ROUTE PLANNING FOR UNMANNED AIR VEHICLES

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

BY

KAMİL TULUM

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

SEPTEMBER 2009

ROUTE PLANNING FOR UNMANNED AIR VEHICLES

submitted by **KAMİL TULUM** in partial fulfillment of the requirements for the degree of **Master of Science in Mechanical Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan ÖZGEN ________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Süha ORAL ________________
Head of Department, **Mechanical Engineering**

Prof. Dr. S. Kemal İDER
Supervisor, **Mechanical Engineering Dept., METU** ________________

Dr. Umut DURAK
Co-Supervisor, **TÜBİTAK - SAGE** ________________

Examining Committee Members:

Prof. Dr. Reşit SOYLU ______________________
Mechanical Engineering Dept., METU

Prof. Dr. S. Kemal İDER ______________________
Mechanical Engineering Dept., METU

Prof. Dr. Tuna BALKAN ______________________
Mechanical Engineering Dept., METU

Assist. Prof. Dr. Yiğit YAZICIOĞLU ______________________
Mechanical Engineering Dept., METU

Dr. Umut DURAK ______________________
TÜBİTAK - SAGE

Date: 07.09.2009

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, Last Name : Kamil TULUM

Signature :

ABSTRACT

ROUTE PLANNING FOR UNMANNED AIR VEHICLES

TULUM, Kamil

M.S., Department of Mechanical Engineering

Supervisor: Prof. Dr. S. Kemal İDER

Co-Supervisor: Dr. Umut DURAK

September 2009, 78 pages

In this thesis, automatic routing technologies for unmanned air vehicles are investigated. A route planner that minimizes the fuel consumption and maximizes the survivability is developed. While planning the route, using more than one objective entails the auto-routing problem to multi-objective optimization considerations. In this work, these considerations are handled with search algorithms.

In order to assess the route options, a fuel consumption model and a survivability model are utilized for the route planner. As the assessment models are established, required computational time is taken into account without deteriorating the fidelity.

Keywords: Mission Route Planning, Unmanned Air Vehicles, A* Algorithm

ÖZ

İNSANSIZ HAVA ARAÇLARI İÇİN ROTA PLANLAMA

TULUM, Kamil

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. S. Kemal İder

Ortak Tez Yöneticisi: Dr. Umut DURAK

Eylül 2009, 78 sayfa

Bu tezde, insansız hava araçları için otomatik rotalama teknolojileri araştırılmıştır. Yakıt sarfiyatını azaltıp, uçuş bekasını artıran rota planlayıcı geliştirilmiştir. Rota planlamasında birden fazla kriterin dikkate alınması, oto rotalama probleminde çok amaçlı eniyileme değerlendirmelerinin dahil edilmesine neden olmuştur.

Rota planlayıcı rota seçeneklerini değerlendirilmesinde yakıt sarfiyatı modeli ve uçuş bekası modellerinden yararlanmaktadır. Değerlendirme modellerinin oluşturulmasında sadakati dikkate alacak şekilde işlemler için gerekli zaman dikkate alınmıştır.

Anahtar Kelimeler: Görev Rota Planlaması, İnsansız Hava Aracı, A* Arama Algoritması.

To all whom dedicate their lives to do something that is useful…

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to Prof. Dr. S. Kemal İder for his supervision, patience, encouragement and guidance throughout this study.

I would also like to thank to Dr. Umut Durak, for his co-supervision patience, encouragement and guidance throughout this study.

This work has been encouraged by TÜBİTAK-SAGE. I would like to thank all my colleagues for their support and for helpful discussions we have made throughout the study.

Finally, my deepest thanks go to my family who gave me the endless support and love, which made this thesis possible.

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF SYMBOLS

L	Leg length
R	Turn radius
$A_{\min}$	Minimum flight altitude from AGL
E_{max}	Maximum flight elevation from MSL
LF	Look forward distance
C	Total cost of route
C_{norm_fuel}	Normalized cost of fuel consumption
C_{norm_surv}	Normalized cost of survivability
W_{fuel}	Weighting of fuel consumption
W_{surv}	Weighting of survivability
C_{fuel}	Fuel cost of route
C_{surv}	Survivability cost of trajectory
F_{Drag}	Drag Force
F_{Lift}	Lift Force
W	Weight
T	Thrust

c_L	Lift force coefficient
c_D	Drag force coefficient
T_{idle}	Idle thrust of UAV, lower limit of supplied thrusts
$T_{\max}$	Maximum available thrust of UAV, upper limit of supplied thrusts
T_{motor}	Supplied thrust by the UAV's motor
sfc_{motor}	Specific fuel consumption of UAV's motor
$Fuel_{\max}$	Maximum available fuel of UAV
$D_{\max}$	Maximum flight range of UAV
$Geometric\_Cross\_Section$	Geometric cross section towards the radar, (m^2). It is the size of target viewed from the aspect of radar
$Reflectivity$	Ratio between reflected signal power and its original power
$Directivity$	Maximum directive gain of an antenna relative to the ideal isotropic radiator antenna radiating the same amount of total power
P_r	Received echo signal power

P_t	Transmitted signal power
G_t	Antenna gain
A_e	Antenna effective aperture (area)
R_{trgt}	Range from threat to its target
$R_{Detection}$	Maximum range required between target and threat system in order to detect target with a given RCS, σ
R_{Threat}	Maximum detection range of radar for a target with 1 m^2 RCS

Greek Letters

α	Search sweep angle
β	Search division angle
θ_d	Maximum dive angle
θ_c	Maximum climb angle
ϕ	Maneuver angle at a node
ρ_{air}	Density of air at flight altitude
σ	Radar Cross Section, RCS, (m^2)

CHAPTER 1

INTRODUCTION

1.1 Background

UAV is an unpiloted air vehicle. UAV’s are used for missions which are tedious and risky. UAV’s were developed for military purposes. The most primitive ones were used in American civil war. Balloons loaded with explosive and combustible munitions launched to fall on enemy munitions stores. Although these balloons are not considered as an UAV at first glance, their mission concurs with one of the today’s UAV’s, “cruise missile” mission. Presently, in military, UAV’s are designed for long range annihilation and espionage purposes with high endurance. After the first practice, usage of UAV is extended to civilian applications. For civilian usage, the aims are surveillance against crimes, studies to minimize the hazardous effects of natural disasters, etc. As the UAV utilization advances, technology implemented in the UAV's needs to be improved. Route planning is one of the most annealing subjects drawing much attraction.

Basically, a route is a series of points which link a start location with an end location. For a given area, there will be several alternative route candidates between a start and an end point. Route planning process arises the better one among route options. The adjective term, better, represent different distinguishable attributes of routes according to mission of the route.

Mission can be in a civilian or in a military area. For military purpose, route should have minimum risk of detection by threats. Similarly, in a civilian application where the UAV is passing through a contaminated area the route must have minimum cumulative radiation damage. Also, a route with minimum fuel consumption is usually desired. The shortest route can be demanded, etc. One can expand the list of criteria.

In literature, there are several researches related to this subject. In this thesis instead of discussing all of them at the beginning, they will be adverted throughout the thesis while discussing the individual aspects of the routing problem. One of the referenced works is the research of Gudaitis [1]. He studied the route planning problem from the aspects of more than one criterion. In his thesis, there is a well prepared literature survey. He discusses different search method alternatives used for route planning.

1.2 Motivation and Objectives

Low Observable (LO) technologies are critical for unmanned air vehicles to provide them will the capability to transit through enemy air defense, penetrate to well defended areas and attack high value targets without any engagement from any enemy defense [2].

For UAVs that are usually lack of any active counter measures, LO can only be accomplished either by developing a LO platform that results low RCS or by planning a LO route for the UAV. This thesis aims at the latter alternative. The motivation of this study is to enhance the low observability characteristics of UAVs by planning LO routes for their missions.

The primary objective of this thesis is to develop a route planner that will compute the least observable route with the lowest fuel cost. This will be accomplished by firstly constructing a search network used for route planning. Then the constructed network will be used to search the optimal route for a given start and end location as an output. Optimality of a route is considered in terms of consumed fuel and survivability of UAV against threats while tracking this route.

The other objective is to construct a simulation platform which includes the models of UAV, threats and environment. Model of UAV represents the characteristics affecting the flight trajectory, fuel performance and radar visibility. Threat model includes the threat coverage calculations with target detection capabilities. Environment model includes the usage of geographic coordinates and terrain information.

To construct evaluation model will be another focus to rate a route with respect to fuel consumption and survivability titles.

Lastly a search method is utilized in order to direct the search while seeking the optimal route. The search method should not affect the optimality and it should give admissible results in a short time.

1.3 Thesis Overview

The theory behind the route planning process is reviewed in Chapter 2 in which search space representation, route and trajectory construction are discussed. Also, implementation of the theory in evaluation of a route and models utilized for evaluation are presented in that chapter. Moreover, there is a section related with the search methods in Chapter 2. After this theory chapter, results of sample runs are shown in Chapter 3. Lastly, conclusions are presented in Chapter 4.

CHAPTER 2

METHODOLOGY

This chapter covers the theory behind the route planning process and its implementation in this thesis. Initially, representation of the space is discussed. This representation is important in terms of the navigation during search process. After that, route and trajectory construction processes are presented. Evaluation of a route option in terms of fuel consumption and survivability is given under the Evaluation Route title. Lastly, search methods which direct the search navigation during the search are discussed.

2.1 Space Representation

Space representation is the envisioning of real world's geographic attributes. Space representation is done with using a Geographic Coordinate System. In this coordinate system, a location is defined with latitude and longitude values using WGS84. Altitude value of a location is given as a distance from the Mean Sea Level (MSL). Elevation of a location on the ground is represented with Digital Terrain Elevation Data (DTED) [3, 4]. It gives the terrain elevation values with respect to MSL.

DTED presents the area in grid form. Each grid has an altitude which gives the average of altitude of area in the related grid. The resolution of DTED data varies according to level of the data. As the level of DTED increases the accuracy of the data and the size of it increases. There are 6 levels of DTED data [5]. These are given as the following:

Table 1 DTED Levels and Their Resolutions

Level	Resolution of Grid
0	~1000 m
1	~100 m
2	~30 m
3	~10 m
4	~3 m
5	~1 m

2.1.1 Search Space Representation

In this research, search is performed in 2D search space. Search space is represented by nodes and legs (Figure 2.1). A Node represents a geographic location at which UAV will maneuver or will advance with same azimuth. A Leg represents the way between two nodes. At the end of the search process, the route is computed as a series of nodes.

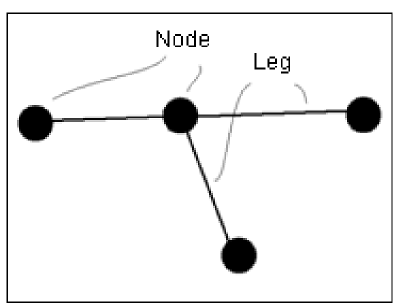

Figure 2.1 Network Elements: Node and Leg

At the search space, there are no previously settled nodes prior to the route planning process. Network is constructed during the search process. Construction of the network begins from the start node and new nodes are sought with respect to presently reached one for each step of the search cycle (Figure 2.2).

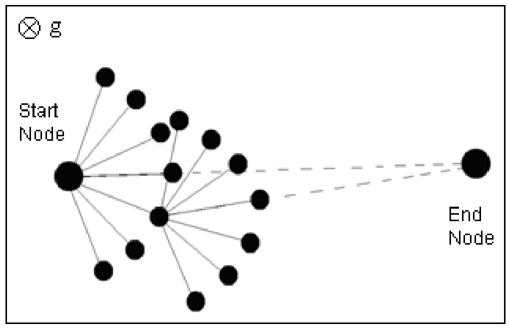

Figure 2.2 Search space

While new nodes are sought from present node and network is constructed, there are several parameters which are taken into consideration (Figure 2.3). These are:

- Leg length, L
- Search sweep angle, α
- Search division angle, β

Following 3 sections discuss L, α and β respectively.

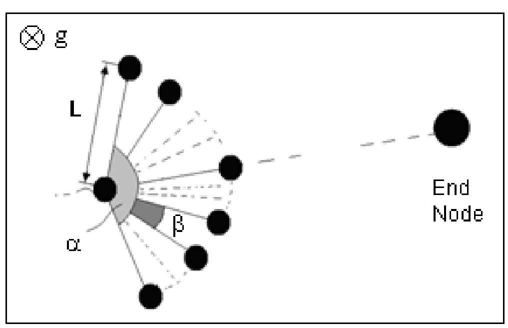

Figure 2.3 Parameters of the search space

2.1.1.1 **Leg Length**

Leg length is the distance to place a new node from current one (Figure 2.3). Its value is set with respect to the UAV's maneuver capabilities. Leg length, L, should be greater than minimum leg length, L_{min}, which represents the minimum distance needed by UAV to perform two consecutive turns. UAV's maneuvers take place with constant turn radius, R. Minimum leg length, L_{min}, and turn radius, R are pictured in Figure 2.4.

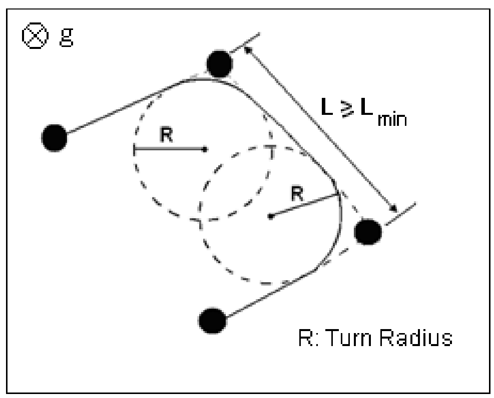

Figure 2.4 Minimum leg length and Turn radius

While searching and constructing the search network, using legs with constant leg lengths constrains the search space and degrade effectiveness of the route planning process in terms of optimality. To decrease this effect and to obtain a variety in the leg size, some extra nodes are used during the search. Those extra nodes are placed along the direction which is formed by the current node and the previous node (Figure 2.5).

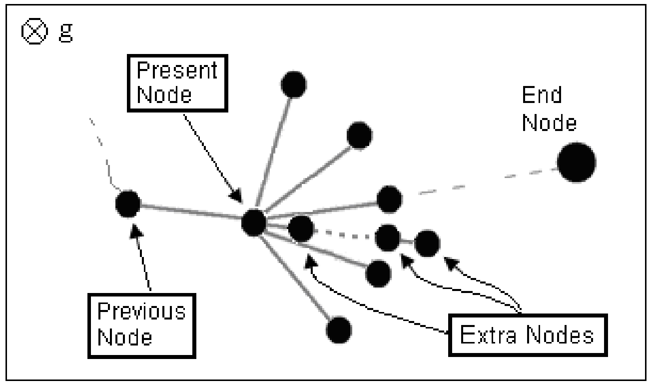

Figure 2.5 Extra nodes

2.1.1.2 **Search Sweep Angle**

Search sweep angle, α, is the span of angle interval to place next nodes from the presently reached one. Its value is limited with the UAV's minimum turn angle, $\gamma_{\min}$ (Figure 2.6). This angle represents the UAV's maneuver limitations. UAV cannot turn sharp angles beyond $\gamma_{\min}$.

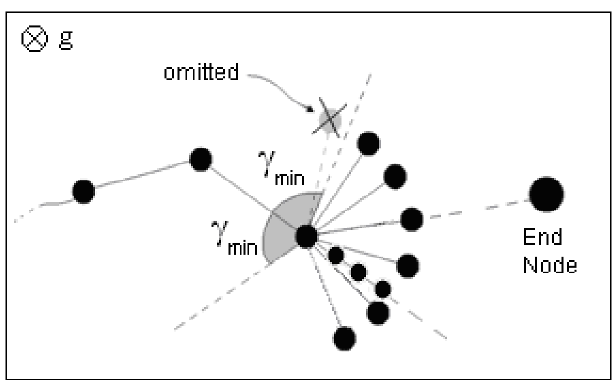

Figure 2.6 Minimum turn angle of UAV

While constructing the search network, the following ruling given in equation (2.1) is taken into consideration for all cases except the start node,

$$\alpha < 2\pi - 2\gamma_{\min} \tag{2.1}$$

where

α : Search sweep angle

$\gamma_{\min}$: Minimum turn angle

2.1.1.3 **Search Division Angle**

Search division angle, β, is the angle between two consecutive legs which are driven from same node. The ratio between search division angle, β, and search sweep angle, α, is related with how deep a search is performed.

Number of new nodes placed as alternative legs from the presently reached node is given as:

$$N_{new\_node} = \left(\frac{\alpha}{\beta} + 1\right) + N_{extra} \tag{2.2}$$

where

N_{new_node} : Number of new nodes from presently reached node

N_{extra} : Number of extra nodes without maneuver from presently reached node

2.2 Route and Trajectory Construction

In this research, the route is constructed over the 2D search space. The third dimension of the route is set according to the terrain which UAV flies over. Reminding that all this routing effort aims at obtaining the low observable route, this research assumes that this route is flown according to a terrain following strategy. UAV performs terrain masking using terrain following as far as its flight performance permits. Outputs of the terrain masking result in 3D trajectory of the route.

Route and Trajectory construction are two consecutive processes that form a pair to evaluate the UAV's navigation. While searching, both construction processes take large times to evaluate alternative route options. In order to obtain a result in a meaningful time, both processes should not include algorithms that require excessive amount of computation time. But, the algorithms should also yield in realistic results at the same time. Therefore, in algorithms, there is a trade-off between computational time and quality of results. In this research, constructions of route and trajectory are done considering this trade of. In the following part, Route and Trajectory Constructions are discussed, respectively.

2.2.1 Route Construction

Search space is composed of nodes which are located during the search process. A Node represents a geographic location on the earth surface without any elevation. While searching, each node in the search space becomes a candidate for being in the route. In Figure 2.7, an example of search space is shown.

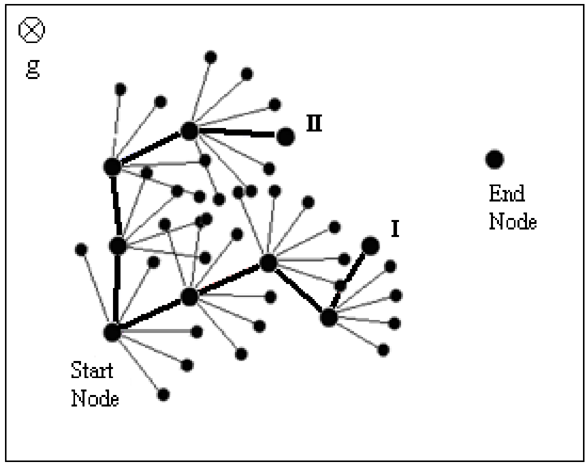

Figure 2.7 Search space and route alternatives

A route is represented as a series of nodes. In Figure 2.7, two routes, I and II, are shown to exemplify how a route is depicted. They are both not completed routes. They are only two route options to reach the End Node.

2.2.2 Trajectory Construction

Trajectory defines the UAV'S flight while it tracks the route. Construction of trajectory is done by using UAV's performance parameters. These parameters are as follows:

- Turn radius, R
- Minimum flight altitude form AGL, $A_{\min}$
- Maximum flight elevation from MSL, E_{max}
- Look forward distance, LF
- Maximum dive angle, θ_d
- Maximum climb angle, θ_c

Turn radius, R, is related with maneuver capability of UAV in horizontal plane. Minimum flight altitude, $A_{\min}$, is the minimum altitude from the above ground level that UAV's trajectory is raised to avoid hitting the ground. Maximum Flight Elevation, E_{max}, is the maximum height from the mean sea level that UAV can fly. Look forward distance, LF, is the distance needed to track between two consecutive vertical maneuvers. Maximum dive angle, θ_d, and maximum climb angle, θ_c, are concerned with UAV's ascending and descending limits. With respect to these parameters, trajectory construction is described below using route "I" given in Figure 2.7.

The route mentioned is composed of 5 nodes (Figure 2.8). At node B, the route continues without a maneuver in the horizontal plane. At nodes C and D, the route has maneuvers. At these nodes, the UAV's trajectory does not pass through the node, but it passes along an arc in the horizontal plane with constant turn radius, R. Figure 2.9 shows how the arc calculation of trajectory maneuver is done. Start and end locations of the arc are found using equation (2.3).

After replacing the corners with arcs, a raw trajectory is obtained (Figure 2.10). The raw trajectory is on the earth surface without elevation.

$$d = R \cdot \tan\left(90 - \frac{\phi}{2}\right) \tag{2.3}$$

where

ϕ : Maneuver angle at node C, $\widehat{BCD}$

d : Distance before the waypoint that maneuver starts.

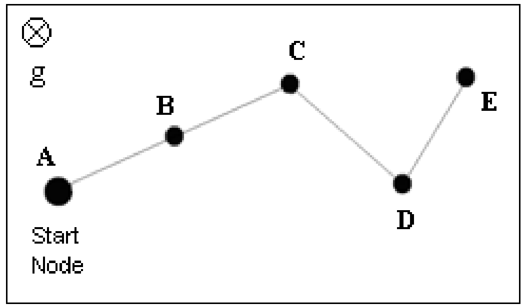

Figure 2.8 Route

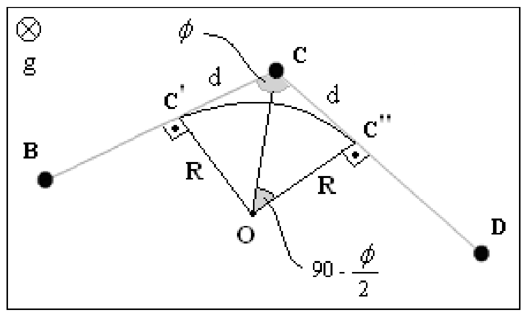

Figure 2.9 Calculation of maneuver on horizontal plane with turn radius, R

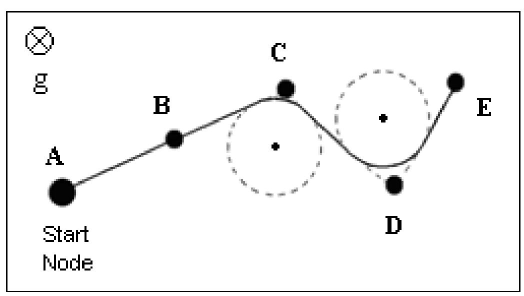

Figure 2.10 Trajectory without elevation

The raw trajectory depicted in Figure 2.10 is composed of continuous straight legs and arcs. It is then put into discrete form to exemplify the terrain below it. Discretization is done by representing the trajectory as a series of subnodes that trajectory passes through (Figure 2.11). A subnode is a geographic location on the earth surface without elevation.

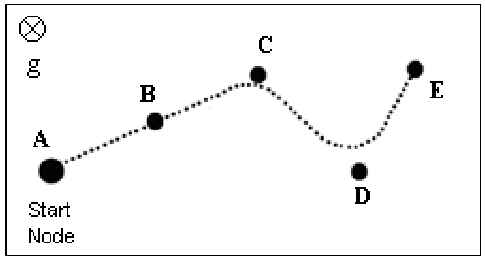

Figure 2.11 Trajectory representation in horizontal plane

Representing the trajectory as a series of subnodes is the end of trajectory construction in horizontal plane. Up to now, geographic coordinates through which the trajectory passes are determined. After that step, elevation calculations start.

Initially, elevation of terrain below the trajectory is obtained from the DTED. According to the value of minimum flight altitude, $A_{\min}$, subnodes are lifted up with respect to terrain elevation (Figure 2.12). For all operations resulting in lifting the trajectory, value of Maximum Flight Elevation, E_{max} is taken into account to avoid rising of trajectory beyond it.

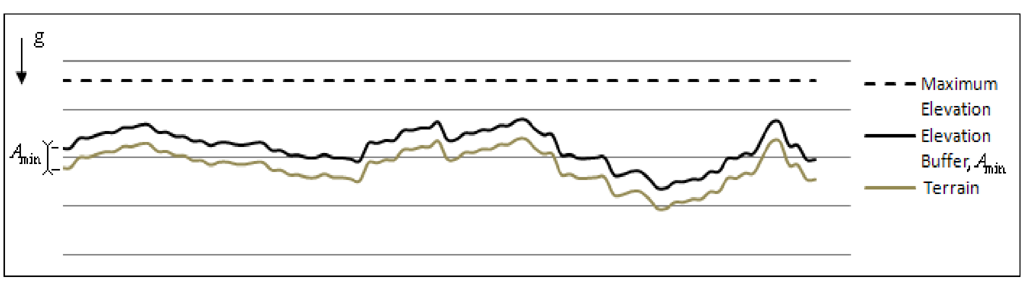

Figure 2.12 Trajectory with minimum flight altitude, $A_{\min}$

Secondly, lifted trajectory is smoothed with Look forward distance, LF , to get rid of consecutive maneuvers that UAV could not track (Figure 2.13).

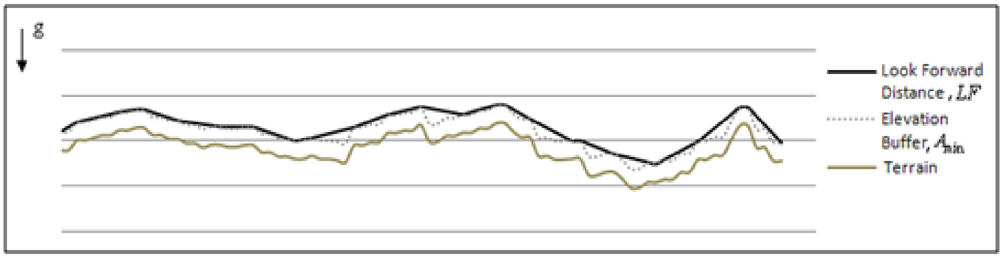

Figure 2.13 Trajectory smoothed with Look forward distance, LF

In the third step, smoothed trajectory is modified with maximum dive angle, θ_d,to avoid sharp descending maneuvers with an angle greater than θ_d.This modification results in raising the end location (Figure 2.14).

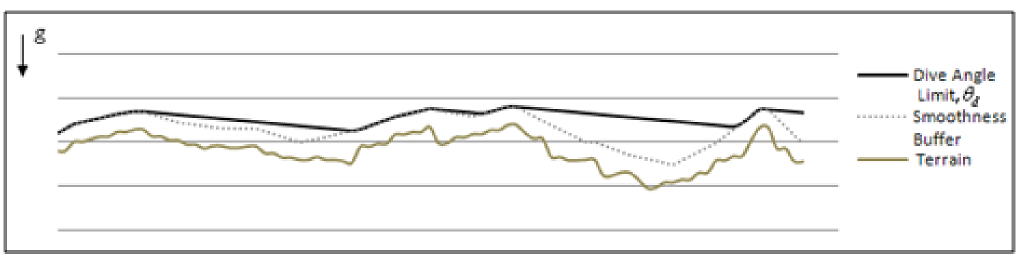

Figure 2.14 Trajectory modified with maximum dive angle, θ_d

Finally, modified trajectory with θ_d is adjusted with maximum climb angle, θ_c,to get rid of sharp ascending maneuvers that UAV is not able to perform (Figure 2.15).

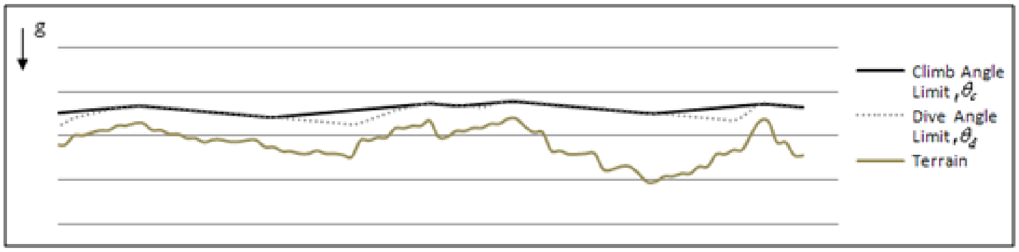

Figure 2.15 Trajectory modified with maximum climb angle, θ_c

At the end of trajectory construction steps, a trajectory that UAV can track with its performance parameters is obtained (Figure 2.16).

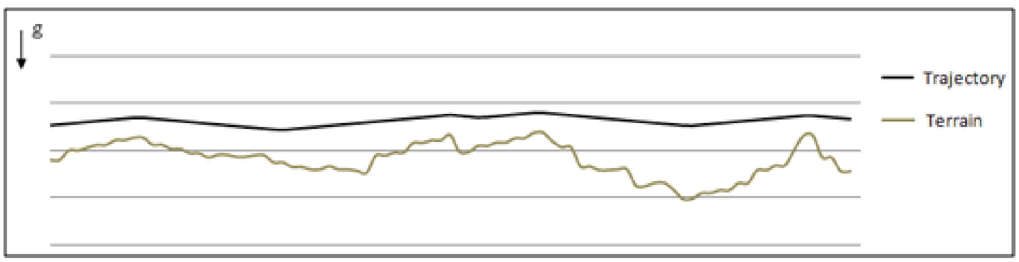

Figure 2.16 Vertical profile of Trajectory

Each step in this construction processes slants in increasing the elevation of subnodes. This introduces a necessity to check whether Maximum Flight Elevation, E_{max} limit is exceeded or not. Exceeding the elevation limit causes elimination of this route.

2.3 Evaluation of Route

The route is evaluated in terms of UAV's fuel consumption and survivability. This evaluation is done using the subnodes of the constructed trajectory. As the required fuel to fly increases, the cost of the route increases and hence the attraction for being an alternative route becomes worse. For survivability case, being detected by the threats makes the route worse and increases the cost.

Fuel consumption is calculated in terms of mass. But, survivability cost is determined with tracked distance by the threat. In order to sum or compare two different quantities, normalized versions of them are required. Therefore, normalization is done before making any cost summations or comparisons.

Evaluation of a route is done by adding the normalized fuel consumption cost quantity and the normalized survivability cost value. This summation determines the whole cost of flying the related route. Actually, according to the mission of the route planning, the relative importance of different costs varies. For example, survivability will be much more essential than consumed fuel for a specific case. Opposite condition will be valid for another case. To reflect this flexibility to the planning process, weighting multipliers are considered in the evaluation function. Therefore, the route is evaluated with respect to weighted sum of normalized cost quantities. This is expressed as equation(2.4).

$$C = W_{fuel} \times C_{norm\_fuel} + W_{surv} \times C_{norm\_surv} \tag{2.4}$$

where

C	: Total cost of route
C_{norm_fuel}	: Normalized cost of fuel consumption
C_{norm_surv}	: Normalized cost of survivability
W_{fuel}	: Weighting of fuel consumption
W_{surv}	: Weighting of survivability

Fuel consumption and Survivability costs are discussed in the following sections, respectively. In these sections, initially, models of fuel consumption and survivability will be introduced, respectively. At the last part of each section, the related cost and its normalization are discussed.

2.3.1 Fuel Consumption

Fuel Consumption of a route is calculated with respect to the computed trajectory that was discussed in Trajectory Construction section. In Figure 2.17, an example trajectory represented by subnodes is given.

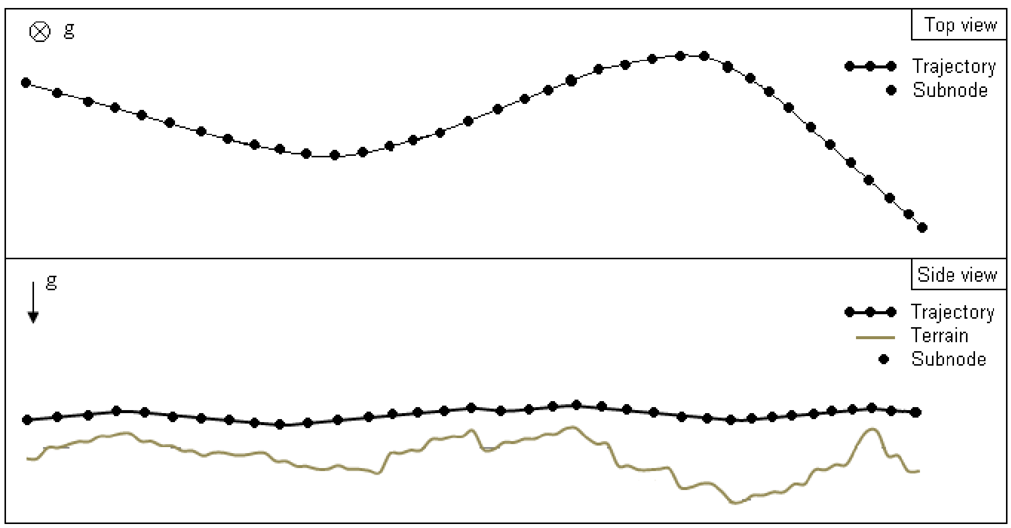

Figure 2.17 Trajectory Example, top view and side view

In fuel consumption computation, initially, the required thrust values to fly between each two consecutive subnodes are calculated. After that, each thrust value is multiplied with specific fuel consumption rate and the distance between the related two nodes. Results of these multiplications present the mass of the consumed fuel between the related consecutive node pairs. Then, all fuel masses so calculated are summed. This summation yields the required fuel to track the route and the fuel consumption cost of this route. Finally, the computed fuel consumption is normalized in order to use it in evaluation of the route.

Calculation of the required thrust between two consecutive subnodes is discussed at Equations of Motion title. Between subnodes, UAV tracks an unaccelerated flight. So, the value of the thrust that is supplied by UAV's motor is equal to the required thrust instantly. If the required thrust goes beyond the UAV's motor's thrust limits, then a deceleration or acceleration will be considered along the flight path. It is significant that while constructing the trajectory, extreme cases are omitted using UAV's maximum dive and climb angles, θ_d and θ_c respectively.

Calculation of fuel consumption cost is discussed in Fuel Consumption Cost title. After this part, normalization of the cost is presented in Fuel Consumption Cost Normalization section.

2.3.1.1 **Equations of Motion**

Free body diagram of UAV is depicted in Figure 2.18. Direction of UAV velocity vector, Flight Path, is inclined with an inclination angle, θ_{UAV}, from the horizontal plane. Chord Line which represents the orientation of UAV with respect to flight path is shown with a geometric angle of attack, α_{UAV}.

Forces acting on the UAV are modeled in 4 main titles:

- Lift force, F_{Lift}.
- Drag force, F_{Drag}
- Weight, W
- Thrust, T

F_{Lift} and F_{Drag} are perpendicular to each other. F_{Drag} is parallel to the flight path. W is towards to the center of the world. T is on the chord line direction.

Lift force, F_{Lift} and Drag force, F_{Drag} are given by the following formulas, respectively:

$$F_{Lift} = \frac{1}{2}.\rho_{air}.S_{UAV}.V_{UAV}^2.c_L \tag{2.5}$$

$$F_{Drag} = \frac{1}{2}.\rho_{air}.S_{UAV}.V_{UAV}^2.c_D \tag{2.6}$$

where

ρ_{air}	: Density of air at flight altitude
S_{UAV}	: Wing reference area
V_{UAV}	: Speed of UAV
c_L	: Lift force coefficient
c_D	: Drag force coefficient

Weight, W, is given as:

$$W = m_{UAV}.g \tag{2.7}$$

where

m_{UAV}	: Mass of UAV
g	: Gravitational acceleration

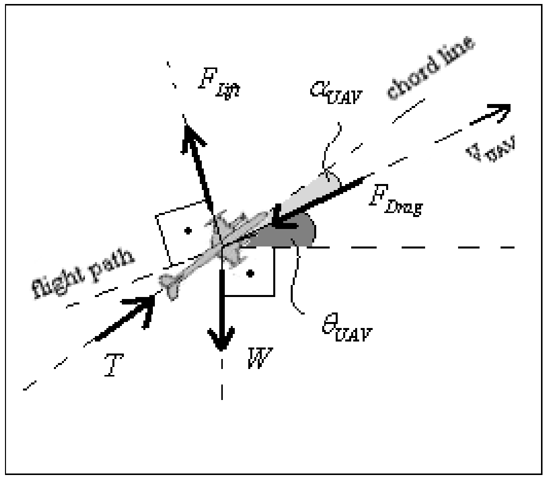

Figure 2.18 Free Body Diagram of the UAV

Free body diagram on UAV is displayed at Figure 2.18. According to this diagram, following force equations are derived:

Summation of forces perpendicular to flight path (positive is upward) is

$$\sum F_{\perp} = F_{Lift} + T.\sin\left(\alpha_{UAV}\right) - W.\cos\left(\theta_{UAV}\right) \tag{2.8}$$

Summation of forces parallel to flight path (positive is right) is

$$\sum F_{\parallel} = T.\cos\left(\alpha_{UAV}\right) - F_{Drag} - W.\sin\left(\theta_{UAV}\right) \tag{2.9}$$

According to Newton’s second law:

$$\sum F_{\perp} = m_{UAV}.\frac{V_{UAV}{}^{2}}{r_{curv}} \tag{2.10}$$

$$\sum F_{\parallel} = m_{UAV}.a_{UAV} \tag{2.11}$$

where

m_{UAV} : Mass of the UAV

V_{UAV} : Speed of UAV

a_{UAV} : Acceleration of UAV

r_c : Radius curvature

Combining (2.8) with (2.10) and (2.9) with (2.11) result in:

$$F_{Lift} + T.\sin(\alpha_{UAV}) - W.\cos(\theta_{UAV}) = m_{UAV} . \frac{V_{UAV}{}^2}{r_{curv}} \tag{2.12}$$

$$T.\cos(\alpha_{UAV}) - F_{Drag} - W.\sin(\theta_{UAV}) = m_{UAV}.a_{UAV} \tag{2.13}$$

At this step the value of geometric angle of attack, α_{UAV}, is assumed to be considerably small, so that magnitudes of $\cos(\alpha_{UAV})$ and $\sin(\alpha_{UAV})$ are taken as 1 and 0, respectively. Also, centrifugal force is neglected. Then reorganization of equations (2.12) and (2.13) yield:

$$F_{Lift} - W.\cos(\theta_{UAV}) = 0 \tag{2.14}$$

$$T - F_{Drag} - W.\sin(\theta_{UAV}) = m_{UAV}.a_{UAV} \tag{2.15}$$

In equation (2.14) F_{Lift} is taken left and Lift force is found as:

$$F_{Lift} = W.\cos(\theta_{UAV}) \tag{2.16}$$

In equation (2.15) T is taken left and Thrust is found as:

$$T = F_{Drag} + W.\sin(\theta_{UAV}) + m_{UAV}.a_{UAV} \tag{2.17}$$

For an unaccelerated flight, Thrust is:

$$T = F_{Drag} + W.\sin(\theta_{UAV}) \tag{2.18}$$

2.3.1.2 **Fuel Consumption Cost**

The approach used to calculate the fuel required to track the route is already discussed at Fuel Consumption title. The physics used in this approach is presented at Equations of Motion section. In this section, the implementation of this approach with its physics and determination of the fuel consumption route cost will be presented.

To model the flight, a state definition is made for UAV. This defined state is used to hold the values of the related parameters of UAV's flight for fuel consumption calculations. Between each two consecutive subnode pairs of trajectory, state of UAV is updated.

State of UAV includes the following parameters:

- m_{UAV} : Mass of UAV
- V_{UAV} : Speed of UAV
- h_{UAV} : MSL(Mean Sea Level) Altitude of UAV
- θ_{UAV} : Inclination angle of UAV
- $Dist_{UAV}$: Travelled distance

m_{UAV} is used to determine acceleration rates of UAV. V_{UAV} is used for the calculation of Lift and Drag Forces. h_{UAV} is used to find density of standard air, and thrust limits of UAV's motor. θ_{UAV} is used to determine required angle of attack,' α_{UAV}', and corresponding and lift – drag force coefficients, $c_L - c_D$. $Dist_{UAV}$ is used to find amount of consumed fuel while multiplying it with fuel consumption rate.

Initially, the state variables are set with UAV's default initial conditions. m_{UAV} is equal to the full mass of UAV. V_{UAV} is set with default flight speed of UAV. h_{UAV}

gets the altitude value of first subnode of trajectory. $Dist_{UAV}$ and θ_{UAV} are calculated with respect to first and second subnode positions. $Dist_{UAV}$ is the geometric distance between given first two subnodes. θ_{UAV} is the angle that results from arctangent of altitude difference verses lateral distance between given subnodes.

After setting the initial state, the required lift force, F_{Lift}, is determined from equation (2.16). From found, F_{Lift}, lift force coefficient, c_L, is obtained from equation (2.5). In order to find, c_L, required value of air density, ρ_{air}, is set with respect to MSL altitude of UAV, h_{UAV}. Reference area, S_{UAV} is specific to the UAV and it is constant.

From c_L value, the corresponding geometric angle of attack, α_{UAV}, is determined in order to find the related force coefficients, c_D. This α_{UAV} determination and related c_D finding are done α_{UAV} versus c_L and α_{UAV} versus c_D tables, respectively. These tables are specific to UAV model.

After obtaining c_D, the Drag Force, F_{Drag}, can be calculated using equation (2.6). In equation (2.6), the values of variables ρ_{air} and S_{UAV} are same as the values previously used in equation (2.5).

F_{Drag} found above is used to calculate the required Thrust, T. As it is decelerated before, UAV tracks an unaccelerated flight. With respect to this supposition, required thrust is calculated with equation (2.18).

Calculated Required Thrust, T, should be in the range of UAV's motor's thrust limits. Thrust limits of the motor can be denoted as:

- T_{idle} : Idle thrust of UAV, lower limit of supplied thrusts.
- $T_{\max}$: Maximum available thrust of UAV, upper limit of supplied thrusts.

Magnitudes of T_{idle} and $T_{\max}$ are related with UAV's flight altitude, h_{UAV}. For each state calculation, both of them are updated with respect to h_{UAV} from the motor characteristics table.

If the required thrust value, T , from the equation (2.18) is in the bounds of the motor thrust limits, then motor runs with calculated thrust, T .

On the other hand, if the thrust value found from equation (2.18) is lower than the idle thrust, T_{idle}, then UAV's motor runs with T_{idle} ($T = T_{idle}$) and UAV accelerates. If the thrust value found from equation (2.18) is higher than the maximum available thrust, $T_{\max}$, then UAV's motor runs with $T_{\max}$ ($T = T_{\max}$) and UAV decelerates. For both of these two cases, acceleration value, α_{UAV}, is found from equation (2.17) . The value of thrust, T , in equation (2.17), is set to T_{idle} and $T_{\max}$, respectively. The acceleration value so found is used to calculate new velocity of UAV for next consecutive subnodes pairs. The model is described below ($acc = a_{UAV}$, $V_{present} = V_{UAV}$, $dist = Dist_{UAV}$):

$$V_{new}^2 = V_{present}^2 + 2.acc.dist \tag{2.19}$$

where

acc : Acceleration

V_{new} : New velocity

$V_{present}$: Present Velocity

$dist$: Distance

If the required thrust value, T , from equation (2.18) is in the bounds of the motor thrust limits but UAV's flight velocity, V_{UAV} , is not equal to default flight velocity, then T will be adjusted in order to justify the value of velocity with its default value. If that is the case, the required acceleration value, a_{UAV} is found with reorganized version of (2.19) ($acc = a_{UAV}$, $V_{new} =$ Default Flight Velocity, $V_{present} = V_{UAV}$, $dist = Dist_{UAV}$):

$$acc = \frac{V_{new}^2 - V_{present}^2}{2.dist} \tag{2.20}$$

a_{UAV} found above is used to calculate the required thrust value, T , using equation (2.17). The previously found thrust is updated with this newly found one. It is significant that, the new thrust value should be within motor's thrust limits. Otherwise, a new acceleration value has to be calculated using the steps discussed at previous paragraphs related to the thrust beyond the motor's thrust limits.

At the end of all these steps, the thrust to be supplied by the UAV's motor, T , is determined. This T is used to calculate the amount of consumed fuel, m_{fuel} , between given subnodes. This calculation is done with the following formula:

$$m_{fuel} = T_{motor}.sfc_{motor}.Dist_{UAV} \tag{2.21}$$

where

m_{fuel} : Consumed fuel to fly $Dist_{UAV}$ distance with thrust T_{motor}

T_{motor} : Supplied thrust by the UAV's motor

sfc_{motor} : Specific fuel consumption of UAV's motor

$Dist_{UAV}$: Flight distance (Distance between two consecutive subnodes)

At the end, the mass of the fuel consumed between two consecutive subnodes pair, m_{fuel}, is obtained. Mass of UAV and Velocity of UAV for next state are updated according to the found m_{fuel} and a_{UAV}, respectively.

Summation of fuel consumptions between each consecutive subnodes pairs in the trajectory gives the required fuel in order to track the given route.

$$M_{Fuel\_route} = \sum_{\text{Trajectory subnodes pairs}} m_{fuel} \tag{2.22}$$

where

M_{Fuel_route}	: Total required fuel for tracking the route
m_{fuel}	: Consumed fuel between two consecutive subnode

During the calculations, if fuel consumption exceeds the UAV's maximum available fuel, then the corresponding route is eliminated. Similarly, if the flight velocity goes beyond the UAV's tolerable velocity limits, $V_{\min}$ and $V_{\max}$, then same conclusion is made: Route is out of use. It won't be an alternative.

As a result, the required fuel mass in order to track the route represents the fuel consumption cost of this route. While evaluating the route options, normalized version of it is used. Normalization of fuel consumption is discussed in the following section.

$$C_{fuel} = M_{Fuel\_route} \tag{2.23}$$

where

C_{fuel}	: Fuel cost of route
M_{Fuel_route}	: Required fuel for tracking the route with found trajectory

2.3.1.3 **Fuel Consumption Cost Normalization**

Fuel cost is normalized with maximum available fuel of UAV.

$$C_{norm\_fuel} = \frac{C_{fuel}}{Fuel_{\max}} \tag{2.24}$$

where

C_{norm_fuel} : Normalized fuel cost

C_{fuel} : Fuel cost of route

$Fuel_{\max}$: Maximum available fuel of UAV

2.3.2 Survivability

At route planning studies, survivability title is considered under several different titles with different levels of detail [1, 2, 6, 7,]. This section deals with modeling of threats and their effects on the route planning.

Actually, threat modeling includes electromagnetic theories and probability calculations. In this research, the model considers only the factors which are significant for route planning. This approach is commonly seen in other works similar to this one as well. Because detailed modeling causes excessive computational workloads. For instance, Gudaitis [1] gives some example works which consider the aircrafts as spheres in terms of radar cross section values. This assumption yields the calculations simpler and avoids the necessity of considering the orientation of the aircraft. Myron[2] discusses survivability from a slightly different point of view. Myron studies route planning process for low observable aircraft and cruise missiles in his work. He mentions threat penetration analysis with route planning during development and test case, and with route planning in an operational environment. Also, he makes considerations on the representation of threat with respect to fidelity of data.

At the next sections, survivability is considered with Radar Cross Section (RCS) Model of UAV and Threat Modeling titles. Threat Modeling consideration is divided into subparts. These are:

- Line of Sight Analysis,
- Radar Equation
- Survivability Cost
- Survivability Cost Normalization

2.3.2.1 **Radar Cross Section (RCS) Model of UAV**

Radar Cross Section (RCS) is defined as a measure of the reflective strength of a radar target [8]. RCS is related to the reflected portion of signals coming from the radar. In literature, it is also called as Scattering Cross Section [10].

Radar Cross Section (RCS) of an object, shown with symbol "σ" is a function of its geometric cross section, reflectivity, and directivity [9].

$$\sigma = Geometric\_Cross\_Section \times Reflectivity + Directivity \tag{2.25}$$

where

σ : Radar Cross Section, RCS, (m2)

$Geometric\_Cross\_Section$: Geometric cross section towards the radar, (m2). It is the size of the target viewed from the aspect of the radar.

$Reflectivity$: Ratio between reflected signal power and its original power

Directivity : Maximum directive gain of an antenna relative to the ideal isotropic radiator antenna radiating the same amount of total power.

RCS values of an object are computed in 3D with offline measurements and analysis. For this research, two dimensional RCS data is used (Figure 2.19). This data represents the radar cross section characteristics of the UAV.

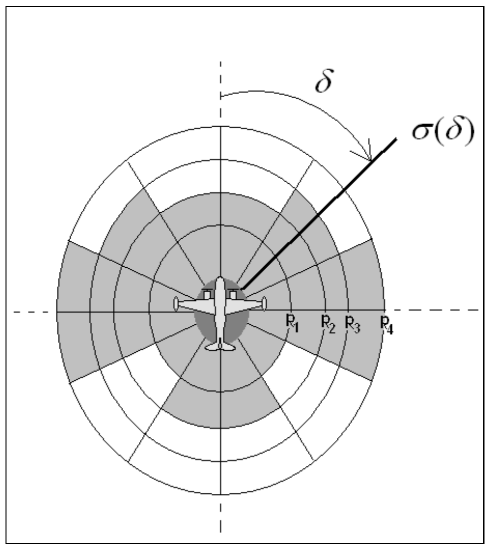

Figure 2.19 Radar Cross Section (RCS) of UAV

2.3.2.2 **Threat Modeling**

For threat modeling, UAV is considered as a target by the threats. The threats are defined with the following parameters (Figure 2.20):

- Maximum detection range of radar for a target with 1 m^2 RCS, R_{Threat}. Detailed consideration is given at Radar Equation title.

- Location
    - Threat latitude, Lat_{Threat}
    - Threat longitude, Lon_{Threat}
    - Threat altitude from above ground level, Alt_{Threat}

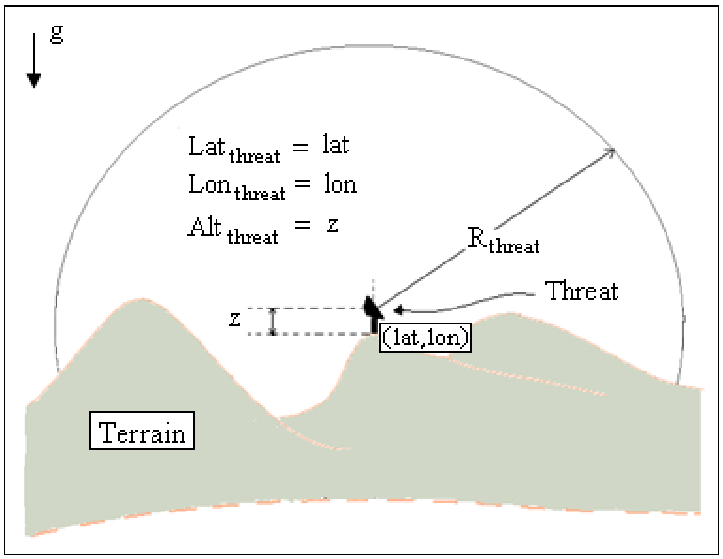

Figure 2.20 Threat Parameters

Threats' Location information is used for Line of Sight Analysis in order to find whether there is any impediment to see the target (UAV) because of the terrain or not. Detection radius is used for radar equation to find whether threat detects the target (UAV) or not.

2.3.2.2.1 Line of Sight Analysis

Line of sight analysis is done using the given threat and target locations and the terrain DTED between these two locations. As an example, UAV1 in Figure 2.21 is in threats' line of sight. But UAV2 in the same figure cannot be seen by the threat because the terrain obstructs threats waves in terms of transmission and reception.

To perform line of sight analysis, threat coverage is calculated with respect to DTED information of terrain around the threat location. This calculated coverage is tabulated and saved for utilization in route planning.

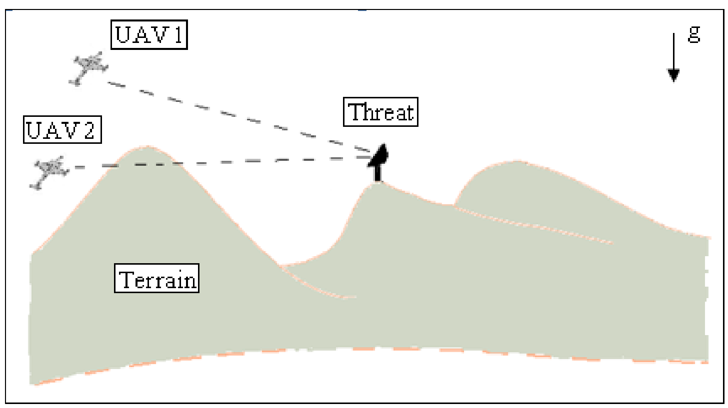

Figure 2.21 Line of sight analysis

2.3.2.2.1.1 Threat Coverage

Threat Coverage which is a two dimensional table holds the line of sight analysis of the threat. First dimension (rows) represents the azimuth values from the north. Second dimension (columns) stands for the range values from the threat location.

There are 'm' many rows and 'n' many columns where:

$$m = \frac{2\pi}{d\theta} \tag{2.26}$$

$$n = \frac{R_{threat}}{dr} \tag{2.27}$$

$d\theta$ and dr are given in Figure 2.22.

Values in tabulated threat coverage table stand for the required above ground level altitudes of geographic locations in order to be seen by threat. Geographic locations are obtained by the table azimuth and range indices. As an example, altitude designations are presented in Figure 2.23 for locations between point 'A' and threat, given in Figure 2.22. As, it is seen in Figure 2.23, seen altitude values of locations are obtained with a step size equal to dr. At some locations, ground surface can be seen directly without any terrain interruption ($h_1 = h_2 = h_3 = \ldots = h_{r-1} = h_r = 0$). But, for the other locations, altitudes from the ground surface should be greater than 0 to be seen by the threat ($h_n > 0$, $h_{n-1} > d0$)

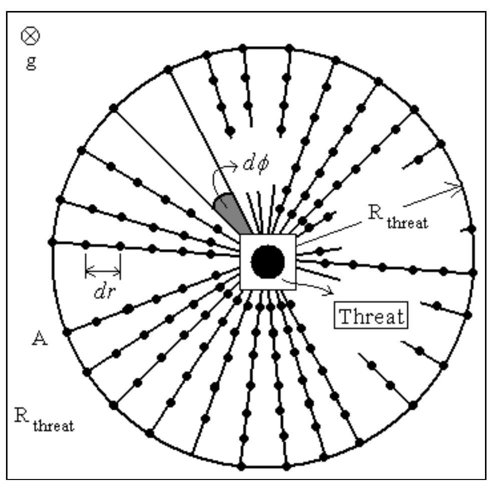

Figure 2.22 Threat Coverage

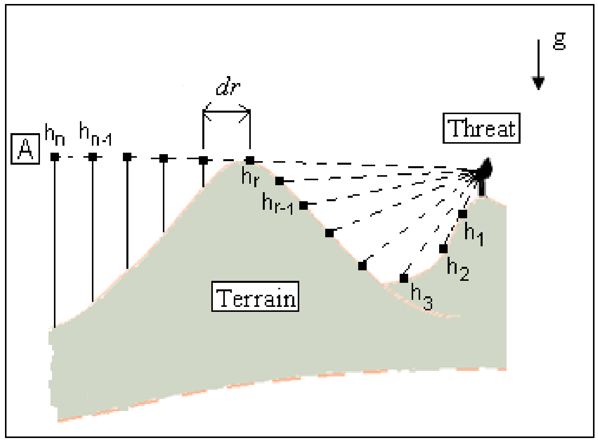

Figure 2.23 Radar Coverage Altitudes

2.3.2.2.2 Radar Equation

Radar Equation is composed of parameters that affect the performance of radar. In literature, Radar Equation is called as Radar Range Equation.

General Radar Equation [8] is given by the following equation:

$$P_r = \frac{P_t.G.A_e.\sigma}{(4\pi)^2.R_{trgt}{}^4} \tag{2.28}$$

where

P_r	: Received echo signal power
P_t	: Transmitted signal power
G_t	: Antenna gain
A_e	: Antenna effective aperture (area)
σ	: Radar Cross Section, RCS, of target
R_{trgt}	: Range from threat to its target

For a radar, in order to detect a target, P_r, should be greater than a threshold value, $P_{r\_Detection}$. In this research, R_{Threat} is used as a maximum radar range value which radar can receive $P_{r\_Detection}$ from a target with a 1 m^2 RCS value. If the range between radar and target with 1 m^2 is smaller than R_{Threat}, then P_r value will be higher than $P_{r\_Detection}$ (i.e. threat will also be detectable for the radar).

$$P_{r\_Detection} = \frac{P_t.G.A_e.1}{(4\pi)^2.R_{Threat}{}^4} \tag{2.29}$$

At equation (2.28), target related parameters are R_{trgt} and σ. The other ones are not directly relevant with target and its position.

When, equation (2.28) is reorganized and both R_{trgt} and σ is taken to the left, hand side, the following equation is obtained:

$$\frac{R_{trgt}}{\sigma^{0.25}} = \left(\frac{P_t.G.A_e}{(4\pi)^2.P_r}\right)^{0.25} \tag{2.30}$$

For the limiting threshold power value, $P_{r\_Detection}$, the right hand side of equation (2.30) will be constant.

$$C_{Radar\_Detection} = \left(\frac{P_t.G.A_e}{(4\pi)^2.P_{r\_Detection}}\right)^{0.25} \tag{2.31}$$

$$\frac{R_{Detection}}{\sigma^{0.25}} = C_{Radar\_Detection} \tag{2.32}$$

where

$R_{Detection}$: Maximum range required between target and threat system in order to detect target with a given RCS, σ.

For 1 m^2 RCS, σ=1,

$$R_{Threat} = R_{Detection} = C_{Radar\_Detection} \tag{2.33}$$

When equations (2.32) and (2.33) are combined, and $R_{Detection}$ is taken to the left hand side, one obtains the following equation:

$$R_{Detection} = \sigma^{0.25}.R_{Threat} \tag{2.34}$$

2.3.2.3 **Survivability Cost**

As fuel consumption cost calculations, survivability cost of a route is considered on subnode representation of related route's trajectory (Figure 2.26). Derivation of trajectory's subnode representation is discussed at Trajectory Construction title.

Survivability cost calculations include radar equation consideration with RCS comparison, and line of sight consideration. For each node, these considerations give the answer of whether subnode's location is seen from the subnode's location or not. If a subnode is seen, then survivability cost of the route is increased. The increment value of cost is determined with respect to visibility conditions of previous subnodes for the same threat.

Visibility of i^{th} subnode (Figure 2.24) with respect to a threat is determined as the following steps:

> Step 1: Angle, δ, between the direction of UAV and the line which connects i^{th} subnode's location with threats location on the horizontal plane is found. (Figure 2.24, top view)

Step 2: For the angle, δ, RCS value of UAV, σ, is determined. This determination is discussed at Radar Cross Section (RCS) Model of UAV title.

Step 3: Using RCS value, σ, maximum range required between target and threat in order to detect the UAV, $R_{Detection}$, is calculated with equation (2.34)

Step 4: $R_{Detection}$ is compared with the range between UAV and threat, R (Figure 2.26, top view). If $R_{Detection}$ is smaller than R, then UAV is not seen by threat. Otherwise, there is a possibility of being seen by threat.

Step 5: Whether UAV is seen or not is revealed with line of sight analysis. According to the comparison between minimum altitude value from the ground in order to be seen, $h_i^{'}$, and flight altitude, h_i. If h_i is greater than $h_i^{'}$, then UAV is seen by threat. Otherwise, it is not seen because of the terrain blockage.

In Figure 2.25, same trajectory is depicted with 3 different threat visibility cases. Although the number of seen subnodes, 4, is equal for each case, their order is not same. For case I, the seen nodes are aligned consecutively. For case II, continuity is disrupted once. For case III, there is no consecutiveness for the seen subnodes. For threats, tracking its targets without any interruption makes easy to acquire the targets location and attack it. Therefore, from the point of view of UAV's (targets of threats), survivability cost values comparisons for cases given in Figure 2.25 is as the following:

$$C_{surv-I} > C_{surv-II} > C_{surv-IIII} \tag{2.35}$$

where

C_{surv-I}, $C_{surv-II}$, $C_{surv-III}$ are the survivability costs of trajectory cases I, II, III given in Figure 2.25, respectively.

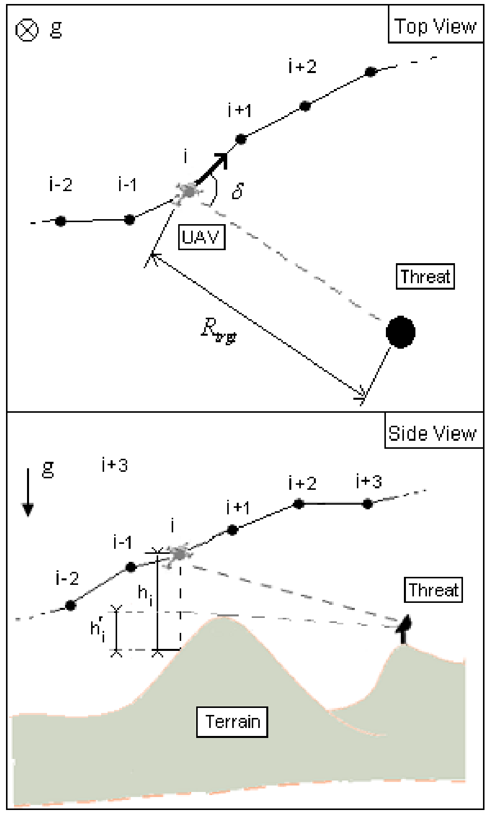

Figure 2.24. Survivability consideration with subnodes of trajectory

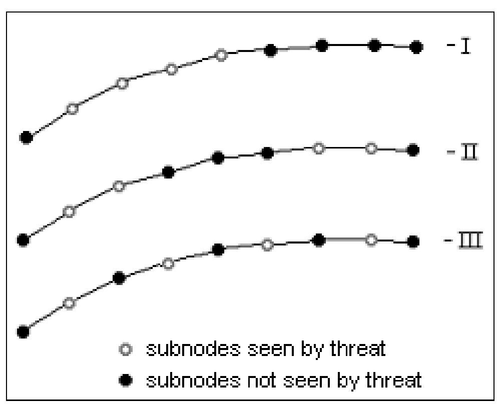

Figure 2.25 Trajectories with threat visibility

According to above considerations, survivability cost of a trajectory against a single threat is formulized with the following equation:

$$C_{surv} = \sum_{i=0}^{N-1} c_i . d_{i,i+1} . V_i \tag{2.36}$$

where

C_{surv} : Survivability cost of trajectory for a single threat

c_i : Multiplier of ith subnode due to tracking of threat

$d_{i,i+1}$: Distance between i^{th} subnode and next subnode

V_i : Visibility of i^{th} subnode. If subnode is seen by threat, it is 1. Otherwise, it is 0.

N : Number of subnodes that represent the trajectory.

For multiple threat cases, each threat is considered independently and each corresponding survivability cost is summed.

$$C_{surv} = \sum_{k}^{M} \sum_{i=0}^{N-1} c_{i,k} . d_{i,i+1} . V_{i,k} \tag{2.37}$$

where

C_{surv} : Survivability cost of trajectory

$c_{i,k}$: Multiplier of i^{th} subnode due to tracking of k^{th} threat

$d_{i,i+1}$: Distance between i^{th} subnode and next subnode

$V_{i,k}$: Visibility of i^{th} subnode with respect to k^{th} threat. If subnode is seen by k^{th} threat, it is 1. Otherwise, it is 0.

N : Number of subnodes that represent the trajectory.

M : Number of threats.

As a result, calculated survivability cost of trajectory represents the survivability cost of related route. While evaluating the route options, normalized version of it is used. Normalization of survivability cost is discussed in the following section.

2.3.2.4 Survivability Cost Normalization

Survivability cost is normalized with maximum flight range of UAV.

$$C_{norm\_surv} = \frac{C_{surv}}{D_{\max}} \tag{2.38}$$

where

C_{norm_surv} : Normalized survivability cost

C_{surv} : Survivability cost

$D_{\max}$: Maximum flight range of UAV

2.4 Search Methods

2.4.1 Literature

Gudaitis [1] groups methods used in route planning in to two titles as deterministic methods and stochastic methods:

- Deterministic Methods
    1. Breadth First
    2. Depth First
    3. Best First
    4. Dynamic Programming
- Stochastic Methods

1. Potential Fields
2. Genetic Algorithms
3. Simulated Annealing
4. Monte Carlo and Las Vegas

For deterministic methods, Gudaitis emphasizes on its predictable and repeatable search behavior for a given set of inputs. On the other hand, probabilistic sides of stochastic methods result in different outputs for the same set of inputs. By using deterministic methods, it is possible to find optimal results, but stochastic methods do not guarantee optimality [1].

The claim on non guaranteed optimality with stochastic methods will be discussed from the aspect of route planning. In theory, stochastic methods will also find optimal results if sufficiently much possibility is taken into account. But in reality, theoretically possible results will not be practical when the required computation time is considered [1].

2.4.1.1 **Deterministic Methods**

Deterministic methods guarantee to give optimal results if enough computational time is present. They are differentiated between each other in terms of their solution approaches and required computational time – work load.

In the following four sections, deterministic methods are discussed.

2.4.1.1.1 Breadth First Search and Depth First Search

Breadth First Search and Depth First Search algorithm have same characteristics in terms of their solution approach. They are both “Uninformed Graph Search Algorithm”. While searching, algorithms do not make use of the information beyond the currently spanned part of the search area. Skiena [11] explains the difference between the Breadth First Search and Depth First Search as the exploration order of zones in the spaces.

Breadth First Search Algorithm directs its search with a uniform progress at all solution alternatives. It intends to look for a result while monitoring every solution alternative and directing the search from reached zones to not accessed ones. Moreover, Skiena [11] uses an expression: “undirected graph” while discussing the breadth first search algorithm. He utters that in a breadth first search of an undirected graph, direction is assigned as from discovered state to undiscovered one.

Breadth First Search Algorithm examines the nodes in search area systematically with their chronologically reaching order (Figure 2.26).

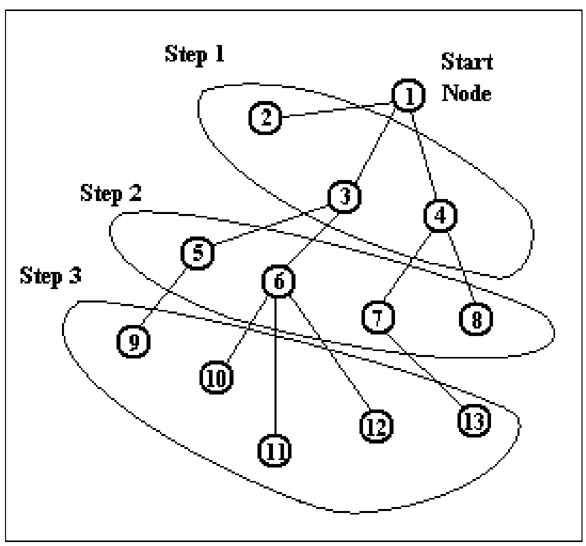

Figure 2.26 Breath First Search

At first step, the algorithm looks for neighbors of start node in a systematic way. Then, all neighbors of start node are examined with the same systematic manner.

Search process continues with looking for neighbors to each of the newly found nodes. Search is ended when end node is reached.

Breadth First Search does not give any guaranty to find the best route between any two locations. To guarantee the best route, it needs examining all route alternatives in the search area. This fact is a significant drawback for Breadth First Search to use in route planning processes. Because, searching all space results in an excessive amount of computational work load. But, the algorithm can yield the optimal result for some special cases. For example, if the cost between two consecutive nodes is constant for all node pairs in the search area, then it is possible to guarantee that the found route is the best route available in the search area.

Although both Depth First Search Algorithm and Breadth First Search Algorithm is uninformed search, their methodologies are different. Depth First Search Algorithm proceeds deeper on one route alternative at each step, instead of examining each of present route alternatives (Figure 2.27).

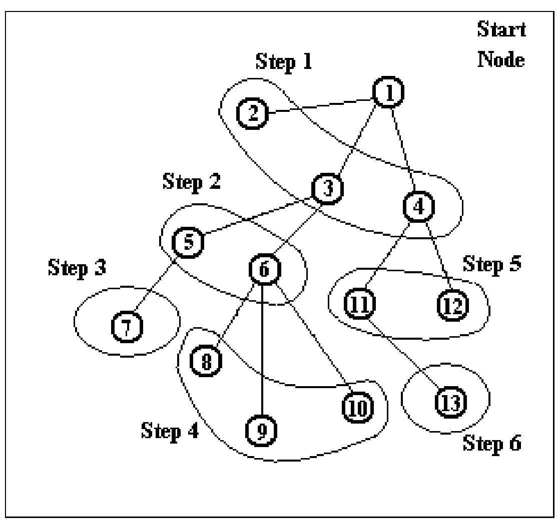

Figure 2.27 Depth First Search

In Depth First Search Algorithm, proceeding on one of the route options continues until new neighbors cannot be found for the proceeded route. If that is the case, then another route option is followed to reach the end node. This search process stops when end node is reached.

Similar to Breadth First Search, Depth First Search does not give guaranty to find the best route without searching the whole search area. So, Depth First Search is also not convenient for route planning.

2.4.1.1.2 Best First Search

Different from breadth first and depth first searches, in literature best first search is discussed under the Informed Search title [12]. Best First Search Algorithm is defined as an algorithm that directs the search from the most convenient node of presently reached ones. Node's "most convenient" property is determined in terms of previously specified rules.

In literature, there are several versions of best first search algorithm in terms of their perceptions of the term, "best" and its usage. These versions are called with different names. Common types of best first search algorithm are: Greedy Best First Search Algorithm, Beam Search Algorithm, Dijkstra's Algorithm, and A* (called as "A star") Algorithm.

Jones [12] discusses Greedy Best First Search and Beam Search algorithms as the variants of best first search. He dedicates a separate title for A*. Narashimhan and Smid [13] discussed the Dijkstra's Algorithm under the Dijkstra's Shortest Path Algorithm title.

Before discussing the methods, definition of a general term related to the best first search will be given. This term is "Heuristic". Moustakas [14] gives the basic definitions of heuristic. Heuristic comes from the Greek word "Heuriskein". Meaning of "Heuriskein" is to discover or to find. Moustakas says that this meaning refers to *"a process of internal search through which one discovers the nature and meaning of experience and develops methods and procedures for further investigation and analysis"*. For route planning case, heuristic is used to orient the searching process towards the goal state. The aim of this orientation is to reduce search time.

Both Greedy Best First Search and Beam Search algorithms direct their searches with an objective function. Both of them use the heuristic as a single factor to determine their objective functions. Preferred heuristic gives an idea to predict how close the end of a path is to a solution. But, since the heuristic is used only for objective function, both algorithms will not give the realistic results if the heuristic values are not admissible [12]. The difference between Greedy Best First Search and Beam Search is that Beam Search uses only a limited set of candidate solutions to continue the searching. This set is composed of best solution alternatives. Eliminating worse candidates and not using all ones make the Beam Search more efficient to reach the goal state. But, it is possible that the real optimum solution may be disregarded while searching.

Both Dijkstra's Algorithm and A* Algorithm uses similar searching approaches. Each of them uses the lowest cost solution alternative to direct the search. For Dijkstra's Algorithm, cost is calculated from the start to reached end. For A* algorithm, cost value consists of a summation of two separate quantities. First quantity is same with the cost in Dijkstra's Algorithm. Second quantity is the heuristic cost from the reached end to goal state. With respect to cost calculations, Dijkstra's Algorithm can be considered as an A* Algorithm with heuristics equal to zero in value. In this research, A* Algorithm is chosen as a search method. So, detailed consideration is given in a separate section.

2.4.1.1.3 Dynamic Programming

Dynamic Programming is defined as a formulation of Breadth First Search that provides expressions in analytical form for optimal cost or optimal policy [1]. In Dynamic Programming, problem solution approach is based on division of the problem into subparts and combination of those subpart's solutions that are got separately.

The aim of the division is to shorten the computational time and to relieve the work load. This is because of that division into subparts prevents recalculations of same problems that are encountered multiple times.

Thomes H. Cormen [15] remarks about the assembling of results which are found as the solutions for subdivided parts of the main problem. With this explanation, he associates dynamic programming with Divide and Conquer algorithm. Divide and Conquer method defined as a method that divides the problem into independent parts and combines the solutions of these independent parts. Different from Divide and Conquer method, Dynamic Programming can solve the problems easily, even if the subparts are not independent from each other. Thomas thinks that this facility of Dynamic programming comes from saving and tabulating the previously solved problem's answers. Saving previous results avoids recomputation of the same problems. Same property is discussed at the work of Yanhong [16]. Yanhong refers to two terms: "Memoization" and "Tabulation". Memoization is described as saving each solution found and hence shortening the computation time of repeatedly solved problems. Tabulation is defined as the statically determination of required table for the calculations. Yanhong [16] discusses affirmative and privative sides of this method.

Gudaitis [1] gives DARTII, Desktop Auto Routing Tool II, as an example of a tool that uses Dynamic Programming method. Moreover, referring to the reported results, Gudaitis declares that for autonomous under water vehicles, Dynamic

Programming's performance is better than that of heuristic algorithms, such as A* Algorithm. Dynamic programming needs less memory and computations takes less time. In spite of this fact, Gudaitis makes remark in terms of this comparison. He points that performance of both methods relies on different sources. Dynamic Programming's performance is related with the optimality of substructure of path problem. But performance of A* Algorithm is connected with the goodness of heuristic function. Better heuristic will result in better performance for A* Algorithm. So, Gudaitis says that general inspirations on performance comparisons cannot be made without knowing the details of objective functions

2.4.1.2 **Stochastic Methods**

The aim of Stochastic Methods is to find convenient results for a given set of inputs and boundary. For route planning process, the significant property of the results is "feasible" rather than "best". So, if the essential thing is "feasibility of found route", then stochastic methods will be used because of their performance with respect to the deterministic methods. But, results of stochastic methods can not be repeated and predicted for the same set of inputs. This uncertainty, which is related with the probabilistic sides of the chosen method, makes the stochastic methods less preferable compared to the deterministic methods.

2.4.1.2.1 Potential Fields

Potential Field method is commonly used in robotic applications. Goodrich [17] presents details of this method in his paper with respect to robotic implementations.

Novy [6] discusses potential method for the purpose of route planning under the Analogous – Problem Formulation title. Analogous- problem formulation is defined as the way of getting results by transforming the present problem to another one which is already solved. Related to the Analogous Problem Formulation, in potential Filed method, properties of search field are simulated with their corresponding physical principles. Search field is represented with two different field types: Attraction Field and Repulsion Field.

Novy [6] presents the Potential Field method with adverting to work of Burtoff [18]. Briefly, the method is presented as the following: Path is assumed to be a chain of masses. These masses are interconnected by springs and dampers. Relation between the threats and the path is modeled by virtual force fields that push the mass-spring-damper systems. Resultant equilibrium locations of the masses stand for the way points that the path is passed on.

Gudaitis [1] discusses potential field method with electromagnetism concepts. Shortly, opposite charges attract and identical charges repel each other. Gudaitis explains how electromagnetism is utilized in path planning as the follows: Goal point is set with a strong high potential value. Start point, which stands for the source, is given a lesser potential value than the goal potential. There is a moving point that has the same potential value with the source. In addition to these, threats, which stand for the obstacles, are given potential values lesser than the source. After setting the potentials, the moving point is repelled by the start point (source) and threats (obstacles); at the same time it is attracted by the goal point.

Gudaitis [1] emphasizes the drawbacks of Potential Fields method. A major drawback is the lack of guarantee to find the optimum result. Also, there is a possibility to be stuck at local minima zones in the search field and not reaching the goal point.

2.4.1.2.2 Genetic Algorithms

Genetic Algorithm is an optimization method that works with principles resembling Evolutionary Theory. It was developed by John Holland [19] in 1960's. In literature, Genetic Algorithm is described as a method used for problems that has a difficultly solved solution model.

Kurt and Şemetay[20] summarize the algorithm steps as the following:

- Initialization: Population generation with n chromosomes (generation of alternative possible solutions).
- Selection: For each selected x chromosomes, evaluation of fitness is considered.
- Reproduction: Regeneration of alternative populations from the selected ones through crossover and mutation.
- Termination: Continue until time limit is reached or sufficient suitability is reached.
    1. Test: If the results satisfy the desired criteria, the algorithm run is terminated. Last population is accepted as final solution.
    2. Loop: If the results do not satisfy the desired criteria, the algorithm run is continued from "Selection" step.

In his work, Miles B. Pellazar [21] discusses route planning process with Genetic Algorithm for flying vehicles. He compares the results found with Genetic Algorithm and the results got with Dynamic Programming. To get an idea about the Genetic Algorithm results, he performs it nearly 10 times for same set of input and initial conditions. Pellazar explains that this repetition arises from the stochastic character of the genetic algorithm.

Miles B. Pellazar emphasizes that Genetic Algorithm outputs are "nearly ideal" results. Also, Pellazar regards the needed computational time as considerably much. But, for "complex regions", difficultly modeling regions, genetic algorithm will be used to get nearly ideal solutions.

2.4.1.2.3 Simulated Annealing

Simulated Annealing is characterized as "Combinatorial Optimization". The common trait of this type optimization is that the results consist of discrete solutions and discrete modeling outputs.

Simulated Annealing is derived by Kirkoatrick [22] and Cerny [23] separately. The name of the method is taken from "annealing" process in metallurgy, because of the similarities between the solution approach and the real metallurgical annealing process. Gudaitis [1] emphasizes that locally optimal results are obtained instead of global ones.

Annealing is a technique of heating and controlled cooling to get rid of defects in materials. Simulated Annealing implementation is as the following: Searched space is assigned with a value (resembles hot temperature). Threat regions are set with considerably higher values according to the common value in search space. While the method runs, the value at search space is decreased (cooled) with respect to previously determined criteria. By the time of process, search space reaches the steady state which is considered as convenient route.

In Simulated annealing method, adjusting the initial value and setting the decreasing criteria are both significant in order to find solutions. Initial value adjusting should be done with a value that does not affect the equilibrium state value. Also, decreasing criteria should be chosen as criteria which direct the system to reach the equilibrium state.

2.4.1.2.4 Monte Carlo and Las Vegas

Although Monte Carlo Method and Las Vegas Method are handled together, their solutions have different characteristics. It is possible to get fast results with Monte Carlo Algorithm, but the exactness of the results cannot be guaranteed. For the Las

Vegas Algorithm, although correctness of the results is guaranteed, required computational time is high.

In Monte Carlo Algorithm, errors of results are diminished, as the computations are replicated. But Gudaitis [1] emphasizes that the correctness level of results cannot be known in Monte Carlo Algorithm. Therefore, using this method for route planning is not favorable.

Similarly, Gudaitis [1] does not see the Las Vegas Algorithm as an admissible method for route planning practices. Related to the computational work load, the required computational time is considerably different for a problem with same set of inputs.

2.4.2 Search Method - A* Algorithm

A* Algorithm is a heuristic, best first, graph search algorithm. At the end of the search process the algorithm finds the path with least cost to reach goal state.

For route planning process, its usage is very common. The reason is the existence of knowledge about the search space before search. Utilization of this knowledge shortens the search time. Heuristic side of the A* Algorithm permits route planning process to add knowledge of search space in the search.

The evaluation function of A* Algorithm is given by at equation(2.39),

$$F = G + H \tag{2.39}$$

where

F : Total Estimated Cost

G : Real Cost

H : Heuristic Cost

During the search, this evaluation function is calculated for each created node in the search space. The implementation of the evaluation function for i^{th} node, shown in Figure 2.28 is given as equation (2.40).

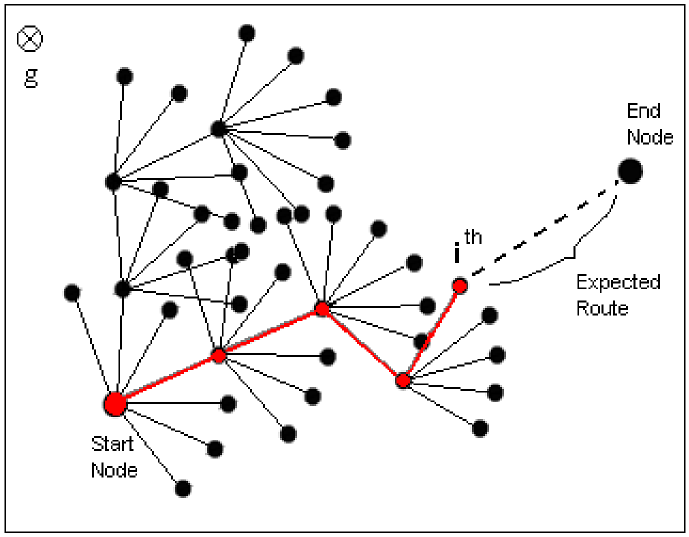

Figure 2.28 Route for i^{th} node.

$$F_i = G_i + H_i \tag{2.40}$$

where

F_i : Total Estimated Cost of route that starts from Start node, passes through ith node and ends with End node.

G_i : Real Cost of route from start node to ith node

H_i : Heuristic (estimated) Cost of route from i^{th} node to End node

In order to guarantee finding the best route between a start and an end node, the following rule should be satisfied while using A* Algorithm. "Heuristic part of the evaluation function, "H_i", should be equal to or smaller than the actual cost of the related distance. If the value of H_i is somewhat equal to the real case, then A* algorithm will have the maximum performance to find the best result. Dechter [24] gives some proofs and proposals on this rule in his work.

G_i, representing actual cost of the route reaching ith node, is calculated with using equation (2.4) given at Evaluation of Route section. The whole normalized and weighted threat cost and fuel cost up to ith node are calculated and summed.

H_i, representing the heuristic side of the evaluation function, is calculated while considering the expected route given in Figure 2.28. While setting the value of H_i, guarantee of optimality in results is tried to be provided. In literature, for search algorithms, guarantee of optimality is discussed with the term admissibility. Nilsson [25] explains the admissibility of A* Algorithm with detailed discussions under the "Algorithm A" and "The Admissibility of A*" titles. Moreover, Bradly [28] presents same considerations in terms of Multi Objective A* Algorithm (MOA*). Briefly, A* algorithm guarantees to terminate in optimal results if heuristic cost between any two nodes is equal or smaller than the actual cost between the same nodes. According to quoted reference [29], detailed considerations are given at Appendix A

$$H_{A,B} \leq G_{A,B} \tag{2.41}$$

where

$H_{A,B}$: Heuristic cost between node A and node B

$G_{A,B}$: Actual (real) cost between node A and node B

In this work, H_i stands for the $H_{i,E}$ which is the heuristic cost between ith and end node. The value of H_i is determined with respect to normalized version of geometric distance between location of ith and location of end node (2.42). Normalized value is got with the ratio of distance with maximum flight range of UAV.

$$H_i = \frac{D_{i,E}}{D_{\max}} \tag{2.43}$$

where

H_i : Heuristic cost

$D_{i,E}$: Geometric distance between ith nodes and end node.

$D_{\max}$: Maximum flight range of UAV

2.4.2.1 **Node Elimination**

When the A* Algorithm runs without any node elimination, there will be excessive amount of time needed to reach the most efficient route. Even if the start search space parameters are set with values resulting in coarse search, stretching speed of search network towards the end is decelerated and the algorithm balks to proceed. Figure 2.29 and Figure 2.30 show such an example for this case.

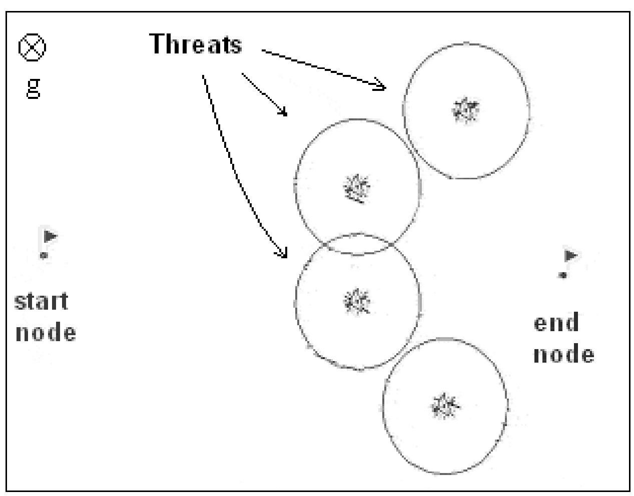

Figure 2.29 Route planning problem example

As it is seen from Figure 2.30, search space expansion starts with coarse legs, but there is excessive number of nodes placed near the threats. This means that speed of search space expansion slows down after it reaches boundaries of threats converges.

One of the reasons for this retardation is related with the heuristic side of A* Algorithm. If the results of heuristic costs are not close enough to real ones, then the number of candidate routes will increase and unsuited routes options also could be examined to satisfy the best route statue of found route.

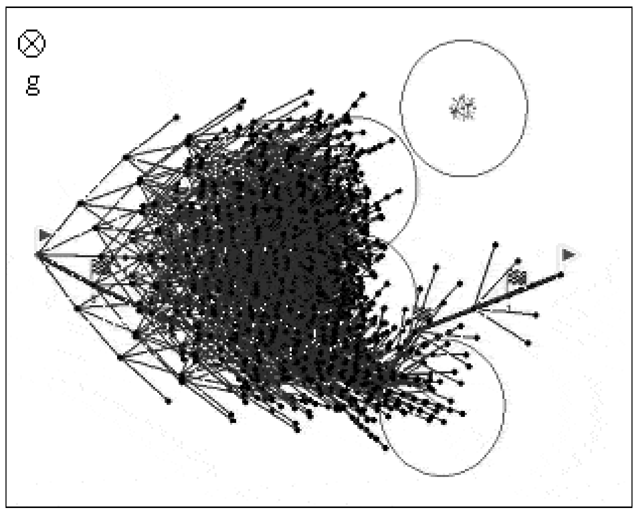

Figure 2.30 Constructed search network for the route planning problem example

The other reason is the calculations of alternative routes which reach same nodes or nodes placed close to the each other (Figure 2.31). Reaching same node case is represented with node "A" in Figure 2.31. Node "B" and "C" are the examples of nodes placed close to each other.

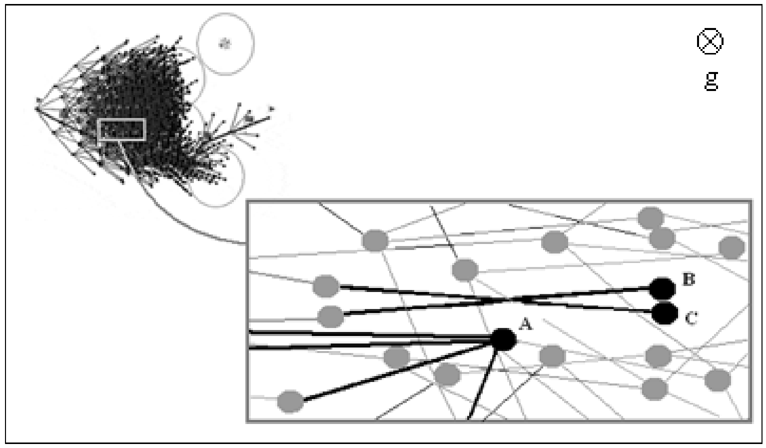

Figure 2.31 Search Network

To make node elimination, the distance between reached nodes, "∂d", and angle difference between upcoming leg directions to those reached nodes, "$\partial\varphi$" (Figure 2.32) are considered. If both values of ∂d and $\partial\varphi$ are at negligible magnitudes, then one of the related routes is chosen to continue searching. The other ones are stopped. Putting differently, related reached nodes are removed from the search space.

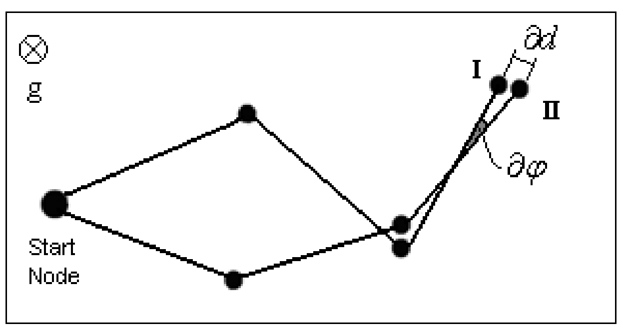

Figure 2.32 Neglected Distance, ∂d and Neglected Angle Difference, $\partial\varphi$

2.4.2.2 **Optimality of Search Method's Results**

At this section, optimality of results which are obtained with search method, Best First Search (A*) Algorithm, is discussed. Considerations related to the proofs are quoted from the "Informed Search and Exploration" chapter of Russel's book [29].

Russel defines best first search algorithm as an instance of the general tree-search or graph-search algorithm in which a node is selected for expansion based on an evaluation function, $f(n)$. Generally, the search tends to proceed from the node with lowest evaluation function value. For A* Algorithm, Russel designates the function, $f(n)$ as the following equation:

$$f(n) = g(n) + h(n) \tag{2.44}$$

where

$g(n)$: The cost to reach node (n)

$h(n)$: The cost to get from node (n) to the goal

Initially, Russel discusses the optimality of A* Algorithm used with tree search. At further discussions, he gives the optimality of A* Algorithm used with graph search. For Tree Search Case, A* is optimal if $h(n)$ is an "admissible" heuristic-that is, provided that $h(n)$ never overestimates the cost to reach the goal. Since $g(n)$ is the exact cost to reach node, "n", $f(n)$ never overestimates the true cost of a solution through node, "n".

Suppose that there is a suboptimal goal node, "G_2" that exist as an alternative solution. Also, let assume that the cost of the optimal solution is C^*.

Because $h(G_2) = 0$ (valid for goal nodes), evaluation function is as the following:

$$f(G_2) = g(G_2) + h(G_2) = g(G_2) + 0 = g(G_2) \quad (2.45)$$

Since, G_2 is suboptimal:

$$f(G_2) = g(G_2) > C^* \quad (2.46)$$

Let's consider a node n which is on an optimal solution path. If $h(n)$ is an admissible heuristic, provided that it does not overestimate the cost of path up to the goal, then evaluation function is as the following:

$$f(n) = g(n) + h(n) \leq C^* \quad (2.47)$$

Then,

$$f(n) \leq C^* < f(G_2) \quad (2.48)$$

Therefore, G_2 will not be expanded. This is extracted that A* using a Tree Search returns an optimal solution, if $h(n)$ is an admissible.

After the discussion of tree search case, Russel handles the optimality of A* Algorithm used with graph search. At first, it is emphasized that suboptimal solutions can be obtained because of the fact that graph search can discard the optimal path to a repeated state if it is not the first one generated. Therefore, in order to guarantee the optimality, Russel proposes two different choices. The first choice proposed is to extend graph-search so that it discards the more expensive of any two paths found to the same node. Although it requires extra bookkeeping and it is messy, but it does guarantee optimality.

The second choice is to ensure that the optimal path to any repeated state is always the first one followed. This is related with the consistency property of heuristic (monotonicity). Consistency of a heuristic is described as the following: A heuristic $h(n)$ is consistent if, for every node n and every successor $n^{'}$ of n generated by any action a, the estimated cost of reaching the goal from n is no greater than the step cost of getting to $n^{'}$ plus the estimated cost of reaching the goal from n

$$f(n) < c(n, a, n^{'}) + h(n^{'}) \tag{2.49}$$

In this work, construction of search network is discussed at Search Space Representation section in depth. In that section, used parameters in order to place nodes and subsequently to build search tree are considered. Moreover, flight between any two consecutive nodes or through a series of nodes (route) is considered at Trajectory Construction section.

Placement of the start and end node pair and assignment of the values of search parameters used during the search means the settlement of locations of all possible candidate nodes, implicitly. Therefore, settlement of the search tree is accomplished before the search. At first glance, it is possible to be supposed that A* Algorithm used in this work makes use of a tree search. But, actually, it makes use of a graph search instead of a tree search, because flight trajectories between the nodes are determined during the search according to proposed node series. For instance, Figure 2.33 exemplifies the case. The path, between node B and node C, changes according to the location of previous node.

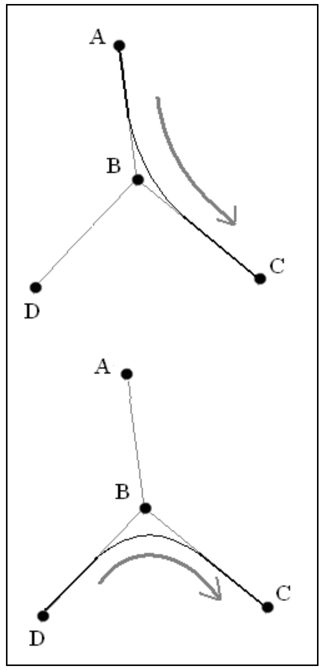

Figure 2.33 Graph Search Repeated States

The used heuristic in this work is admissible. Geometric distance which is the minimum range value is considered as a heuristic. Actual distance cannot be less than this one. Therefore, the rule of using not overestimated heuristic is satisfied. Moreover, as it is discussed at Node Elimination section, repeated states are not omitted but taken in considerations during the search. If such as case shown at Figure 2.33 is come up, both path options is considered and the one with least evaluation cost is chosen in order to continue to the search from Node C. Therefore, the first condition stated by Russel for graph search is satisfied.

2.4.2.3 **Flow Chart of A* Algorithm**

In Figure 2.34, flow chart of A* Algorithm used in this research is presented.

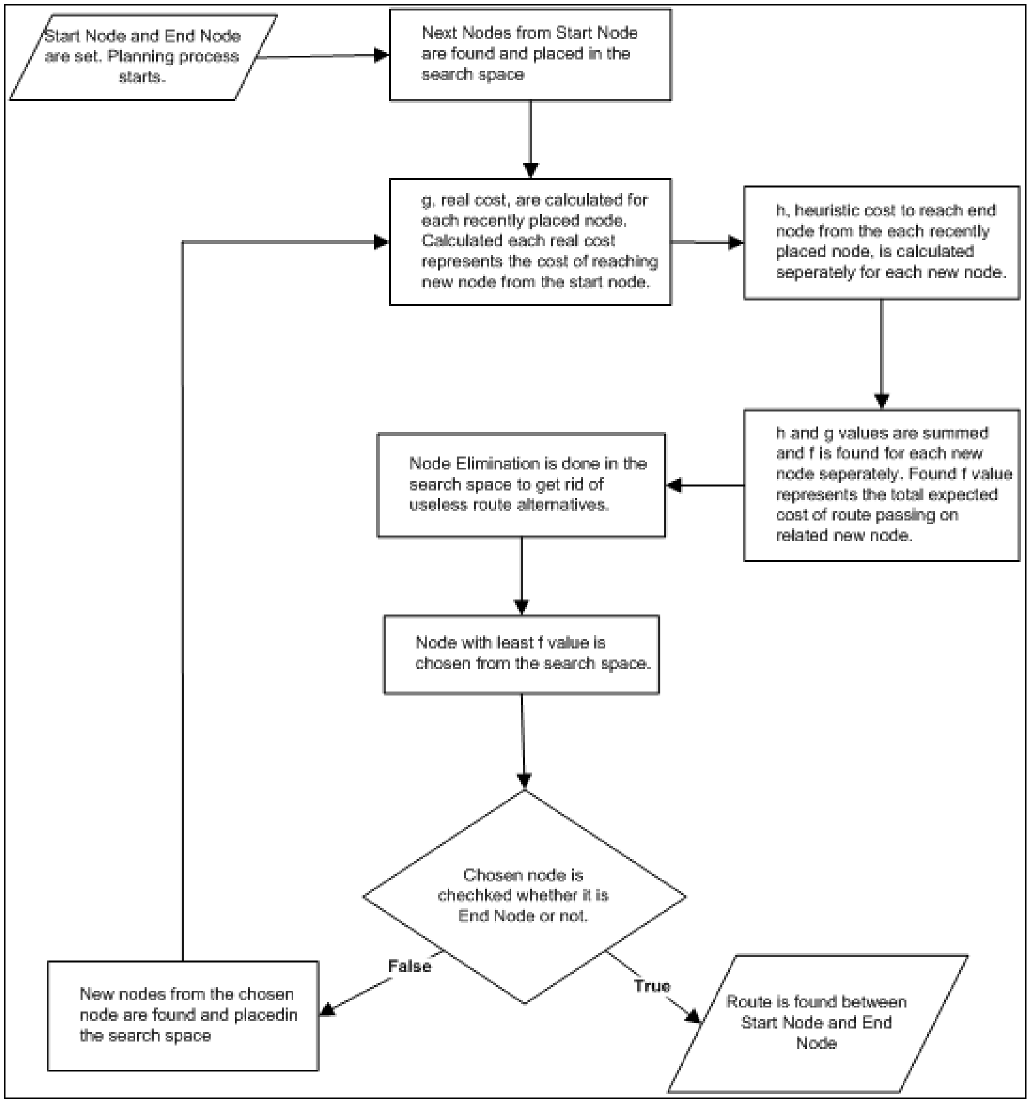

Figure 2.34 Flow chart of A* Algorithm

CHAPTER 3

SAMPLE RUNS AND RESULTS

In this chapter sample runs are presented to show the performance of the route planner developed. There are four different scenarios, all representing different performance characteristics. The first one aims at showing the minimum distance property, the second one presents the minimum fuel characteristic, the third one is depicting the survivability performance and finally the last one is a typical operational scenario.

In the minimum distance case, the start and end points of the route are located over the sea away from shore. As expected, the route planner as shown in Figure 3.1 computed the route over the minimum distance line from start to end point. Supplementary figures of this problem including search network and vertical trajectory profile are provided in Appendix A.

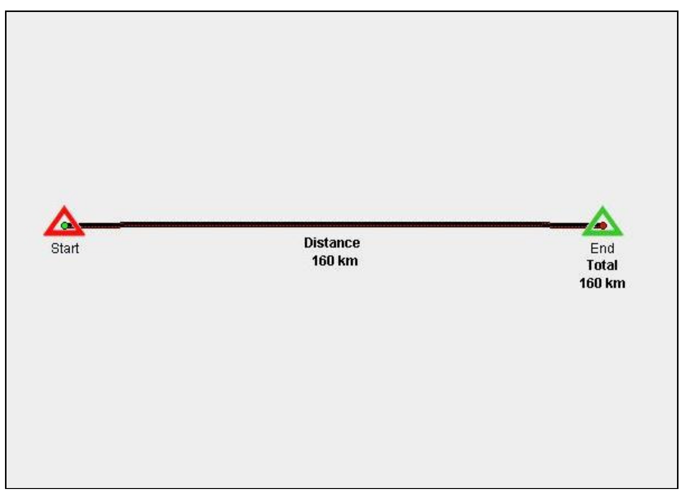

Figure 3.1 Minimum Distance Case

In minimum fuel case, the start and end points of the route are located over an inland region. As expected, the route planner as shown in Figure 3.2 computed the route over the area with less fuel consumption. i.e. over the area where there are less ascending and descending. Supplementary figures of this problem including search network and vertical trajectory profile are provided in Appendix A.

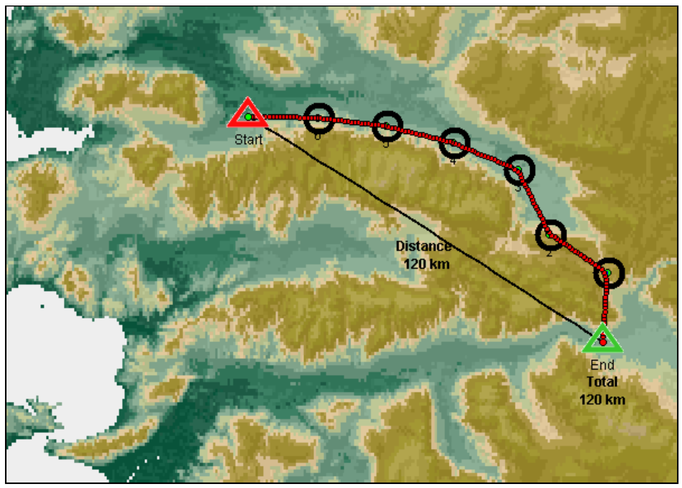

Figure 3.2 Minimum Fuel Case

In survivability case, the start and end points of the route are located over the sea away from shore. Between start and end there are threats each having a specific detection radius. As expected, the route planner as shown in Figure 3.3 computed the route circumventing and bewaring from threats. The route planner tried to minimize the length of route at the same time. Supplementary figures of this problem including search network and vertical trajectory profile are provided in Appendix A.

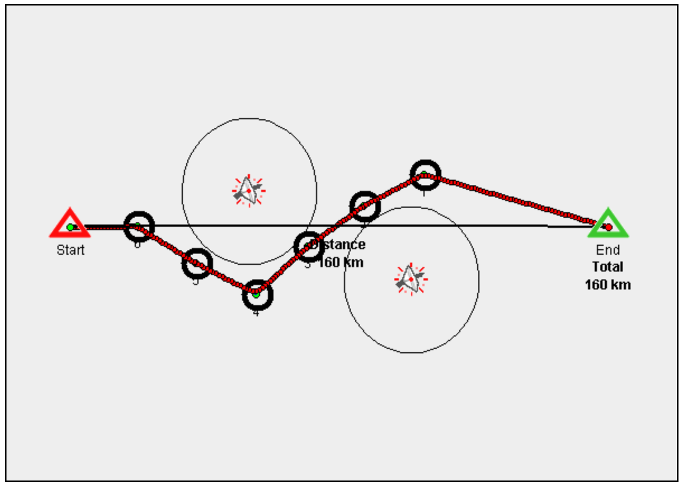

Figure 3.3 Survivability Case

In survivability case, the start and end points of the route are located over an inland region. Between start and end there are threats each having a specific detection radius. As expected, the route planner as shown in Figure 3.4 computed the route bewaring from threats and minimizing the fuel consumption simultaneously. Supplementary figures of this problem including search network and vertical trajectory profile are provided in Appendix A.

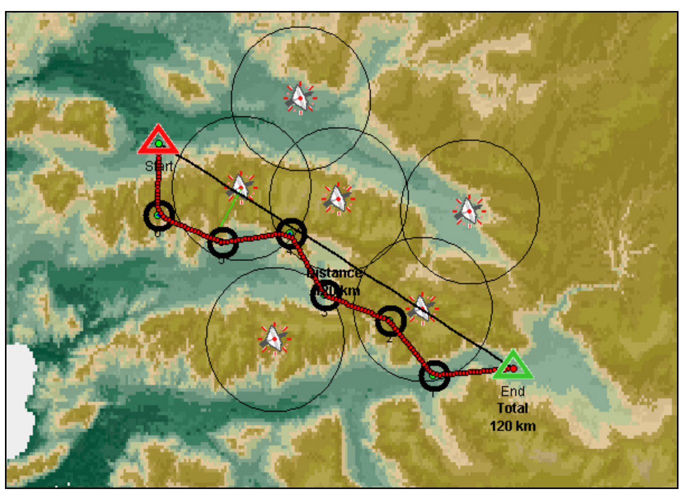

Figure 3.4 Typical Operational Scenario Case

CHAPTER 4

CONCLUSION AND FUTURE WORK

In this research enhancing the low observability characteristics of UAVs by planning LO routes for their missions is aimed. In order to achieve this, a route planner that will compute the least observable route with the lowest fuel cost is developed.

The route planner built includes a search network which yields an optimal route between a given start and end location at the end of its search. Optimality of resultant route is derived from the consideration of fuel consumption and survivability comparison of UAV against threats.

Search network has two main appointments in order to find the optimal route. The former one is to evaluate a single route in terms of fuel consumption and survivability. The further one is to direct the search systematically while comparing the evaluated values of route options between given start and end locations. Modality for both issues is discussed at METHODOLOGY chapter of this thesis.

In METHODOLOGY chapter, initial discussions are related with space representation subject. Under this title, how the search space is handled and which parameters are used in order to navigate in a 2D search space are discussed. Construction of search space is done with geographic coordinate systems instead of using a cartesian coordinates. This conformation enables the usage of standard digital terrain data.

Navigation of search is directly related to the parameters discussed under the search space representation title. The values of Leg Length, L, Search Sweep Angle, α, and Search Division Angle, β affect the precision of optimality and the computation time. The computational time required to reach the solution is less if either the Search Sweep Angle, α, is small or Search Division Angle, β, is high or

Leg Length, L, is long. But, this case may result in missing route options with better performance. For the opposite case, the results will be vice versa. At this point, some questions arise: is there a necessity for a deeper search to get better results, when does this necessity emerge, what are the search space parameters statuses in these judgments? For these questions, a study was conducted related to this thesis, and its outputs were presented at 2009 IEEE Aerospace Conference [26]. The solution approach on setting the search parameters apart from the leg length is discussed at the following paragraph briefly. Detailed considerations can be seen in the above mentioned paper.

In reference [26], the values of search sweep angle and search division angle are associated with the situation between and "around" the searched route's start and end locations. This is named with the words "Situation Awareness". Initially, the value of search sweep angle is correlated with the range between start and end locations. If this range is close to the maximum flight range of UAV, then UAV cannot maneuver so much in order to reach the end; i.e. a search with wide search sweep angle, α, will be meaningless. Secondly, an operation zone is defined in order to embody the interpretation of the word "around". Operation Zone assigns the limits of the area to get observation about the threats and their coverage. According to the gotten observation, search division angle is determined. If the threats coverage is not uniform or variance of it is high, then a deep search will be preferable to get a better route; i.e. search division angle, β, is set with a low value.

Beside the considerations handled at previous paragraph, there is one more title about the search space parameters. That is Leg Length, L. In this work, leg definition is given as the way of any two nodes. The only restriction that exists is its minimum length limit (Figure 2.4).

As it is emphasized under the related title, search space is represented in 2D space with nodes which are located during the search. These located nodes constitute the route options to reach end location from the start one. Routes get their third dimension while the trajectories are set. In reality, there will be infinitely many trajectory choices to track a route. In this research, since planning of LO routes are aimed, a terrain following strategy is adopted in order to obtain a trajectory from a given set of node, route. Route and trajectory considerations are handled under the Route and Trajectory Construction title. At this part, trajectory construction is given with respect to UAV's performance parameters.

In this research, UAV's performance parameters are assumed to be constant quantities. But in practice, they will be functions of several different variables, such as altitude or velocity. For example, at Figure 2.9, maneuver of the UAV is calculated with respect to UAV's Turn Radius, R (maneuvers performance constant). But, normally UAV's maneuver capability will increase directly with the increasing of elevation from the mean sea level. Since the density of air decreases, sharper maneuvers will take place at high elevations. Therefore, instead of using a constant turn radius, a tabulated version which is constituted with respect to elevation can be used. Although this contention is very legitimate, it is not practical for present trajectory construction approach, because, maneuver calculation is the first step of trajectory construction process. Also, elevation information does not exist up to that step. In order to implement the variable turn radius case, either the present approach should be abandoned or more dynamic dealing has to be established or some iterative steps are inserted to the present trajectory construction calculations.

Same judgments which are discussed at previous paragraph can be made on the other UAV's performance parameters. There are some significant points which make the presently used approach admissible. First point is that our scope is limited to "Planning of route". In literature there is one more step after the planning, called as "Mission Rehearsal" [28]. This step is focused on the analysis of the plan outputs. Such deep considerations are addressed at this rehearsal step. Second point is that during the route planning process, assumptions or ignorance are valid for all alternative solutions. If an assumption or ignorance is enforced for evaluations of all alternatives, then differences resulting from that assumption or ignorance will not be so much misleading. Third but not the least important point is that route planning process should be terminated in a reasonable time. Including extra iterative steps slows down the planning. In order to get an idea whether extra iterations are needed or whether present approach should be updated, a mission rehearsal study should be done as a future work. From the outputs of this study, reconsideration may be made on the present approach.

In this thesis, another significant part is the "Evaluation of Route" title. Briefly, evaluation of a route is done using its trajectory. The evaluation criteria are consumed fuel during flight and the survivability of UAV against being tracked by threats present at the search space. As a future work, these evaluation criteria can be enlarged. For example GPS (Global Positioning System) performance of a route may be discussed for evaluating against the other route candidates.

For fuel consumption calculations, a dynamic model is used. From the required lift and thrust force, the required instant fuel consumption rates are found. These consumption rates are obtained using the required thrust values and the corresponding specific fuel consumption data of UAV's motor. As an alternative approach, same fuel consumption will be obtained using fuel consumption tables. These tables will be "n" dimensional tables. Dimensions of the table will be formed by the flight parameters, such as elevation, speed, mass, displacements, etc. For tabulated case, fuel consumption between any given two locations is calculated with n dimensional straight forward iterations. Although this method expedites the computations, the results will not be exactly correct because of the rounding or computational errors coming from the iterations. But as it is emphasized before, if an omission is enforced for evaluations of all alternatives, then differences resulting from that omission will not be so much misleading.

For survivability evaluations, three main subjects in literature are dealt with: RCS modeling, line of sight analysis and radar equation. RCS modeling is discussed on the horizontal plane (Figure 2.19). It needs the RCS values as a function of azimuth angle around the UAV. As a future work this representation can be extended by adding elevation angle to the RCS model. Line of sight analysis includes the threat coverage calculations around each threat. Therefore, line of sight analysis needs terrain profile (DTED) around threats. Radar equation is used with its simplified version so that the detection radius of threat for a specific RCS value is calculated. In order to use this equation, maximum detection range of each threat against a target with 1 m^2 RCS is required. In order to improve present survivability considerations, higher fidelity approaches can be studied as a future work. Instead of using being seen or not seen by threat knowledge, numeric value of probability related with being seen will be taken into account. As a future work, survivability cost will also be divided into sub parts: Being detected cost, being tracked cost, being engaged cost. These are the steps of a threat so as to counteract an UAV.

In this research, several different search methods are briefly discussed from the aspect of route planning. Among these methods, Best First Search (A* Algorithm) is chosen as a method so as to direct the seeking of optimal route. A* algorithm is a heuristic graph search algorithm. The most attractive side of this method is its heuristic facility. This attraction comes from the fact that heuristic impels the search towards the goal.

In A* Algorithm, impellent performance due to heuristic is related with the closeness of heuristic value with actual one. But, admissibility considerations should be taken into account so as to avoid spoiling the route's optimality [25]. In this research, chosen heuristic is relatively common. Remaining geometric distance to reach the end is regarded. Regarding geometric distance gives reasonable intention in order to reach the goal without extending the search towards relatively irrelevant regions. But, contribution of this heuristic becomes worse for some specific cases. One of them is the case where the end location is surrounded with threats. The other one is the case where the variance of terrain elevation is considerably high. Common property of these two cases is that the relation between the actual cost and the geometric distance is considerably weak. So, heuristic impellent to reach the end will not be so much effective. As a future work, heuristic function can be updated so as to increase its contribution. As an alternative solution, the search approach can be rearranged such that the route is searched from the end location to the start one.

REFERENCES

1 Michael, S. Gudaitis, *Multicriteria Mission Route Planning Using a Parallel A* Search*, Air Force Institute of Technology WPAFB OH 45433-6583, 1994.

2 Myron, Hura and Gary McLeod, *Route Planning Issues for LO Aircraft and Cruise Missiles*, Implications for Intelligence Community, United States Air Force, 1993.

3 National Digital Elevation Program, *Guidelines for Digital Elevation Data*, *Version 1.0*, 2004.

4 National Geospatial-Intelligence Agency (NGA), *Performance Specification Digital Terrain Elevation Data (Dted)*, MIL-PRF-89020B, 23 May 2000.

5 Federation of American Scientists, *Digital Terrain Elevation Data (DTED)*, 1725 DeSales Street, NW 6th Floor Washington, DC 20036, updated: 24 January 2000, http://www.fas.org/irp/program/core/dted.htm, accessed: 02.06.2009

6 Novy, M.C, *Air Vehicle Optimal Trajectories for Minimization of Radar Exposure*, PhD dissertation, Graduate School of Engineering and Management, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, March 2001.

7 Pierre, T. Kabamba, *Optimal Path Planning for Unmanned Combat Aerial Vehicles to Defeat Radar Tracking*, Journal Of Guidance, Control, And Dynamics, Vol. 29, No. 2, March–April 2006.

8 Merril, I. Skolnik, *Radar*, Reference Data for Engineers: Radio, Electronics, Computers and Communications, chapter 36. Ninth Edition, 2002.

9 David, L. Adamy, *Radar Cross Section*, EW102 A Second Course in Electronic Warfare, Artech House, Norwood, Massachusetts, 2004.

10 Eugene, F., *Radar Cross Section*, *2nd Ed.*, SciTech Publishing, 2004.

11 Skiena, Steven S., *The Algorithm Design Manual*, *2nd Ed.*, London, Springer, 2008.

12 Tim, M. Jones, *Artificial Intelligence: A Systems Approach*, Infinity Science Press, p. 56 -58, 2007.

13 Giri Narasimhan, and Michiel Smid, *Geometric Spanner Networks*, Cambirage University Press, 32 Avenues of the Americas, New York, NY 10013-2473 USA, p. 31, 2007.

14 Clark Moustakas, *Heuristic Research: Design, Methodology, and Applications*, Sage Publications, Newbury Park London New Delh, p. 9, 1990.

15 Thomas, H. Cormen, *Introduction to Algorithms*, Massachusetts Institute of Techonology, Cambiridge, Massachusetts 02142, MIT Press, 2001.

16 Yanhong, A. Liu, Scott, D. Stooler, *Dynamic Programming via Static Incrementalization*, Higher-Order and Symbolic Computation, 16, 37–62, Kluwer Academic Publishers, Manufactured in The Netherlands, 2003.

17 Michael, A. Goodrich, *Potential Fields Tutorial*, Brigham Young University, http://www.ee.byu.edu/ugrad/srprojects/robotsoccer/papers/goodrich_potential_fields.pdf, accessed: 01.06.2009.

18 Bortoff, A. Scott, *Path-Planning For Unmanned Air Vehicles*. Unpublished Report, Wright-Patterson AFB OH: AFRLIVAAD, 1999.

19 Holland, J., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.

20 Kurt, M., Semetay, C., *Genetik Algoritma ve Uygulama Alanları*, Mühendis ve Makine, sayı 501, 2001.

21 Miles, B. Pellazar, *Vehicle Route Planning with Constraints Using Genetic Algorithms*, Northrop Corporation, 8-2 Division 8900 E. Washington Boulevard, MS: T213/GB Pic0 Rivera, CA 90660, 1994.

22 Kirkpatrick, S., Gellatt, C. D., Vecchi, M. P., *Optimization by simulated annealing*, Science, 220:671—680, 1983.

23 Cerny, V., *Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm*, Journal of Optimization Theory and Applications, 45:41--51. 10, 1985.

24 Dechter, Rina Judea Pearl, *Generalized best-first search strategies and the optimality of A*, Journal of the ACM 32 (3). p. 505 – 536, 1985.

25 Nils J. Nilsson, *Algorithm A*, Principles of Artificial Intelligence, p.76. Birkhäuser, ISBN 3540113401, 9783540113409, 1982.

26 Kamil TULUM, Umut DURAK, S Kemal IDER, *Situation Aware UAV Mission Route Planning*, 2009 IEEE Aerospace Conference. Big Sky, Montana USA, 2009.

27 AGARD-LS-192, *New Advances in Mission Planning and Rehearsal Systems*, NATO Research and Technology Organisation, ISBN-92-835-0723-1, January 1993.

28 Bradley S. Stewart, Chelsea C. White, III, *Multiobjective A**, Journal of the ACM (JACM), v.38 n.4, p.775-814, Oct. 1991.

29 Stuart, J. Russell, and Peter Norvig, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Inc., Upper Saddle River, NJ, p. 99 – 100, 1995.

APPENDIX A

SUPLEMENTARY FIGURES FOR SAMPLE RUNS

1. Minimum Distance Case:

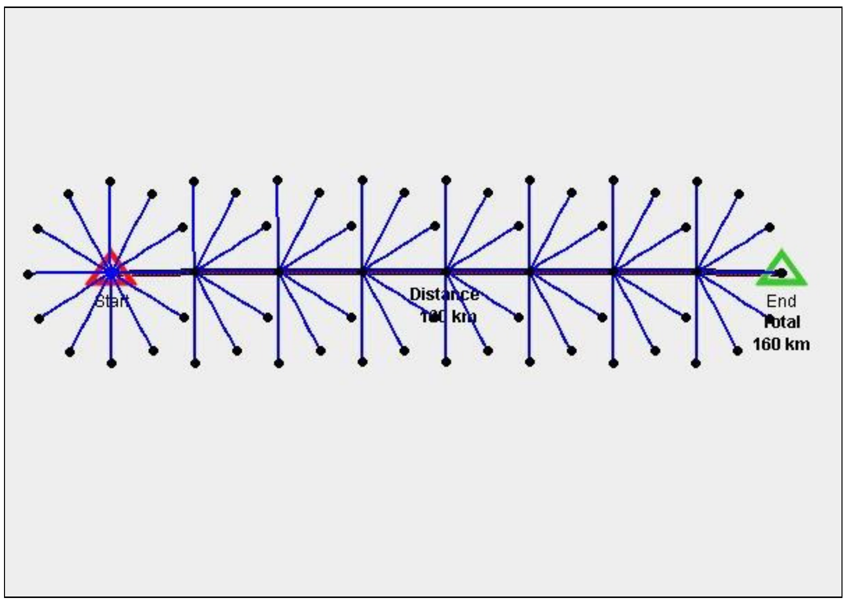

Figure 4.1 Search network for Minimum Distance Case

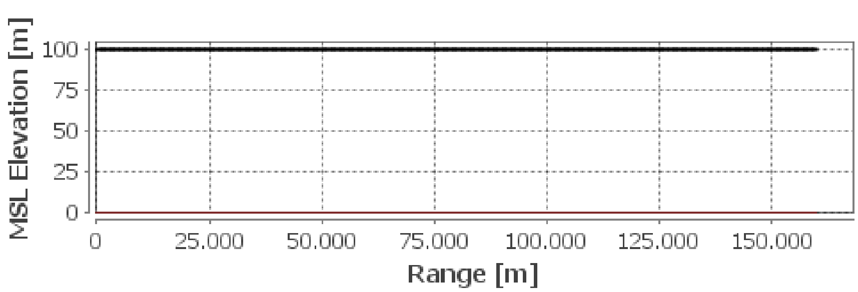

Figure 4.2 Vertical Terrain and Trajectory Profile for Minimum Distance Case

2. Minimum Fuel Case:

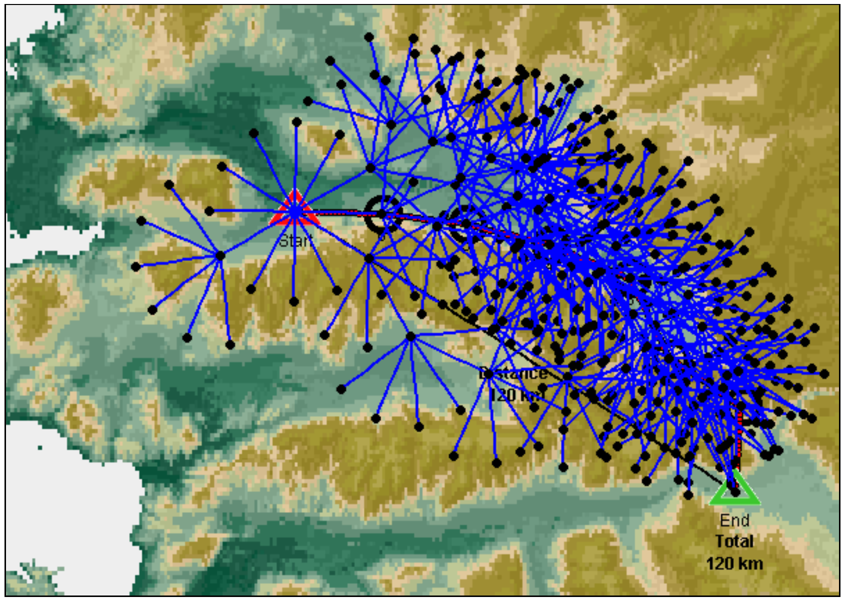

Figure 4.3 Search network for Minimum Fuel Case

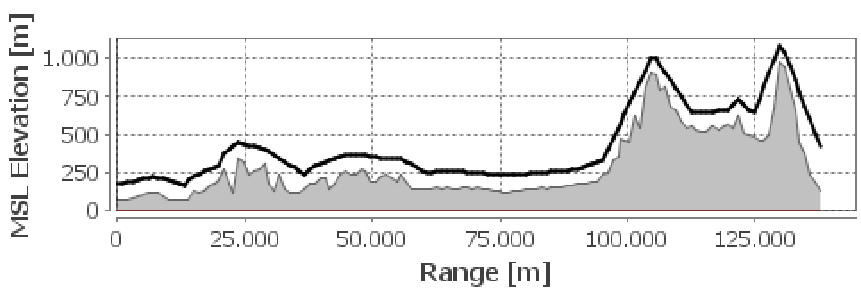

Figure 4.4 Vertical Terrain and Trajectory Profile for Minimum Fuel Case

3. Survivability Case:

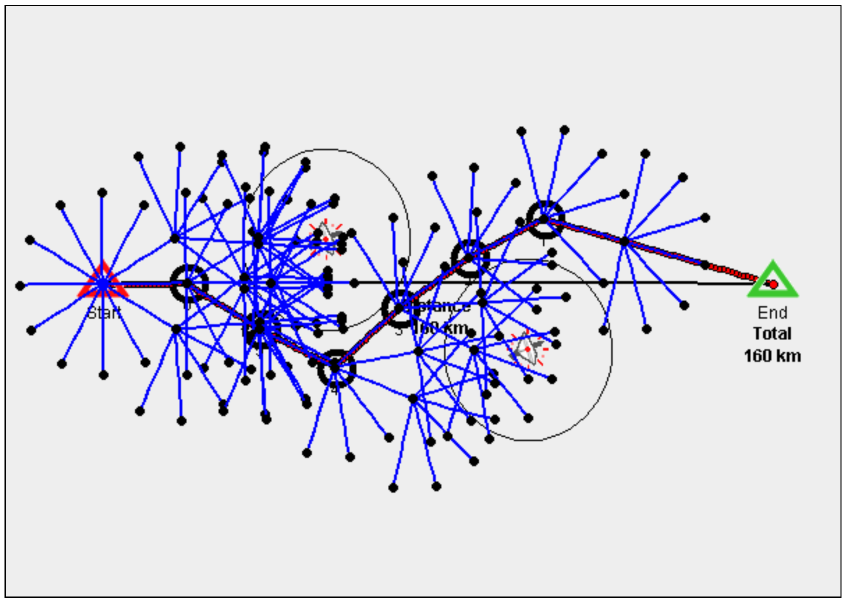

Figure 4.5 Search network for Survivability Case

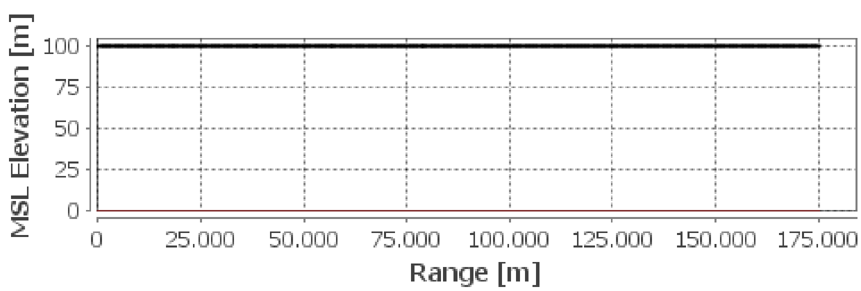

Figure 4.6 Vertical Terrain and Trajectory Profile for Survivability Case

4. Typical Operational Scenario Case:

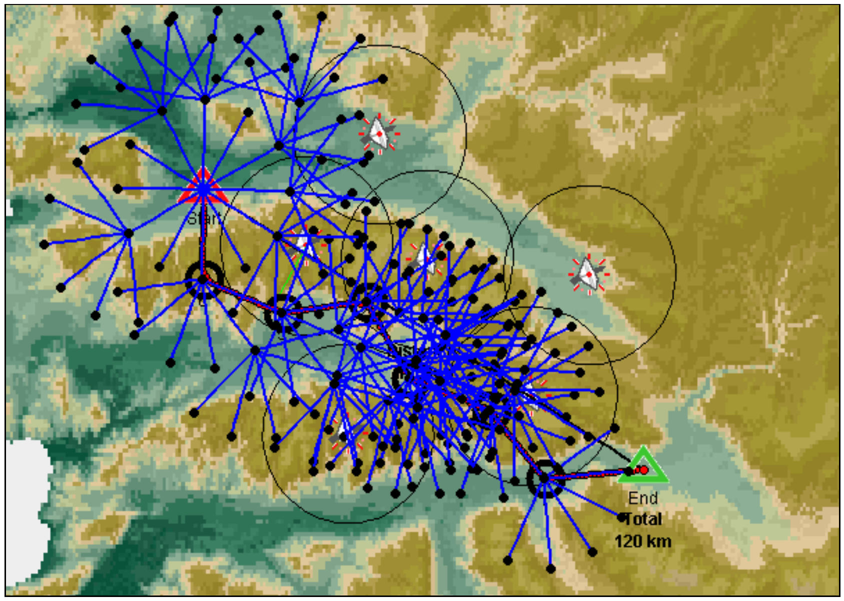

Figure 4.7 Search network for Typical Operational Scenario Case

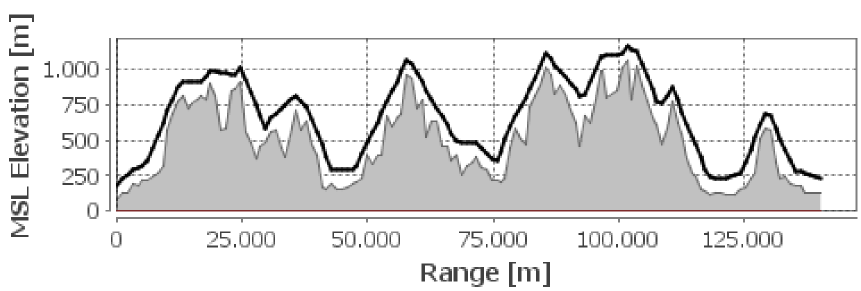

Figure 4.8 Vertical Terrain and Trajectory Profile for Typical Operational Scenario Case