**Figure 9** - Curvature and Radius of Curvature Representation

In Equation 4.2, the general definition of curvature is given. Curvature (κ) measures the rate of turning of the tangent line to the curve as well [27]. Using the geometric relationships shown in Figure 9, Equation 4.5 is obtained, where $d\theta$ the rate of change of the turn angle is and $ds$ is the arc length element. The radius of curvature is the reciprocal of the curvature, which is given in Equation 4.6.

$$\kappa(s) = \left|\frac{d\hat{T}}{ds}\right| \tag{4.2}$$

$$\kappa(s) = \left|\frac{d\hat{T}}{ds}\right| = \left|\frac{d\hat{T}}{d\theta}\right| \left|\frac{d\theta}{ds}\right| \tag{4.3}$$

$$ds = v(t)\, dt = \left|\frac{d}{dt}\vec{r}(s)\right| dt \tag{4.4}$$

From Figure 9, as $\left|d\hat{T}\right| \sim |d\theta|$ for small angles; $\left|\frac{d\hat{T}}{d\theta}\right| = 1$, then

$$\kappa(s) = \left|\frac{d\theta}{ds}\right| \tag{4.5}$$

$$\rho(s) = \frac{1}{\kappa(s)} \tag{4.6}$$

From Equations 4.2-4.6, Equation 4.7 is obtained.

$$\rho = \frac{v}{\dot{\theta}} \tag{4.7}$$

## 4.2 Generating Curvature

The problem is to work with the change of the direction of the velocity vector; in other words to find the heading angle at each instant. Once the detection distance and the radius of the obstacle avoidance zone ($R_{sphere}$) are introduced to the algorithm, with the Pythagorean Theorem, the radius of curvature is found using Equation 4.8 as seen in Figure 10.

$$dd^2 + {R_2}^2 = (R_{sphere} + R_2)^2 \tag{4.8}$$

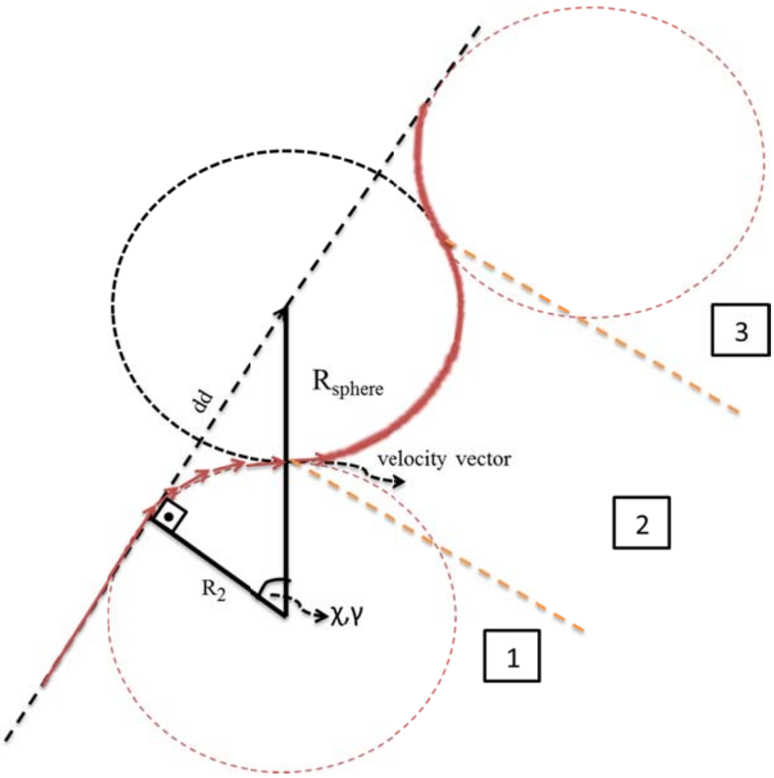


**Figure 10 -** Avoidance Maneuver

Using Equation 4.7, with the radius of curvature and the constant velocity values are known, the rate of change of the elevation or heading angle is considered is obtained. The rate of the turn angle is then integrated to get the turn angle value and the turn angle is fed into the rotation matrix.

As stated, the algorithm works in planar motion, either in horizontal or vertical plane. Therefore, heading and elevation angle rotations are not coupled and carried out in different scenarios.

From Figure 5, the rotation matrix in horizontal plane is;

$$\hat{C}(3, \chi) = \begin{bmatrix} \cos\chi & -\sin\chi & 0 \\ \sin\chi & \cos\chi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.9}$$

New velocity vector with the given heading angle is;

$$\text{V\_vector\_new} = \hat{C}(3, \chi) * \text{V\_vector} \tag{4.10}$$

Similarly, the rotation matrix in vertical plane is;

$$\hat{C}(2, \gamma) = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix} \tag{4.11}$$

New velocity vector with the given elevation angle is;

$$\text{V\_vector\_new} = \hat{C}(2, \gamma) * \text{V\_vector} \tag{4.12}$$

$$\hat{C}\ (B, NED) = \hat{C}(2, \gamma) * \hat{C}(3, \chi) \tag{4.13}$$

$$\hat{C}\ (B, NED) = \begin{bmatrix} \cos\gamma cos\chi & -sin\chi & sin\gamma cos\chi \\ \cos\gamma sin\chi & cos\chi & sin\chi sin\gamma \\ -sin\gamma & 0 & cos\gamma \end{bmatrix} \tag{4.14}$$

$\hat{C}\ (B, NED)$ is the matrix to transform the vehicle from body to inertial (NED) frame. Even though, the algorithm is executed in a planar motion, the conflict detection part runs as a coupled model, with the horizontal and vertical planes. The detection distance is evaluated as it is in only horizontal axis of the vehicle and it is integrated in the algorithm in inertial plane using Equation 4.15.

$$\text{detection distance} = \hat{C}\ (B, NED) * \text{dd} \tag{4.15}$$

After the detection of the obstacle, the algorithm is formulated in three parts. The division to the parts is shown in Figure 10.

At first, the obstacle is detected from a predetermined detection distance. If the distance between the obstacle and the vehicle is equal to or smaller than the detection distance, the obstacle is assumed to be detected. Besides, the vehicle is approaching the obstacle and the closing time (Equation 4.16) must be decreasing ($t_{closing} \leq 0$).

$$t_{closing} = -\frac{\vec{r}-\vec{r}_{obstacle}}{\vec{v}} \qquad (4.16)$$

The algorithm is made up of three circular paths. The intersection point between the first and the second circle lies on the line that connects the center points of the spheres. The distance between this point and the commanded trajectory is found from the relationship shown in Figure 10 and in Equations 4.17-4.19. The second circle is run until the distance between the vehicle and the commanded trajectory is smaller than the length of the intersection line. Subsequently, the third circle (identical to the first) is used as a path.

$$\delta = 90° - \chi \qquad (4.17)$$

$$\delta = \sin^{-1} \frac{R_2}{R_2+R_{sphere}} \qquad (4.18)$$

$$\ell = R_{sphere} * \sin(\delta) \qquad (4.19)$$

## 4.3 Simulation Results

Results of obstacle avoidance algorithm in horizontal and vertical plane using the CCM are shown in this part.

The obstacle avoidance in horizontal plane is shown in Figure 11-Figure 15 with the trajectory, heading angle change, lateral acceleration during the maneuver and velocity graphs. The parameters used for the simulation are given in Table 1.

**Table 1** – Parameters for the Simulation in Horizontal Plane

| | |
|---|---|
| Initial Point of The Vehicle [m] | [0 0 10] |
| Position of The Obstacle [m] | [150 150 10] |
| $\vec{V}$ [m/s] | [30 30 0] |
| $R_{sphere}$ [m] | 50 |
| Detection Distance [m] | 75 |

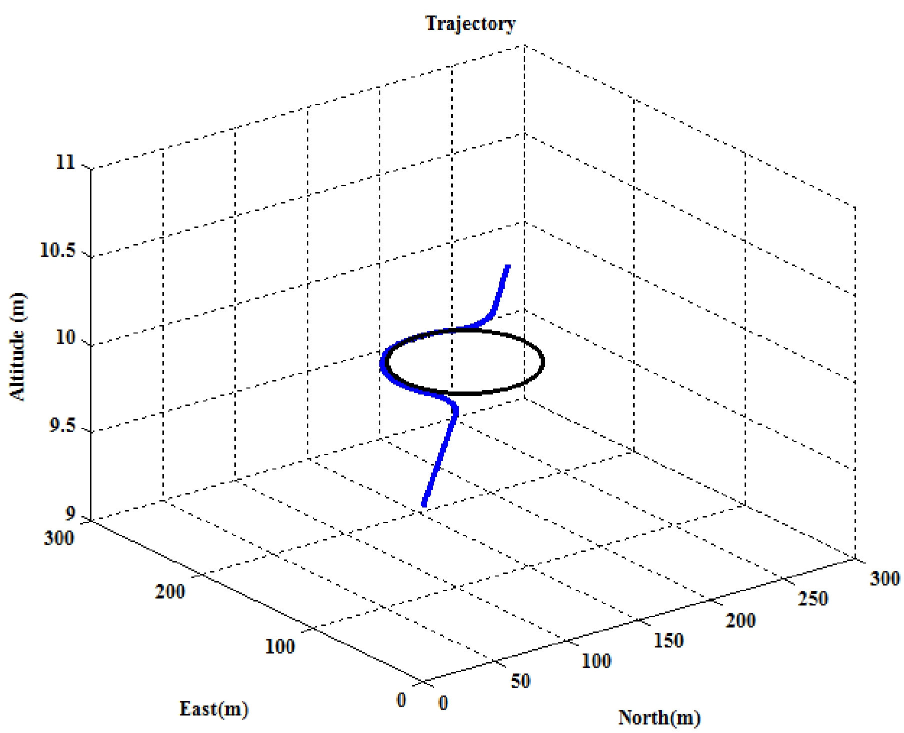


**Figure 11** – Trajectory of Obstacle Avoidance in Horizontal Axis

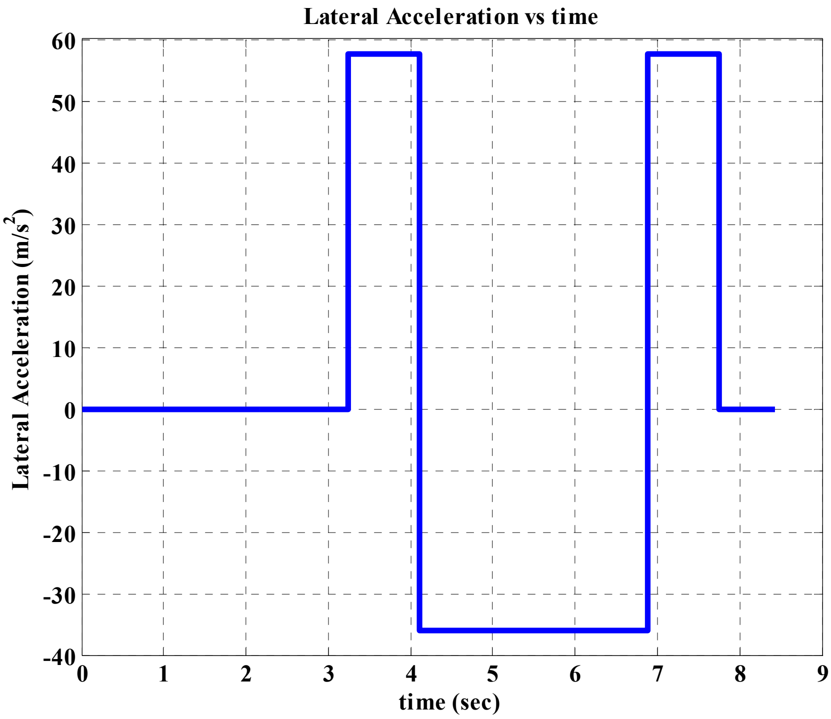


**Figure 12 –** Lateral Acceleration of the Vehicle in Obstacle Avoidance in Horizontal Axis

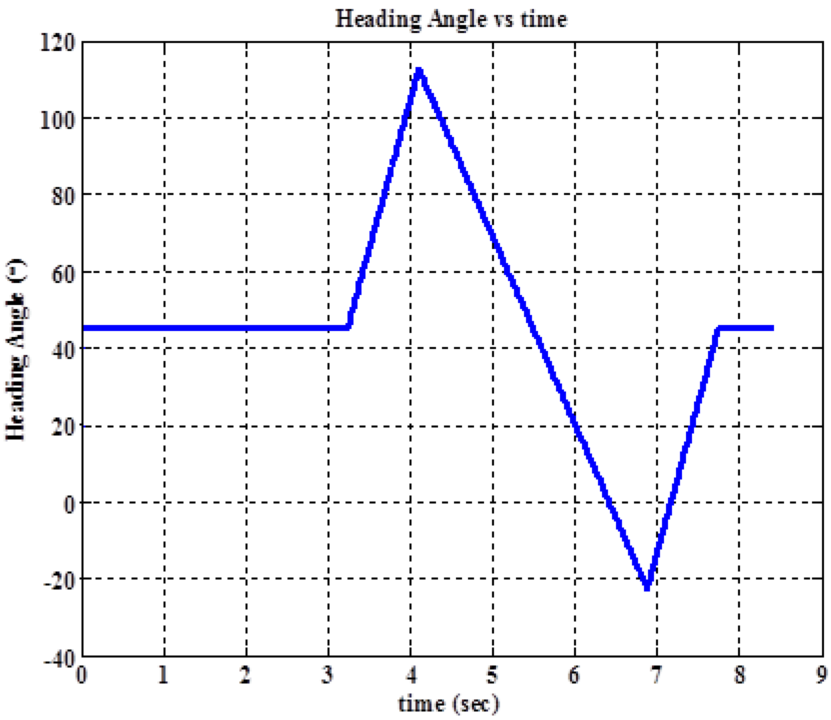


**Figure 13 –** Heading Angle of the Vehicle in Obstacle Avoidance in Horizontal Axis

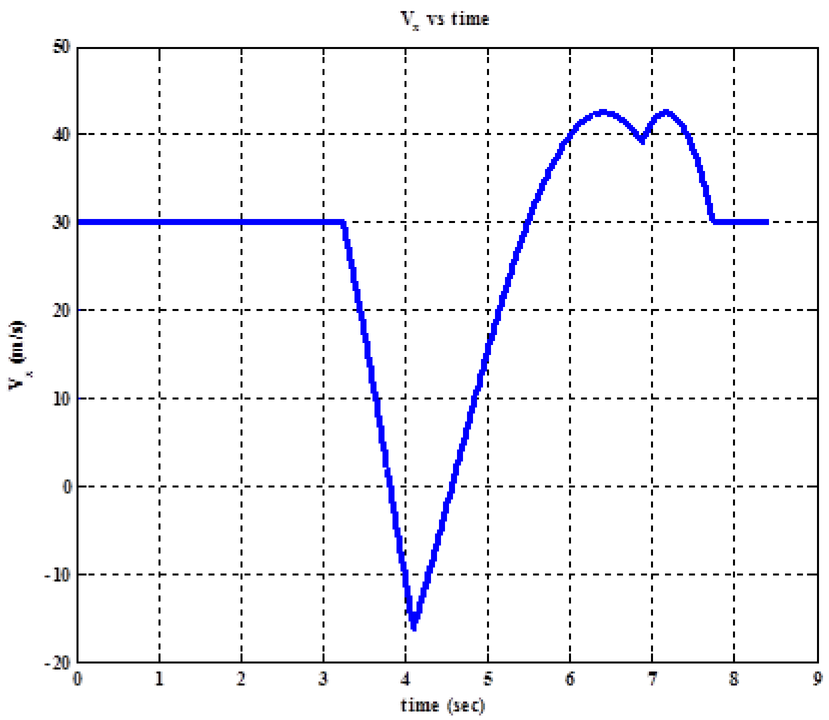


**Figure 14 –** Downrange Velocity of the Vehicle in Obstacle Avoidance in Horizontal Axis

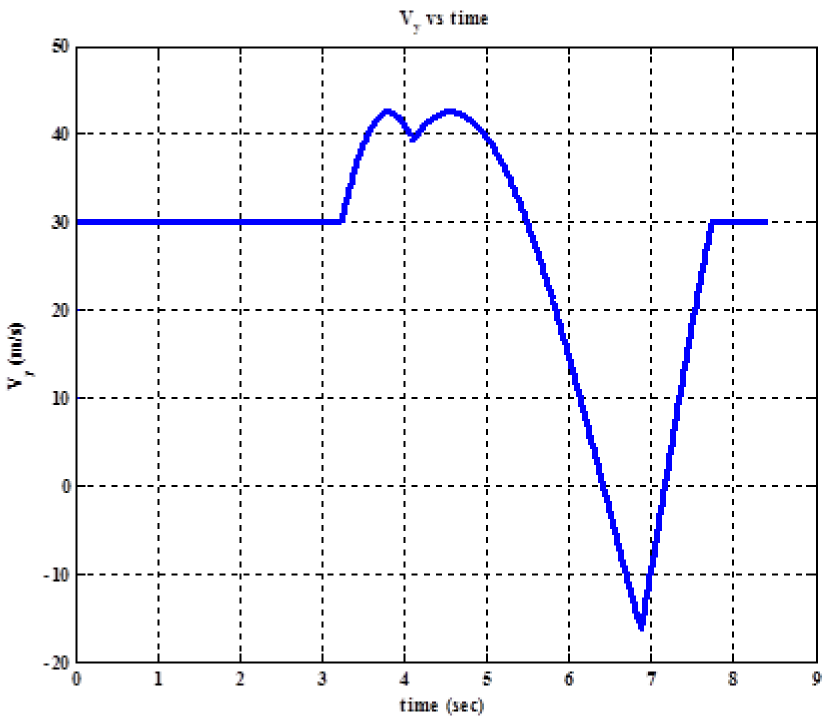


**Figure 15 –** Crossrange Velocity of the Vehicle in Obstacle Avoidance in Horizontal Axis

In Figure 12-Figure 15, the effects of the maneuvers in three regions, which are shown in Figure 10, can be seen.

The obstacle avoidance in vertical plane is shown in Figure 16 - Figure 20 with the trajectory, elevation angle change, lateral acceleration during the maneuver and velocity graphs. The parameters used for the simulation are given in Table 2.

**Table 2 -** Parameters for the Simulation in Vertical Plane

| | |
|---|---|
| Initial Point of The Vehicle [m] | [0 0 0] |
| Position of The Obstacle [m] | [150 0 150] |
| $\vec{V}$ [m/s] | [30 0 30] |
| $R_{sphere}$ [m] | 50 |
| Detection Distance [m] | 75 |

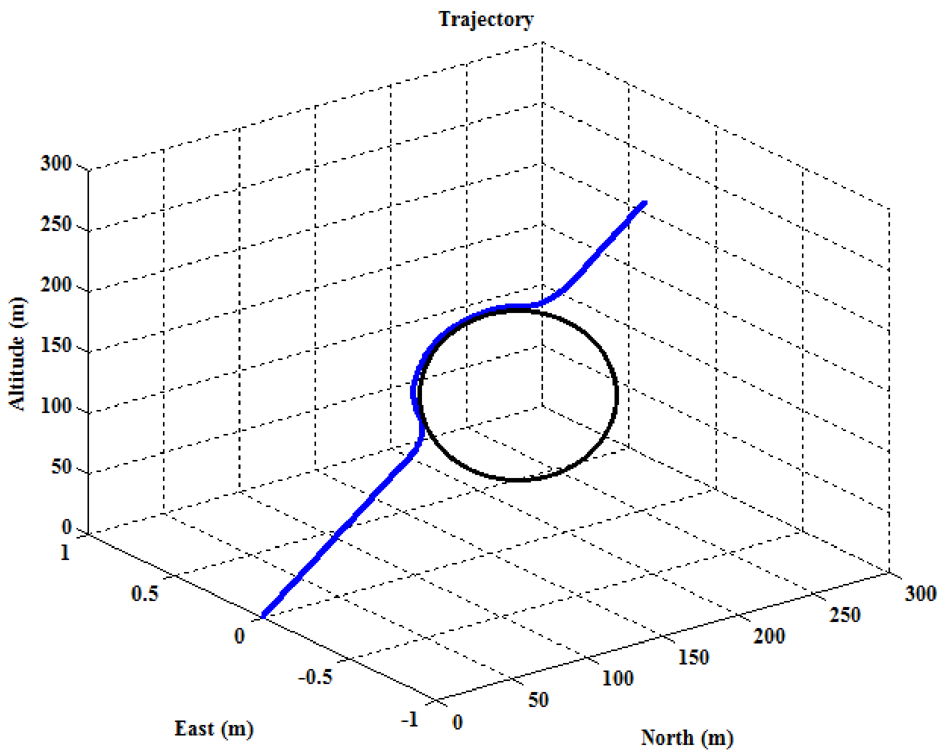


**Figure 16 -** Trajectory of the Vehicle in Obstacle Avoidance in Vertical Axis

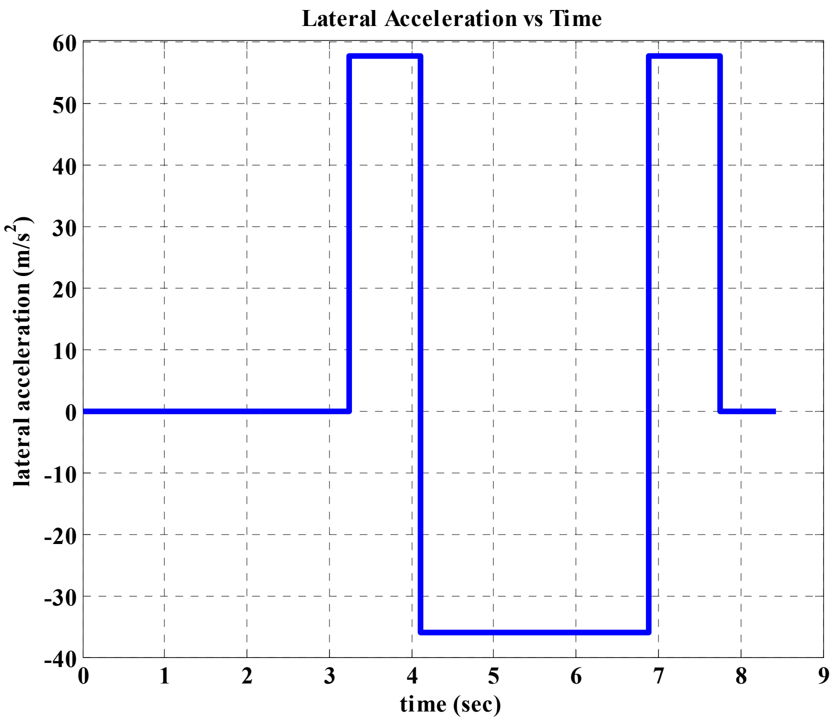


**Figure 17 –** Lateral Acceleration of the Vehicle in Obstacle Avoidance in Vertical Axis

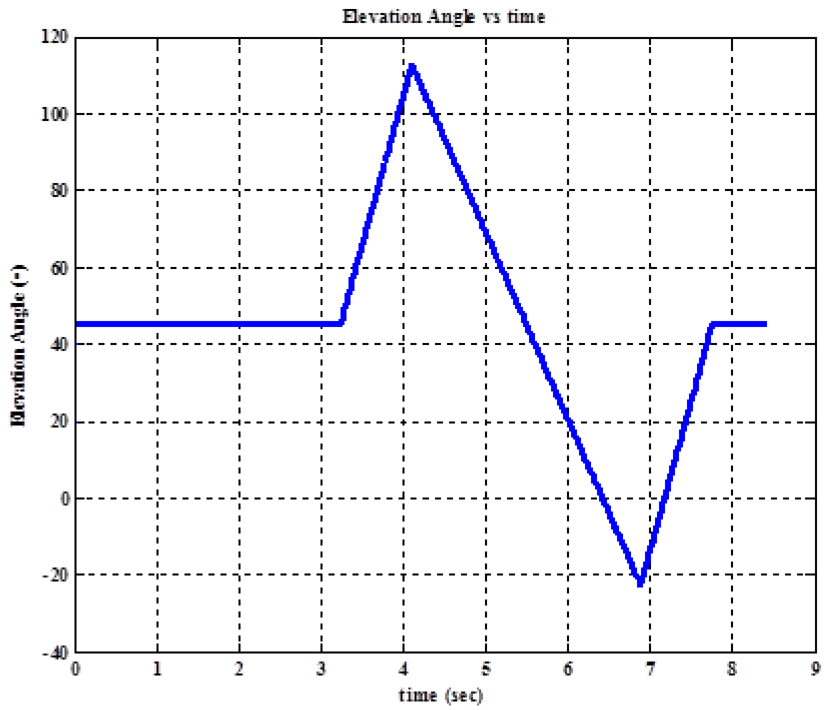


**Figure 18 –** Elevation Angle of the Vehicle in Obstacle Avoidance in Vertical Axis

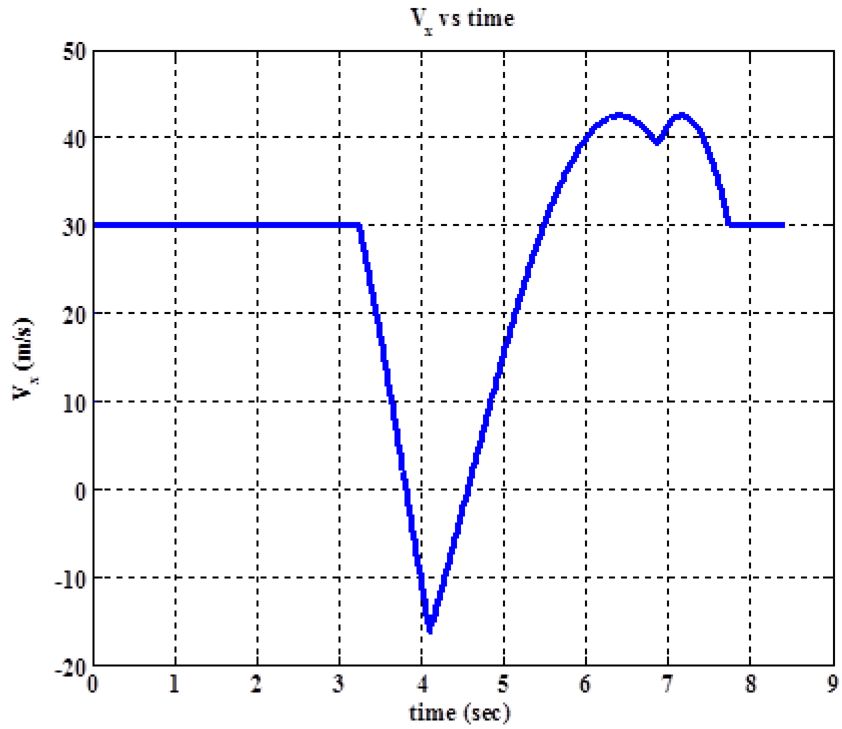


**Figure 19 –** Downrange Velocity of the Vehicle in Obstacle Avoidance in Vertical Axis

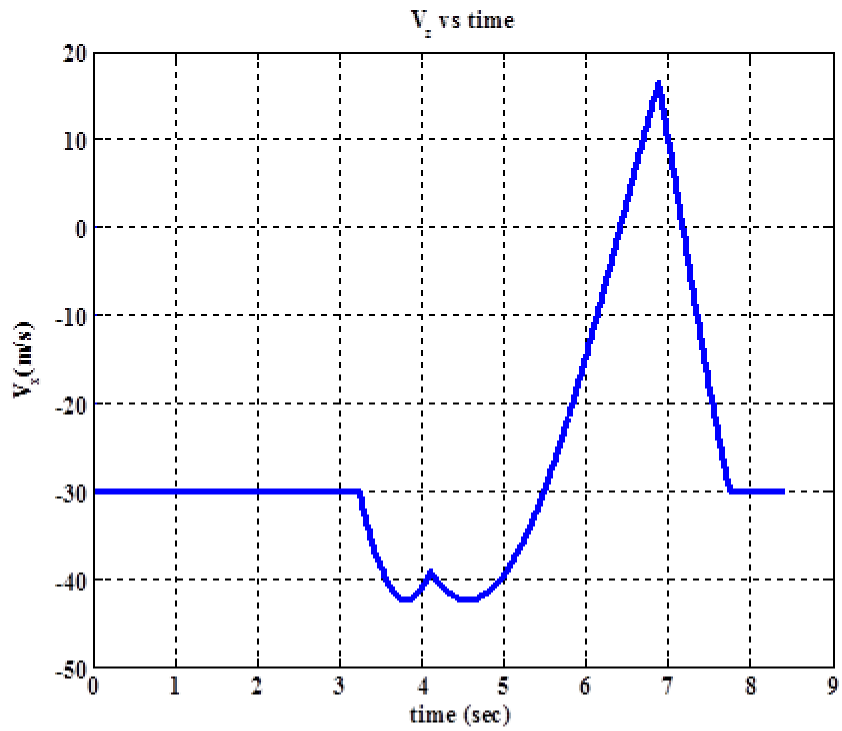


**Figure 20 -** Vertical Velocity of the Vehicle in Obstacle Avoidance in Vertical Axis

# CHAPTER 5

# 3D COLLISION CONE METHOD

The 2D obstacle avoidance method with constant velocity and predetermined avoidance zone is explained in Chapter 4. In this chapter, the method is extended to 3D.

## 5.1 3D CCM Application

In this part, the algorithm in part 4.2 is extended to 3D. In the 3D problem, the axes are coupled. The navigation equations for the Earth model problems are modified and used for this algorithm.

The rate of change of the elevation and heading angles are modified using a sphere are in equations 5.1-5.2 can be shown in Figure 21, which is referenced from [28].

$$\dot{\varphi} = \frac{V_N}{R} \tag{5.1}$$

$$\dot{\mu} = \frac{V_E}{Rcos(\varphi)} \tag{5.2}$$

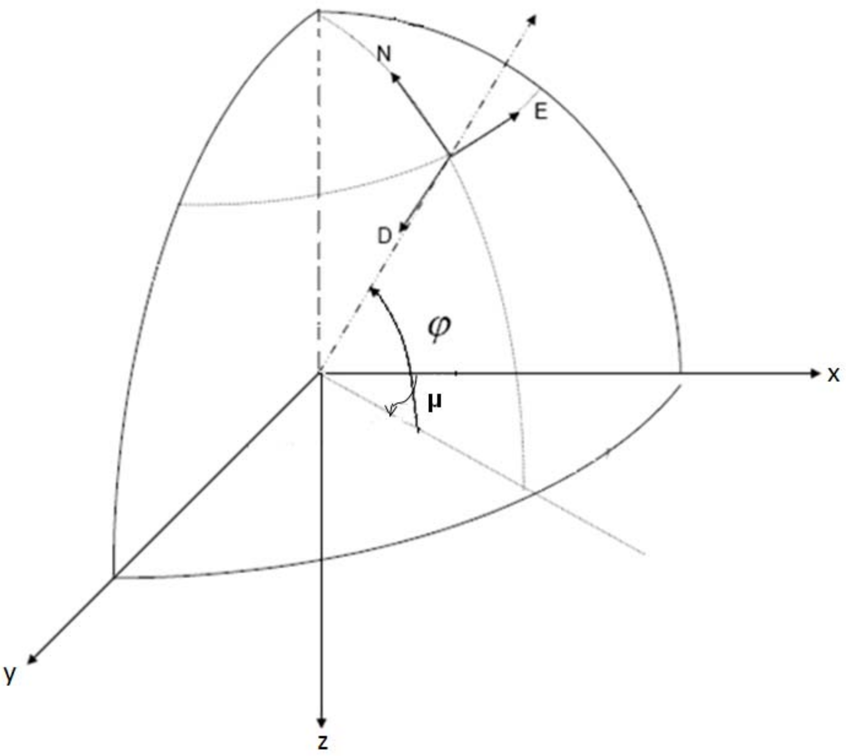


**Figure 21 –** Coordinate Frame and Angular Displacement on a Sphere

In the navigation equations, the motion around a sphere is modeled. However, the motion in this problem is considered in Cartesian coordinates. Therefore, the motion is determined as an elevation in the velocity in z axis, $V_z$ or $V_D$, instead of $V_N$ in navigation. The effect of $V_D$ is settled as magnitude in the problem and equation 5.1 is changed to equation 5.3.

$$\dot{\varphi} = \frac{V_N \tan \varphi}{R} \tag{5.3}$$

The heading and elevation angles are found from equation 5.4.

$$\begin{bmatrix} \varphi \\ \mu \end{bmatrix} = \int \omega_{NED} \tag{5.4}$$

The velocity vector is rotated to its new orientation with the transformation matrix in equation 4.14 as in equation 5.5.

$$\text{V\_vector\_new} = \hat{C}(\text{B, NED}) * \text{V\_vector} \tag{5.5}$$

There is a difference in the transition logic when compared to part 4.2. After the avoidance is completed, the vehicle settles on the commanded trajectory. The transition is handled by settling the vehicle on the trajectory when the vehicle is a detection distance away from the obstacle. This logic works on a 2D motion with only elevation or heading plane. Therefore, in 3D, the vehicle is in the commanded trajectory when the heading and elevation angles are the same as in the commanded trajectory.

In this algorithm, the vehicle avoids the obstacle on its trajectory and the avoidance is carried out in the same plane of motion. For example, the simulation result in 5.2 is with a heading of 60° and elevation of 30°, therefore, the avoidance is carried out on a plane with 60° and 30° elevation.

There are two issues, which result from $\tan\varphi$ term in equation 5.3. One of them is the case where both of the heading and elevation angles are 0°. In this case, the avoidance cannot be carried out, as the vehicle is defined no direction of avoidance. The solution is to feed 1 instead of $\tan\varphi$ and let the vehicle avoid the obstacle in elevation plane. The second case is 90° elevation angle, where $\tan\varphi$ goes to infinity. This problem is removed by defining the 90° elevation angle in an error bound around 90°, not the exact value.

## 5.2 Simulation Results

Simulation results of a 3D motion is shown in this part with graphs of trajectory, heading angle change, elevation angle change and velocity in Figure 22 - Figure 27.

The parameters used for the simulation are given in Table 3.

**Table 3 –** Parameters for the 3D Simulation

| | |
|---|---|
| Initial Point of The Vehicle [m] | [0 0 0] |
| Heading Angle [°] | 60 |
| Elevation Angle [°] | 30 |
| Position of The Obstacle on the Direction of Vehicle [m] | 200 |
| $\vec{V}$ [m/s] | 30 |
| $R_{sphere}$ [m] | 50 |
| Detection Distance [m] | 100 |

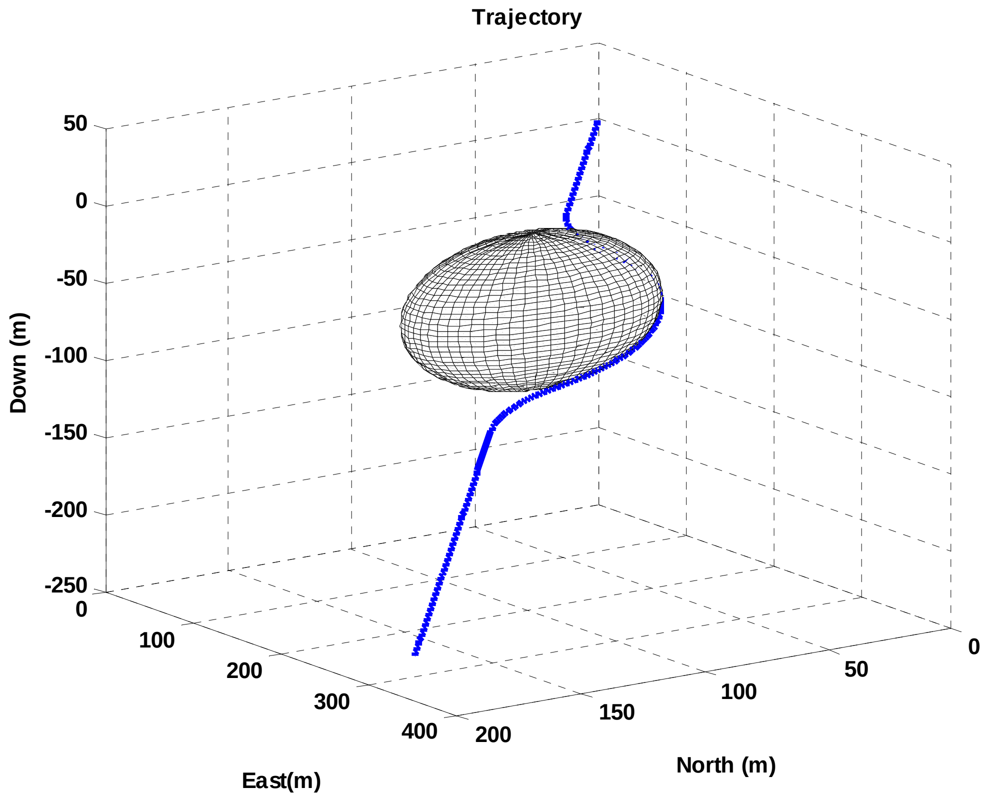


**Figure 22 -** Trajectory of the Vehicle in 3D Obstacle Avoidance

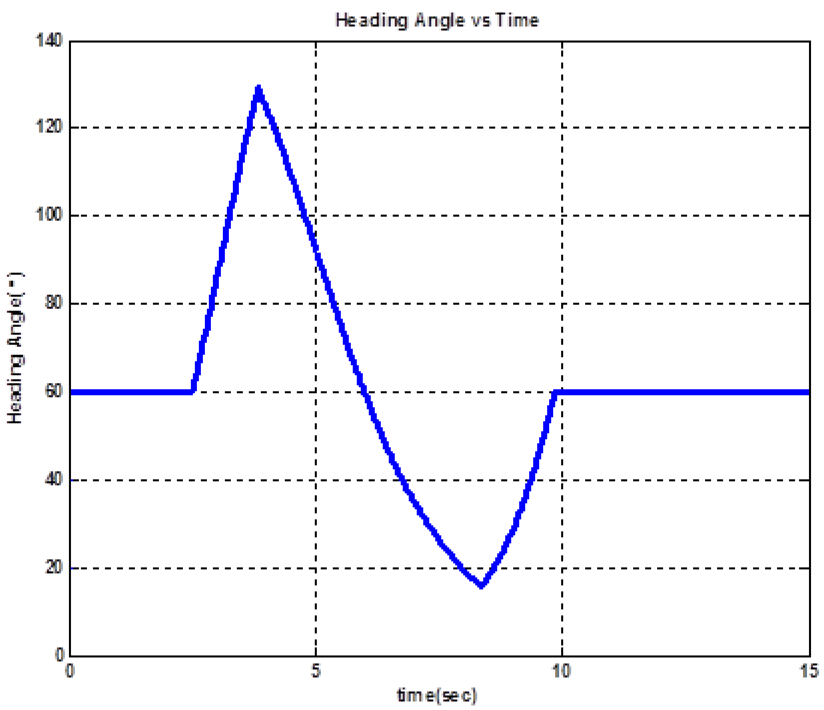


**Figure 23 -** Heading Angle of the Vehicle in 3D Obstacle Avoidance

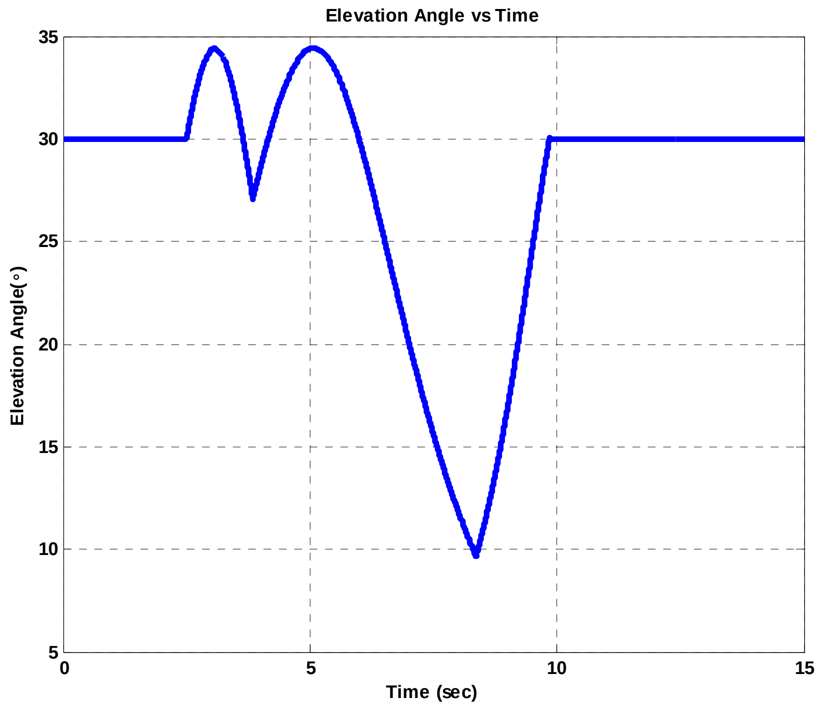


**Figure 24 -** Elevation Angle of the Vehicle in 3D Obstacle Avoidance

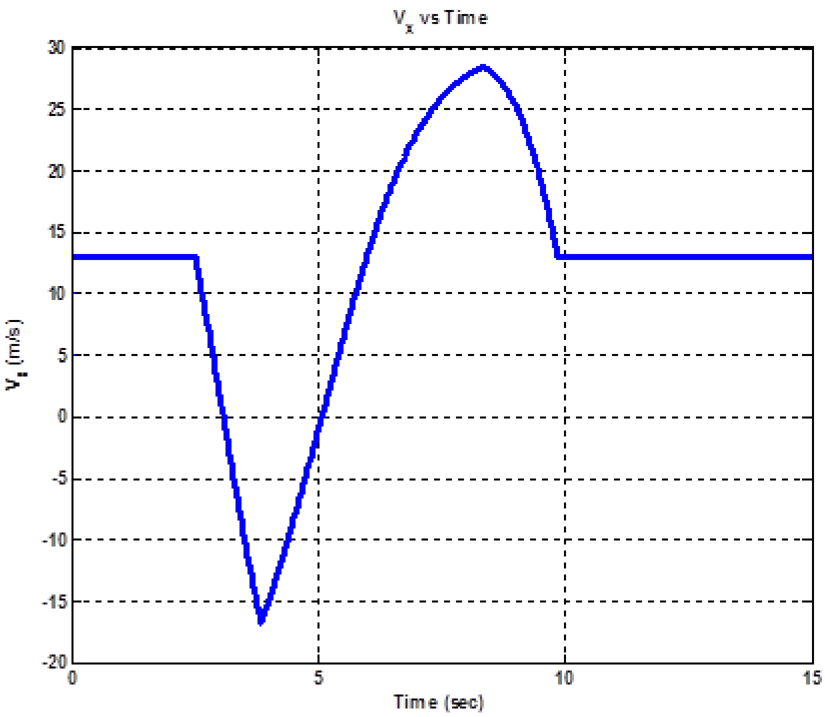


**Figure 25 -** Downrange Velocity of the Vehicle in 3D Obstacle Avoidance

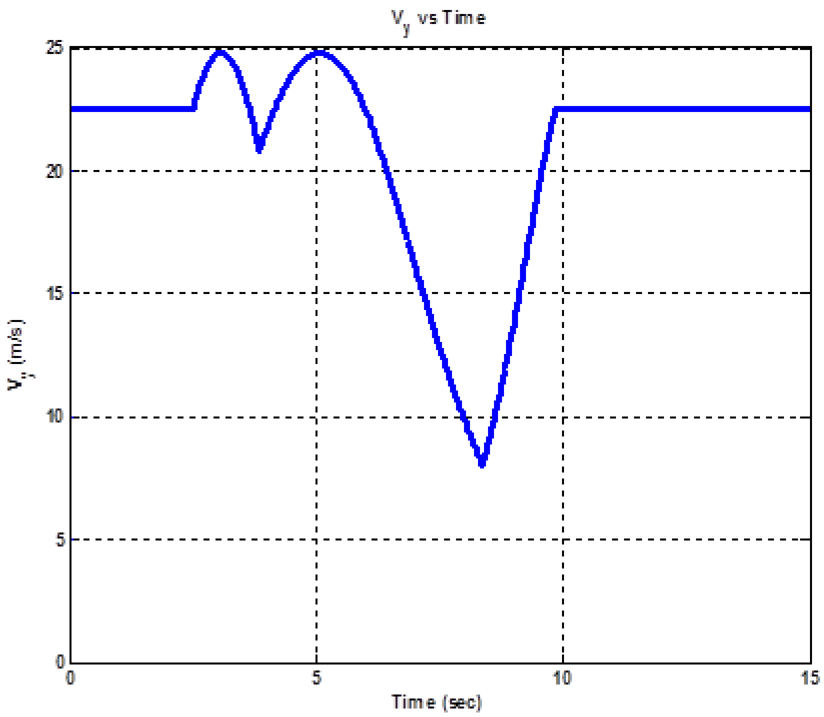


**Figure 26 -** Crossrange Velocity of the Vehicle in 3D Obstacle Avoidance

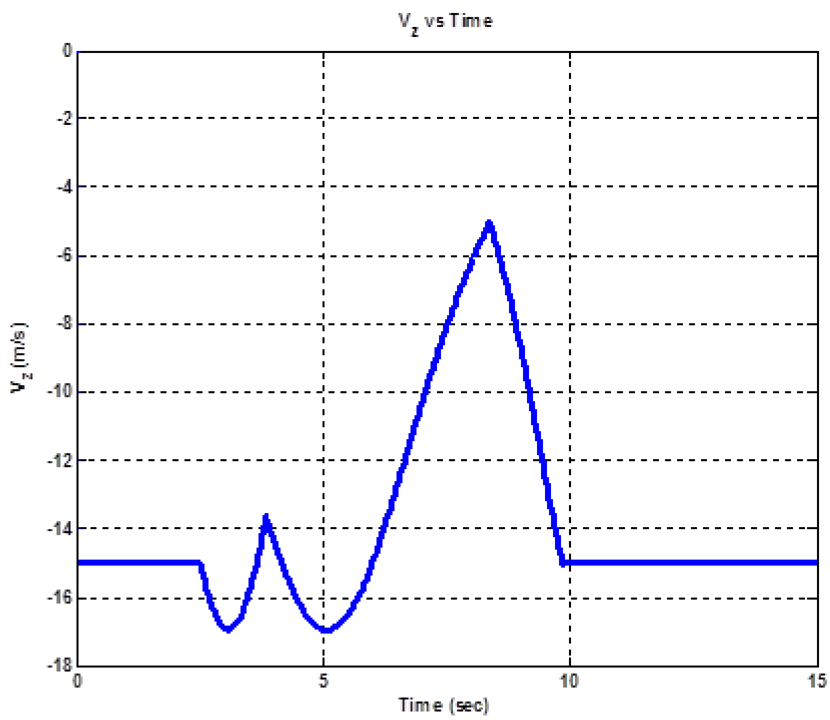


**Figure 27 -** Vertical Velocity of the Vehicle in 3D Obstacle Avoidance

In Figure 24, the effect of coupling between the horizontal and vertical axis can be seen. As the vehicle avoids the obstacle in 3D, the trend of the vertical and horizontal velocity in Figure 26 and Figure 27 are almost the same.

# CHAPTER 6

# OPTIMIZED COLLISION CONE METHOD WITH CONSTRAINTS

In this chapter, the obstacle avoidance is optimized in terms of minimum energy and minimum separation distance. The flight envelope limits are modeled in terms of acceleration and velocity limits. Therefore, the avoidance maneuver is optimized to avoid these limits as well.

## 6.1 Basic Optimization Problem

The main goal of optimization is to find the optimum solution of a problem by obtaining the appropriate control variables for the problem. A performance index (a cost function to be optimized) for the problem has to be set. The optimization problem may involve equality or inequality constraints; then the problem is said to be a constrained optimization problem. If there are no constraints, then the problem is unconstrained.

$$\mathrm{L}(x, u) = \text{performance index}$$

$$u = \begin{bmatrix} u_1 \\ \cdot \\ \cdot \\ \cdot \\ u_m \end{bmatrix} = \text{control vector}, \quad x = \begin{bmatrix} x_1 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix} = \text{state vector}, \quad f = \begin{bmatrix} f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_n \end{bmatrix} = \text{constraint vector}$$

minimize $\mathrm{L}(x, u)$

subject to $f_1(x, u) = 0$ (equality constraint)

$f_2(x, u) \leq 0$ (inequality constraint)

## 6.2 Optimization Methods

Optimization methods are in general classified as analytical, experimental, graphical and numerical [29].

Analytical methods apply the rules of differential calculus to the optimization problem. The performance index, constraint and state equations are clearly defined and the set of equations are derived by the parameters. Afterwards, the derivative functions are set to zero in order to optimize the problem. The advantage of analytical methods is their simplicity and minimum need for the use of computers. However, for the highly nonlinear problems and for the problems with several independent parameters, the method is not applicable.

Another method to apply is the graphical method. Graphical methods are easy by means of finding the optimum values of the problem illustratively from graphs.

Experimental methods provide the optimum solution as well. After the problem is described, the possible values of parameters within the constraints/limits are experimented one by one by carrying out batch analysis.

Numerical methods are the most commonly used optimization methods. These methods use iterations to find the solution by making progress on the results of every single iteration. A proper initial estimate to start the iteration process is needed and when the solutions converge to a trend, the algorithm stops. The advantage of this method is its effectiveness on highly nonlinear problems. The algorithm of this method can be programmed by using computer as well.

Numerical methods are classified as linear programming, integer programming, quadratic programming, nonlinear programming, stochastic programming and dynamic programing methods as described in [29] and [30]. To briefly describe these methods; they differ by the way of the mathematical behavior, the form of the performance index and constraint functions. If the cost function and the constraint functions are linear, then the optimization method is called linear programming, if the cost function is quadratic; then the method is selected as quadratic programming method. When the parameters need to be integer values, it is integer programming. Nonlinear programming is the method for solving the problems with nonlinear cost and constraint functions. Stochastic programming method is used when some of the constraints depend on random variables. Dynamic programming is a method for obtaining the global solution of complex or multipart problems. The problem can be solved in smaller local parts as well to get the whole result as the subparts may interact or may be in sequence with each other. Other numerical methods described above can be used for dynamic programming.

## 6.3 Constrained Optimization Methods

Many dynamic systems have limits as in real life they deal with the physical laws. Therefore, many of the optimization problems are constrained. Common methods applied to constrained optimization problems are Lagrange multipliers formulation for optimization, penalty-barrier function method, quadratic programming and gradient based methods.

These methods can be briefly explained on example 6.1 as in [31].

$$\begin{array}{llll} & \text{minimize } f(x) & & \\ \text{subject to} & g_i(x) = 0 & i=1,\ldots,m & \\ & h_j(x) \leq 0 & j=1,\ldots,n & \end{array} \qquad (6.1)$$

The problem can be solved using classical methods. As it has inequality constraints, Karush-Kuhn-Tucker conditions are used.

$$L(x,\lambda) = f(x) + \sum_{j=1}^{n} \lambda_{jh}^T h_j(x) + \sum_{i=1}^{m} \lambda_{ig}^T g_i(x) \qquad (6.2)$$

$$\frac{\partial L}{\partial x_{i,j}} = 0 \tag{6.3}$$

$$\frac{\partial L}{\partial \lambda_{i,j}} = 0 \tag{6.4}$$

Equations 6.2-6.4 are solved using linear algebra to find the appropriate x value-control variable to minimize the cost function.

The approach with penalty-barrier method, the problem 6.1 is set in the form of;

$$\text{minimize } P(x)$$

$$P(x, \eta, \beta) = f(x) + \sum_{j=1}^{n} \eta h_j^2\,(x) + \sum_{i=1}^{m} \beta g_i^2\,(x) \tag{6.5}$$

where η and β are the penalty parameters

In this method, it is important to get the appropriate value of the penalty parameters without getting very small or big numbers. Therefore, sequential unconstrained minimization technique is used as explained in [31] as well.

Quadratic programming is another widely used method of optimizing constrained problems. The problem in equation 6.1 is altered to equation 6.6.

$$f(x) = \frac{1}{2} x^T A x + b^T x + c \tag{6.6}$$

subject to $Cx \leq d$

where A,b,C,d are the matrix form of the coefficients of the variables and constraints

After this operation, Lagrange multiplier or Theil and Van de Panne method is used to solve the problem. Theil and Van de Panne [31] takes the unconstrained minimum of the problem as a first step to determine the active constraints.

Gradient based methods are the modern methods, which are explained in [31] such as gradient projection method and Sequential Quadratic Programming (SQP).

The gradient projection method is a gradient based method as well, which searches for a direction of feasible points starting from an initial feasible point. Then, a projection matrix is produced in order to descent the value of feasible points on the direction of search to the real value of the feasible point. In Figure 28, the method is shown, where the $x^i$ values are the optimized parameters of the problem and $\bar{x}^i$ values are the values on the direction of feasible region, the descent vectors are the projection vectors to the optimized value.

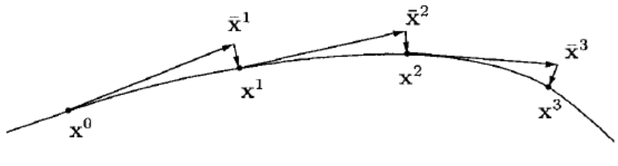


**Figure 28 –** Schematic of gradient projection method

Sequential Quadratic Programming (SQP) method is based on Newton's method and it is an iterative search method, it includes the selection of the active constraints as well.

## 6.4 Optimization Problem in This Thesis

Obstacle avoidance methods can be treated as a part of trajectory optimization problems. Most of the problems in literature are related to dynamic programming method and the trajectory is optimized in the whole problem. However, in this thesis, the obstacle avoidance problem is CCM and it is aimed to optimize the main parameters in this problem, velocity (V) and the radius of obstacle avoidance zone ($R_{sphere}$).

Optimization is carried out according to the objective functions, which are mainly divided into three categories; minimum time, minimum fuel and minimum energy. The objective of the optimization problem in this thesis is to minimize the lateral acceleration of the vehicle while turning around an obstacle. The performance index of a minimum energy problem is stated as in 6.7;

$$L = \int u^2 dt \tag{6.7}$$

where u stands for acceleration for problems related to dynamic and mechanical systems.

The performance index for the problem of this thesis is;

$$L = \int \left( \left( \frac{V^2}{R_{sphere}} \right)^2 + \left( \frac{V^2}{R_2} \right)^2 \right) dt \tag{6.8}$$

The lateral acceleration contributes to the energy consumption of the vehicle and this can be minimized using the lateral acceleration parameter as well.

$$\left|\frac{V^2}{R_{sphere}}\right| \leq a_{max} \tag{6.9}$$

$$\left|\frac{V^2}{R_2}\right| \leq a_{max} \tag{6.10}$$

$$r_{obstacle} < R_{sphere} < dd \tag{6.11}$$

$$V_{stall} \leq V \leq V_{max} \tag{6.12}$$

For the optimization problem, Lagrange multipliers approach is selected. The inequality constraints in the problem are formulated as Karush-Kuhn-Tucker conditions. The problem can be approached in two aspects; optimization of the avoidance zone radius by setting constant velocity and optimization of the velocity and the avoidance zone radius.

### 6.4.1 Avoidance Zone Optimization

For avoidance zone optimization, the constraints in equations 6.9 - 6.11 can be combined in one equation using the relationship between $R_{sphere}$ and $R_2$, which is shown in equation 4.8.

The avoidance zone radius constraint is set by using the lateral acceleration limits in equations 6.9 – 6.10 and new radius constraint is defined by;

$$R_{min} < R_{sphere} < R_{max} \qquad (6.13)$$

The Lagrangian equation for this problem is;

$$L(R_{sphere}, \lambda_1, \lambda_2) = \left(\frac{V^2}{R_{sphere}}\right)^2 + \left(\frac{V^2}{R_2}\right)^2 + \lambda_1(R_{min} - R_{sphere}) + \lambda_2(R_{sphere} - R_{max}) \qquad (6.14)$$

The Karush-Kuhn-Tucker conditions are;

$$\frac{\partial L}{\partial R_{sphere}} = 0 \qquad (6.15)$$

$$\frac{\partial L}{\partial \lambda_{1,2}} = 0 \qquad (6.16)$$

$$R_{min} - R_{sphere} \le 0 \qquad (6.17)$$

$$R_{sphere} - R_{max} \le 0 \qquad (6.18)$$

As the Lagrange multipliers, $\lambda_1$ and $\lambda_2$ should be nonnegative, equations 6.17-6.18 can be rewritten as;

$$\lambda_1(R_{min} - R_{sphere}) = 0 \tag{6.19}$$

$$\lambda_2(R_{sphere} - R_{max}) = 0 \tag{6.20}$$

$$\lambda_1, \lambda_2 \geq 0 \tag{6.21}$$

The active Lagrange multipliers are defined according to the radius constraint in equations 6.17-6.18. In this problem, there are alternatives of active constraints;

- $\lambda_1, \lambda_2 = 0$ → Optimized avoidance radius is within the limits in equation 6.13.
- $\lambda_1 \neq 0$ → Optimized avoidance radius is smaller than $R_{min}$, therefore, the avoidance zone radius is set to $R_{min.}$
- $\lambda_2 \neq 0$ → Optimized avoidance radius is larger than $R_{max}$, therefore, the avoidance zone radius is set to $R_{max.}$

### 6.4.2 Avoidance Zone and Velocity Optimization

For the optimization of velocity along with the avoidance zone radius, two alternative optimization method are used; optimization with Lagrange multipliers and direct experimentation method.

#### 6.4.2.1 Optimization with Lagrange Multipliers

For the optimization of velocity and the avoidance zone radius, the velocity limits of the vehicle in equation 6.12 should be taken into account. The avoidance zone radius cannot be limited using the lateral acceleration limitations, as the velocity value is not fixed.

The Lagrangian equation is;

$$L(R_{sphere}, V, \lambda_1, \lambda_2, \lambda_3, \lambda_4) = \left(\frac{V^2}{R_{sphere}}\right)^2 + \left(\frac{V^2}{R_2}\right)^2 + \lambda_1(R_{obstacle} - R_{sphere}) + \lambda_2(R_{sphere} - dd) + \lambda_3(V_{stall} - V) + \lambda_4(V - V_{max}) \quad (6.22)$$

The Karush-Kuhn-Tucker conditions are;

$$\frac{\partial L}{\partial R_{sphere}} = 0 \quad (6.23)$$

$$\frac{\partial L}{\partial V} = 0 \quad (6.24)$$

$$\frac{\partial L}{\partial \lambda_{1,2,3,4}} = 0 \quad (6.25)$$

$$R_{obstacle} - R_{sphere} \leq 0 \quad (6.26)$$

$$R_{sphere} - dd \leq 0 \quad (6.27)$$

$$V_{stall} - V \leq 0 \quad (6.28)$$

$$V - V_{max} \leq 0 \quad (6.29)$$

As the Lagrange multipliers, $\lambda_1$, $\lambda_2$, $\lambda_3$, $\lambda_4$ should be nonnegative, equations 6.26-6.29 can be rewritten as;

$$\lambda_1(R_{obstacle} - R_{sphere}) = 0 \quad (6.30)$$

$$\lambda_2(R_{sphere} - dd) = 0 \quad (6.31)$$

$$\lambda_3(V_{stall} - V) = 0 \quad (6.32)$$

$$\lambda_4(V - V_{max}) = 0 \quad (6.33)$$

$$\lambda_1, \lambda_2, \lambda_3, \lambda_4 \geq 0 \quad (6.34)$$

As explained in part 6.4.1, the active constraints of the problem are set. In addition to the avoidance zone and velocity limit constraints shown in equations 6.11 and 6.12; there are lateral acceleration limits of the vehicle, which is defined in equations 6.9 and 6.10. However, lateral acceleration values depend on both of avoidance zone radius and velocity and the lateral acceleration limits should be checked in every step of optimization process. This increases the complexity of the problem as well.

#### 6.4.2.2 Optimization Using Direct Experimentation

In the direct experimentation method, avoidance zone radius and velocity values that are within the limits defined in equations 6.11 and 6.12 are experimented in batch simulations according to the lateral acceleration limits shown in equations 6.9 and 6.10. Then, the optimum simulation values of V and $R_{sphere}$, which has the minimum of the sum of the squares of the lateral acceleration values in regions 1 and 2 in Figure 9, is selected as the velocity and the avoidance zone radius for the obstacle avoidance simulation.

### 6.4.3 Simulation Results for Optimized Obstacle Avoidance

Simulations are carried out for the methods, which are explained in detail in parts 6.4.1 and 6.4.2. In this part, simulation results are shown.

#### 6.4.3.1 Avoidance Zone Optimization

Avoidance zone optimization is analysed in two aspects; different detection distance values and different velocities.

The first analysis is carried out in order to see the effect of detection distance to the avoidance zone. The constant parameters used in the simulation are shown in Table 4. For the analysis, detection distance value is changed from 100m to 1000m and the velocity limits are set as 20 m/s and 200 m/s. When the lateral acceleration values exceed the limit, the code generates no solution. An example is given in Table 4 and Table 5

**Table 4 –** Parameters used in Simulation for Different Detection Distances

| | |
|---|---|
| Obstacle Size as Radius (m) | 50 |
| Maximum Lateral Acceleration (g) | 5 |
| Velocity (m/s) | 75 |
| Detection Distance (m) | 100:100:1000 |

As seen from the tabulated results of the simulation in Table 5, except the case with detection distance value of 100m, in all of the cases, the avoidance zone is within the limits and as early as the vehicle detects the obstacle, less acceleration the vehicle pulls.

**Table 5 –** Avoidance Zone Optimization Results for Different Detection Distances

| Detection Distance (m) | Minimum Avoidance Radius (m) | Maximum Avoidance Radius (m) | Avoidance Zone Radius (m) | Lateral Acceleration in Region 2 (g) | Lateral Acceleration in Regions 1&3 (g) | L ($g^2$) |
|---|---|---|---|---|---|---|
| 100 | No Solution | | | | | |
| 200 | 115 | 115 | 115 | 4.98 | 4.92 | 49 |
| 300 | 115 | 206 | 157 | 3.65 | 2.75 | 21 |
| 400 | 115 | 301 | 210 | 2.73 | 2.07 | 11.8 |
| 500 | 115 | 398 | 262 | 2.19 | 1.65 | 7.5 |
| 600 | 115 | 496 | 314 | 1.83 | 1.38 | 5.23 |
| 700 | 115 | 594 | 366 | 1.57 | 1.18 | 3.85 |
| 800 | 115 | 693 | 419 | 1.37 | 1.04 | 2.94 |
| 900 | 115 | 792 | 471 | 1.22 | 0.92 | 2.32 |
| 1000 | 115 | 891 | 523 | 1.1 | 0.83 | 1.88 |

The results for detection distance values of 300-500-1000m are plotted in Figure 29 for 2D motion and Figure 30 for 3D motion. The dotted lines in the figures show the optimized avoidance zone. In Figure 31, for the detection distance value of 300m, the maximum and minimum avoidance zones are shown as dark circles and dotted line is the avoidance zone. As stated in the logic in part 4.2, at the time the vehicle detects the obstacle, the avoidance maneuver is started. Therefore, the detection distance in the graphs can be evaluated as the point to start the maneuver. The lateral acceleration values for the same detection distances can be seen from Figure 32. In the simulations, the obstacle is on [1100- 0- 0] m.

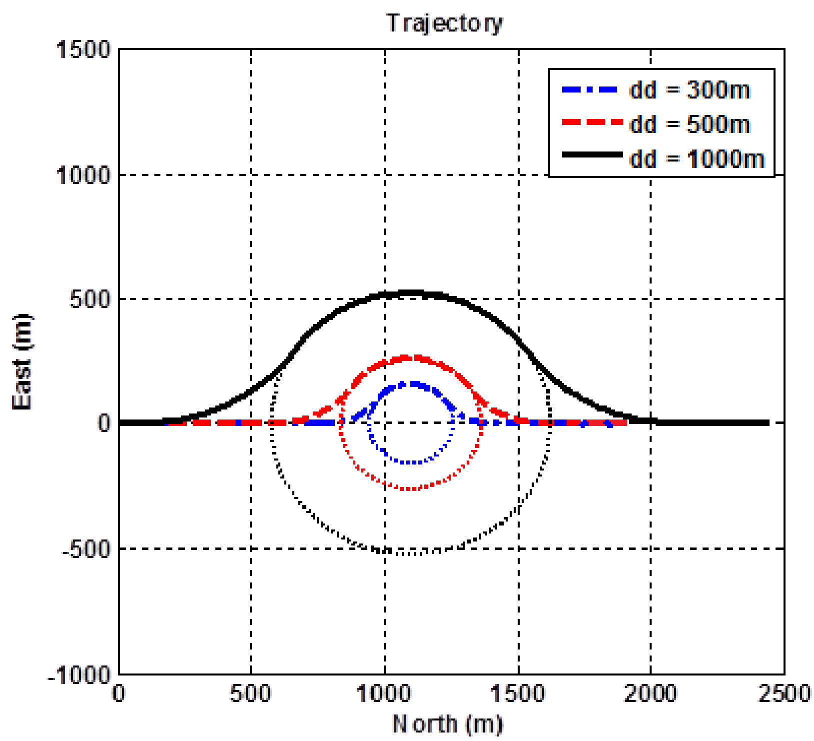


**Figure 29 –** 2D Trajectories for Different Detection Distance Values

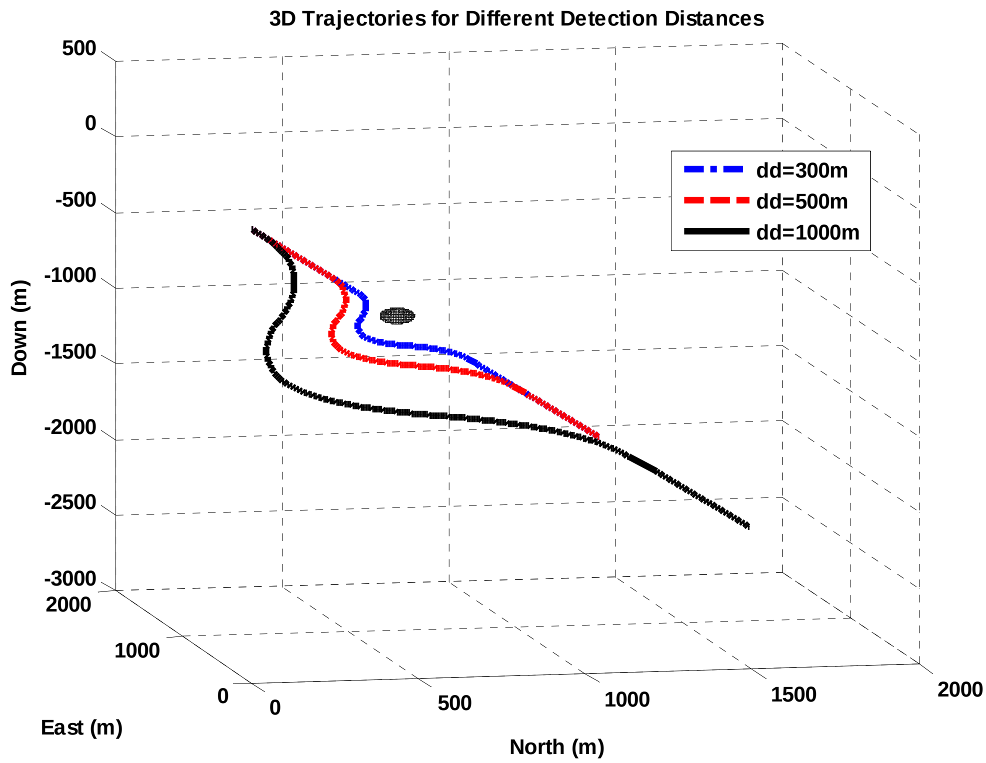


**Figure 30** – 3D Trajectories for Different Detection Distance Values

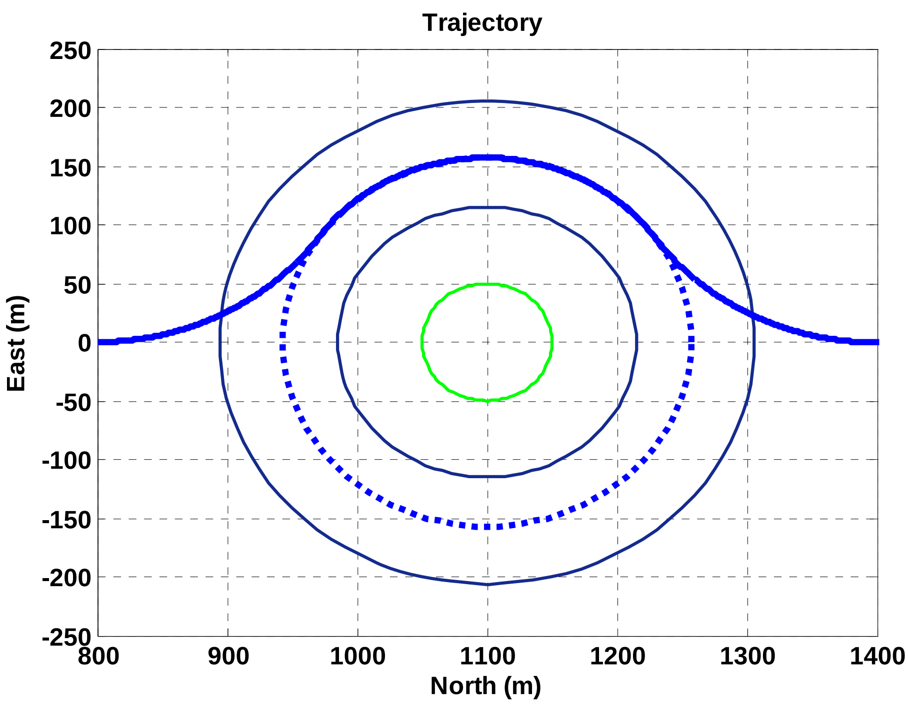


**Figure 31** – Trajectory and Avoidance Zone Limitations

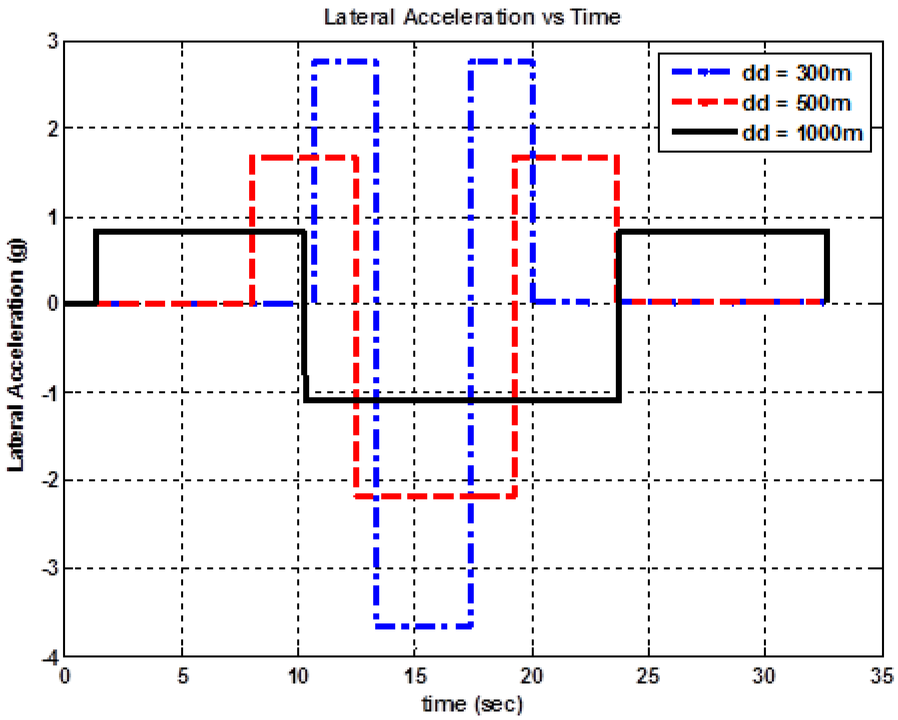


**Figure 32 –** Lateral Acceleration Values for Different Detection Distance Values

The second analysis for avoidance zone optimization is in the aspect of different velocities. The parameters used in sample simulation can be found from Table 6.

**Table 6** - Parameters used in Simulation for Different Velocities

| | |
|---|---|
| Obstacle Size as Radius (m) | 50 |
| Maximum Lateral Acceleration (g) | 5 |
| Detection Distance (m) | 200 |
| Velocity (m/s) | 20:10:100 |

The simulation results are tabulated and can be seen from Table 7. For the simulations with lateral acceleration values that exceed the limit, no solution is displayed. As expected, when the vehicle has higher velocity, the acceleration value the vehicle pulls gets bigger. It is seen from Table 7 for the same detection distance, change of the velocity doesn't affect the optimized avoidance zone. As seen from Figure 34, the change of velocity effects the time to complete the avoidance maneuver; slower the vehicle, more long-lasting the maneuver.

**Table 7 –** Avoidance Zone Optimization Results for Different Velocities

| Velocity (m/s) | Minimum Avoidance Radius (m) | Maximum Avoidance Radius (m) | Avoidance Zone Radius (m) | Lateral Acceleration in Region 2 (g) | Lateral Acceleration in Regions 1&3 (g) | L ($g^2$) |
|---|---|---|---|---|---|---|
| 20 | 50 | 192 | 104.7 | 0.39 | 0.3 | 0.24 |
| 30 | 50 | 182 | 104.7 | 0.88 | 0.67 | 1.2 |
| 40 | 50 | 170 | 104.7 | 1.56 | 1.18 | 3.81 |
| 50 | 51 | 155 | 104.7 | 2.43 | 1.83 | 9.3 |
| 60 | 74 | 139 | 104.7 | 3.5 | 2.65 | 19.3 |
| 70 | 100 | 123 | 104.7 | 4.77 | 3.6 | 35.8 |
| 80 | No Solution | | | | | |
| 90 | No Solution | | | | | |
| 100 | No Solution | | | | | |

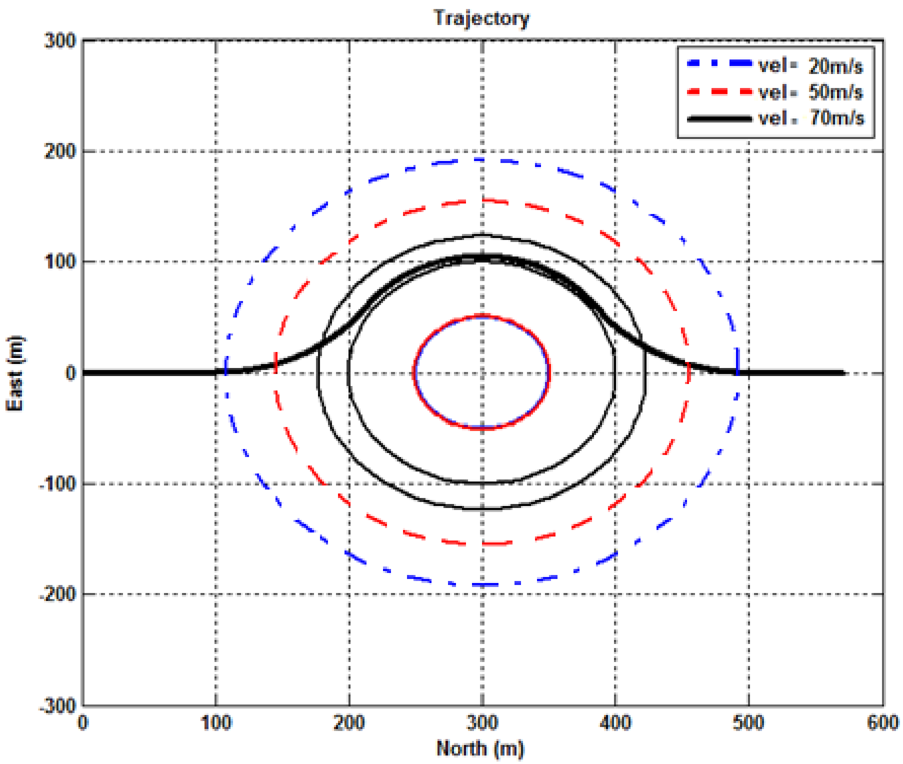


**Figure 33** – Trajectory and Avoidance Zone Limitations for Different Velocities

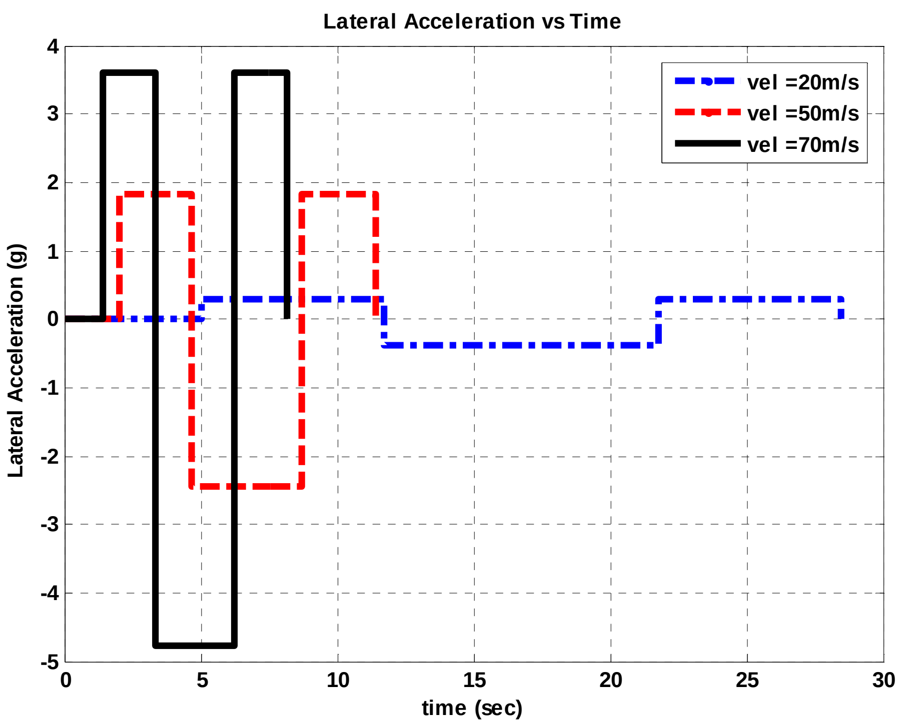


**Figure 34** – Lateral Acceleration Values for Different Velocities

In Figure 33, it is seen that the trajectory for different velocities is the same, for the reason that the avoidance zone is the same. However, the avoidance zone limitations, which are shown in the figure with thinner circles, are different. The lateral acceleration values are plotted on Figure 34. Another point in Figure 34 is the difference of the avoidance time. When the vehicle is slower, the time to avoid the obstacle increases.

#### 6.4.3.2 Avoidance Zone and Velocity Optimization

Avoidance zone and velocity optimization is carried out in two approaches; Lagrange multipliers and direct experimentation. With the Lagrange multipliers method with Karush-Kuhn-Tucker conditions, no unique solution is found. As in equation 6.14, the cost function has fourth order V term and third order R term; there is infinite number of solutions for every velocity and radius value.

The direct experimentation is carried out with velocity range of 20-200 m/s and avoidance zone radius range starting from the size of the obstacle to the detection distance. The lateral acceleration limits are set as well. However, for every velocity value, there are many avoidance zone radius values within the acceleration limit. The optimized value for minimum energy is for the minimum velocity value, which could not be preferred at all time. Therefore, the velocity and avoidance zone could not be optimized together for this problem.

# CHAPTER 7

# CONCLUSION&DISCUSSIONS

This thesis presents a local and reactive obstacle avoidance algorithm. The collision cone approach is used for obstacle avoidance. In this thesis, an obstacle avoidance algorithm is developed in MATLAB/Simulink environment using a point mass simulation model.

The Collision Cone algorithm is formed to avoid the obstacles at a predefined detection distance by changing heading or elevation angle. The sizes of the obstacles are set in the algorithm and the avoidance zone radius is defined accordingly. The main criterion in this method is not to exceed the aerodynamic and structural limits of the vehicle. Acceleration of the vehicle during the avoidance is the main element in defining these limits.

The algorithm is extended into a three dimensional space. In the extension algorithm, navigation equations for the Earth model are modified and used. However, the vehicle is not treated as moving on Earth or a sphere. While forming the point mass simulation, flat and nonmoving environment is assumed, since the avoidance maneuver takes place in a limited region, in which the effect of spherical environment cannot be seen. The problem is considered in Cartesian coordinates. Therefore, necessary modification on angular velocity term in elevation plane is done by setting the magnitude of $V_D$ into $V_N$. The transformation matrix from body frame to navigation

(inertial) frame is used instead of the Earth frame to navigation frame, which is used on navigation equations. In 3D algorithm, the avoidance maneuver is carried out in the direction of the commanded trajectory. In other words, initial angles of the vehicle while avoiding the obstacle are the angles of the vehicle in the trajectory.

Next, the avoidance algorithm is advanced to optimize the avoidance zone radius based on the velocity and lateral acceleration limits and in order to minimize the energy of the vehicle during maneuver. The problem is a constrained optimization problem with the radius, velocity and acceleration limits. Among the constrained optimization methods, Lagrange Multipliers with Karush-Kuhn-Tucker conditions method is chosen, which is a basic optimization method and commonly used for inequality constraints. The avoidance zone limits are the size of the obstacle and the detection distance. However, in this problem, the radius is limited with the acceleration limit of the vehicle as well. Therefore, the avoidance zone radius range is constricted. With the simulations, the change of optimized radius with velocity and detection distance is analyzed. It is seen that detection distance or the start point of the maneuver has an obvious effect on optimized radius, whereas velocity has no effect on optimized radius, except the change of lateral acceleration.

The optimization is carried out for both velocity and avoidance zone. Lagrange Multipliers with Karush-Kuhn-Tucker conditions and direct experimentation methods are applied. Since the cost function contains highly nonlinear terms of velocity and avoidance zone radius, a unique solution could not be found. For each of the velocity values, there is an optimum radius value. The velocity value to minimize the energy is for minimum velocity and this is not preferable. Therefore, to optimize this problem in terms of energy, one of the velocity or avoidance zone radius values should be decided to be optimized while having the other value as a selected constant parameter.

## 7.1 Future Work

As a future work, this algorithm can be improved in terms of optimum avoidance direction. In this thesis, the avoidance is in the same plane with the direction of the vehicle. The avoidance maneuver may be optimized in terms of potential or kinetic energy separately and this affects the avoidance direction of the vehicle.

The algorithm can be implemented on an autonomous vehicle. The transition between the regions should be modified according to the vehicle limits and maneuverability. Point mass simulation in this thesis is given no vehicle dynamics and carries out all the commanded maneuvers. However, every vehicle cannot change direction in a sharp way or in short time. Therefore, the transition logic should be improved according to the vehicle dynamics.

Detection part is not modelled in this work, a perfect sensor is assumed. An obstacle detection algorithm can be modelled as well and obstacle avoidance algorithm can be improved for pop-up obstacles.

The algorithm is developed for stationary obstacles. An improvement can be done for moving obstacles as well.

# REFERENCES


[1] R. Frampton, “UAV Autonomy”, Defence Codex, Issue 1, summer 2008

[2] A. Muhumdar, R. Padhi, “Evolving Philosophies on Autonomous Obstacle/Collision Avoidance of Unmanned Aerial Vehicles ”, Journal of Aerospace Computing, Information and Communication, Vol 8, February 2011

[3] Ferbach, P., “A Method of Progressive Constraints for Nonholonomic Motion Planning,” IEEE Transactions on Robotics and Automation, Vol. 14, No. 1, Feb. 1998, pp. 172–179

[4] LaValle, S. M., Planning Algorithms, Cambridge University Press, Cambridge, 2006

[5] Lozano-Pérez, Tomás; Wesley, Michael A., "An algorithm for planning collision-free paths among polyhedral obstacles", Communications of the ACM 22 (10): 560–570, 1979

[6] M. Restelli, “Path Planning – Robotica for Computer Engineering Students”, A.A. 2006/2007

[7] S.M.LaValle, “Rapidly Exploring Random Trees : A New Tool for Path Planning”, Computer Science Department, Iowa State University, October 1998

[8] J. Saunders, B. Call, A. Curtis, R. Beard, and T. McLain, “Static and Dynamic Obstacle Avoidance in Miniature Air Vehicles,” Proceedings of the Infotech@Aerospace, AIAA, Arlington, Virginia, Sept. 2005; AIAA paper 2005-6950

[9] Khatib, O., “Real Time Obstacle Avoidance for Manipulators and Mobile Robots,” The International Journal of Robotics Research, Vol. 5, No. 1, March 1986, pp. 90–98

[10] P.Zarchan, “Tactical and Strategic Missile Guidance” AIAA,1994

[11] Watanabe, Y., Calise, A. J., and Johnson E. N., “Minimum Effort Guidance for Vision-Based Collision Avoidance,” Proceedings of the Atmospheric Flight Mechanics Conference and Exhibit, AIAA, Keystone, Colorado, 21–24 Aug. 2006; AIAA Paper 2006-6641

[12] H. Choi, Y. Kim, I. Hwank, “Vision-Based Reactive Collision Avoidance Algoritm for Unmanned Aerial Vehicle” , AIAA Guidance, Navigation and Control Conference and Exhibit, August 2011

[13] Chakravarthy, A., and Ghose, D., “Obstacle Avoidance in a Dynamic Environment: A Collision Cone Approach,” IEEE Transactions on

Systems, Man and Cybernetics—Part A: Systems and Humans, Vol. 28, No. 5, Sept. 1998, pp. 562–574

[14] Camacho, E.F., and Bordons, C., Model Predictive Control, Springer, 1996

[15] Wang, X., Yadav, V., and Balakrishnan, S. N., “Cooperative UAV Formation Flying with Obstacle/Collision Avoidance” IEEE Transactions on Control Systems Technology, Vol. 15, No. 4, July 2007, pp. 672–679

[16] Kuchar, J., andYang, L., “Review of Conflict Detection and Resolution Modeling Methods,” IEEETransactions on Intelligent Transportation Systems, Vol. 1, No. 4, Dec. 2000, pp. 179–189

[17] Y. Watanabe, A. Calise, E.N. Johnson, “Vision-Based Obstacle Avoidance for UAVs”, AIAA Guidance, Navigation and Control Conference and Exhibit, August 2007

[18] Park, J. W., Oh, H. D., and Tahk, M. J., “UAV Collision Avoidance Based on Geometric Approach,” SICE Annual Conference, IEEE, Tokyo, Japan, 20–22 Aug. 2008, pp. 2122–2126

[19] J. Gross, R. Rajvanshi and K. Subbarao, "Aircraft Conflict Detection and Resolution using Mixed Geometric and Collision Cone Approaches", AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, Rhode Island, 2004

[20] C. Carbone, U. Ciniglio, F. Corraro, S. Luongo, “A Novel 3D Geometric Algorithm for Aircraft Autonomous Collision Avoidance”, IEEE Conference on Decision & Control, San Diego, USA, December 13-15, 2006

[21] Movarec, H.P. and Elfes, A.,“High Resolution Maps from Wide Angle Sonar”, IEEE Conference on Robotics and Automation, Washington D.C., pp. 116-121, 1985

[22] Borenstein, J. and Koren, Y., “The vector field histogram – fast obstacle avoidance for mobile robots”, IEEE Journal of Robotics and Automation, 1991, vol. 7, n. 3, pp. 278-288

[23] Ulrich, I. and Borenstein, J., “VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots”, IEEE International Conference on Robotics & Automation, Leuven, Belgium, May 1998, pp. 1572-1577

[24] Ulrich, I. and Borenstein, J., 2000, “VFH*: Local Obstacle Avoidance with look-ahead Verification”, IEEE International Conference on Robotics & Automation, San Francisco, CA, April 2000

[25] S.Quinlan, O.Khatib, “Elastic Bands: Connecting Path Planning and Control”, IEEE International Conference on Robotics and Automation, Vol 2, pp. 802-807, Atlanta, USA, 1993

[26] Fox, D., Burgard, W. and Thrun, S., “The Dynamic Window Approach to Collision Avoidance”, IEEE Robotics & Automation Magazine, March 1997, pp. 23-33

[27] R.A.Adams, Calculus : A Complete Course, Fifth Edition, pg 676-678

[28] R. M. Rogers, Applied Mathematics in Integrated Navigation Systems, Second Edition, AIAA Education Series

[29] A. Antoniou, W.-S. Lu, Practical Optimization, Algorithms and Engineering Applications, Springer, 2007

[30] D. N. Kumar, Classical and Advanced Techniques for Optimization

[31] J.A. Snyman, Practical Mathematical Optimization - An Introduction to Basic Optimization Theory and Classical and New Gradient-Based Algorithms

[32] M. Becker, C.M.Dantas, W.P.Macedo, "Obstacle Avoidance Procedure for Mobile Robots", ABCM Symposium Series in Mechatronics, Vol. 2, pp.250-257

[33] J. Moon, J.V.R. Prasad, " Minimum-time Approach to Obstacle Avoidance Constrained by Envelope Protection for Autonomous UAVs "

[34] M. Athans, P. L. Falb, Optimal Control – An Introduction to the Theory and its Applications, Dover Publications, 2007

[35] A. E. Bryson, Y-C. Ho, Applied Optimal Control - Optimization, Estimation and Control, Hemisphere Publishing Corporation, 1975