

Cranfield University

Madhavan Shanmugavel

Path Planning of Multiple  
Autonomous Vehicles

DEFENCE COLLEGE OF MANAGEMENT AND TECHNOLOGY  
DEPARTMENT OF AEROSPACE, POWER & SENSORS  
CRANFIELD UNIVERSITY  
SHRIVENHAM

PhD



CRANFIELD UNIVERSITY

Defence College of Management and Technology  
Department of Aerospace, Power & Sensors

PhD Thesis  
Academic Year 2007

MADHAVAN SHANMUGAVEL

Author email: s.madhav1@yahoo.co.uk

PATH PLANNING OF MULTIPLE  
AUTONOMOUS VEHICLES

Supervisors  
Dr. Antonios Tsourdos  
Prof. Brian A. White

22nd May 2007



# Abstract

Safe and simultaneous arrival of constant speed, constant altitude UAVs on target is solved by design of paths of equal lengths. The starting point of the solution is the well-known Dubins path which is composed of circular arcs and line segments, thus requiring only one simple manoeuvre - constant rate turn. An explicit bound can be imposed on the rate during the design and the resulting paths are the minimum time solution of the problem. However, transition between arc and line segment entails discontinuous changes in lateral accelerations (latax), making this approach impractical for real fixed wing UAVs. Therefore, the Dubins solution is replaced with clothoid and also a novel one, based on quintic Pythagorean Hodograph (PH) curves, whose latax demand is continuous. The clothoid solution is direct as in the case of the Dubins path. The PH path is chosen for its rational functional form. The clothoid and the PH paths are designed to have lengths close to the lengths of the Dubins paths to stay close to the minimum time solution.

To derive the clothoid and the PH paths that way, the Dubins solution is first interpreted in terms of Differential Geometry of curves using the path length and curvature as the key parameters. The curvature of a Dubins path is a piecewise constant and discontinuous function of its path length, which is a differential geometric expression of the discontinuous latax demand involved in transitions between the arc and the line segment. By contrast, the curvature of the PH path is a fifth order polynomial of its path length. This is not only continuous, also has enough design parameters (polynomial coefficients) to meet the latax (curvature) constraints (bounds) and to make the PH solution close to the minimum time one. The offset curves of the PH path are used to design a safety region along each path.

The solution is simplified by dividing path planning into two phases. The first phase produces flyable paths while the second phase produces safe paths. Three types of paths are used: Dubins, clothoid and Pythagorean Hodograph (PH). The paths are produced both in  $2D$  and  $3D$ . In two dimensions, the Dubins path is generated using Euclidean and Differential geometric principles. It is shown that the principles of Differential geometry are convenient to generalize the path with the curvature. Due to the lack of curvature continuity of the Dubins path, paths with curvature continuity are considered. In this respect, initially the solution with the Dubins path is

---

extended to produce clothoid path. Latter the PH path is produced using interpolation technique. Flyable paths in three dimensions are produced with the spatial Dubins and PH paths.

In the second phase, the flyable paths are tuned for simultaneous arrival on target. The simultaneous arrival is achieved by producing the paths of equal lengths. Two safety conditions: *(i)* minimum separation distance and *(ii)* non-intersection of paths at equal distance are defined to maneuver in free space. In a cluttered space, an additional condition, threat detection and avoidance is defined to produce safe paths. The tuning is achieved by increasing the curvature of the paths and by creating an intermediate way-point. Instead of imposing safety constraints, the flyable paths are tested for meeting the constraints. The path is replanned either by creating a new way-point or by increasing the curvature between the way-points under consideration. The path lengths are made equal to that of a reference path.

# Acknowledgements

*I express my sincere and heartfelt gratitude to my respectable Supervisors **Prof. Brian White & Dr. Antonios Tsourdos**, whose guidance, and encouragement enunciated throughout my research programme.*

*I gratefully acknowledge **Dr. Rafał Żbikowski**, Reader, DAPS, for his valuable guidance and critical suggestions given during the study period.*

*I thank Cranfield University, Shrivenham for the merit scholarship awarded to me.*

*I take this opportunity to express my deep sense of gratitude and obligation to **Asst. Prof. João Silva Sequeira**, and **Nelson Gonçalves**, PhD student IST/ISR, Lisbon, Portugal during implementation of codes in robots.*

*My heartfelt thanks to **Immanuel, Rajnikanth, Ramesh** who were helpful in my very early days of PhD and **Samuel** for timely help since we met.*

*My thanks will be out of tune, if I dont extend my thanks to all my colleagues in Heaviside lab and in Shrivenham campus for sharing their views on technical and non-technical matters during my PhD.*

*I am with great humility to place my sincere thanks to my new friends and their family members in the UK for their help during my stay.*

*I will be failing with my responsibility if I dont acknowledge the love and affection of caring **family members & in-laws** for their blessings.*

*My final thanks are reserved for my wife **Kasthuri Madhavan**, who provided the love, and support that I needed most during the course of my carrier.*

*Above all, I like to thank God(s) for leading me towards success throughout my life.*

*Last but not the least, I pray the almighty to shower blessings on my thesis committee members and their family members and do wish them a happy and prosperous life.*

UK,  
May 2007

Madhavan Shanmugavel





*To all my teachers (Gurus) who enriched me with knowledge throughout  
my academic carrier since my school days*





---

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Path planning - An overview . . . . .	2
1.2	Coordinated Guidance . . . . .	5
1.2.1	Coordinated Path Planning of Multiple UAVs . . . . .	5
1.3	Problem Statement . . . . .	9
1.4	Solution Approach . . . . .	9
1.5	Thesis contributions . . . . .	11
1.6	Disseminations, presentations from this thesis . . . . .	12
1.6.1	Conference papers . . . . .	12
1.6.2	Journal paper . . . . .	13
1.7	Organization of the thesis . . . . .	13
<b>2</b>	<b>Producing flyable paths -2D</b>	<b>16</b>
2.1	Producing Flyable Paths - Dubins . . . . .	16
2.1.1	Using Euclidean Geometry . . . . .	17
2.1.2	Existence of Dubins paths . . . . .	21
2.1.3	Length of the Dubins paths . . . . .	21

---

2.1.4	Using Differential Geometry . . . . .	21
2.2	Paths of continuous curvature . . . . .	25
2.3	Producing flyable path - Clothoid . . . . .	27
2.4	Producing Flyable Path - Pythagorean Hodograph . . . . .	34
2.4.1	Flyable path -PH . . . . .	36
2.5	Summary . . . . .	39
<b>3</b>	<b>Producing flyable path - 3D</b>	<b>40</b>
3.1	Dubins Path-3D . . . . .	41
3.2	Extension of 2D Dubins to 3D manoeuver . . . . .	41
3.3	Dubins arc for the 2D coplanar manoeuvre . . . . .	42
3.4	Composite Dubins arc for the 3D manoeuvre . . . . .	47
3.5	Path Length - Dubins 3D . . . . .	51
3.6	Pythagorean Hodograph Path-3D . . . . .	51
3.6.1	Spatial PH Curve . . . . .	52
3.7	Design of Flyable Path using PH curve . . . . .	53
3.7.1	Design of flyable path . . . . .	54
3.8	Summary . . . . .	56
<b>4</b>	<b>Solution to Simultaneous Arrival</b>	<b>57</b>
4.1	Flyable Paths . . . . .	57
4.2	Feasible Paths . . . . .	58
4.2.1	Minimum Separation Distance . . . . .	58
4.2.2	Non-intersection of Paths at Equal Length . . . . .	59
4.3	Paths of Equal Length . . . . .	61
4.3.1	Reference Path . . . . .	62
4.3.2	Equal path lengths . . . . .	62
4.4	Algorithm - free space . . . . .	62
4.5	Summary . . . . .	63
<b>5</b>	<b>Path Planning in cluttered environment</b>	<b>64</b>

---

5.1	Threat Detection and Avoidance . . . . .	65
5.1.1	Safety-Region Inclusion . . . . .	65
5.1.2	Safety Distance . . . . .	66
5.2	Replanning the path . . . . .	67
5.3	Algorithm - Cluttered space . . . . .	70
5.4	Summary . . . . .	71
<b>6</b>	<b>Simulations and Results - 2D</b>	<b>72</b>
6.1	Simulations and Results - Dubins . . . . .	72
6.2	Simulations and Results - Clothoid . . . . .	75
6.3	Simulations and Results - PH . . . . .	78
6.4	Path planning in cluttered space . . . . .	83
6.5	Summary . . . . .	86
<b>7</b>	<b>Simulations and Results - 3D</b>	<b>87</b>
7.1	Simulations and Results - Dubins . . . . .	87
7.2	Simulations and Results - PH . . . . .	95
7.3	Summary . . . . .	96
<b>8</b>	<b>Conclusions and Future Work</b>	<b>100</b>
8.1	Discussions and Conclusions . . . . .	100
8.2	Future Work . . . . .	101
<b>A</b>	<b>Differential geometry</b>	<b>103</b>
A.1	Frenet Serret equations . . . . .	105
A.2	Importance of Curvature and Torsion . . . . .	106
	<b>References</b>	<b>103</b>
<b>B</b>	<b>Pythagorean Hodograph</b>	<b>107</b>
<b>C</b>	<b>Quaternion</b>	<b>109</b>
C.1	Properties of quaternion . . . . .	109



---

# List of Figures

1.1	Hierarchical character of co-operative controller of multiple UAVs . . . .	6
1.2	A block diagram approach to path planing . . . . .	7
1.3	Path planning of UAVs . . . . .	7
1.4	Problem formulation . . . . .	10
2.1	Dubins - CLC & CCC paths . . . . .	17
2.2	Dubins - Design of CLC path . . . . .	18
2.3	Tangent Circles . . . . .	20
2.4	Dubins paths with $\theta_f$ as a free variable . . . . .	20
2.5	Dubins Arc Geometry . . . . .	22
2.6	Curvature profiles of Dubins paths . . . . .	26
2.7	Curvature profile of a clothoid . . . . .	27
2.8	Path with clothoid Arc Geometry . . . . .	30
2.9	Dubins path vs. PH path . . . . .	36
2.10	Evolution of PH path of curvature continuity . . . . .	38
3.1	3D Dubins Manoeuver of UAV . . . . .	43
3.2	Dubins Arc Geometry . . . . .	44

## LIST OF FIGURES

4.1	Safe Flight Path . . . . .	60
5.1	UAVs in cluttered environment . . . . .	65
5.2	Threat Detection by intersection . . . . .	67
5.3	Measuring safety distance . . . . .	68
5.4	Flyable paths of two UAVs in a cluttered environment . . . . .	69
5.5	Replanning the path of UAV2 by curvature adjustment . . . . .	69
5.6	Replanning the path of UAV2 by creating an intermediate way-point . . . . .	70
6.1	Shortest flyable paths of UAVs - Dubins 2D . . . . .	73
6.2	Paths of equal length -Dubins 2D . . . . .	74
6.3	Separation distance for paths of first four combinations - Dubins 2D . . . . .	75
6.4	Separation distance for paths of second four combinations - Dubins 2D . . . . .	76
6.5	Separation distance for paths of last two combinations - Dubins 2D . . . . .	77
6.6	Flyable paths of UAVs - Clothoid 2D . . . . .	77
6.7	Paths of equal lengths - Clothoid 2D . . . . .	78
6.8	Dubins and PH Paths of UAV1 . . . . .	79
6.9	Dubins and PH Paths of UAV2 . . . . .	79
6.10	Dubins and PH Paths of UAV3 . . . . .	80
6.11	Flyable PH with offset paths of UAV1 . . . . .	80
6.12	Flyable PH with offset paths of UAV2 . . . . .	81
6.13	Flyable PH with offset paths of UAV3 . . . . .	81
6.14	PH paths of equal lengths . . . . .	82
6.15	PH Paths of UAVs, equal lengths with offset paths and safety rings . . . . .	82
6.16	PH Paths of UAVs, equal lengths elevated at constant altitude . . . . .	83
6.17	Initial paths (only tangent continuity) - PH 2D in cluttered space . . . . .	84
6.18	Flyable paths - PH 2D in cluttered space . . . . .	84
6.19	Feasible (safe and flyable) paths - PH 2D in cluttered space . . . . .	85
6.20	Paths of equal lengths - PH 2D in cluttered space . . . . .	85
7.1	Flyable Paths of UAVs - Dubins 3D . . . . .	88

## LIST OF FIGURES

---

7.2	Path of equal lengths - Dubins 3D . . . . .	89
7.3	Flight path with safety tubes - Dubins 3D . . . . .	90
7.4	Paths of equal lengths - UAV1 & UAV2 - Dubins 3D . . . . .	91
7.5	Paths of equal lengths - UAV1 & UAV3 - Dubins 3D . . . . .	92
7.6	Paths of equal lengths - UAV2 & UAV3 - Dubins 3D . . . . .	93
7.7	Flight path intersections - Dubins 3D . . . . .	94
7.8	Curvatures variation of Initial path of UAV1- PH 3D . . . . .	95
7.9	Curvatures variation of Initial path of UAV2 - PH 3D . . . . .	96
7.10	Flyable path of UAV1- PH 3D . . . . .	97
7.11	Flyable path of UAV2- PH 3D . . . . .	97
7.12	Curvatures variation: Flyable path of UAV1- PH 3D . . . . .	98
7.13	Curvatures variation: Flyable path of UAV2- PH 3D . . . . .	98
7.14	Curvatures variation: Feasible path of UAV2 - PH 3D (Length is equal to that of UAV1) . . . . .	99
7.15	Paths of equal lengths - PH 3D . . . . .	99
A.1	Frenet-Serret Frame . . . . .	105



---

# Introduction

**T**he momentum is increasing to consider using UAVs in a wide range of applications like weather and atmospheric research, reconnaissance and surveillance, conventional combat roles and innovative roles that were not previously possible (e.g., dull, dirty, and dangerous missions), such as operations in chemical and biological weapons environments and operations that require micro air vehicles [1]. The forecast applications of unmanned aerial vehicles in the military accelerate the growth of UAV markets in commercial and academic sectors by increasing research opportunities [2]. Autonomous vehicles on land, on air, in space or in water - together called Autonomous Systems will play a major role in near future.

Advances in avionics, GPS-based navigation, and flight control techniques fuelled the use of Unmanned Aerial Vehicles (UAVs) in commercial and military applications. Unmanned Air Vehicles of the future will be more autonomous than the remotely piloted reconnaissance platforms in use today. One of the open issues in their development is path planning. A path planning algorithm produces one or more safe flyable paths for UAVs. The path has to be of minimal length, subject to the stealthy constraint. As the UAV has limited range, the time spent surveying should be minimized, so the path length should always be a factor in the algorithm. Also, the path should be feasible for the aircraft to follow. The trajectory has to meet the speed and turn limits of the UAVs. The path-planning algorithm must be compatible with the cooperative nature envisioned for the UAV. Finally, path-planning algorithms are expected to be coded in software that runs on an airborne processor. Thus, they must

be computationally efficient and real-time, enabling the UAV to re-plan its trajectory if needed.

### 1.1 Path planning - An overview

Any autonomous vehicle, in fact any autonomous system involving mobility needs path-planning. Path-planning is widely documented in ground robotics and manipulator systems. However, the technological advances in the field of robotics extend its horizon: on land - Unmanned Ground Robot, in water - Unmanned Underwater Robot, in Air - Unmanned Aerial Robot. In all these applications, path planning has an integral part and plays an important role. This is understood from various references, for example, [3, 4] in ground robotics, [5, 6, 7, 8] on aerial vehicles, [9, 10, 11, 12] in underwater vehicles, and [13, 14] in space. The classic example of two-dimensional case of path planning is that of a mobile robot. In fact, the idea of the path-planning was originated from the field of robotics.

Early approaches to solving path planning problem were focused on *(i)* road map methods such as: Visibility graph, Voronoi diagram, *(ii)* cell decomposition and *(iii)* potential field method. The road map methods work on configuration space, where the robot shrinks to a point while the workspace grows. The start and goal points are connected by a network of lines. A\* algorithm [15] is used to find the shortest path. In the cell decomposition method, the space is divided into small regions, called cells. A connectivity graph is created among the free cells, which connect the start and goal nodes. The potential field method is based on the principles of electric potential theory that like fields repulse each other while the unlike fields attract each other. In this manner, a potential field is produced to attract the robot to the goal. One of the problems in this approach is local minima, which may be produced in the positions of obstacles. A detailed work on these methods can be found in the book by Latombe [16]. An important observation is that all these approaches produce only a route to reach the goal point. But, a route may not be flyable, driveable or maneuverable. A flyable path meets the kinematic constraints and the imposed dynamics of the robot. Therefore, the attention turned towards the development of paths which can be driveable, flyable and maneuverable.

Dubins [17] showed in his work that the shortest path between two vectors in a plane and meets minimum bound on turning radius is a composite path formed by the segments of line and circular arcs. This paper got a wide-spread attention by the research community and is extensively cited in ground robotic works [16, 18, 19, 20]

and airborne problems [21, 22, 23, 24]. Later the real time incident - parking of a car motivates the development of the shortest path for vehicle which can move forward and backward. Reeds and Shepp [25] developed the shortest path for a vehicle which can move both forward and backward. Laumond [26] addresses the path planning for car-like vehicles using composite path made of circular and clothoid arcs. B-splines [27], quintic polynomials [28], polar splines [29], clothoid [30], cubic spirals [31],  $G^2$  splines [32] have been used for path planning of mobile robots. Robot path planning using the Voronoi diagram has been studied widely since the mid-1980s [33], and in late 1990s, the focus on coordinated path planning of multiple robots began. Though, much of the work done on path planning is carried out in ground robotics, the approaches could not directly be applied to the Unmanned Aerial Vehicles. Because, the path of a UAV is limited by the high-G turns and also that it has a threshold speed below which it can not fly.

Optimization techniques such as probabilistic methods, mixed integer linear programming, and genetic programming are applied to path planning of UAVs. These techniques produce paths by optimizing certain cost function. The cost functions differ based on the applications such as minimum time arrival, optimizing fuel consumption and coordinated attack. They are mostly the search algorithms. Probabilistic Road Maps (PRMs) [34, 35] connect the starting point to the goal point by adding successive trajectory to a pre-computed route. In another approach called Rapidly-exploring Random Trees (RRTs) [36, 37], extends a tree of trajectory segments from the start point to the goal point. The every successive trajectory in the tree is selected randomly by connecting to a closest point in the existing tree. The potential field algorithm [38] solves the path planning by generating an attractive field towards the goal point and repulsive field at the obstacles. These approaches are randomized path planning approaches mainly involved search algorithms. They result in a route planning. But the route cannot always be flyable.

In another approach [39, 40], a Dijkstra- like method is suggested for solving a continuous-space shortest path problem in  $2D$  plane by optimization. An analytical and discrete optimization approaches is used for optimal risk path generation in two-dimensional space with constant Radar Cross Section, arbitrary number of sensors and a constraint on path length [8]. Probabilistic method is applied to path planning considering positional uncertainty of threat regions [41]. The final path is refined with circular arcs at the points of line joining. Use of Mixed Integer Linear Programming (MILP) for path planning applications can be found in [42, 43, 44, 45]. MILP is an application of the operational research method, called Linear Programming with integer or binary constraints. These constraints are used for logical decisions such

## 1. INTRODUCTION

---

as turn left, move up. This method produces safe route for UAVs. But, the route has to be smoothed further to make it flyable. Also, the optimization methods are associated with high computational time. Accomplishing the mission objectives with physical and functional limitations of UAVs further increase the complexity of solution to path planning problem [46, 47]. An overview of coordinated control of UAVs and their complexities can be found in [48].

Another widespread approach is the use of Voronoi diagram. Voronoi diagram is used to produce polygonal paths connecting start and goal locations for each UAV by minimizing radar detection. Latter the path is refined by adding fillets of minimum turning radius. The simultaneous arrival is coordinated by a high level manager based on the sensitivity function (cost vs time of arrival) sent by each UAV [49]. Similar approach is adopted in [50] where an analogy of a chain connected by sequences of spring-mass-damper system to the UAV path is used. The ends of the chain are located at the initial and final configurations. The threats induce a repulsive force which cause the masses in the chain to move away from the threats. However, this method involves complexity in solving ODEs with curvature constraints. Also, accumulation of only a few masses around the threat location will lead to coarse path resolution which is undesirable. The above approach is extended by replacing the spring-damper system with rigid links between masses to eliminate sharp corners [51]. However, this method does not guarantee that the resultant path is flyable by an UAV. Later in [52], the Voronoi path is interpolated with a series of cubic splines assigning a cost to each obstacle/threat position.

The Voronoi diagram produces route for each UAV and the routes are refined to make them flyable. Also, in the optimization approaches, the final outcome is a route planning, satisfying certain constraints. If the route is refined by adding fillets, the resulting path is a series of lines and arcs, which is a subset of Dubins path [17]. The optimization methods, randomized search approaches, and Voronoi diagram approach use an exhaustive search and computational methods which result in route planning. The route planning does not consider the kinematic constraints of the path. Also, reactive behavior of the UAV needs a flyable path at any point of its flight. In such situation the route planning would be a handicap. For this reason it appeared reasonable to attempt to use the curves directly in path planning. In this manner, planar and spatial Dubins path [22, 53], Pythagorean Hodograph [54, 55] and 2D clothoid [56] are used to solve the problem of simultaneous arrival on target. In contrast to other approaches, this approach divides the path planning into two phases: (*i*) producing flyable paths and (*ii*) producing safe, flyable (feasible) paths. In the first phase a flyable path is produced which satisfies the kinematic and dynamic constraints of

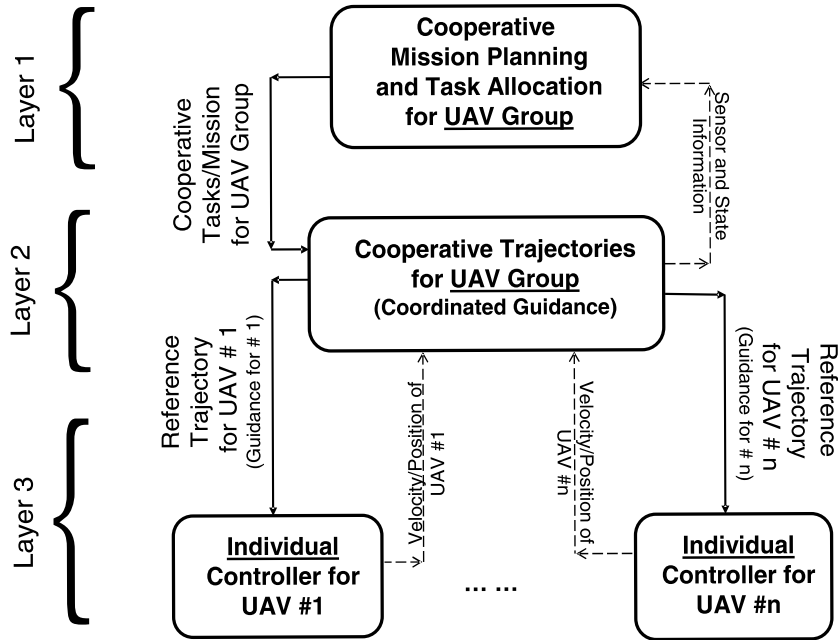
the UAVs. The flyable path connects the way-points, thus produces sequence of flyable paths. This path is also useful in reactive path planning, where the UAV needs to take an evasive maneuver during conflicts. In the second phase, the flyable paths are tuned to achieve the mission.

## 1.2 Coordinated Guidance

While employing a group of UAVs for a mission, it is important for UAVs to cooperate among themselves and obey the constraints of environment. The coordination or cooperation is established either by a preplanned course of actions or set by communication and feedback. In this context, path planning for a swarm of UAVs can be considered as a part of coordinated guidance and control. Figure (1.1) shows a schematic sketch of various levels in hierarchical character of coordinated closed-loop guidance and control of multiple UAVs. The mission objective and task allocation of a group of UAVs are decided in layer 1, which is high-level planning. In practice, the mission or tasks are defined by a human operator, interacting with the co-operative controller in layer 2. The decision making in the layer 1 results in generation of co-operative trajectories in layer 2. The layer 2 produces coordinated trajectories for a swarm of UAVs, under which a reference trajectories are produced for each UAV. This is called cooperative path plan. Each  $i^{th}$  reference trajectory (guidance demand) generated in Layer 2 is followed by the individual controller of the  $i^{th}$  UAV in Layer 3. Thus, the overall controller is obtained by co-operation decided on level 1, and defined by the trajectory tracking requirements in level 2. Here we are interested in Layer 2, where the path planner produces feasible paths/trajectories for the UAVs.

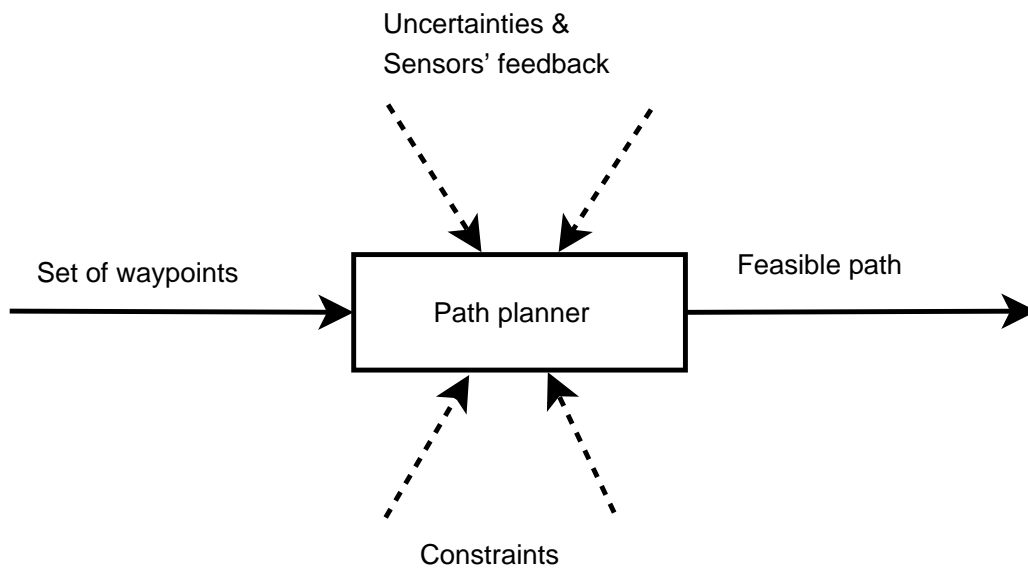
### 1.2.1 Coordinated Path Planning of Multiple UAVs

A path-planner connects points of interests by a path for an autonomous vehicle. In general, an autonomous vehicle can either be on land, in water, in air or in space. In this work, only Unmanned Aerial Vehicles are considered. Generally, the points of interests are predefined or can be generated by sensor feedback or obtained from a Voronoi diagram. So, the input to the path planner is a set of points or poses. A pose or configuration is a set of position and orientation variables. The outcome of the path planner is a path connecting the input data. Therefore, it is worthwhile to consider the path planner as a black-box with set of points as input and with path as an output. Figure (1.2) illustrates the black diagram approach to path planning. For every set of inputs (a set of way-points/poses), it produces a feasible path con-



*Figure 1.1: Hierarchical character of co-operative controller of multiple UAVs*

sidering the uncertainties, constraints and feedback. As the properties of the path influence the motion of the vehicle, it is necessary to discuss the characteristics of the path. Any autonomous vehicle needs a path to move from one location to another. Once the path is produced, it is necessary to ensure the path is safe to fly and also it guarantees the UAVs accomplish their mission. Basically, a plan is needed to achieve the mission safely. Here a question arises: How and When to plan. One way is to integrate the path generation with the planning. This approach is similar to the route planning by optimization methods discussed in section (1.1). But, this approach needs further refinement to produce flyable paths and also this method is computationally intensive. Another possibility is to separate the two processes: path generation and planning. Now two possibilities arise: when to plan - either before or after the path generation. This question is answered in section 1.4. This section defines the path planning. In general, path planning algorithm produces a feasible path for a UAV to fly from one location to another. The initial and final locations are characterized with poses, also called configurations. A pose is a set of position and orientation variable. For example  $P(x, y, z, \theta, \phi)$  is a pose  $P$ , where  $(x, y, z)$  is position coordinate, and  $(\theta, \phi)$  is orientation (The position coordinates are assumed to lie at the centre of a sphere of minimum turning radius. The direction of the tangent vector is specified in spherical coordinates, which reduce the number of input variables



**Figure 1.2:** A block diagram approach to path planing



**Figure 1.3:** UAVs have to fly from one location to another either independently or in coordination with one another. The starting and finishing locations, respectively, are called as base and target. However, the locations can be any two way-points



## 1. INTRODUCTION

---

to the path planner). The operating environment of the UAVs may be clutter-free (figure 1.3) or cluttered (figure 5.1). The path planning produces one or more feasible path(s) connecting two or more poses/configurations. The feasible path is (i) flyable, that is it meets the kinematic constraints and (ii) safe to fly, that is it guarantees the safety of the UAVs while achieving the mission.

Mathematically, a path can be characterized by a curve. Therefore it can be argued that geometrically, the path planning can be considered as geometric evolution of curve. The path planning connects an initial pose  $P_s(x_s, y_s, z_s, \phi_s, \theta_s)$ , to a final pose  $P_f(x_f, y_f, z_f, \phi_f, \theta_f)$  by a feasible path  $r(t)$ , which satisfies maximum curvatures bound  $\kappa_{max}$ , maximum torsion bound  $\tau_{max}$ , and constraint  $\mathbb{I}$ . The values of  $\kappa_{max}$  and  $\tau_{max}$  define the kinematic limits of the UAV in space. Also, these are the only two parameters which determine a curve in space. In two dimensions, only curvature determines the curve [[57], [58]]. A curve satisfying the curvature constraints imposed by the dynamics of the UAV is called flyable path. The importance of curvatures is discussed in appendix A. A feasible path is both flyable (meets kinematic and dynamic constraints) and safe to fly (no collisions). The safety constraints is represented by  $\mathbb{II}$ . The safety constraints are discussed in chapter 4.

$$\boxed{P_s(x_s, y_s, \phi_s, \theta_s) \xrightarrow{r(t)} P_f(x_f, y_f, z_f, \phi_f, \theta_f),} \quad (1.2.1)$$

$$|\kappa(t)| < \kappa_{max}, |\tau(t)| < \tau_{max}, \text{ and } \mathbb{II}$$

where  $t$  is a parameter.

Extending the above equation (3.0.1) for a group of  $N$  UAVs

$$\boxed{P_{si}(x_{si}, y_{si}, z_{si}, \phi_{si}, \theta_{si}) \xrightarrow{r_i(t)} P_{fi}(x_{fi}, y_{fi}, z_{fi}, \phi_{fi}, \theta_{fi}),} \quad (1.2.2)$$

$$|\kappa_i(t)| < \kappa_{i,max}, |\tau_i(t)| < \tau_{i,max}, \text{ and } \mathbb{II}$$

where the suffix  $i$  represents the  $i^{th}$  UAV,  $i = 1 \dots N$ .

The path  $r_i(t)$  in equations (1.2.1, and 1.2.2) is either a single polynomial curve or a composite curve and its properties change with  $t$ . Such a path is useful in predicting the future position and attitude of the UAVs. Also, it helps the path planner to consider the kinematic limits at the early phase of the path planning. Besides, the dynamics can be estimated by coupling the kinematic parameters with inertial properties of the UAVs.



### 1.3 Problem Statement

Consider  $N$  UAVs deployed for simultaneous arrival on target. All the UAVs leave the base at time  $t_{base}$  and have to reach the target at the time  $t_{target}$ , where  $t_{base} < t_{target}$ . The base and target can be connected through a set of way-points. The problem is simplified by taking the base and the target as two successive way-points. The poses of each UAV at the base and the target are predefined. Figure (1.4) shows the schematic of the mission. The UAVs are assumed to have equal kinematic and dynamic capabilities and flying at equal speed in a free space. Each UAV is assumed to lie at the centre of two concentric spheres. The inner sphere is called safety sphere with radius  $R_s$ , while the outer one is called com-sphere of radius  $R_r$ , which represents the sensor range such that  $R_r \gg R_s > (1/\kappa_{max})$ . For any two paths, if the intersection of safety-spheres is empty, the paths are safe to fly. Otherwise, the path(s) has to be replanned or adjusted to avoid inter-collision of UAVs. However, this safety criterion can be extended to obstacle or threat avoidance. Let the configurations of  $i^{th}$  UAV at the base and target respectively be  $P_i(x_{si}, y_{si}, z_{si}, \phi_{si}, \theta_{si})$  and  $P_f(x_{fi}, y_{fi}, z_{fi}, \phi_{fi}, \theta_{fi})$ . With the  $k$  number of constraints, the problem is formulated as:

$$\boxed{\begin{aligned} P_{si}(x_{si}, y_{si}, z_{si}, \phi_{si}, \theta_{si}) \xrightarrow{r_i(t)} P_{fi}(x_{fi}, y_{fi}, z_{fi}, \phi_{fi}, \theta_{fi}), \\ |\kappa_i(t)| < \kappa_{i,max}, \quad |\tau_i(t)| < \tau_{i,max} \quad \text{and} \quad \prod_k \end{aligned}} \quad (1.3.1)$$

### 1.4 Solution Approach

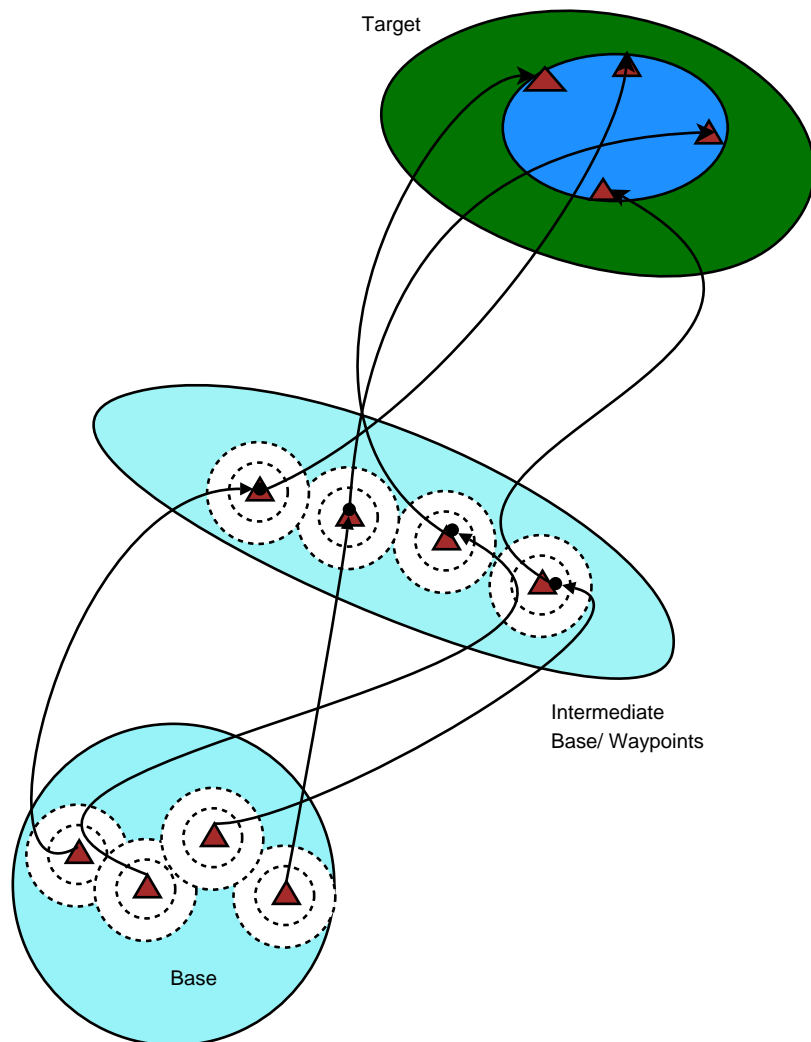
The simultaneous arrival can be achieved by producing paths equal in length for constant speed UAVs or the paths of unequal lengths for variable speed UAVs. With the constant speed UAVs, the simultaneous arrival is achieved with constant speed profile, while the variable speed UAVs use variable speed profile. In this thesis, only constant speed UAVs are considered. As all the UAVs are flying at same constant speed, producing paths of equal length ensures the simultaneous arrival. Accordingly, the equation (1.3.1) changes into:

$$\boxed{\begin{aligned} P_{si}(x_{si}, y_{si}, z_{si}, \phi_{si}, \theta_{si}) \xrightarrow{r_i(t)} P_{fi}(x_{fi}, y_{fi}, z_{fi}, \phi_{fi}, \theta_{fi}), \\ |\kappa_i(t)| < \kappa_{i,max}, \quad |\tau_i(t)| < \tau_{i,max}, \quad s_i(t) = s_j(t) \quad \text{and} \quad \prod_k \end{aligned}} \quad (1.4.1)$$

where  $s_i(t)$  and  $s_j(t)$  are the path lengths of  $i^{th}$  for  $j^{th}$  UAVs, and  $i, j = 1 \dots N$ .

## 1. INTRODUCTION

---



**Figure 1.4:** Schematic figure showing Problem formulation. Each UAV is assumed to be surrounded by two concentric circles: inner one is called safety-circle and the outer is called com-circle. The path planning can be between any two way-points

The path-length  $s(t)$  of the path  $r(t) = \{x(t), y(t), z(t)\}$  is:

$$s(t) = \int_{t_1}^{t_2} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} dt, \quad t \in [t_1, t_2] \quad (1.4.2)$$

where  $\dot{x}(t) = \frac{dx}{dt}$ ,  $\dot{z}(t) = \frac{dz}{dt}$  and  $\dot{y}(t) = \frac{dy}{dt}$  are hodographs.

The equation (1.4.1) can be solved by optimization techniques. But it will be computationally intensive. And also the resultant path need not be an optimal one. Considering these difficulties, the solution is divided into two phases: (i) Producing flyable paths and (ii) Producing safe flyable (feasible) paths of same lengths. A flyable path meets the kinematic constraints of the UAV. A safe flyable path is a flyable path that guarantees the safety of the UAV. In the first phase, a flyable path is produced, which is a polynomial curve may single or composite. In the second phase, the flyable paths are tuned to produce feasible paths of equal lengths. In short, the solution involves first the generation of flyable paths and is followed by planning of flyable paths. Therefore it is convenient to write:

**PATH + PLANNING  $\longrightarrow$  PATH PLANNING**

First, the flyable path is produced for each UAV. Three types of paths are studied here: (i) Dubins path, (ii) Pythagorean Hodograph (PH) path and (iii) clothoid path. The Dubins and Clothid are produced as composite paths, while the PH is a single path.

$$\begin{aligned} P_{si}(x_{si}, y_{si}, z_{si}, \phi_{si}, \theta_{si}) \xrightarrow{r_i(t)} P_{fi}(x_{fi}, y_{fi}, z_{fi}, \phi_{fi}, \theta_{fi}), \\ |\kappa_i(t)| < \kappa_{i,max} \quad |\tau_i(t)| < \tau_{i,max} \end{aligned} \quad (1.4.3)$$

In the second phase of the path planning, the flyable paths are tuned to meet the safety conditions by satisfying the equation (1.3.1). Finally, the safe paths are made equal in length by satisfying equation (1.4.1) for simultaneous arrival on target.

## 1.5 Thesis contributions

- ☞ This thesis focusses on path planning of multiple UAVs for simultaneous arrival on target (mission). The solution to path planning divided into two phases. In the first phase, the paths are produced to meet the curvature constraints, called flyable paths. In the second phase, the flyable paths are tuned to meet the mission. In contrast to the existing approaches, this approach uses the flyable path directly into the path planning. This is advantageous in producing the flyable path between any two way-points or poses obtained by feedback or

## 1. INTRODUCTION

---

commanded by the mission planner. Also, it needs a set of points as an input to produce the paths. Hence it is not computationally intensive.

- ☞ The flyable paths are produced using Dubins, clothoid - composite curves and Pythagorean Hodograph - single curve. The Dubins path is generated using the principles of Euclidean and Differential geometries. It is shown that the results obtained by both methods are equivalent. However, the differential geometry is advantageous in generalizing the paths as it has only maximum of two parameters to define the path: curvature and torsion in 3D and only curvature in 2D. The PH path is produced by curvature optimization which is done by increasing the length of the boundary tangent vectors.
- ☞ Also, the flyable paths are generated in three dimensions. The principles used in producing 2D Dubins path by differential geometry are extended into three dimensional Dubins path. The 3D path is obtained by an initial rotation and is followed by 2D Dubins path in a common intersecting plane connecting the initial and final poses. The spatial PH path is developed for curvature continuity. The multiple constraints are met by increasing the boundary tangent vectors.
- ☞ Throughout the thesis, the fundamental principle - curvatures determine a path and its properties is used. This principle is used to produce the flyable path and also to tune the path to meet the various constraints.
- ☞ In a free space maneuver, two safety conditions are defined to avoid inter-collision avoidance. Two approaches are defined for path planning in cluttered space. The threat detection and avoidance involves the detection by intersection of path with threat regions and is avoided by replanning the path either by increasing the curvature or by creating an intermediate way-point.
- ☞ The simultaneous arrival on target is solved by producing paths of equal lengths. However, the proposed method can also be applied to variable speed UAVs. Because, this approach is based on paths connecting the any set of poses.

## 1.6 Disseminations, presentations from this thesis

### 1.6.1 Conference papers

1. M.Shanmugavel and A.Tsourdos and R.Żbikowski and B.A.White. *Path Planning of Multiple UAVs Using Dubins Sets*. AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California, Aug. 15 – 18, 2005.

2. Madhavan Shanmugavel, A.Tsourdos, R.Żbikowski, and B.A.White. *Path planning of multiple UAVs in an environment of restricted regions*, IMECE2005-79682. Proceedings of IMECE2005, ASME International Mechanical Engineering Congress and Exposition November 5 – 11, 2005, Orlando, Florida, USA, 2005.
3. Madhavan Shanmugavel, A.Tsourdos, R.Żbikowski, B.A.White, C.A.Rabbath, and N.Lechevin. *A solution to simultaneous arrival of multiple UAVs using Pythagorean Hodograph curves*. American Control Conference, Minneapolis, June 14 – 16, 2006.
4. Madhavan Shanmugavel, Antonios Tsourdos, Rafał Żbikowski, and Brian.A.White. *3D Dubins Sets Based Coordinated Path Planning for Swarm of UAVs*, AIAA-2006-6211. AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado, August 21 – 24, 2006.
5. Madhavan Shanmugavel, A.Tsourdos, R.Żbikowski, and B.A.White. *Path planning of multiple UAVs with Clothoid curves*. IFAC ACA 2007 (accepted).
6. Madhavan Shanmugavel, A.Tsourdos, R.Żbikowski, and B.A.White. *3D path planning for multiple UAVs using Pythagorean Hodograph curves*. AIAA GNC 2007 (accepted).
7. Antonios Tsourdos, Brian White, Rafał Żbikowski, Peter Silson, Suresh Jeyaraman and Madhavan Shanmugavel *A Formal Model Approach for the Analysis and Validation of the Cooperative Path Planning of a UAV Team*. IEE Seminar on Autonomous Agents in Control. p. 67 – 73, 2005 .
8. Nelson Gonçalves, Madhavan Shanmugavel, João Sequeira, Antonios Tsourdos, Brian White, and M.Isabel Ribeiro, *Indoor active surveillance*. 13<sup>th</sup> IEEE International Conference on Methods and Models in Automation and Robotics. Szczecin, Poland on 2730, August 2007 (accepted).

### 1.6.2 Journal paper

1. Madhavan Shanmugavel, A.Tsourdos, R.Żbikowski, B.A.White. *Differential Geometric Path Planning of Multiple UAVs*. ASME JDME 2007 (accepted).

## 1.7 Organization of the thesis

The thesis is divided into three major parts: (i) Producing flyable paths, (ii) Planning to meet the mission objective and (iii) simulations, results and conclusions. It is divided into eight chapters.

## 1. INTRODUCTION

---

**Chapter 2** deals with producing flyable paths in two dimensions. Three types of paths are considered. The chapter begins with the design of Dubins path, because, this is the shortest path between two poses in 2D and also it is simple. The Dubins path is produced using the principles of Euclidean and Differential geometries. Besides, the equivalence of results from both approaches, it is shown that the differential geometric principles are advantageous in generalization of the path. The lack of curvature continuity of the Dubins path motivates the use of other paths. In this respect, a single path - Pythagorean Hodograph and a composite path - clothoid and line segment are considered. The circular arcs in 2D Dubins path are approximated with clothoid segments to produce a smooth path. The last part is dealt with the Pythagorean Hodograph (PH) curve known for its rational properties. A procedure is established to derive a PH path of curvature continuity.

**Chapter 3** discusses three dimensional path planning. It extends the principles used in chapter 2. The Dubins path is produced in 3D using the principles of differential geometry. The clothoid path is not discussed as the design is similar to that the 3D Dubins path. PH path is developed in 3D for the use in path planning. The composite versions of Dubins path is generated by finding the common intersecting plane between the initial and final poses with an initial rotation at the start pose. The spatial PH path is developed with quaternion and the curvature and torsion are met by increasing the tangent vectors at the initial and final poses.

**Chapter 4** discusses the solution to the simultaneous arrival on target. The previous two chapters discuss how to produce flyable paths. This first part of this chapter details the various constraints of path planning. The curvature constraint which defines the kinematics of the UAV and the safety constraints for inter-collision avoidance are discussed. The flyable paths are tested for safety conditions. A solution is achieved by increasing the lengths to that of a reference path.

**Chapter 5** describes algorithms for detecting and avoiding the threats or obstacles. Detection precedes avoidance. The region of known threat is detected by testing whether the path intersects the boundary of the threat region. This is simply implemented by testing whether the path is inclusive of the region. In the case of unknown threat region, it is necessary to locate the threat. This needs the relative distance of the threat region with respect to the UAVs. However, the measurement is not necessary unless UAVs has to conduct mapping. As the mission is the simultaneous arrival on target, the UAVs have to detect and avoid the threat rather than mapping. In this respect, the com-circle discussed in the chapter 1 is used to detect the threat and the safety circle is used to test and avoid the threat region. The PH path uses its

offset path in 2D and tubes in 3D to define the safety region (circular in shape). This chapter describes the threat avoidance with an example of entry into the restrictive regions.

**Chapter 6** describes the simulation results in two dimensions. It is assumed that the UAVs are flying at constant altitude. The simulation results of composite paths, Dubins and Clothoid and the PH path are discussed in this chapter. The rings around the PH paths are generated to visualize the use of offset paths defining safety.

**Chapter 7** discusses the simulation results of the Dubins and PH paths. The clothoid path is not discussed as it is similar to that of the Dubins path. The tube around the paths are generated to visualize the use of canal surfaces defining safety.

**Chapter 8** discusses the conclusions and future work. As every research does not have an end, there is always a scope of further work.

---

## Producing flyable paths -2D

**F**lyable paths satisfy maximum curvature bound of the UAVs. In this chapter UAVs flying at constant altitude are considered. The constant altitude flight have coplanar trajectory. Therefore, two dimensional paths can handle this situation. A flyable path may be a single or a composite path. Three types of flyable paths are studied. Firstly the Dubins path which is the shortest path is studied and is followed by Dubins-like path but with Clothoid arcs. Finally, the Pythagorean Hodograph which is a single path known for its rational properties. A simple case of producing path between two poses is considered. This can be extended into any number of way-points/poses. The pose in 2D composes the position coordinates  $(x, y)$ , and orientation  $\theta$ . The general equation of producing flyable paths (equation 1.4.3) reduces into:

$$P_s(x_s, y_s, \theta_s) \xrightarrow{r(t)} P_f(x_f, y_f, \theta_f), \quad |\kappa(t)| < \kappa_{max} \quad (2.0.1)$$

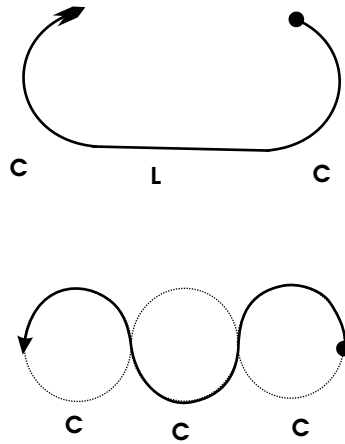
### 2.1 Producing Flyable Paths - Dubins

Motion in a plane composes rectilinear and turning or angular motions. A straight line provides the shortest distance for the rectilinear motion and the circular arc provides the shortest distance for an angular motion. Also, the arc provides the constant turning radius, which satisfies the maximum curvature constraint. This is the basic idea of Dubins path [17]. The Dubins path is the shortest path between two vectors in a plane and the path meets the minimum bound on turning radius. The Dubins



## 2.1 Producing Flyable Paths - Dubins

path provides the shortest path for forward moving vehicle. The Dubins path is a composite path formed either by two circular arcs connected by a common tangent or three consecutive tangential circular arcs, or a subset of either of these two. The first path is CLC path and the second one is CCC path and the last one is either a CL, LC or CC, where 'C' stands for Circular segment and 'L' stands for Line segment. Combining these two curves, obviously forms the shortest path between two poses. In this work, we focus on Dubins path of CLC type. Here, two approaches of producing the Dubins path are studied. In the first approach, principles of Euclidean geometry are used and in the second, principles of differential geometry are used. From this point, Dubins path connotes the CLC path.



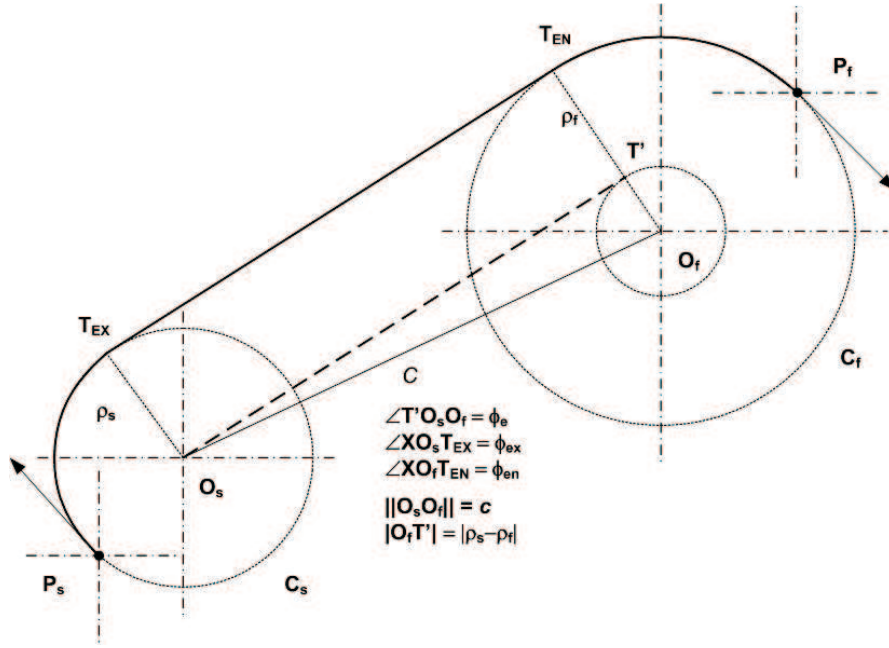
*Figure 2.1: Dubins - CLC & CCC paths*

### 2.1.1 Producing Dubins Path using principles of Euclidean Geometry

In Euclidean geometry, the Dubins path is produced by drawing common tangents between two circular arcs. The common tangents connect the arcs externally and internally (diagonally), respectively called external and internal tangents. Here the Dubins path produced by an external tangent is explained. The case of internal tangent is analogous. Following are the procedures to produce a Dubins path geometrically. Refer figure (2.1.1). Consider the following input parameters.

- i) Initial pose:  $P_s(x_s, y_s, \theta_s)$
- ii) Final pose:  $P_f(x_f, y_f, \theta_f)$
- iii) Initial turning radius:  $\rho_s (= \frac{1}{\kappa_s})$  and
- iv) Final turning radius:  $\rho_f (= \frac{1}{\kappa_f})$

## 2. PRODUCING FLYABLE PATHS -2D



**Figure 2.2:** Dubins - Design of CLC path

1. Find the centres of turning circles  $O_s(x_{cs}, y_{cs})$  and  $O_f(x_{cf}, y_{cf})$ :

$$(x_{cs}, y_{cs}) = (x_s \pm \rho_s \cos(\theta_s \pm \pi/2), y_s \pm \rho_s \sin(\theta_s \pm \pi/2)) \quad (2.1.1a)$$

$$(x_{cf}, y_{cf}) = (x_f \pm \rho_f \cos(\theta_f \pm \pi/2), y_f \pm \rho_f \sin(\theta_f \pm \pi/2)) \quad (2.1.1b)$$

where  $O_s$  and  $O_f$  are called primary circles represented by  $C_s$  and  $C_f$  respectively.

2. Draw a secondary circle of radius  $|\rho_f - \rho_s|$  at  $O_f$  for  $\rho_s \leq \rho_f$ .
3. Connect the centres  $O_s$  and  $O_f$  forms a line  $c$ , called centre line, where  $|c| = \sqrt{(x_{cs} - x_{cf})^2 + (y_{cs} - y_{cf})^2}$ .
4. Draw a perpendicular to  $c$  at  $O_f$ , which intersects the secondary circle at  $T'$  and the primary circle  $C_f$  at  $T_{EN}$ , called tangent entry point.
5. Connect the points  $O_s$  and  $T'$ .
6. Draw a line from  $O_s$  parallel to  $O_f T_{EN}$  which meets the  $C_s$  at  $T_{EX}$ , called tangent exit point.
7. Draw a line by connecting the points  $T_{EX}$  and  $T_{EN}$  which is parallel to the line  $O_s T'$ .
8. Connect the points  $P_s$  and  $T_{EX}$  by an arc of radius  $\rho_s$  and  $T_{EN}$  and  $P_f$  by an arc of radius  $\rho_f$ .
9. The composite path formed by the starting arc  $P_s T_{EX}$ , followed by the external tangent line  $T_{EX} T_{EN}$  and the ending arc  $T_{EN} P_f$ .

Calculations:

## 2.1 Producing Flyable Paths - Dubins

From figure (2.1.1), the triangle  $\triangle O_s O_f T'$  is a right angled triangle with hypotenuse  $O_s O_f$  and the other two sides are  $O_f T'$  and  $O_s T'$ , where  $\|O_s T'\| = |\rho_f - \rho_s|$ .

The included angle between  $O_s O_f$  and  $O_s T'$  is  $\phi_e$ .

$$\phi_e = \arcsin\left(\frac{\rho_f - \rho_s}{|c|}\right) \quad (2.1.2)$$

The slope of the line  $c$  is  $\psi$ .

$$\psi = \arctan\left(\frac{y_{cf} - y_{cs}}{x_{cf} - x_{cs}}\right) \quad (2.1.3)$$

The angles  $\phi_{ex} = \angle(XO_s T_{EX})$  and  $\phi_{en} = \angle(XO_f T_{EN})$  are calculated from the table (2.1.1).

**Table 2.1:** Calculation of tangent exit and entry points

Start-Turn	Finish-Turn	$\phi_e$	$\phi_{ex}$	$\phi_{en}$
Right	Right	$\arcsin\left(\frac{\rho_f - \rho_s}{c}\right)$	$\phi_e + \frac{\pi}{2} + \psi$	$\phi_e + \frac{\pi}{2} + \psi$
Left	Left	$\arcsin\left(\frac{\rho_f - \rho_s}{c}\right)$	$\phi_e - \frac{\pi}{2} + \psi$	$\phi_e - \frac{\pi}{2} + \psi$

The values of  $\phi_{ex}$  and  $\phi_{en}$ , the tangent exit and entry points are calculated as:

$$T_{EX} = (x_{cs} + \rho_s \cos(\phi_{ex}), y_{cs} + \rho_s \sin(\phi_{ex})) \quad (2.1.4a)$$

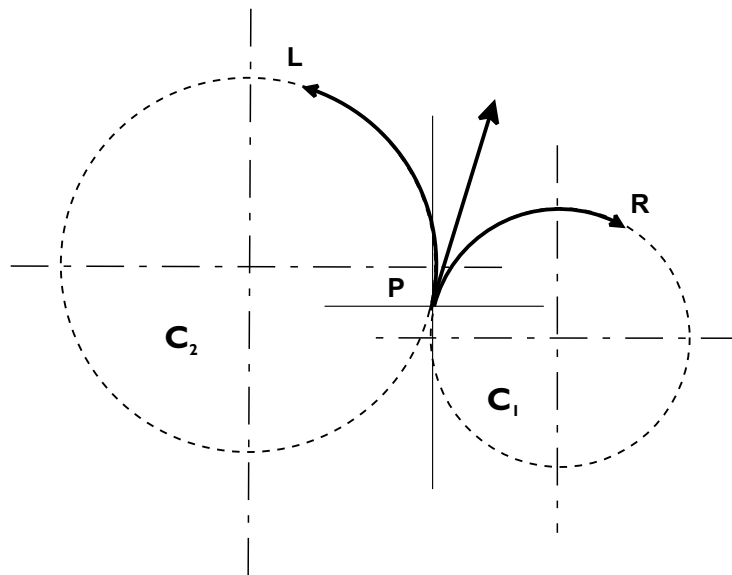
$$T_{EN} = (x_{cf} + \rho_f \cos(\phi_{en}), y_{cf} + \rho_f \sin(\phi_{en})) \quad (2.1.4b)$$

All the angles are assumed positive in counterclockwise direction. The path is also called RSR path owing to its Right turns at the ends. A similar procedure can be adopted to LSL path by drawing the secondary circle of radius  $|\rho_s - \rho_f|$ , where 'L' represents Left turn. The other two Dubins paths with internal tangents are RSL and LSR. These paths can be produced with secondary circle of radius  $|\rho_s + \rho_f|$ .

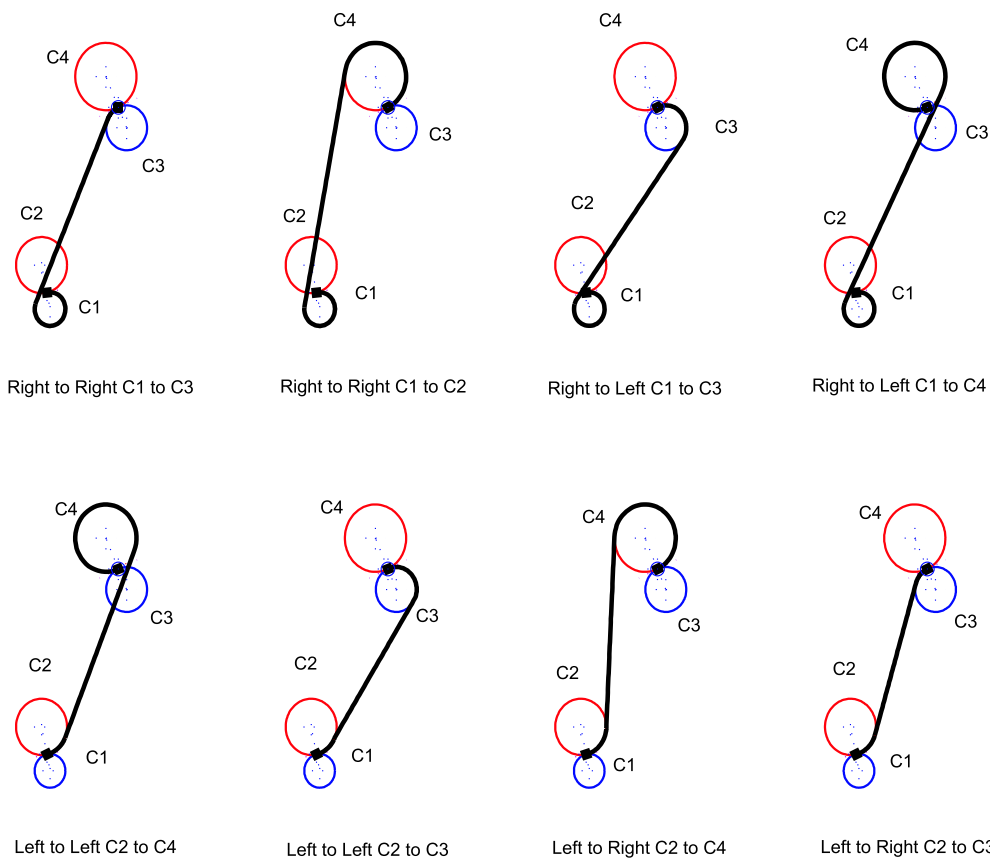
It is worth pointing out that the calculation of the tangent exit and entry points  $T_{EX}$  and  $T_{EN}$  is central in producing the Dubins path.

For a given pose, there are two circles tangent to it. Referring to the figure (2.3), the pose  $P$  have a right turn  $R$  on the arc  $C_1$  and a left turn  $L$  on the arc  $C_2$ . If either  $\theta_s$  or  $\theta_f$  is a free variable, a set of eight paths can be produced (figure 2.1.1). If both the orientations are free variables, a set of sixteen paths can be produced. The shortest path can be selected from the set of available paths.

## 2. PRODUCING FLYABLE PATHS -2D



**Figure 2.3:** Tangent Circles. For a given pose, there are two possible turns: Left and Right turn. Thus for a set of poses four possible turns are possible. This forms a set of Dubins path or simply called Dubins set



**Figure 2.4:** Dubins paths with  $\theta_f$  as a free variable. The path starts with either clockwise or counter clockwise direction. But it finishes with eight possible turns, hence a set of eight paths is produced

### 2.1.2 Existence of Dubins paths

Before calculating the path, it is important to know whether there exists a path between two poses. This saves the computational time. From the section (2.1.1) it is apparent that the existence of the Dubins path between two poses is determined by the existence of common tangents between the turning arcs. The common external tangent determines the existence of RSR and LSL paths, while the existence of RSL and LSR paths are determined by the common internal tangent.

The external tangent vanishes when the primary circles are inclusive of each other. The internal tangent vanishes when the primary circles intersect with each other. Both the conditions are determined by the central distance  $c$  and the turning radii  $\rho_s$  and  $\rho_f$ . But the centres of the primary circles are fixed by the radii of the arcs. Hence, the existence of the Dubins path for a pair of poses is simply a function of their turning radii.

$$\text{External tangent} : (c + \rho_s) > \rho_f, \rho_f > \rho_s \quad (2.1.5a)$$

$$\text{Internal tangent} : c > (\rho_s + \rho_f), \rho_f > \rho_s \quad (2.1.5b)$$

### 2.1.3 Length of the Dubins paths

The Dubins path is a composite path of two circular arcs and a straight line. Hence the path length is the sum of the lengths of individual path segments. Since the length of the common tangent connecting the arcs are decided by radii of the arcs, the length is also the function of the turning radii. Hence, the length of the path can be varied by changing the radii (curvatures). Also, any two paths can be made equal in length by simply varying the curvature of the arcs.

$$L_{Dubins} = L_{\text{arc,start}} + L_{\text{tangent}} + L_{\text{arc,finish}} \quad (2.1.6a)$$

$$L_{CLC} = \rho_s \alpha_s + L_t + \rho_f \alpha_f \quad (2.1.6b)$$

$$L_{Dubins} = f(\rho_s, \rho_f) \quad (2.1.6c)$$

where  $L_{Dubins}$  is length of the Dubins path,  $\alpha_s$  and  $\alpha_f$  are the included angles,  $\alpha_s = \phi_{ex}$ ,  $\alpha_f = \phi_{en}$  and  $L_t = ||T_{EX}T_{EN}||$ .

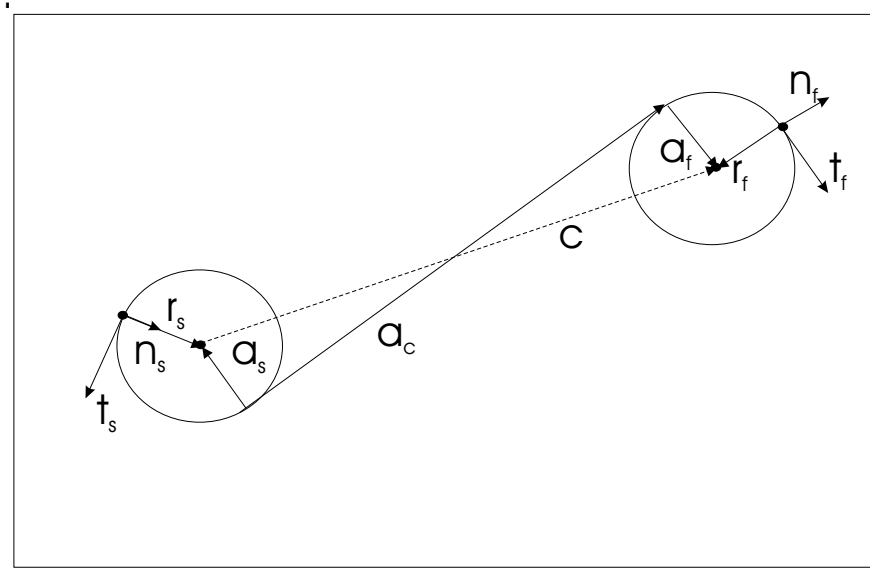
### 2.1.4 Producing Dubins Path using principles of Differential Geometry

The basic idea of using the principles of differential geometry is that the path can be determined by curvature in two dimensions and by curvature and torsion in three

## 2. PRODUCING FLYABLE PATHS -2D

dimensions. Also, the path is coordinate independent. The turning and twisting of the path is given by a moving trihedron called Frenet-Serret frame along the path. The Frenet-Serret frame is formed by tangent, normal and binormal unit vectors perpendicular to one another. Refer the appendix A for details.

For a two dimensional manoeuvre, the initial and final tangent vectors are coplanar, hence the initial and final turning circles and the connecting tangent lie in the plane. A 2D Dubins path is shown in figure (2.5). The sign of the initial and final manoeuvre



*Figure 2.5: Dubins Arc Geometry*

can be determined by designating either a left or right turn. Viewed from each position, a positive or negative rotation will define the sign of the curvature for each manoeuvre. Also, from the figure, we have:

$$\begin{aligned} \mathbf{r}_s &= e_s \begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_s} \end{pmatrix} \\ e_s &= \begin{bmatrix} \mathbf{t}_s & \mathbf{n}_s \end{bmatrix} \end{aligned} \quad (2.1.7)$$

where  $\kappa_s$  is the curvature of the initial manoeuvre and:

$$\begin{aligned} \mathbf{r}_f &= e_f \begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_f} \end{pmatrix} \\ e_f &= \begin{bmatrix} \mathbf{t}_f & \mathbf{n}_f \end{bmatrix} \end{aligned} \quad (2.1.8)$$

where  $\kappa_f$  is the curvature of the final manoeuvre. The initial and final manoeuvre vectors  $\mathbf{t}_s$  and  $\mathbf{t}_f$  are related by:

$$\mathbf{t}_f = \mathbf{R}(\theta)\mathbf{t}_s \quad (2.1.9)$$

## 2.1 Producing Flyable Paths - Dubins

---

where  $R(\theta)$  is the rotation matrix required to change the axis set from initial to final axes, see also (2.1.19) below. Hence, we have:

$$\cos(\theta) = \mathbf{t}'_f \mathbf{t}_s \quad (2.1.10)$$

The connecting vectors  $\mathbf{a}_s$ ,  $\mathbf{a}_f$  and  $\mathbf{a}_c$  form an orthogonal set of vectors. In order to determine the vectors, first define the connecting vector  $\mathbf{a}_c$  as:

$$\mathbf{t}_c = \mathbf{R}(\theta_s) \mathbf{t}_s \quad (2.1.11)$$

where  $\mathbf{t}_c$  is the basis vector defining the connecting vector. If the position of the final point  $\mathbf{p}_f$  relative to the start position  $\mathbf{p}_s$  is measured in start axes  $\mathbf{e}_s$ , we have:

$$\begin{aligned} \mathbf{p}_f - \mathbf{p}_s &= \mathbf{e}_s \mathbf{p} \\ \mathbf{p} &= \begin{pmatrix} p_t \\ p_n \end{pmatrix} \end{aligned} \quad (2.1.12)$$

Hence, the vector sum for the position vector in start axes is given by:

$$\begin{aligned} \mathbf{p} &= \mathbf{r}_s - \mathbf{a}_s + \mathbf{a}_c + \mathbf{a}_f - \mathbf{r}_f \\ \mathbf{p} - \mathbf{r}_s + \mathbf{r}_f &= -\mathbf{a}_s + \mathbf{a}_c + \mathbf{a}_f \end{aligned} \quad (2.1.13)$$

The left hand side of this equation represents the vector connecting the centres of the turn circles. Hence:

$$c \mathbf{t}_c = -\mathbf{a}_s + \mathbf{a}_c + \mathbf{a}_f \quad (2.1.14)$$

where  $c$  is the length of the centre vector. The remaining connecting vectors  $\mathbf{a}_s$ ,  $\mathbf{a}_f$  and  $\mathbf{a}_c$  can be written in terms of the start basis vectors, as:

$$\begin{aligned} \mathbf{a}_s &= \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_s} \end{pmatrix} \\ \mathbf{a}_f &= \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_f} \end{pmatrix} \\ \mathbf{a}_c &= \mathbf{R}(\theta_s)' \begin{pmatrix} a \\ 0 \end{pmatrix} \end{aligned} \quad (2.1.15)$$

The centre vector equation (3.3.9), now becomes:

$$\begin{aligned} c \mathbf{t}_c &= -\mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_s} \end{pmatrix} + \mathbf{R}(\theta_s)' \begin{pmatrix} a \\ 0 \end{pmatrix} + \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \frac{\pm 1}{\kappa_f} \end{pmatrix} \\ &= \mathbf{R}(\theta_s)' \begin{pmatrix} a \\ \frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s} \end{pmatrix} \end{aligned}$$

## 2. PRODUCING FLYABLE PATHS -2D

---

This is a rotation equation, hence the right hand vector must have the same magnitude as the left, to give:

$$\left| \frac{1}{c} \begin{pmatrix} a \\ \frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s} \end{pmatrix} \right| = 1 \quad (2.1.16)$$

or:

$$\begin{aligned} \left(\frac{a}{c}\right)^2 + \frac{1}{c^2} \left(\frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s}\right)^2 &= 1 \\ \left(\frac{a}{c}\right)^2 &= 1 - \frac{1}{c^2} \left(\frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s}\right)^2 \end{aligned} \quad (2.1.17)$$

This can be used to test for a feasible solution, by:

$$\boxed{1 - \frac{1}{c^2} \left(\frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s}\right)^2 > 0} \quad (2.1.18)$$

In order to compute the rotation angle  $\theta_s$ , the equation can be written in the form:

$$\begin{aligned} \mathbf{t}_c &= \mathbf{R}(\theta_s)' \begin{pmatrix} \frac{\sqrt{c^2 - \left(\frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s}\right)^2}}{c} \\ \frac{\left(\frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s}\right)}{c} \end{pmatrix} \\ \mathbf{R}(\theta_s) &= \begin{pmatrix} \cos(\theta_s) & -\sin(\theta_s) \\ \sin(\theta_s) & \cos(\theta_s) \end{pmatrix} \end{aligned} \quad (2.1.19)$$

Solving for  $\theta_s$  gives:

$$\begin{pmatrix} \cos(\theta_s) \\ \sin(\theta_s) \end{pmatrix} = \mathbf{R}(c, \kappa_s, \kappa_f) \mathbf{t}_c \quad (2.1.20)$$

where:

$$\boxed{\mathbf{R}(c, \kappa_s, \kappa_f) = \frac{1}{c} \begin{pmatrix} \sqrt{c^2 - \left(\frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s}\right)^2} & -\left(\frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s}\right) \\ \left(\frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s}\right) & \sqrt{c^2 - \left(\frac{\pm 1}{\kappa_f} - \frac{\pm 1}{\kappa_s}\right)^2} \end{pmatrix}} \quad (2.1.21)$$

The final angle  $\theta_f$  can then be determined using:

$$\begin{aligned} \theta &= \theta_s + \theta_f \\ \theta_f &= \theta - \theta_s \end{aligned} \quad (2.1.22)$$

The path length of the CLC path is calculated by summation of arc lengths and connecting tangent length.

$$\begin{aligned} L &= L_{\text{arc,start}} + L_{\text{tangent}} + L_{\text{arc,finish}} \\ &= \frac{\theta_s}{\kappa_s} + a + \frac{\theta_f}{\kappa_f} \end{aligned} \quad (2.1.23)$$



The important point to note here is that the results obtained from both the approaches are equivalent. The equation of path length (2.1.23) is analogous to (2.1.6b), and the condition for existence of the Dubins path (2.1.18) is analogous to (2.1.5a) and (2.1.5b). However, as stated in the beginning of this section the method derived by differential geometry is simple and easy to generalize, e.g. to polynomial curve such as Pythagorean Hodograph curve.

## 2.2 Paths of continuous curvature

The Dubins path is simple to produce and easy to implement because it composes of arcs and their tangents of low order polynomials. The line and arc are connected tangentially. This holds good as long as there is no change in the direction of motion. A change in direction induces lateral acceleration which acts in a direction perpendicular to that of the linear acceleration, acting along the tangent. This can not be directly handled by the Dubins path unless the UAV reduced its speed while approaching the arc from the line and vice-versa. Otherwise, a sudden change in acceleration will occur, which is not desirable. However, the piecewise smooth motion of the Dubins path may be used possibly for a rotorcraft, but not for a fixed wing UAV. Hence, it is important for the UAVs to have paths which provide smooth motion. A smooth motion has a continuous acceleration profile.

From the principles of physics, in time domain, the second derivative of a curve represents acceleration. Hence, a smooth motion in a plane requires at least non-vanishing first and second derivatives. By the principles of differential geometry, a path in plane is completely determined by its curvature. Also, it is proportional to the lateral acceleration of a moving vehicle. Thus, a smooth acceleration profile can be generated from a path of continuous curvature without any sudden reversal or jump. The curvature  $\kappa(t)$  of a curve,  $r(t) = (x(t), y(t))$  with  $t$  as a parameter is

$$\kappa(t) = \frac{\dot{r} \times \ddot{r}}{|\dot{r}|^3} \quad (2.2.1)$$

$$\kappa(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{(\sqrt{\dot{x}^2 + \dot{y}^2})^3} \quad (2.2.2)$$

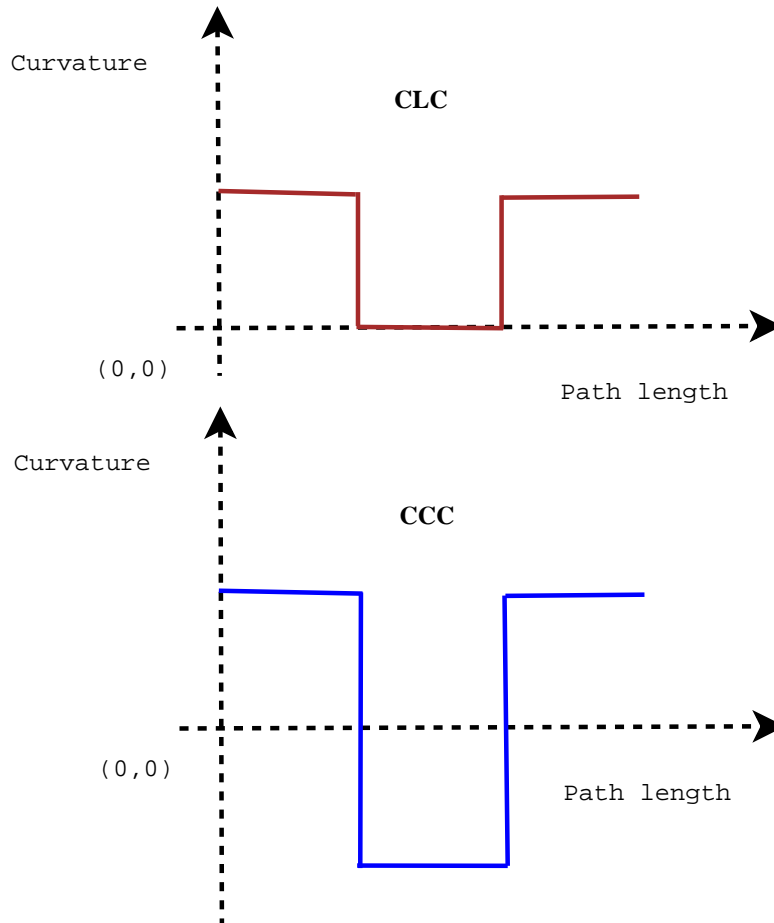
where  $\dot{x} = \frac{dx}{dt}$ ,  $\dot{y} = \frac{dy}{dt}$ ,  $\ddot{x} = \frac{d^2x}{dt^2}$  and  $\ddot{y} = \frac{d^2y}{dt^2}$ .

From the equation (A.2.1), the curvature is a function of first two derivatives of a curve, so the path needs to be at least twice continuously differentiable, that is  $C^2$  continuity.

In the Dubins path, the arc has a constant curvature, thus provides  $C^2$  continuity and

## 2. PRODUCING FLYABLE PATHS -2D

the line has zero continuity, thus provides  $C^1$  continuity.<sup>1</sup> Hence the point joining the arc and the line could not provide the curvature continuity. The curvature profile of the Dubins path of CLC and CCC types are shown in figure (2.6). The CLC path has a transition from a constant curvature to zero curvature and vice-versa, while the CCC path has a jump from a positive to a negative curvature and vice-versa. Both profiles cause an abrupt change in acceleration which is undesirable in practice. A



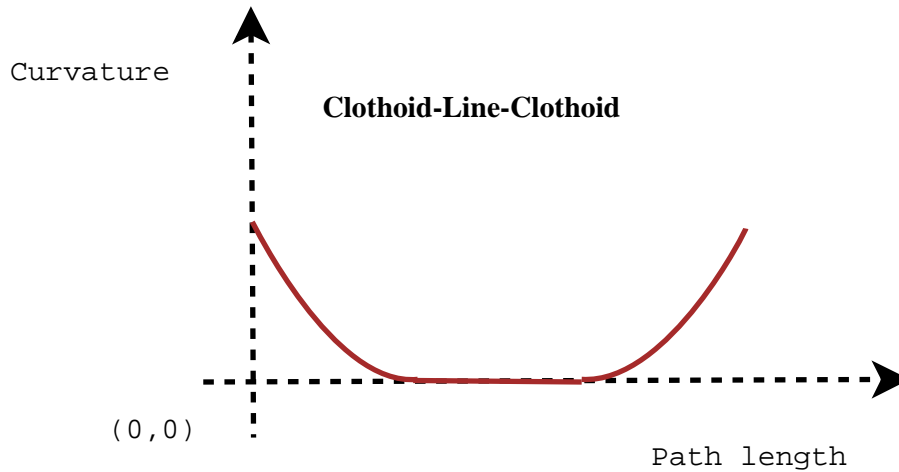
**Figure 2.6:** Curvature profiles of Dubins paths

path of curvature continuity can be produced either by a single curve of  $C^2$  continuity or by a composite path formed by joining pieces of curves of curvature continuities. To begin with, the arcs in the 2D Dubins path are replaced with the clothoid arcs, thus producing a composite path is discussed in the following section (2.3). This follows design of flyable path using Pythagorean Hodograph (PH) in the section (2.4).

<sup>1</sup> $C^2$  represents continuity upto second derivative and  $C^1$  represents continuity upto first derivative

## 2.3 Producing flyable path - Clothoid

A clothoid path has a property that its curvature varies linearly with the path length. Its curvature profile is shown in figure (2.7). In this section a flyable path is produced with the clothoid and the line segments. The clothoid path is generated with the zero curvature at the point of joining with the line segment. Differential geometric principles are used to produce the path. The principles employed in producing the flyable composite clothoid path is same as that of the Dubins path in section (2.1.4). The only difference is that the clothoid segment is produced by calculating Fresnel integrals which is discussed below. The circular arcs are replaced with the clothoid arcs. The curvature profile of the path is shown in figure (2.7). Note the curvature varies from a maximum to zero for a clothoid path and remains zero for the straight line segment and increases from zero to maximum for the final clothoid segment. The linear variation of curvature with path length of the clothoid enable a smooth transition to and from the line segment. For a clothoid arc, the arc angle varying



**Figure 2.7:** Curvature profile of a clothoid. Notice the difference with respect the curvature profile of the Dubins path in figure (2.6). Dubins path has a step variation in the curvature profile while the clothoid has a ramp variation

along the trajectory is given by:

$$\begin{aligned}\theta(t) &= \int_0^t \kappa \frac{\tau}{s} d\tau \\ &= \frac{\kappa}{2s} t^2\end{aligned}\tag{2.3.1}$$

where  $\kappa$  is the curvature at arc length  $s$  and  $t$  is the arc length variable, such that  $s = |\vec{v}|t$ , where  $\vec{v}$  is the velocity. The position vector of the end point is given by the  $x$

## 2. PRODUCING FLYABLE PATHS -2D

---

and  $y$  positions. These are obtained by integration:

$$\begin{aligned} x(s) &= \int_0^s \cos(\theta) dt \\ y(s) &= \int_0^s \sin(\theta) dt \end{aligned} \quad (2.3.2)$$

The angle  $\theta_t$  through which the trajectory moves over the total arc length  $s$  is  $\theta_t = \kappa \frac{s}{2}$ .

Hence:

$$\begin{aligned} x(s) &= \int_0^s \cos\left(\frac{\kappa}{2s}t^2\right) dt \\ y(s) &= \int_0^s \sin\left(\frac{\kappa}{2s}t^2\right) dt \end{aligned} \quad (2.3.3)$$

These integrals are scaled Fresnel Integrals and are given by:

$$\begin{aligned} \mathbf{C}(s) &= \int_0^s \cos\left(\frac{\kappa}{2s}t^2\right) dt \\ \mathbf{S}(s) &= \int_0^s \sin\left(\frac{\kappa}{2s}t^2\right) dt \end{aligned} \quad (2.3.4)$$

Hence:

$$\begin{aligned} x(s) &= \mathbf{C}(s) \\ y(s) &= \mathbf{S}(s) \end{aligned} \quad (2.3.5)$$

The integrals can be evaluated more easily by a change of variable, given by:

$$\bar{t} = \sqrt{\frac{\kappa}{2s}} t$$

Hence:

$$dt = \sqrt{\frac{2s}{\kappa}} d\bar{t} \quad (2.3.6)$$

and the integrals can be rewritten in the form:

$$\mathbf{C}(s) = \sqrt{\frac{2s}{\kappa}} \int_0^{\bar{s}} \cos((\bar{t})^2) d\bar{t} \quad (2.3.7a)$$

$$\mathbf{S}(s) = \sqrt{\frac{2s}{\kappa}} \int_0^{\bar{s}} \sin((\bar{t})^2) d\bar{t} \quad (2.3.7b)$$

Reconstructing this from the radius vector  $\mathbf{v}_r$  and the connecting vector  $\mathbf{v}_a$ , we have:

$$\begin{aligned} \mathbf{p} &= \mathbf{v}_r + \mathbf{v}_a \\ &= \rho \mathbf{t}_r + \alpha \mathbf{t}_a \end{aligned} \quad (2.3.8)$$

## 2.3 Producing flyable path - Clothoid

---

where  $\rho$  and  $\alpha$  are the lengths of the two vectors. As both  $\mathbf{t}_r$  and  $\mathbf{t}_a$  are basis vectors, they are of unit length. From figure (2.8), we have:

$$\begin{aligned}\mathbf{t}_r &= \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \mathbf{t}_a &= \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \end{pmatrix}\end{aligned}\quad (2.3.9)$$

Hence, we have:

$$\begin{aligned}\mathbf{p} &= \begin{pmatrix} \mathbf{C}(s) \\ \mathbf{S}(s) \end{pmatrix} \\ &= \rho \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \alpha \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \end{pmatrix}\end{aligned}\quad (2.3.10)$$

This gives:

$$\begin{aligned}\alpha &= \frac{\mathbf{C}(s)}{\sin(\theta)} \\ \rho &= \left( \mathbf{S}(s) + \frac{1}{\tan(\theta)} \mathbf{C}(s) \right)\end{aligned}\quad (2.3.11)$$

for  $\theta > 0$ .

Now:

$$\theta_t = \frac{\kappa}{2}s \quad (2.3.12)$$

Converting to angles using  $\theta_t = \frac{\kappa}{2}s$ ,

$$\begin{aligned}\alpha &= \frac{\mathbf{C}\left(\frac{2\theta_t}{\kappa}\right)}{\sin(\theta_t)} \\ \rho &= \left[ \mathbf{S}\left(\frac{2\theta_t}{\kappa}\right) + \frac{1}{\tan(\theta_t)} \mathbf{C}\left(\frac{2\theta_t}{\kappa}\right) \right]\end{aligned}\quad (2.3.13)$$

where

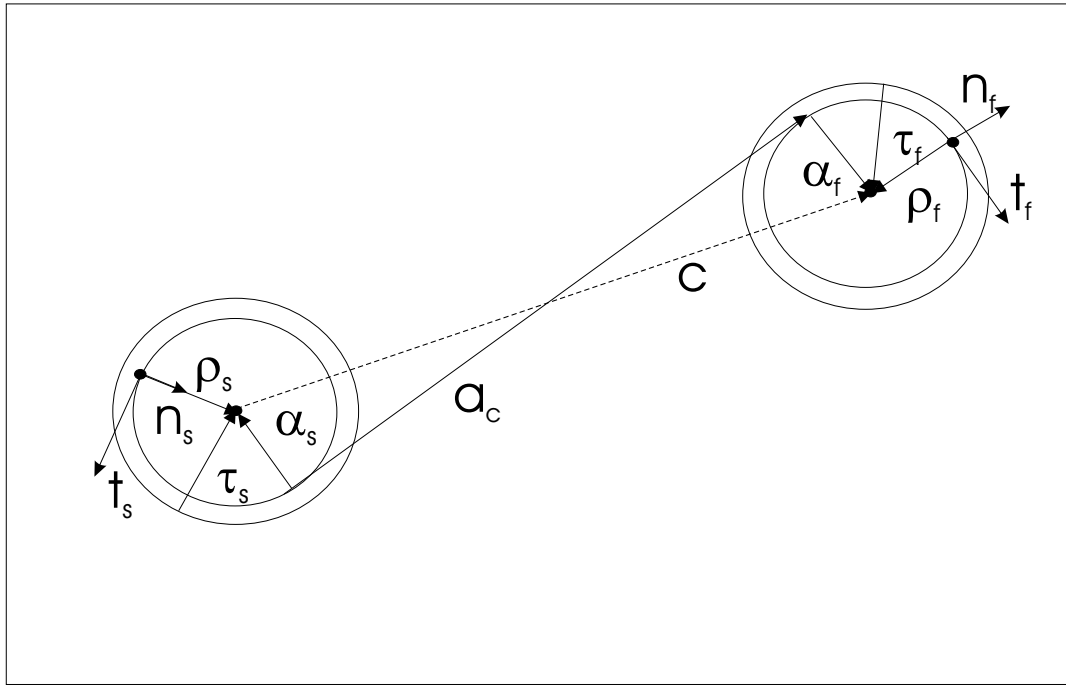
$$\mathbf{C}(\theta_t) = \frac{2}{\kappa} \sqrt{\theta_t} \int_0^{\sqrt{\theta_t}} \cos((\bar{\mathbf{t}})^2) d\bar{\mathbf{t}} \quad (2.3.14a)$$

$$\mathbf{S}(\theta_t) = \frac{2}{\kappa} \sqrt{\theta_t} \int_0^{\sqrt{\theta_t}} \sin((\bar{\mathbf{t}})^2) d\bar{\mathbf{t}} \quad (2.3.14b)$$

This implies that there is no closed form solution to the clothoid trajectory. In order to compute a solution, the tangent and normal vectors have to be computed.

For a two dimensional manoeuvre, the initial and final tangent vectors are coplanar and the straight line manoeuvre is not uniquely defined for this case and must be

## 2. PRODUCING FLYABLE PATHS -2D



**Figure 2.8:** Path with clothoid Arc Geometry

calculated. The 2D Clothoid arc is shown in figure (2.8). The derivation is similar to that of 2D Dubins path, but the circular arc is replaced with the clothoid arcs. However, the derivation is repeated here for convenience. The figure shows two circles of radius  $\rho$  and  $\tau$ . Also, from the figure, the sign of the manoeuvre can be determined by considering the centre line between the two positions. Viewed from each position a positive or negative rotation from the tangent vector to the centre vector will define the sign of the curvature for each manoeuvre. Also, from the figure, we have:

$$\begin{aligned} \mathbf{r}_i &= \mathbf{e}_i \begin{pmatrix} 0 \\ \pm\rho_s \end{pmatrix} \\ \mathbf{e}_i &= \begin{bmatrix} \mathbf{t}_i & \mathbf{n}_i \end{bmatrix} \end{aligned} \quad (2.3.15)$$

where  $\rho_s$  is the radius of the initial manoeuvre.

Similarly:

$$\begin{aligned} \mathbf{r}_f &= \mathbf{e}_f \begin{pmatrix} 0 \\ \pm\rho_f \end{pmatrix} \\ \mathbf{e}_f &= \begin{bmatrix} \mathbf{t}_f & \mathbf{n}_f \end{bmatrix} \end{aligned} \quad (2.3.16)$$

where  $\rho_f$  is the radius of the final manoeuvre. The Frenet basis vectors are related by:

$$\mathbf{e}_f = \mathbf{R}(\theta)\mathbf{e}_s \quad (2.3.17)$$

## 2.3 Producing flyable path - Clothoid

---

where  $R(\theta)$  is the rotation matrix required to change the axis set from start to finish axes. Hence, we have:

$$R(\theta) = e_s e_f' \quad (2.3.18)$$

The connecting vectors  $a_s$ ,  $a_f$  and  $a_c$  form an orthogonal set of vectors. In order to determine the vectors, first define the connecting vector  $a_c$  in both initial and final axes, as:

$$\begin{aligned} e_c &= R(\theta_s) e_s \\ e_f &= R(\theta_f) e_c \end{aligned} \quad (2.3.19)$$

where  $e_c$  is the basis set defining the connecting vector. Hence, the total rotation matrix  $R(\theta)$  is given by:

$$R(\theta) = R(\theta_f) R(\theta_s) \quad (2.3.20)$$

If the position of the final point  $p_f$  relative to the start position  $p_s$  is measured in start axes  $e_s$ , we have:

$$\begin{aligned} p_f - p_s &= e_s p \\ p &= \begin{pmatrix} p_t \\ p_n \end{pmatrix} \end{aligned} \quad (2.3.21)$$

Hence, the vector sum for the position vector in start axes is given by:

$$\begin{aligned} p &= \rho_s - \alpha_s + a_c + \alpha_f - \rho_f \\ p - \rho_s + \rho_f &= -\alpha_s + a_c + \alpha_f \end{aligned} \quad (2.3.22)$$

The left hand side of this equation represents the vector connecting the centres of the turn circles. Hence:

$$c = -\alpha_s + a_c + \alpha_f \quad (2.3.23)$$

The centre vector  $c$  can be written in start axes, to give:

$$\begin{aligned} c &= c t_c \\ &= e_{ct} \begin{pmatrix} c \\ 0 \end{pmatrix} \\ e_{ct} &= \begin{bmatrix} t_c & n_c \end{bmatrix} \end{aligned} \quad (2.3.24)$$

where  $e_{ct}$  is the basis vector set of the centre vector.

## 2. PRODUCING FLYABLE PATHS -2D

---

The remaining connecting vectors  $\alpha_s$ ,  $\alpha_f$  and  $a_c$  can be written in terms of the start basis vectors, as:

$$\begin{aligned}\alpha_s &= \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \pm\rho_s \end{pmatrix} \\ \alpha_f &= \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \pm\rho_f \end{pmatrix} \\ a_c &= \mathbf{R}(\theta_s)' \begin{pmatrix} a \\ 0 \end{pmatrix}\end{aligned}\tag{2.3.25}$$

The centre vector equation (2.3.23), now becomes:

$$\begin{aligned}ct_{ct} &= -\mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \pm\rho_s \end{pmatrix} + \mathbf{R}(\theta_s)' \begin{pmatrix} a \\ 0 \end{pmatrix} + \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \pm\rho_f \end{pmatrix} \\ &= \mathbf{R}(\theta_s)' \begin{pmatrix} a \\ \pm\rho_f - \pm\rho_s \end{pmatrix}\end{aligned}\tag{2.3.26}$$

Normalizing the centre vector to unit magnitude, gives:

$$t_{ct} = \mathbf{R}(\theta_s)' \frac{1}{c} \begin{pmatrix} a \\ \pm\rho_f - \pm\rho_s \end{pmatrix}\tag{2.3.27}$$

This is a rotation equation, that represents the rotation of a unit vector. Hence, the right hand vector must have unit magnitude, to give:

$$\left| \frac{1}{c} \begin{pmatrix} a \\ \pm\rho_f - \pm\rho_s \end{pmatrix} \right| = 1\tag{2.3.28}$$

or:

$$\begin{aligned}\left(\frac{a}{c}\right)^2 + \frac{1}{c^2} (\pm\rho_f - \pm\rho_s)^2 &= 1 \\ \left(\frac{a}{c}\right)^2 &= 1 - \frac{1}{c^2} (\pm\rho_f - \pm\rho_s)^2\end{aligned}\tag{2.3.29}$$

This can be used to test for a feasible solution, by:

$$\boxed{1 - \frac{1}{c^2} (\pm\rho_f - \pm\rho_s)^2 > 0}\tag{2.3.30}$$

In order to compute the rotation angle  $\theta_s$ , the equation can be written in the form:

$$\begin{aligned}t_{ct} &= \mathbf{R}(\theta_s)' \frac{1}{c} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} \\ \mathbf{R}(\theta_s) &= \begin{pmatrix} \cos(\theta_s) & -\sin(\theta_s) \\ \sin(\theta_s) & \cos(\theta_s) \end{pmatrix}\end{aligned}\tag{2.3.31}$$



## 2.3 Producing flyable path - Clothoid

---

where  $\beta = \sqrt{c^2 - (\pm\rho_f - \pm\rho_s)^2}$  and  $\gamma = (\pm\rho_f - \pm\rho_s)$ .

Expanding this and solving for  $\theta_s$  gives:

$$\begin{aligned} \cos(\theta_s) \frac{\sqrt{c^2 - (\pm\rho_f - \pm\rho_s)^2}}{c} + \sin(\theta_s) \frac{(\pm\rho_f - \pm\rho_s)}{c} &= t_{ct1} \\ -\sin(\theta_s) \frac{\sqrt{c^2 - (\pm\rho_f - \pm\rho_s)^2}}{c} + \cos(\theta_s) \frac{(\pm\rho_f - \pm\rho_s)}{c} &= t_{ct2} \end{aligned} \quad (2.3.32)$$

or:

$$\frac{1}{c} \begin{pmatrix} \sqrt{c^2 - (\pm\rho_f - \pm\rho_s)^2} & (\pm\rho_f - \pm\rho_s) \\ -(\pm\rho_f - \pm\rho_s) & \sqrt{c^2 - (\pm\rho_f - \pm\rho_s)^2} \end{pmatrix} \begin{pmatrix} \cos(\theta_s) \\ \sin(\theta_s) \end{pmatrix} = \mathbf{t}_{ct} \quad (2.3.33)$$

Solving for  $\theta_s$  gives:

$$\begin{pmatrix} \cos(\theta_s) \\ \sin(\theta_s) \end{pmatrix} = \frac{1}{c} \begin{pmatrix} \beta & -\gamma \\ \gamma & \beta \end{pmatrix} \mathbf{t}_{ct}$$

Hence:

$$\theta_s = \tan^{-1}(\sin(\theta_s), \cos(\theta_s)) \quad (2.3.34)$$

The final angle  $\theta_f$  can then be determined using:

$$\begin{aligned} \theta &= \theta_s + \theta_f \\ \theta_f &= \theta - \theta_s \end{aligned} \quad (2.3.35)$$

An alternate solution is:

$$\mathbf{R}(\theta_s) \mathbf{t}_{ct} = \begin{pmatrix} \frac{1}{c} \\ \gamma \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} \quad (2.3.36)$$

Expanding this gives:

$$\begin{aligned} \begin{pmatrix} \cos(\theta_s) \\ \sin(\theta_s) \end{pmatrix} &= \frac{1}{\Delta} \begin{pmatrix} t_{ct1} & t_{ct2} \\ -t_{ct2} & t_{ct1} \end{pmatrix} \begin{pmatrix} \frac{1}{c} \\ \gamma \end{pmatrix} \begin{pmatrix} \beta \\ \gamma \end{pmatrix} \\ \Delta &= t_{ct1}^2 + t_{ct2}^2 \\ &= 1 \end{aligned} \quad (2.3.37)$$

The related Fresnel Integrals are given by:

$$\mathbf{C}(s) = \int_0^{sf} \cos(s^2) ds \quad (2.3.38a)$$

$$\mathbf{S}(s) = \int_0^{sf} \sin(s^2) ds \quad (2.3.38b)$$

## 2. PRODUCING FLYABLE PATHS -2D

---

Series expansions are given by:

$$\mathbf{C}(s) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!(4n+3)} s^{4n+3} \quad (2.3.39a)$$

$$\mathbf{S}(s) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!(4n+1)} s^{4n+1} \quad (2.3.39b)$$

### 2.4 Producing Flyable Path - Pythagorean Hodograph

Pythagorean Hodograph (PH) was first introduced by Farouki [59]. This is a polynomial curve known for its rational properties. As the name implies, the PH path has its hodograph satisfy pythagorean condition. The first derivative of a curve is its hodograph. The Pythagorean condition is that the sum of the squares of the sides of right angle triangle is equal to the square of its hypotenuse. In time domain, the hodograph is called velocity vector which is always parallel to the tangent of the path. However, the derivation of the PH path arose from the definition of path length. The length of curve  $r(t)$  with parameter  $t$  is:

$$s(t) = \int_{t_1}^{t_2} \|\dot{r}(t)\| dt \quad (2.4.1a)$$

$$= \int_{t_1}^{t_2} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} dt \quad (2.4.1b)$$

where  $t \in [t_1, t_2]$  and  $\dot{x}(t) = \frac{dx}{dt}$  and  $\dot{y}(t) = \frac{dy}{dt}$  are hodographs.

The calculation of path length requires solution to the integral in equation (2.4.1). The presence of square root term in the equation may not result in closed form solution. This requires numerical approximation which is less desirable in practice. To rectify this problem a perfect solution to the path length is required. Here comes the Pythagorean Hodograph. Note that the term inside the square root of equation (2.4.1) is the sum of the square of the hodographs. If it is possible to represent the term inside the square root as a perfect square, say  $\sigma(t)^2$ , then the solution to the path length will be an integral of a polynomial equation  $\sigma(t)$ .

$$\boxed{\sigma(t)^2 = \dot{x}(t)^2 + \dot{y}(t)^2} \quad (2.4.2)$$

$$s(t) = \int_{t_1}^{t_2} |\sigma(t)| dt \quad (2.4.3)$$

This is equivalent to satisfying the Pythagorean law of right angle triangle taking polynomials  $\sigma(t)$  as hypotenuse,  $\dot{x}(t)$  and  $\dot{y}(t)$  as two other sides. A polynomial curve whose hodographs meet the condition (equation 2.4.2) is called Pythagorean Hodograph. The PH path is designed by selecting the suitable polynomials  $u(t)$ ,  $v(t)$ , and

## 2.4 Producing Flyable Path - Pythagorean Hodograph

---

$w(t)$  so that the hodographs  $\dot{x}(t)$  and  $\dot{y}(t)$  meet the condition (equation 2.4.2). Now, the problem is reduced to finding the coefficients of the polynomials  $u(t)$ ,  $v(t)$ , &  $w(t)$ . The advantage of this idea is not only elimination of radical form in the equation (2.4.1), but also equal distribution of points on the path. In other words, there is an equal increment of path length for an equal increment of the parameter  $t$ .

$$\dot{x}(t) = w(t)[u(t)^2 - v(t)^2] \quad (2.4.4)$$

$$\dot{y}(t) = 2w(t)u(t)v(t) \quad (2.4.5)$$

$$\begin{aligned} \implies \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} &= w(t)[u(t)^2 + v(t)^2] \\ &= |\sigma(t)| \end{aligned}$$

where  $u(t)$  and  $v(t)$  are relatively prime polynomials,  $w(t) = 1$ , and  $\sigma(t)$  is a polynomial of degree  $(n - 1)$ .

The parametric speed  $\dot{s}$  and the curvature  $\kappa$  and the offset curve at a distance  $\pm d$  of the PH curve are:

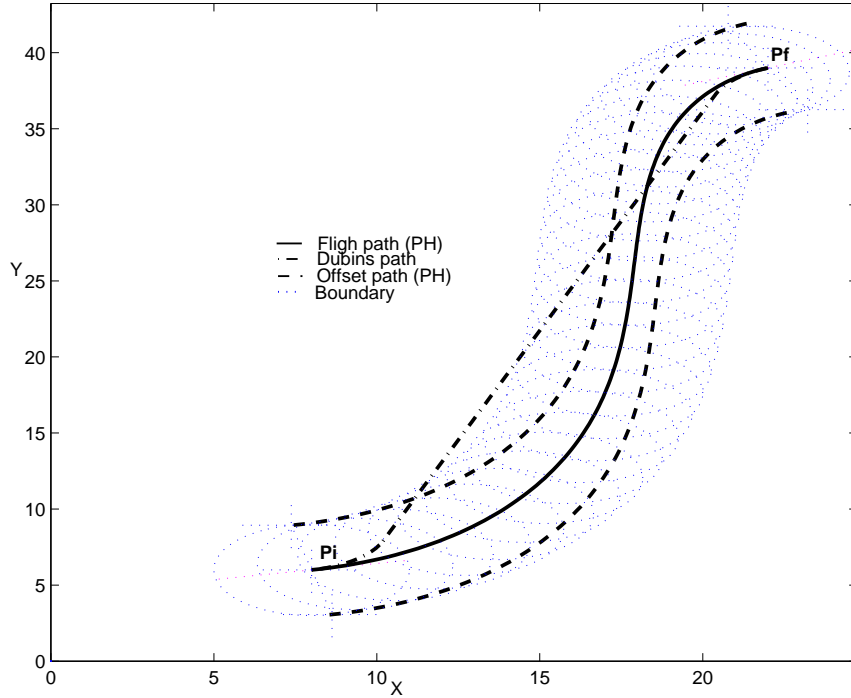
$$\dot{s}(t) = |\sigma(t)| \quad (2.4.6)$$

$$\kappa = \frac{2(u(t)\dot{v}(t) - \dot{u}(t)v(t))}{w(t)(u(t)^2 + v(t)^2)} \quad (2.4.7)$$

$$r_d(t) = r(t) \pm d\mathbf{N}(t) \quad (2.4.8)$$

where  $\dot{s}(t) = \frac{ds}{dt}$  and  $\mathbf{N}(t)$  is unit normal to the curve  $\mathbf{r}(t)$ .

From above three equations, the parametric speed of the PH curve is simply a root-finding problem of a polynomial. The curvature is in rational form. Also, the offset curves of the PH curve can be represented exactly by rational parametric curves of order  $(2n - 1)$ . The offset curve can be used to define a safety region or sensor range or uncertainty along the path. The offset curve self-intersects when the path is too convex or too concave. However, the self-intersection can be eliminated by choosing the value of offset distance less than the local radius of curvature [60]. Figure (2.9) shows a comparative visualization of a smooth PH path and a Dubins path for same maximum bound on curvature. It is evident from this figure that the curvature continuity of the PH path is achieved with the sacrifice on the path length. The length of the PH path is greater than that of the Dubins path. However, the PH path possesses the rational offset which shown as the dotted circular tube around the path. Though it has equal offset distance, at the point of higher curvature value it tends to diminishes. This is evident from the middle portion of the path.



**Figure 2.9:** A visual comparison of Dubins path with PH path. The Dubins path (-.) is the shortest path between the poses  $P_i$  and  $P_f$ . But it lacks the curvature continuity. On the other hand the PH path (-) has continuity but has more length for the same curvature bound. This is a tradeoff

### 2.4.1 Flyable path -PH

The minimum order of polynomial which exhibits the PH behavior is three, called cubic PH. However, the lowest order of the PH path which has a point of inflexion is five [59]. The presence of inflexion point allows the path to have more flexibility so that the path can easily be manipulated. Hence, the quintic PH curve is used for path planning. From now on, a PH path denotes a quintic PH curve. The initial and final positions respectively are  $(x_s, y_s)$  and  $(x_f, y_f)$  and corresponding orientations (tangential directions) are  $\theta_s$  and  $\theta_f$ . These are boundary values.

The PH path is represented in Bézier form for numerical stability. The general equation for  $n^{\text{th}}$  order polynomial in Bézier form is:

$$r(t) = \sum_{k=0}^n b_k \binom{n}{k} t^k (1-t)^{(n-k)}, \quad t \in [0, 1] \quad (2.4.9)$$

where  $b_k = (x_k, y_k)$   $k = 1 \dots n$  are control points,  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ , and  $r(t)|_{(t=0)}$  &  $r(t)|_{(t=1)}$  respectively represent starting and ending points of the path.

## 2.4 Producing Flyable Path - Pythagorean Hodograph

The  $r^{th}$  derivative of  $n^{th}$  order Bézier curve is:

$$\boxed{\frac{d^r r(t)}{dt^r} = \frac{n!}{(n-r)!} \sum_{j=0}^{n-r} \Delta^r b_j \binom{n-r}{j} t^j (1-t)^{(n-r-j)}} \quad (2.4.10)$$

where  $\Delta^r b_i = \sum_{j=0}^r \binom{r}{j} (-1)^{r-j} b_{i+j}$ ,  $\Delta b_j = b_{j+1} - b_j$  and  $\Delta^0 b_j = b_j$

For a quintic path,  $n = 5$ . Hence the equation (2.4.9) becomes:

$$\boxed{r(t) = \sum_{k=0}^5 b_k \binom{5}{k} t^k (1-t)^{(5-k)}} \quad (2.4.11)$$

$$r(t) = b_0(1-t)^5 + 5b_1t(1-t)^4 + 10b_2t^2(1-t)^3 + 10b_3t^3(1-t)^2 + 5b_4t^4(1-t) + b_5t^5 \quad (2.4.12)$$

From equation (2.4.10), the first derivative of the path  $r(t)$  is:

$$\boxed{\frac{dr(t)}{dt} = 5 \sum_{j=0}^4 \binom{4}{j} \Delta^1 b_j t^j (1-t)^{4-j}} \quad (2.4.13)$$

$$\frac{dr(t)}{dt} = 5(b_1 - b_0)(1-t)^4 + 20(b_2 - b_1)t(1-t)^3 + 30(b_4 - b_3)t^3(1-t) + 5(b_5 - b_4)t^4 \quad (2.4.14)$$

Now comes the interpolation. As the position and direction at initial and final locations are known, first order Hermite interpolation is used. Substituting the position coordinates at  $t = 0$  and at  $t = 1$  in equation (2.4.10), and from the first derivative of the path, the control points  $b_0, b_1, b_5$  &  $b_4$  are calculated as below:

$$b_0 = (x_s, y_s) \quad (2.4.15a)$$

$$b_5 = (x_f, y_f) \quad (2.4.15b)$$

$$d_0 = (\cos(\theta_s), \sin(\theta_s)) \quad (2.4.15c)$$

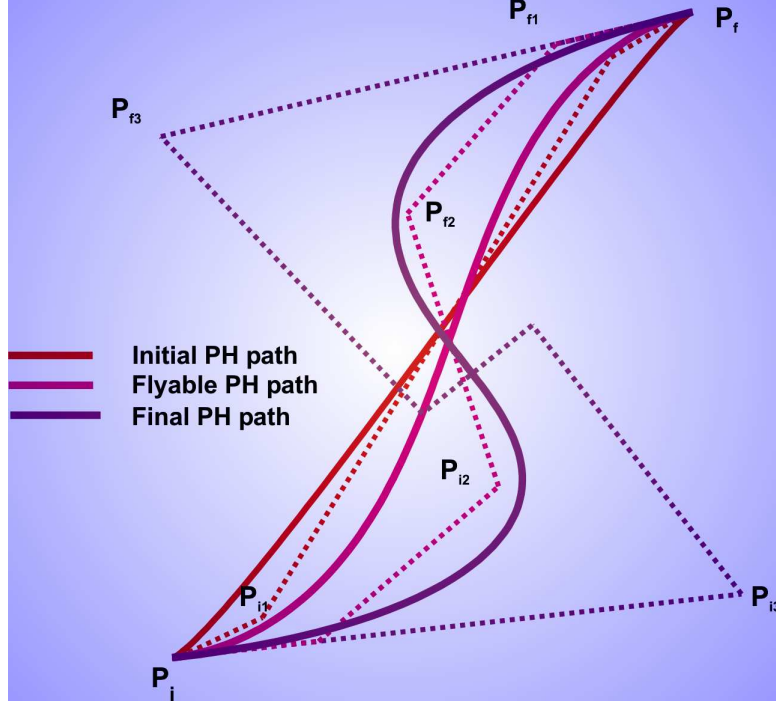
$$d_5 = (\cos(\theta_f), \sin(\theta_f)) \quad (2.4.15d)$$

$$b_1 = b_0 + (1/5) * d_0 \quad (2.4.15e)$$

$$b_4 = b_5 - (1/5) * d_5 \quad (2.4.15f)$$

Thus, the control points  $(b_0, b_1, b_4, b_5)$  in the equation (2.4.15) are fixed by the poses. Now the problem is reduced to finding the control points  $b_2$  and  $b_3$  so that the equation (2.4.9) satisfies the PH condition (2.4.2). This results in four solutions [61]. A minimum energy curve [62] which has smooth variation of curvature is used for path planning. As an original development, the PH curve provides only the tangent continuity. Or in other words, the initial development of the curve is based on tangent continuity at the end points. For a flyable path, the tangent continuity is not enough. Hence to have curvature continuity, it needs further refinement. One approach found

## 2. PRODUCING FLYABLE PATHS -2D



**Figure 2.10:** Evolution of a PH path from the tangent continuity into curvature continuity. The initial PH path has tangent continuity. The flyable path has curvature continuity. The final path has more curvature than the previous two showing the flexibility of the path

in [63] achieves the curvature continuity by increasing the length of the boundary vectors. The length of the tangent vectors are increased by approximating the term  $\frac{\partial \kappa}{\partial c_t}$ , where  $c_t$  is the magnitude of the tangent vector. However, there is no closed form solution. Here, the length of the tangent vectors are increased directly by modifying the equations (2.4.15c) and (2.4.15d) into:

$$d_0 = c_0(\cos(\theta_s), \sin(\theta_s)) \quad (2.4.16a)$$

$$d_5 = c_5(\cos(\theta_f), \sin(\theta_f)) \quad (2.4.16b)$$

where  $c_0 \in [1, \infty]$  and  $c_5 \in [1, \infty]$ .

Increasing the values of  $c_0$  and  $c_5$  will increase the length of tangent vectors  $\overrightarrow{b_0 b_1}$  and  $\overrightarrow{b_5 b_4}$  and in turn  $b_2$  and  $b_3$  are changed to meet the PH condition. As there is no closed form solution available, an iterative method is sought to arrive at the flyable PH path with the path length close to that of the Dubins path. Thus the curvature constraint is met and a flyable path is produced. The figure (2.10) shows the flexibility and evolution of the PH path from curvature continuity from the tangent continuity. The initial PH path does have only tangent continuity. The control points  $P_{i1}$  and  $P_{f1}$  are shifted respectively to  $P_{i2}$  and  $P_{f2}$  by increasing the length of the boundary tangent vectors  $\overrightarrow{P_i P_{i1}}$  and  $\overrightarrow{P_f P_{f1}}$ . This results in flyable path which meets the maximum cur-

vature bound. Again shifting of the control points to  $P_{i3}$  and  $P_{f3}$  shows the flexibility of the PH path.

## 2.5 Summary

This chapter describes how to produce paths for two-dimensional maneuver. Three types of path are discussed. The Dubins path is derived both using the principles of Euclidean and Differential geometries. Also, it is shown that the results obtained by both the principles are equivalent. A path in 2D is determined only by its curvature profile. It is shown that with principles of differential geometry, it will be easy to generalize the path for applications such as path tracking and trajectory generation. Later in the section (2.2), the importance of curvature continuity is explained. Also, the limitation of the Dubins path due to the lack of curvature continuity is explained with the curvature profile.

The latter part of the chapter deals with generation of continuous curvature paths. A single path - Pythagorean Hodograph and a composite path - clothoid with line segments are discussed. Fundamentals properties and the generation of the PH path by first order hermite interpolation are described in section (2.4). The last section (2.3) explains how to produce the composite path - clothoid with line segment. It is important to note that the derivation of Clothoid-composite path is similar to that of Dubins path. In fact, the clothoid path is formed by approximation of the Dubins path.

---

## Producing flyable path - 3D

**T**he real time operation of the UAVs occurs in a three dimensional space. Therefore, it is important to have three dimensional trajectories for path planning. In this chapter, UAVs flying in three dimensional space are considered. The condition of constant altitude flight is relaxed such that the initial and final poses do not lie in a plane. Two paths are studied for 3D maneuvers: (i) Dubins and (ii) PH. The flyable paths need to meet the curvature and torsion constraints. A similar concept employed in designing 2D Dubins trajectories in the previous chapter are extended for 3D maneuver. However, the solution involves finding a common intersecting plane for a smooth motion. The spatial PH path is obtained by first order Hermite interpolation. The resulting path is made flyable increasing the lengths of boundary tangent vectors. A three dimensional flyable path is produced by solving the equation (3.0.1). It is repeated here for the convenience.

$$P_s(x_s, y_s, \phi_s, \theta_s) \xrightarrow{\tau(t)} P_f(x_f, y_f, z_f, \phi_f, \theta_f), |\kappa(t)| < \kappa_{max}, |\tau(t)| < \tau_{max} \quad (3.0.1)$$

where  $\kappa(t)$  is the curvature and  $\tau(t)$  is the torsion.

One of the classical paths used for aircraft maneuver is circular helix, whose projection on  $X - Y$  plane is a circle. The path can be visualized as wound on the surface of a cylinder, stand vertically in  $X - Y$  plane. An important property of this curve is that the ratio of its curvature to torsion is constant. Comparing with the Dubins and PH path, the path length of the helix will be more for any two poses owing to spiral in nature. In contrast, a 3D Dubins path is produced with an initial maneuver followed by a 2D Dubins maneuver in the common intersecting plane between the two poses.



The PH path is produced by interpolation. Both the paths would have path length less than that of a helix for the same poses.

### **3.1 Dubins Path-3D**

The design and derivation of Dubins path in 2D is shown in section (2.1) The Dubins path has two circular maneuvers and a straight line maneuver, where all three maneuvers are in the same plane. Hence it is easy to find a common tangent between the initial and final poses. The same approach can be extended only into a part of the solution to a three dimensional space maneuver, because a general maneuver in 3D cannot be confined into a single plane. For the 2D manoeuvre, the tangent and normal vectors of the start and finish configurations are co-planar. But for a 3D manoeuvre, the tangent and normal vectors may not lie in the same plane. Hence the path generation in 3D space is not simple as the case with 2D. Hence, the Frenet-Serret frame of unit tangent, unit normal and unit binormal vectors (see appendix A) are used to define the path.

### **3.2 Extension of 2D Dubins to 3D manoeuvre**

The theory behind the construction of a Dubins arc consisting of a circular arc at either end of a straight line arc, that connects two points in space with prescribed pose follows. In order to perform such a manoeuvre, two planes have to be defined. The first is the start manoeuvre plane, which contains the tangent vector  $t_s$  and the normal vector  $n_s$ . These are completed by defining a right handed set to give the binormal vector  $b_s$ . This triple is  $\begin{bmatrix} t_s & n_s & b_s \end{bmatrix}$ . The second manoeuvre plane is the finish manoeuvre plane defined by the triple  $\begin{bmatrix} t_f & n_f & b_f \end{bmatrix}$ . Both of these frames are known as Frenet frames from the principles of differential geometry.

The manoeuvre from the start position and pose to the finish position and pose will consist of an initial circular manoeuvre in the start manoeuvre plane, followed by a straight manoeuvre along the line that is the intersection of the two manoeuvre planes and a final circular manoeuvre in the finish manoeuvre plane. This follows as the straight line manoeuvre must be a tangent to both the initial and final circular manoeuvre, and hence the straight manoeuvre must lie in both manoeuvre planes. The only common line between the planes is the intersection line. Figure (3.1) shows the 3D Dubins path. The initial configuration is  $P_i$  and the final configuration is  $P_f$ . The UAV flies from  $P_i$  to  $P_f$ . First, it makes an initial maneuver at  $P_s$ . It is fol-

### 3. PRODUCING FLYABLE PATH - 3D

---

lowed by a 2D Dubins maneuver in a plane formed by the final tangent vector with the initial position. The calculation starts with finding a coplanar connecting the final tangent vector and initial position, which is used for 2D Dubins maneuver and is followed by an initial rotational maneuver at starting pose to connect the first 2D maneuver. To define the Dubins manoeuvre, the start manoeuvre circle with defined curvature  $\tau_s$ , and the finish manoeuvre circle with defined curvature  $\tau_f$ , must be connected by the straight line manoeuvre. The geometry that defines such a manoeuvre is given in the previous chapter 2. However, this is repeated in the next section for convenience.

### 3.3 Dubins arc for the 2D coplanar manoeuvre

For a two dimensional manoeuvre, the initial and final tangent vectors are coplanar and the straight line manoeuvre is not uniquely defined for this case and must be calculated. The 2D Dubins arc is shown in (3.2). The sign of the manoeuvre can be determined by considering the centre line between the two positions. Viewed from each position a positive or negative rotation from the tangent vector to the centre vector will define the sign of the curvature for each manoeuvre. Also, from the figure (3.2), we have:

$$\begin{aligned} \mathbf{r}_i &= \mathbf{e}_i \begin{pmatrix} 0 \\ \frac{\pm 1}{\tau_s} \end{pmatrix} \\ \mathbf{e}_i &= \begin{bmatrix} \mathbf{t}_i & \mathbf{n}_i \end{bmatrix} \end{aligned} \quad (3.3.1)$$

where,  $\tau_i$  is the curvature of the initial manoeuvre.

Similarly:

$$\begin{aligned} \mathbf{r}_f &= \mathbf{e}_f \begin{pmatrix} 0 \\ \frac{\pm 1}{\tau_f} \end{pmatrix} \\ \mathbf{e}_f &= \begin{bmatrix} \mathbf{t}_f & \mathbf{n}_f \end{bmatrix} \end{aligned} \quad (3.3.2)$$

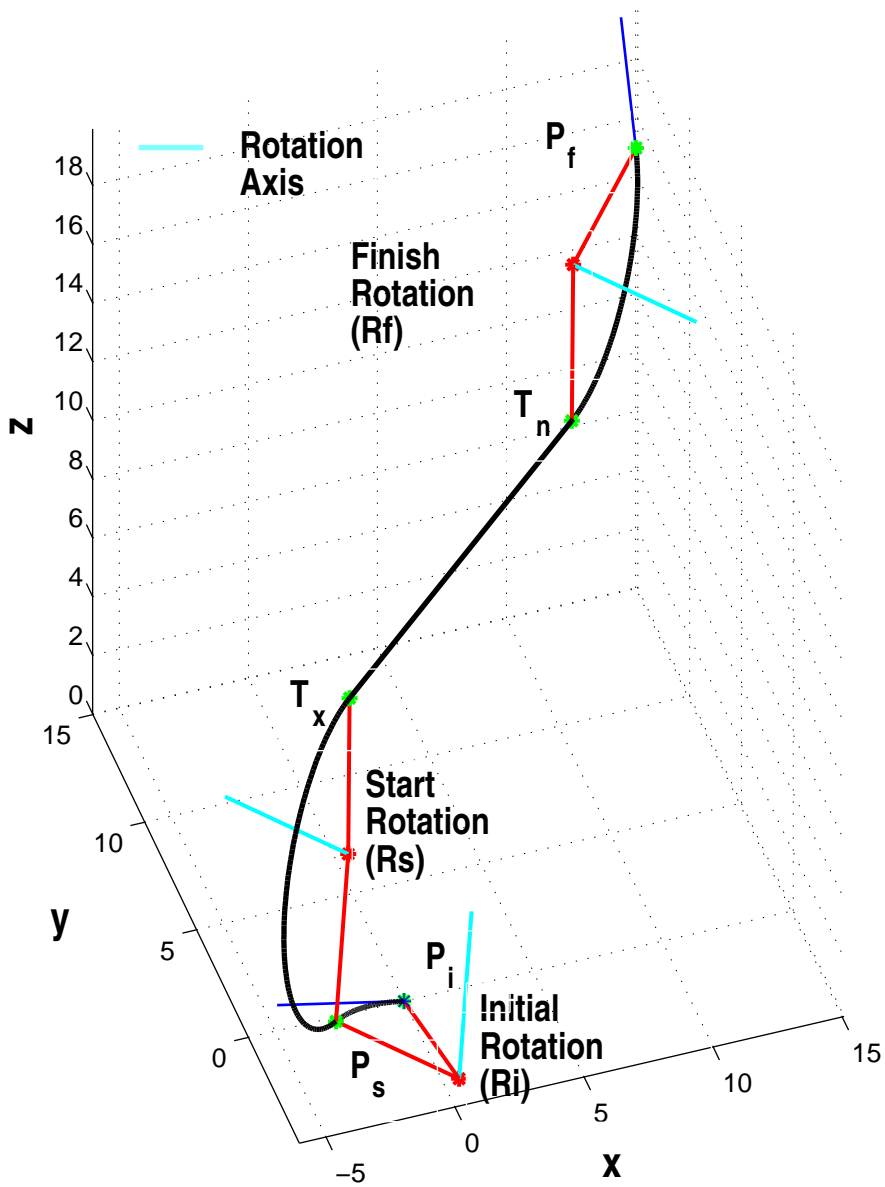
where  $\tau_f$  is the curvature of the final manoeuvre.

The Frenet basis vectors are related by:

$$\mathbf{e}_f = \mathbf{R}(\theta) \mathbf{e}_s \quad (3.3.3)$$

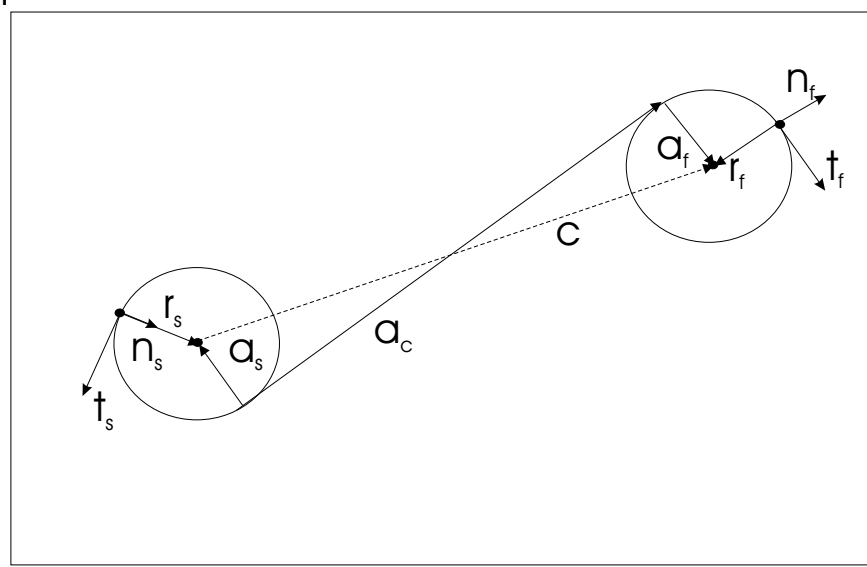
where  $\mathbf{R}(\theta)$  is the rotation matrix required to change the axis set from start to finish axes.

### 3.3 Dubins arc for the 2D coplanar manoeuvre



*Figure 3.1: 3D Dubins Manoeuvre of UAV*

### 3. PRODUCING FLYABLE PATH - 3D



**Figure 3.2:** Dubins Arc Geometry

Hence, we have:

$$\mathbf{R}(\theta) = \mathbf{e}_s \mathbf{e}'_f \quad (3.3.4)$$

The connecting vectors  $\mathbf{a}_s$ ,  $\mathbf{a}_f$  and  $\mathbf{a}_c$  form an orthogonal set of vectors. In order to determine the vectors, first define the connecting vector  $\mathbf{a}_c$  in both initial and final axes, as:

$$\begin{aligned} \mathbf{e}_c &= \mathbf{R}(\theta_s) \mathbf{e}_s \\ \mathbf{e}_f &= \mathbf{R}(\theta_f) \mathbf{e}_c \end{aligned} \quad (3.3.5)$$

where  $\mathbf{e}_c$  is the basis set defining the connecting vector.

Hence, the total rotation matrix  $\mathbf{R}(\theta)$  is given by:

$$\mathbf{R}(\theta) = \mathbf{R}(\theta_f) \mathbf{R}(\theta_s) \quad (3.3.6)$$

If the position of the end point  $\mathbf{p}_f$  relative to the initial position  $\mathbf{p}_s$  is measured in start axes  $\mathbf{e}_s$ , we have:

$$\begin{aligned} \mathbf{p}_f - \mathbf{p}_s &= \mathbf{e}_s \mathbf{p} \\ \mathbf{p} &= \begin{pmatrix} p_t \\ p_n \end{pmatrix} \end{aligned} \quad (3.3.7)$$

Hence, the vector sum for the position vector in start axes is given by:

$$\begin{aligned} \mathbf{p} &= \mathbf{r}_s - \mathbf{a}_s + \mathbf{a}_c + \mathbf{a}_f - \mathbf{r}_f \\ \mathbf{p} - \mathbf{r}_s + \mathbf{r}_f &= -\mathbf{a}_s + \mathbf{a}_c + \mathbf{a}_f \end{aligned} \quad (3.3.8)$$

### 3.3 Dubins arc for the 2D coplanar manoeuvre

---

The left hand side of this equation represents the vector connecting the centres of the turn circles. Hence:

$$\mathbf{c} = -\mathbf{a}_s + \mathbf{a}_c + \mathbf{a}_f \quad (3.3.9)$$

The centre vector  $\mathbf{c}$  can be written in start axes, to give:

$$\begin{aligned} \mathbf{c} &= c\mathbf{t}_c \\ &= \mathbf{e}_{ct} \begin{pmatrix} c \\ 0 \end{pmatrix} \\ \mathbf{e}_{ct} &= \begin{bmatrix} \mathbf{t}_c & \mathbf{n}_c \end{bmatrix} \end{aligned} \quad (3.3.10)$$

where  $\mathbf{e}_{tc}$  is the basis vector set of the centre vector.

The remaining connecting vectors  $\mathbf{a}_s$ ,  $\mathbf{a}_f$  and  $\mathbf{a}_c$  can be written in terms of the start basis vectors, as:

$$\begin{aligned} \mathbf{a}_s &= \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \frac{\pm 1}{\tau_s} \end{pmatrix} \\ \mathbf{a}_f &= \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \frac{\pm 1}{\tau_f} \end{pmatrix} \\ \mathbf{a}_c &= \mathbf{R}(\theta_s)' \begin{pmatrix} a \\ 0 \end{pmatrix} \end{aligned} \quad (3.3.11)$$

The centre vector equation (3.3.9), now becomes:

$$\begin{aligned} c\mathbf{t}_{ct} &= -\mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \frac{\pm 1}{\tau_s} \end{pmatrix} + \mathbf{R}(\theta_s)' \begin{pmatrix} a \\ 0 \end{pmatrix} + \mathbf{R}(\theta_s)' \begin{pmatrix} 0 \\ \frac{\pm 1}{\tau_f} \end{pmatrix} \\ &= \mathbf{R}(\theta_s)' \begin{pmatrix} a \\ \frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s} \end{pmatrix} \end{aligned} \quad (3.3.12)$$

Normalizing the centre vector to unit magnitude, gives:

$$\mathbf{t}_{ct} = \mathbf{R}(\theta_s)' \frac{1}{c} \begin{pmatrix} a \\ \frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s} \end{pmatrix} \quad (3.3.13)$$

This is a rotation equation, that represents the rotation of a unit vector. Hence, the right hand vector must have unit magnitude, to give:

$$\left| \frac{1}{c} \begin{pmatrix} a \\ \frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s} \end{pmatrix} \right| = 1 \quad (3.3.14)$$

### 3. PRODUCING FLYABLE PATH - 3D

---

or:

$$\begin{aligned} \left(\frac{a}{c}\right)^2 + \frac{1}{c^2} \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2 &= 1 \\ \left(\frac{a}{c}\right)^2 &= 1 - \frac{1}{c^2} \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2 \end{aligned} \quad (3.3.15)$$

This can be used to test for a feasible solution, by:

$$\boxed{1 - \frac{1}{c^2} \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2 > 0} \quad (3.3.16)$$

In order to compute the rotation angle  $\theta_s$ , the equation can be written in the form:

$$\begin{aligned} \mathbf{t}_{ct} &= \mathbf{R}(\theta_s)' \begin{pmatrix} \frac{\sqrt{c^2 - \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2}}{c} \\ \frac{\left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)}{c} \end{pmatrix} \\ \mathbf{R}(\theta_s) &= \begin{pmatrix} \cos(\theta_s) & -\sin(\theta_s) \\ \sin(\theta_s) & \cos(\theta_s) \end{pmatrix} \end{aligned} \quad (3.3.17)$$

Expanding this gives:

$$\begin{aligned} \cos(\theta_s) \frac{\sqrt{c^2 - \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2}}{c} + \sin(\theta_s) \frac{\left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)}{c} &= t_{ct1} \\ -\sin(\theta_s) \frac{\sqrt{c^2 - \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2}}{c} + \cos(\theta_s) \frac{\left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)}{c} &= t_{ct2} \end{aligned} \quad (3.3.18)$$

or:

$$\frac{1}{c} \begin{pmatrix} \sqrt{c^2 - \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2} & \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right) \\ -\left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right) & \sqrt{c^2 - \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2} \end{pmatrix} \begin{pmatrix} \cos(\theta_s) \\ \sin(\theta_s) \end{pmatrix} = \mathbf{t}_{ct} \quad (3.3.19)$$

Solving for  $\theta_s$  gives:

$$\begin{pmatrix} \cos(\theta_s) \\ \sin(\theta_s) \end{pmatrix} = \frac{1}{c} \begin{pmatrix} \sqrt{c^2 - \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2} & -\left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right) \\ \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right) & \sqrt{c^2 - \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2} \end{pmatrix} \mathbf{t}_{ct} \quad (3.3.20)$$

Hence:

$$\theta_s = \tan^{-1}(\sin(\theta_s), \cos(\theta_s)) \quad (3.3.21)$$

The final angle  $\theta_f$  can then be determined using:

$$\begin{aligned} \theta &= \theta_s + \theta_f \\ \theta_f &= \theta - \theta_s \end{aligned} \quad (3.3.22)$$

### 3.4 Composite Dubins arc for the 3D manoeuvre

---

An alternate solution is:

$$R(\theta_s)t_{ct} = \begin{pmatrix} \sqrt{c^2 - \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2} \\ \frac{c}{\left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)} \end{pmatrix} \quad (3.3.23)$$

Expanding this gives:

$$\begin{pmatrix} \cos(\theta_s) \\ \sin(\theta_s) \end{pmatrix} = \frac{1}{\Delta} \begin{pmatrix} t_{ct1} & t_{ct2} \\ -t_{ct2} & t_{ct1} \end{pmatrix} \begin{pmatrix} \sqrt{c^2 - \left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)^2} \\ \frac{c}{\left(\frac{\pm 1}{\tau_f} - \frac{\pm 1}{\tau_s}\right)} \end{pmatrix}$$

$$\begin{aligned} \Delta &= t_{ct1}^2 + t_{ct2}^2 \\ &= 1 \end{aligned} \quad (3.3.24)$$

### 3.4 Composite Dubins arc for the 3D manoeuvre

In order to produce a navigation solution for 3D an additional manoeuvre is required. This manoeuvre will take the UAV into a manoeuvre plane defined by the target tangent vector and the sightline vector. This is shown in figure (3.1).

For the three dimensional plane manoeuvre, the start and finish manoeuvre plane are not coincident, which implies that the start and finish binormal vectors are not parallel. Hence the intersection of the start and finish manoeuvre planes is a line, Hence the straight line manoeuvre is uniquely defined for this case. It is a function of the position of the manoeuvre planes, which are not known. The sign of the manoeuvre can be determined by considering the centre line between the two positions. Viewed from each position a positive or negative rotation from the tangent vector to the centre vector will define the sign of the curvature for each manoeuvre. The start manoeuvre and the finish manoeuvre plane is obtained by rotation about the tangent vector  $t$ . So, we have:

$$\begin{aligned} \begin{bmatrix} t_{ms} & n_{ms} & b_{ms} \end{bmatrix} &= \begin{bmatrix} t_s & n_s & b_s \end{bmatrix} R_s \\ \begin{bmatrix} t_{mf} & n_{mf} & b_{mf} \end{bmatrix} &= \begin{bmatrix} t_f & n_f & b_f \end{bmatrix} R_f \end{aligned} \quad (3.4.1)$$

### 3. PRODUCING FLYABLE PATH - 3D

---

where:

$$\begin{aligned} \mathbf{R}_s &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_s) & -\sin(\theta_s) \\ 0 & \sin(\theta_s) & \cos(\theta_s) \end{pmatrix} \\ \mathbf{R}_f &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_f) & -\sin(\theta_f) \\ 0 & \sin(\theta_f) & \cos(\theta_f) \end{pmatrix} \end{aligned} \quad (3.4.2)$$

and where  $\theta_s$  and  $\theta_f$  are the rotation angles for the start and finish manoeuvre planes.

The radius vectors can then be defined in the manoeuvre planes as:

$$\mathbf{r}_s = \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} \begin{pmatrix} 0 \\ \frac{\pm 1}{\tau_s} \\ 0 \end{pmatrix} \quad (3.4.3)$$

and, similarly:

$$\mathbf{r}_f = \begin{bmatrix} \mathbf{t}_{mf} & \mathbf{n}_{mf} & \mathbf{b}_{mf} \end{bmatrix} \begin{pmatrix} 0 \\ \frac{\pm 1}{\tau_f} \\ 0 \end{pmatrix} \quad (3.4.4)$$

The Frenet basis vectors are related by:

$$\begin{bmatrix} \mathbf{t}_f & \mathbf{n}_f & \mathbf{b}_f \end{bmatrix} = \begin{bmatrix} \mathbf{t}_s & \mathbf{n}_s & \mathbf{b}_s \end{bmatrix} \mathbf{R} \quad (3.4.5)$$

where  $\mathbf{R}$  is the rotation matrix required to change the axis set from start to finish axes.

Hence, we have:

$$\mathbf{R} = \begin{pmatrix} \mathbf{t}_f & \mathbf{n}_f & \mathbf{b}_f \end{pmatrix} \cdot \begin{pmatrix} \mathbf{t}_s & \mathbf{n}_s & \mathbf{b}_s \end{pmatrix} \quad (3.4.6)$$

giving

$$\mathbf{R} = \begin{pmatrix} \mathbf{t}_f \cdot \mathbf{t}_s & \mathbf{t}_f \cdot \mathbf{n}_s & \mathbf{t}_f \cdot \mathbf{b}_s \\ \mathbf{n}_f \cdot \mathbf{t}_s & \mathbf{n}_f \cdot \mathbf{n}_s & \mathbf{n}_f \cdot \mathbf{b}_s \\ \mathbf{b}_f \cdot \mathbf{t}_s & \mathbf{b}_f \cdot \mathbf{n}_s & \mathbf{b}_f \cdot \mathbf{b}_s \end{pmatrix} \quad (3.4.7)$$

The connecting vectors  $\mathbf{a}_s$ ,  $\mathbf{a}_f$  and  $\mathbf{a}_c$  form an orthogonal set of vectors. The connecting vectors  $\mathbf{a}_s$  and  $\mathbf{a}_f$  are normal to the vector  $\mathbf{a}_c$ , but are not parallel. Each vector lies in the appropriate manoeuvre plane, which are not coincident. The internal connecting vector  $\mathbf{a}_c$  is common to both manoeuvre planes. It can be thus written in the



### 3.4 Composite Dubins arc for the 3D manoeuvre

---

form:

$$\begin{aligned}
 \boldsymbol{\alpha}_c &= a \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} \boldsymbol{\alpha}_s \\
 &= a \begin{bmatrix} \mathbf{t}_{mf} & \mathbf{n}_{mf} & \mathbf{b}_{mf} \end{bmatrix} \boldsymbol{\alpha}_f \\
 \boldsymbol{\alpha}_s &= \begin{pmatrix} \alpha_{ts} \\ \alpha_{ns} \\ \alpha_{bs} \end{pmatrix} \\
 \boldsymbol{\alpha}_f &= \begin{pmatrix} \alpha_{tf} \\ \alpha_{nf} \\ \alpha_{bf} \end{pmatrix}
 \end{aligned} \tag{3.4.8}$$

The Frenet frame for both manoeuvre planes can be related by

$$\begin{aligned}
 \begin{bmatrix} \mathbf{t}_f & \mathbf{n}_f & \mathbf{b}_f \end{bmatrix} &= \begin{bmatrix} \mathbf{t}_s & \mathbf{n}_s & \mathbf{b}_s \end{bmatrix} \mathbf{R} \\
 \begin{bmatrix} \mathbf{t}_{mf} & \mathbf{n}_{mf} & \mathbf{b}_{mf} \end{bmatrix} &= \begin{bmatrix} \mathbf{t}_f & \mathbf{n}_f & \mathbf{b}_f \end{bmatrix} \mathbf{R}_f \\
 \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} &= \begin{bmatrix} \mathbf{t}_s & \mathbf{n}_s & \mathbf{b}_s \end{bmatrix} \mathbf{R}_s
 \end{aligned} \tag{3.4.9}$$

Hence:

$$\begin{aligned}
 \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} \mathbf{R}'_s &= \begin{bmatrix} \mathbf{t}_s & \mathbf{n}_s & \mathbf{b}_s \end{bmatrix} \\
 \begin{bmatrix} \mathbf{t}_{mf} & \mathbf{n}_{mf} & \mathbf{b}_{mf} \end{bmatrix} \mathbf{R}'_f &= \begin{bmatrix} \mathbf{t}_f & \mathbf{n}_f & \mathbf{b}_f \end{bmatrix} \\
 &= \begin{bmatrix} \mathbf{t}_s & \mathbf{n}_s & \mathbf{b}_s \end{bmatrix} \mathbf{R} \\
 &= \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} \mathbf{R}'_s \mathbf{R}
 \end{aligned} \tag{3.4.10}$$

and so:

$$\begin{bmatrix} \mathbf{t}_{mf} & \mathbf{n}_{mf} & \mathbf{b}_{mf} \end{bmatrix} = \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} \mathbf{R}'_s \mathbf{R} \mathbf{R}_f \tag{3.4.11}$$

This implies:

$$\begin{aligned}
 \boldsymbol{\alpha}_s &= \mathbf{R}'_s \mathbf{R} \mathbf{R}_f \boldsymbol{\alpha}_f \\
 \boldsymbol{\alpha}_f &= \mathbf{R}'_f \mathbf{R} \mathbf{R}_s \boldsymbol{\alpha}_s
 \end{aligned} \tag{3.4.12}$$

The radius vectors  $\mathbf{r}_s$  and  $\mathbf{r}_f$  can also be described in start manoeuvre axes, to give:

$$\begin{aligned}
 \mathbf{r}_s &= \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} \begin{pmatrix} 0 \\ \pm 1/\tau_s \\ 0 \end{pmatrix} \\
 \mathbf{r}_f &= \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} \mathbf{R}'_s \mathbf{R} \mathbf{R}_f \begin{pmatrix} 0 \\ \pm 1/\tau_f \\ 0 \end{pmatrix}
 \end{aligned} \tag{3.4.13}$$

### 3. PRODUCING FLYABLE PATH - 3D

Now, the vectors  $\mathbf{a}_s$  and  $\mathbf{a}_f$  lie in the manoeuvre planes and are normal to the connecting vector  $\mathbf{a}_c$ . These can also be defined in start manoeuvre axes, in the form:

$$\begin{aligned}\mathbf{a}_s &= \frac{\pm 1}{\tau_s} \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} \boldsymbol{\beta}_s \\ \mathbf{a}_f &= \frac{\pm 1}{\tau_f} \begin{bmatrix} \mathbf{t}_{mf} & \mathbf{n}_{mf} & \mathbf{b}_{mf} \end{bmatrix} \boldsymbol{\beta}_f \\ &= \frac{\pm 1}{\tau_f} \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} \mathbf{R}'_s \mathbf{R} \mathbf{R}_f \boldsymbol{\beta}_f\end{aligned}\quad (3.4.14)$$

where, to ensure that the connection vectors lie in the manoeuvre plane, and are normal to the internal connection vector  $\mathbf{a}_c$ , we have:

$$\begin{aligned}\boldsymbol{\beta}_s &= \frac{1}{b_s} \begin{pmatrix} -\alpha_{ns} \\ \alpha_{ts} \\ 0 \end{pmatrix} \\ \boldsymbol{\beta}_f &= \frac{1}{b_f} \begin{pmatrix} -\alpha_{nf} \\ \alpha_{tf} \\ 0 \end{pmatrix} \\ b_s &= \sqrt{\alpha_{ns}^2 + \alpha_{ts}^2} \\ b_f &= \sqrt{\alpha_{nf}^2 + \alpha_{tf}^2} \\ \boldsymbol{\beta}_s \boldsymbol{\alpha}_s &= 0 \\ \boldsymbol{\beta}_f \boldsymbol{\alpha}_f &= 0\end{aligned}\quad (3.4.15)$$

The position of the finish point  $\mathbf{p}_f$  relative to the start position  $\mathbf{p}_s$  is measured in start plane axes  $[\mathbf{t}_s \ \mathbf{n}_s \ \mathbf{b}_s]$ , so that:

$$\begin{aligned}\mathbf{p}_f - \mathbf{p}_s &= \begin{bmatrix} \mathbf{t}_s & \mathbf{n}_s & \mathbf{b}_s \end{bmatrix} \mathbf{p} \\ &= \begin{bmatrix} \mathbf{t}_{ms} & \mathbf{n}_{ms} & \mathbf{b}_{ms} \end{bmatrix} \mathbf{R}'_s \mathbf{p} \\ \mathbf{p}_m &= \mathbf{R}'_s \mathbf{p} \\ \mathbf{p} &= \begin{pmatrix} p_t \\ p_n \\ p_b \end{pmatrix}\end{aligned}\quad (3.4.16)$$

then, the vector sum for the position vector is given by:

$$\begin{aligned}\mathbf{p} &= -\mathbf{r}_s + \mathbf{a}_s + \mathbf{a}_c - \mathbf{a}_f + \mathbf{r}_f \\ \mathbf{p} + \mathbf{r}_s - \mathbf{r}_f &= \mathbf{a}_s + \mathbf{a}_c - \mathbf{a}_f\end{aligned}\quad (3.4.17)$$

Re-writing this in the start manoeuvre axes, gives:

$$\boxed{\mathbf{R}'_s \mathbf{p} + \mathbf{r}_s - \mathbf{R}'_s \mathbf{R} \mathbf{R}'_f \mathbf{r}_f = \mathbf{a}_s + \mathbf{a}_c - \mathbf{R}'_s \mathbf{R} \mathbf{R}_f \mathbf{a}_f}\quad (3.4.18)$$

Re-writing this in the start plane axes, gives:

$$\boxed{p + R_s r_s - RR'_f r_f = R_s a_s + R_s a_c - RR_f a_f} \quad (3.4.19)$$

### 3.5 Path Length - Dubins 3D

As the Dubins path in 3D forms a composite path of four segments, the path length is the sum of the length of these segments and is given by:

Length of Dubins path:

$$\begin{aligned} L &= L_i + L_s + L_t + L_f \\ &= \frac{\alpha_i}{\kappa_s} + \frac{\alpha_s}{\kappa_s} + \sqrt{(\Delta X)^2 + (\Delta Y)^2} + \frac{\alpha_f}{\kappa_f} \end{aligned} \quad (3.5.1)$$

where  $L_i$ ,  $L_s$ ,  $L_t$  and  $L_f$  are the lengths of initial arc, start arc, tangent line and the final arc respectively.  $\Delta X$  and  $\Delta Y$  are difference of  $x$  and  $y$  coordinates of tangent vectors respectively and  $\alpha_i$ ,  $\alpha$  and  $\kappa$  are the included angle of the arc and its curvature respectively. The suffix  $i$ ,  $s$  and  $f$  respectively represent the initial, start and end turns. It is apparent from the equation that the length of the Dubins path is simple and is not computationally intensive.

Application to a swarm of moving vehicles requires that their path length shall be controlled. That is the length of paths shall be adjustable. This condition can be met by Dubins path. Consider the equation 3.5.1. In this equation, the values of  $\Delta X$  and  $\Delta Y$  are fixed by the curvature  $\kappa$  of the circular arcs (refer section 2.1). Hence, the length of the Dubins path is completely determined by the curvature of circular arcs. Thus by simply varying the curvature of the turn, we can easily control the length of the path.

### 3.6 Pythagorean Hodograph Path-3D

A Pythagorean Hodograph is a polynomial curve first introduced by Farouki [59], [64], and [65]. Here we give a brief introduction of the PH path. Here a fifth order PH curve is used as this is the lowest order curve which has inflexion points that can provide sufficient flexibility [59]. The PH path provides exact calculation of path length, it's curvature and the offset curve are rational. Substituting an appropriate polynomial  $\sigma(t)$  such that  $\sigma(t)^2 = \dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2$  in the equation (3.6.1) produces a path length  $s(t)$  and speed  $\dot{s}(t)$  which are reduced to an integral of the polynomial  $\sigma(t)$  and the polynomial itself respectively. The canal tube around the path is also

### 3. PRODUCING FLYABLE PATH - 3D

---

rational which is used to define the safety region around each UAV. The basics of the PH curve is given in appendix B.

The path-length  $s(t)$  of the curve  $r(t) = \{x(t), y(t), z(t)\}$  is:

$$s(t) = \int_{t_1}^{t_2} \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} dt \quad t \in [t_1, t_2] \quad (3.6.1)$$

The term inside the square root in equation (3.6.1) is the sum of the squares of the hodographs. If  $x(t)$ ,  $y(t)$ , and  $z(t)$  are polynomial functions of  $t$  and we could make this term a perfect square, then the path-length would simply be an integral of a polynomial  $\sigma(t)$ :

$$\boxed{\sigma(t)^2 = \dot{x}(t)^2 + \dot{y}(t)^2 + \dot{z}(t)^2} \quad (3.6.2)$$

$$s(t) = \int_{t_1}^{t_2} |\sigma(t)| dt \quad (3.6.3)$$

For any polynomial curve, if its hodographs meet the Pythagorean condition, the curve is called a Pythagorean Hodograph curve. Now, for a polynomial curve, i.e. when  $x(t)$ ,  $y(t)$  and  $z(t)$  are polynomials, a useful PH path is designed by selecting suitable polynomials for the hodographs,  $\dot{x}(t)$ ,  $\dot{y}(t)$ , and  $\dot{z}(t)$ . The main advantages of this formulation are (i) Calculation of path length without any approximation, (ii) Equal increment of distance traveled along the curve for equal increment of the parameter  $t$ , (iii) Rational parametric speed, and (iv) Rational intrinsic properties (curvature, torsion and canal surface).

#### 3.6.1 Spatial PH Curve

Consider a polynomial space curve  $r(t) = (x(t), y(t), z(t))$  represented in pure quaternion form:  $r(t) = (x(t)i + y(t)j + z(t)k)$ . The curve  $r(t)$  is a PH curve only if

$$\frac{dr}{dt} = Q(t)iQ^*(t) \quad (3.6.4)$$

for some quaternion polynomial  $Q(t) = u(t) + iv(t) + jp(t) + kq(t)$ . The hodographs of  $r(t)$  satisfies:

$$\dot{x}(t) = u(t)^2 + v(t)^2 - p(t)^2 - q(t)^2 \quad (3.6.5a)$$

$$\dot{y}(t) = 2(u(t)q(t) + v(t)p(t)) \quad (3.6.5b)$$

$$\dot{z}(t) = 2(v(t)q(t) - u(t)p(t)) \quad (3.6.5c)$$

$$\sigma(t) = u(t)^2 + v(t)^2 + p(t)^2 + q(t)^2 \quad (3.6.5d)$$

### 3.7 Design of Flyable Path using PH curve

---

These equations provide sufficient and necessary conditions for a polynomial space curve to be PH. The quaternion  $Q(t)$  in Bézier form is:

$$Q(t) = \sum_{i=0}^2 Q_i \binom{2}{i} t^i (1-t)^{2-i}; \quad t \in [0, 1] \quad (3.6.6)$$

The coefficients  $Q_0$ ,  $Q_1$  and  $Q_2$  have to be found out by Hermite interpolation as explained in references [[64], [65]]. The length of the curve  $s(t)$  is:

$$s(t) = \int_{t_1}^{t_2} |Q(t)|^2 dt \quad (3.6.7)$$

A PH curve designed by (3.6.6) is obtained by interpolating for positions and directions. This is not a smooth path. Hence the path has to be smoothed for curvature continuity.

### 3.7 Design of Flyable Path using PH curve

The equation (3.6.6) is quintic polynomial designed by interpolating the free vectors at the boundaries. The free vectors have positions  $(x, y, z)$  and direction  $(\phi, \theta)$  in space. A curve interpolating two such vectors is called Hermite interpolation. The resulting curve will have tangent continuity. For real time application, it is essential to have curvature continuity as the curvature is proportional to lateral dynamics of a moving vehicle. In order to have curvature continuity we impose maximum curvature bound. Let  $\kappa_{max}$  is the maximum curvature. In addition to the positions and curvature we interpolate for curvature at the end points. As no closed form solution exists for curvature interpolation an iterative process is adopted to meet the curvature constraint. The PH curve is represented in fifth order Bernstein-Bézier polynomial.

$$r(t) = \sum_{k=0}^5 b_k \binom{5}{k} (1-t)^{(5-k)} t^k; \quad t \in [0, 1] \quad (3.7.1)$$

where  $b_k = (x_k, y_k, z_k)$  are control points, whose vertices define the control polygon or Bézier polygon and  $k = 0 \dots 5$ . The initial and final configurations are  $P_s(x_s, y_s, z_s, \phi_s, \theta_s)$  and  $P_f(x_f, y_f, z_f, \phi_f, \theta_f)$  respectively. The four control points of the Bézier polygons

### 3. PRODUCING FLYABLE PATH - 3D

are calculated by first order Hermite interpolation as follows:

$$b_0 = (x_s, y_s, z_s) \quad (3.7.2a)$$

$$b_5 = (x_f, y_f, z_f) \quad (3.7.2b)$$

$$d_0 = m_0(\cos(\phi_s) \cos(\theta_s), \cos(\phi_s) \sin(\theta_s), \sin(\phi_s)) \quad (3.7.2c)$$

$$d_5 = m_5(\cos(\phi_s) \cos(\theta_s), \cos(\phi_s) \sin(\theta_s), \sin(\phi_s)) \quad (3.7.2d)$$

$$b_1 = b_0 + \frac{1}{5}d_0 \quad (3.7.2e)$$

$$b_4 = b_5 - \frac{1}{5}d_5 \quad (3.7.2f)$$

where  $(x_s, y_s, z_s)$  is initial position,  $(x_f, y_f, z_f)$  is final position,  $(\phi_s, \theta_s)$  is initial orientation and  $(\phi_f, \theta_f)$  is final orientation. The positive constants  $m_0$  and  $m_5$  play a crucial role in path planning. The constants increase the length of the control vectors  $b_0\vec{b}_1$  and  $b_4\vec{b}_5$  which in turn fix the control points  $b_2$  and  $b_3$  satisfying the PH condition (3.6.5). This changes the curvature and torsion of the path with corresponding change in shape. From (3.7.2), the control points  $(b_0, b_1, b_4, b_5)$  are fixed. Now the problem is reduced to finding the control points,  $b_2$  and  $b_3$ . This is found out by (3.6.6).

#### 3.7.1 Design of flyable path

The path resulted from above interpolation is tangent continuous. This path is further required to be interpolated for curvature to make it flyable path. The  $r^{\text{th}}$  derivative of the path is:

$$\frac{d^r r(t)}{dt^r} = \frac{5!}{(5-r)!} \sum_{j=0}^{5-r} \Delta^r B_j \binom{5-r}{j} t^j (1-t)^{5-r-j} \quad (3.7.3)$$

$$j = 0, 1, \dots, 5$$

where  $\Delta^r b_i = \sum_{j=0}^r \binom{r}{j} (-1)^{r-j} b_{i+j}$ ,  $\Delta^0 B_j = B_j$  and  $\Delta^r B_j = \Delta^{k-1} B_{j+1} - \Delta^{k-1} B_j$ ,  $k = 1, 2, \dots, (5-r)$

Using the equation (3.7.3) the values of derivatives of the curve at the boundary points are:

$$\dot{r}(t)_{t=0} = 5B_{01} \quad (3.7.4a)$$

$$\dot{r}(t)_{t=1} = 5B_{54} \quad (3.7.4b)$$

$$\ddot{r}(t)_{t=0} = 20(B_{12} + B_{01}) \quad (3.7.4c)$$

$$\ddot{r}(t)_{t=1} = 20(B_{54} + B_{34}) \quad (3.7.4d)$$

$$\dddot{r}(t)_{t=0} = 120(B_{03} - 3B_{12}) \quad (3.7.4e)$$

$$\dddot{r}(t)_{t=1} = 60(B_{52} - 3B_{43}) \quad (3.7.4f)$$

### 3.7 Design of Flyable Path using PH curve

where  $B_{ij} = B_j - B_i$

$$\begin{bmatrix} \dot{r}(t) & \ddot{r}(t) & \dddot{r}(t) \end{bmatrix}_{t=0} = \begin{bmatrix} 5B_{01} & 20(B_{12} + B_{01}) & 120(B_{03} - 3B_{12}) \end{bmatrix} \quad (3.7.5)$$

$$\begin{bmatrix} \dot{r}(t) & \ddot{r}(t) & \dddot{r}(t) \end{bmatrix}_{t=1} = \begin{bmatrix} 5B_{54} & 20(B_{54} + B_{34}) & 60(B_{52} - 3B_{43}) \end{bmatrix} \quad (3.7.6)$$

where the square brackets in the above equations represents determinants.

Using equation (3.7.4) in (A.2.1)

$$\begin{aligned} |\dot{r}(t) \times \ddot{r}(t)|_{t=0} &= |5B_{01} \times 20B_{12}| \\ &= 100|(B_1 - B_0) \times (B_2 - B_1)| \end{aligned} \quad (3.7.7)$$

$$= 100A_0 \quad (3.7.8)$$

$$\dot{r}(t)_{t=0} = 5(B_1 - B_0) \quad (3.7.9)$$

$$\boxed{|\kappa(t)|_{t=0} = \frac{4}{5} \frac{A_0}{\|B_{01}\|^3}} \quad (3.7.10)$$

where  $A_0$  is the area of triangle formed by the control points  $B_0$ ,  $B_1$  and  $B_2$ .

Similarly the curvature at the end point that is  $t = 1$  is:

$$\begin{aligned} |\dot{r}(t) \times \ddot{r}(t)|_{t=1} &= |5B_{34} \times 20B_{45}| \\ &= 100|(B_4 - B_3) \times (B_5 - B_4)| \end{aligned} \quad (3.7.11)$$

$$= 100A_1 \quad (3.7.12)$$

$$\dot{r}(t)_{t=1} = 5(B_5 - B_4) \quad (3.7.13)$$

$$\boxed{|\kappa(t)|_{t=1} = \frac{4}{5} \frac{A_1}{\|B_{45}\|^3}} \quad (3.7.14)$$

where  $A_1$  is the area of triangle formed by the control points  $B_3$ ,  $B_4$  and  $B_5$ .

For the maximum curvature  $\kappa_{max}$ , the boundary curvature has to satisfy:

$$\kappa_{max} \leq \frac{4}{5} \frac{A_0}{\|B_{01}\|^3} \quad (3.7.15a)$$

$$\kappa_{max} \leq \frac{4}{5} \frac{A_1}{\|B_{45}\|^3} \quad (3.7.15b)$$

Using equations (3.7.4) and (3.7.5) in (A.2.2), the values of torsion at the boundary points becomes:

$$|\tau(t)_{t=0}| = \frac{\begin{bmatrix} 5B_{01} & 20(B_{12} + B_{01}) & 120(B_{03} - 3B_{12}) \end{bmatrix}}{|5B_{01} \times 20B_{12}|^2} \quad (3.7.16)$$

$$|\tau(t)_{t=1}| = \frac{\begin{bmatrix} 5B_{54} & 20(B_{54} + B_{34}) & 60(B_{52} - 3B_{43}) \end{bmatrix}}{|5B_{34} \times 20B_{45}|^2} \quad (3.7.17)$$

### 3. PRODUCING FLYABLE PATH - 3D

---

Substituting equations (3.7.8) and (3.7.12) in the equation (3.7.16), the maximum torsion  $\tau_{max}$  at the boundary points have to satisfy:

$$|\tau(t)_{t=0}| \leq \frac{\left[ 5B_{01} \ 20(B_{12} + B_{01}) \ 120(B_{03} - 3B_{12}) \right]}{A_0^2} \quad (3.7.18a)$$

$$|\tau(t)_{t=1}| \leq \frac{\left[ 5B_{54} \ 20(B_{54} + B_{34}) \ 60(B_{52} - 3B_{43}) \right]}{A_1^2} \quad (3.7.18b)$$

where  $A_0$  is the area of triangle formed by the control points  $B_0, B_1$  and  $B_2$  and  $A_1$  is the area of triangle formed by the control points  $B_5, B_4$  and  $B_3$ .

Thus a flyable path is designed by interpolating the positions, directions and curvature at the end points. The PH path is optimized for the maximum curvature bound (3.7.15). A similar procedure is adopted for the torque optimization. However, the resulting form (3.7.18) can not be interpreted in simple geometrical form as (3.7.15). Hence an iterative procedure is adopted to arrive at an optimal value of the torque. In both cases, the length of the tangent vectors are increased till the path meets the maximum bound on curvature and torque in equations (3.7.18 & 3.7.15). To achieve this, the boundary tangent vector equations are increased by increasing the values of  $m_0$ , and  $m_5$ .

$$d_0 = m_0(\cos(\phi_s) \cos(\theta_s), \cos(\phi_s) \sin(\theta_s), \sin(\phi_s)) \quad (3.7.19a)$$

$$d_5 = m_5(\cos(\phi_f) \cos(\theta_f), \cos(\phi_f) \sin(\theta_f), \sin(\phi_f)) \quad (3.7.19b)$$

where  $m_0 \in [1, \infty]$  and  $m_5 \in [1, \infty]$ .

### 3.8 Summary

This chapter discusses the method of producing flyable paths in three dimensions. The Dubins path is produced by finding a common intersecting plane which provides a maneuver plane where the 2D Dubins path is produced. Another maneuver plane is produced with an initial rotation which contains an arc that connects the Dubins path tangentially. As the procedure to derive the 3D composite path with Clothoid is similar, it is omitted. The flyable PH paths are produced by optimizing the paths of tangent continuity into curvature continuity. This is achieved by increasing the lengths of boundary tangent vectors.



# Solution to Simultaneous Arrival

**T**his chapter discusses the solution to the problem of simultaneous arrival on target. The solution involves how to accomplish the mission by satisfying various constraints. In free space, the main constraints are the maximum bound on the curvatures and safety constraints to avoid inter-collision of UAVs. The threat avoidance in a cluttered environment is discussed in the next chapter. The curvature constraint is solved in the first phase of the path planning where the path of maximum bound on curvatures is produced. The second phase is discussed to produce feasible path.

## 4.1 Flyable Paths

A path satisfying the maximum curvature constraint is a flyable path. As the curvature is proportional to the lateral force acting on the UAV, it is necessary for a path to meet the maximum curvature bound of the UAV. Thus, at any point on the path the curvature shall not be greater than the maximum curvature bound allowed for each UAV. As seen in earlier chapters, the Dubins path provides the path of maximum curvature bound and it is limited to use in rotorcraft. This is due to the hovering ability of rotorcraft. Hence, the possibility of approximating the Dubins path is sought. In this respect, two types of paths: PH and clothoid paths are used to provide shortest path solution with curvature continuity. These paths can be used for both the

## 4. SOLUTION TO SIMULTANEOUS ARRIVAL

---

rotorcraft and fixed-wing UAVs.

$$\boxed{|\kappa_i| < \kappa_{max}} \quad (4.1.1)$$

where  $\kappa_i$  the curvature of  $i^{th}$  path and  $\kappa_{max}$  is the maximum curvature of the path.

### 4.2 Feasible Paths

A feasible path is both flyable and safe. The flyable path satisfy the principal constraint, maximum bound on curvature. However, this path does not guarantee safety to the UAVs. Because, the flyable path does not consider the safety during its design. Hence the flyable paths needs to satisfy additional constraints to ensure the safety of the UAVs. Two safety conditions are defined here for inter-collision avoidance in free space. They are: (i) Minimum separation distance and (ii) Non-intersection of paths at equal lengths.

#### 4.2.1 Minimum Separation Distance

The minimum separation distance  $d_{sep}$  between any two UAVs should at least be equal to the sum of corresponding radii of the safety circles. For homogeneous UAVs this will be two times the radius of the safety circle. Refer figure (4.2.1). The separation between two UAVs is measured by calculating the Euclidean distance between two points on two different paths. The separation distance between  $k^{th}$  path and  $l^{th}$  path at a particular length or time is  $d_{sep} = \sqrt{(z_l - z_k)^2 + (y_l - y_k)^2 + (x_l - x_k)^2}$ , where  $(x_k, y_k, z_k)$  is the point on  $k^{th}$  path and  $(x_l, y_l, z_l)$  is the point on  $l^{th}$  path at that instant or length. The two 2D case is reduced into:  $d_{sep} = \sqrt{(y_l - y_k)^2 + (x_l - x_k)^2}$ . This value should be greater than or equal to  $2R_s$ . It is important to note here that at the points of failure of this condition, the lengths of the paths from their corresponding starting pose  $P_s(x_s, y_s, z_s)$  may differ. This is important in deciding the collision avoidance in the case of constant speed flights. This is explained in detail in the section below.

In general, the minimum separation distance between two paths has to meet the following equation:

$$\boxed{d_{sep,k,l} \geq R_{s,k} + R_{s,l}} \quad (4.2.1)$$

where,  $d_{sep,k,l}$  is the separation distance between the  $k^{th}$  and  $l^{th}$  path,  $R_{s,k}$  is the safety radius of  $k^{th}$  path and  $R_{s,l}$  is the safety radius of  $l^{th}$  path.

For homogenous UAVs, the above equation (4.2.1) is reduces into:

$$d_{sep} \geq 2R_s \quad (4.2.2)$$

For a 2D PH path, the minimum separation distance can be represented by an offset path. The offset path at an offset distance  $d$  is given by:

$$r_d(t) = r(t) \pm d\mathbf{N}(t) \quad (4.2.3)$$

where  $r(t)$  is the PH path and  $r_d(t)$  is its offset path and  $N(t)$  is the normal vector to  $r(t)$ .

Similarly, for a spatial PH path, the canal surface or tube  $r_s(t)$  of radius equal to the minimum separation distance provides a safety margin. The equation of canal surface is:

$$r_s(t) = r(t) + \mathbf{N}(t) \cos(\theta) + \mathbf{B}(t) \sin(\phi) \quad (4.2.4)$$

where  $\mathbf{N}(t)$  is the normal vector and  $\mathbf{B}(t)$  is the binormal vector, which can be analytically computed from curve parameterization.

### 4.2.2 Non-intersection of Paths at Equal Length

Suppose two flyable paths failed to meet the minimum separation distance condition. For constant speed flights, still there is a possibility of no collision. Because, as the UAVs are flying at constant speed at the same time from their initial pose, they traverse equal distance in equal increment of time. Therefore there is a possibility of no collision occurs even if two flyable paths intersect with each other. Consider two paths  $k$  and  $l$  intersect at a point  $X$ . The path length of  $k^{\text{th}}$  path from its start pose to  $X$  is  $L_{k,int}$  and the path length of  $l^{\text{th}}$  path from its start pose to  $X$  is  $L_{l,int}$ . The difference of lengths at the point of intersection is  $d_{int} = |L_{k,int} - L_{l,int}|$ . This value must be at least equal to the sum of radii of safety circles of corresponding UAVs. This shows the possibility of no-collision even if the paths fail to meet the minimum separation distance. Hence the condition minimum-separation-distance, is necessary but not sufficient.

$$\boxed{d_{int,k,l} \geq R_{s,k}^X + R_{s,l}^X} \quad (4.2.5)$$

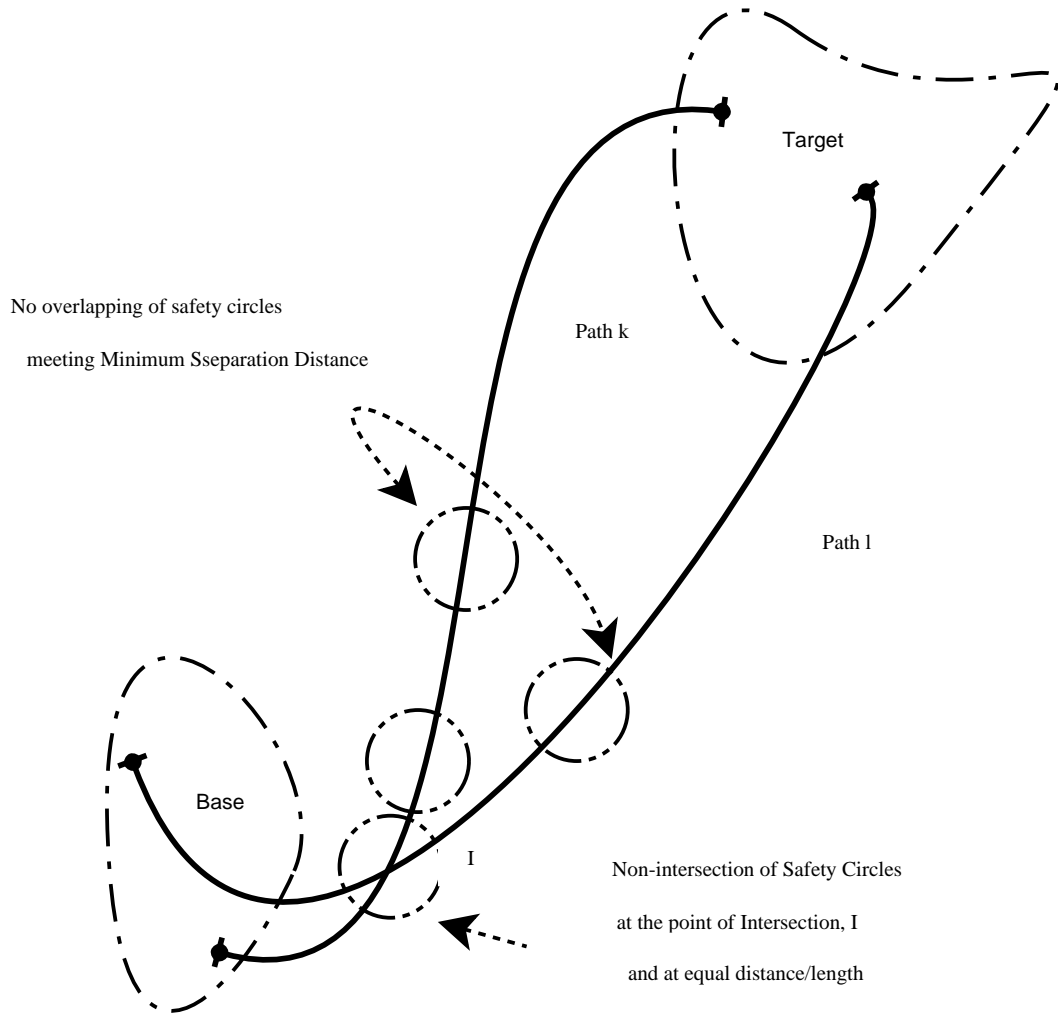
where,  $d_{int,k,l}$  is the difference in lengths of  $k^{\text{th}}$  and  $l^{\text{th}}$  paths at the intersection point  $X$ ,  $R_{s,k}^X$  is the safety radius of  $k^{\text{th}}$  path at  $X$  and  $R_{s,l}^X$  is the safety radius of  $l^{\text{th}}$  path at  $X$ .

For homogenous UAVs, the above equation (4.2.5) is reduces into:

$$d_{int} \geq 2R_s \quad (4.2.6)$$

Figure (4.2.1) shows the schematic of both safety constraints. In two dimensional maneuver, the collision avoidance is tested by a moving safety circle along the path

#### 4. SOLUTION TO SIMULTANEOUS ARRIVAL



**Figure 4.1:** Safe Flight Path explains producing safe flight path by testing the conditions: (i) Minimum separation distance and (ii) Non-intersection at equal distance

and in the case of three dimensional maneuver the safety circle is replaced with a safety sphere. This is simulated with offset curves in the case of 2D PH paths and with canal surface in the case of 3D PH path. The non-overlapping of safety circles of any two paths meet the safety constraints. If the flyable paths meet (4.2.1), the paths are safe to fly and there is no need to replan the path. On failure of this condition, the second condition (4.2.5) is tested for the paths. In the event of failure of both conditions, the replanning is done.

In the presence of stationary threats (refer section 5.1), the threats are detected by intersection of path with the threat region. If the locations of the threat regions are known, it is enough to test for intersection of the threat boundary with the safety circle (equation (4.2.7)). If the intersection is not empty, the replanning can be done

### 4.3 Paths of Equal Length

by increasing the curvature of the path or by creating intermediate way points.

$$\boxed{O_{obs} \cap O_{Rs}} \quad (4.2.7)$$

In the case of Dubins and Clothoid paths, three possibilities are possible: (i) The radius of curvature can be increased to meet the condition or (ii) the next shortest path can be selected from the sets or (iii) a new path can be planned from the point of failure. In the case of the PH path, two possibilities are possible: (i) the radius of curvature is increased by increasing the lengths of the boundary tangent vectors to meet the conditions (refer sections 2.4, and 3.7.1).

There are minimum two conditions are to be tested for every two paths in a flock of UAVs. For a group of  $N$  UAVs, taking  $r$  UAVs at a time, the safety conditions have to be tested for  $n_u$  times, where  $n_u$  is given by:

$$\boxed{n_u = 2 \frac{N!}{r!(N-r)!}} \quad (4.2.8)$$

It is important to note here that the safety conditions are not imposed on the flyable paths. In contrast, the flyable paths are tested to meet these conditions. Thus, the use of search methods are minimized in producing the paths of equal lengths. In the event of failure of these conditions, replanning is done by increasing the curvature of the flyable paths or by creating new way-points.

### 4.3 Paths of Equal Length

As defined in section (1.4), the problem of simultaneous arrival on target is planned to be solved by producing the paths equal lengths. The safe flyable paths that is the feasible paths obtained by satisfying the conditions (4.2.5, 4.2.1, and 4.1.1) may not be equal in lengths. Hence it is essential to make the all the paths to have equal length by adjusting the lengths of each path. The paths of equal length are generated by increasing the length of shorter path to the length of the longest one in the set of  $N$  paths, where  $N$  is the number of UAVs. The lengths of the feasible paths are calculated using (2.1.6b) or (2.1.23) for Dubins path. The path length of the PH path is calculated using (3.6.3). For  $N$  number of UAVs, with the length of each path  $L_i$ , the set of path lengths  $\mathbf{L}$  is:

$$\mathbf{L} = \{L_i\}, \quad i = 1, \dots, N \quad (4.3.1)$$

## 4. SOLUTION TO SIMULTANEOUS ARRIVAL

---

### 4.3.1 Reference Path

The longest path is called as the reference path. This is the longest path from the set of paths in the equation (4.3.1). The length of the reference path is:

$$L_{ref} = \max(\mathbf{L}) \quad (4.3.2)$$

where  $L_{ref}$  represents the length of the reference path.

### 4.3.2 Equal path lengths

The path lengths of  $(N - 1)$  UAVs are increased to the length of the reference path. Lengths of the Dubins paths are increased by increasing the turn radii (equation (2.1.6c)), while that of the PH path is done by increasing the length of boundary tangent vectors (equations (2.4.16a & 3.7.19a)) and (equations (2.4.16b & 3.7.19b)). This condition is implemented as:

$$\text{find } \kappa, \text{ such that } L_i - L_{ref} = 0, \quad i = 1, \dots, N - 1 \quad (4.3.3)$$

## 4.4 Algorithm - free space

The algorithm in this section details the generation of paths of equal lengths by increasing the radius of curvatures of the path. Other methods like replanning from the point of failure of safety conditions and choose the next shortest path from the set of paths can also be implemented in the same way. Also the solution is possible by creating an intermediate way-point.

- (i) Produce flyable paths for each UAV.
- (ii) Change the course of the path to meet the safety constraints.
- (iii) Calculate the length of the paths.
- (iv) Find the reference path.
- (v) Increase the length of the shorter paths to the length of the reference path. This results in paths of equal length.
- (vi) Check again for the paths meeting the safety constraints.
- (vii) If not increase the length of the curvature to meet the safety conditions and produce the paths of equal lengths.

### 4.5 Summary

This chapter discusses a solution approach to the simultaneous arrival on target by multiple UAVs. The curvature constraint, and safety constraints are discussed to produce feasible paths. The flyable paths are produced by the first phase of the path planning. The second phase of producing feasible paths relies on curvatures of the paths. The end-point curvatures of the paths are increased to meet the constraints. This is continued to produce the paths of equal lengths. However, another approach is also possible to produce a feasible path which is done by creating an intermediate way-point. This method is discussed in the following chapter.

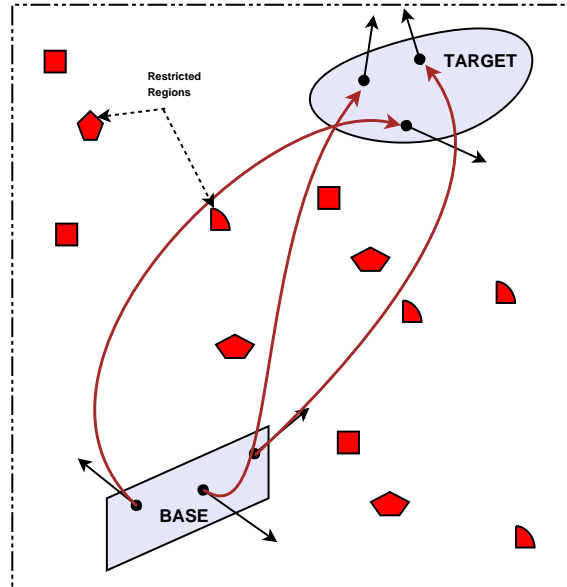
# Path Planning in cluttered environment

Path planning in a cluttered environment is described here. Obstacle avoidance methods are studied by many researchers in the robotic community like potential field function [[66], ] and Voronoi diagram [[67], [68], [51] and [50]]. Yang and Zhao [69] describes a path planning in the midst of known obstacles and conflicts. They used  $A^*$  algorithm for path planning. But the path produced by the algorithm is not flyable as it consists of series of straight lines. The resultant path is not flyable as it does not satisfy the maximum curvature bound. Richards [45] uses Mixed Integer Linear Programming to solve the collision avoidance problems. The resulting path from these approaches does not provide a flyable path. Taking this point as initiation and in contrast to these approaches, flyable paths are used here to solve the problem of collision avoidance. Hence, the problem is reduced to solving the equation (1.3.1). The equation is repeated here for convenience.

$$P_{si}(x_{si}, y_{si}, z_{si}, \phi_{si}, \theta_{si}) \xrightarrow{r_i(t)} P_{fi}(x_{fi}, y_{fi}, z_{fi}, \phi_{fi}, \theta_{fi}), |\kappa_i(t)| < \kappa_{i,max}, |\tau_i(t)| < \tau_{i,max}, \prod_k \quad (5.0.1)$$

A schematic cluttered environment is shown in figure (5.1). The threat regions are modeled as rectangular boxes. The regions are assumed stationary and their positions are known. The threat regions are called as restricted regions. The mission is: simultaneous arrival on target avoiding the threats and also inter-collision avoidance of UAVs. The UAVs set off from the base and have to reach the target





**Figure 5.1:** UAVs in cluttered environment: The restricted regions are the threats which are assumed stationary. The UAVs have to set off from the base and has to reach the target at the same time. The solution approach is by generating the paths of equal lengths. The paths must be flyable and also have to meet the inter-collision avoidance of UAVs and avoid the threat regions

## 5.1 Threat Detection and Avoidance

First lets define the threat region. The threat region is assumed known a priori. The boundary of the region is fixed and is simplified with rectangular in shape. Before avoiding the threats, it is necessary to detect it. In practice the threat is detected by sensor. The sensor is modeled with range. It is the distance at which a sensor can detect a threat. Once a threat is detected, the UAV has to replan the path either by varying the curvature between two way-points under consideration or replan is done with an intermediate way-point.

### 5.1.1 Safety-Region Inclusion

For a threat with known region, it is enough to verify whether the path passes through the threat region. Or in other words, if any part of the path is contained in the threat region. If the intersection of the safety circle with the threat region is empty, then the flyable path is safe to fly. Otherwise, the flyable path needs to be changed for safe flight paths. Representing the restricted region by  $X$  and the safety

## 5. PATH PLANNING IN CLUTTERED ENVIRONMENT

---

circle by  $C_{R_s}$ , the condition for testing the safety of the UAV is:

$$\boxed{\mathbf{X} \cap C_{R_s}} \quad (5.1.1)$$

This method works on the idea that safety is ensured when no point of the path is inclusive of a restricted zone. The idea behind this approach is that if there is an intersection, the UAV has to take a reactive approach to avoid or attack the threat. The restricted region is defined by a set  $\mathbf{X}$  whose values lie in  $\mathfrak{R}$ .

$$\mathbf{X} = \{x \in [a \ b] \ \& \ y \in [c \ d]; \ a, b, d \in \mathfrak{R}\} \quad (5.1.2)$$

The Dubins and clothoid paths are tested for their safety circles intersect with the flyable path. The turning radii of these paths are increased to achieve the threat avoidance. In the case of the PH path, the offset path is used. If any point  $T_o$  on offset-path  $r_o$  falls in the set  $\mathbf{X}$ , there is an intersection. The safety of UAVs is ensured when the resulting set is empty.

$$\boxed{\mathbf{X} \cap T_o = \emptyset} \quad (5.1.3)$$

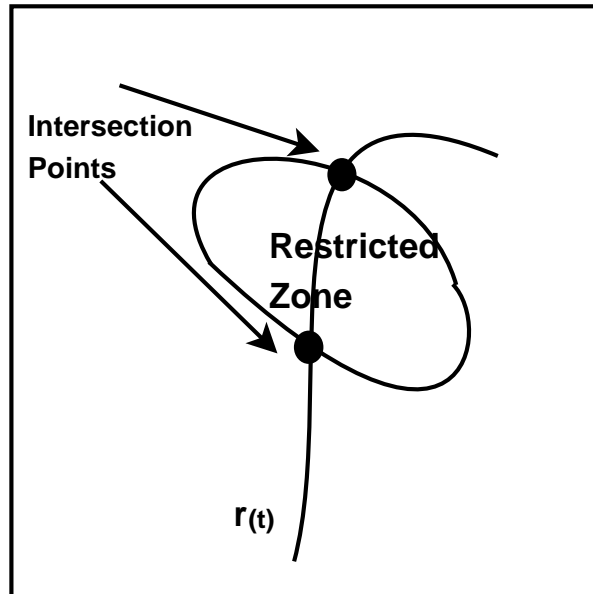
where  $T_o$  is any point on offset-path. Though the offset path is used in this method, sometimes, the offset path intersects with each other due to large curvature. As a matter of precaution, the safety-circles are used to test the intersection with the threat regions. A safe flight path is ensured by satisfying the condition (5.1.3) and is verified iteratively for an empty set. If a non-empty set results, the values  $c_0$  and  $c_5$  in equation (2.4.16) are adjusted until the path meets the safety constraints. The figure (5.1.1) shows the approach.

### 5.1.2 Safety Distance

For a threat region of unknown boundary, the distance between the threat and the UAV needs to be measured to fly safely away from the threat region. This can be measured in the direction of heading and the normal vectors of the flyable path using sensors. Refer figure (5.3). For a path  $r(t)$  the distance to the threat region can be measured along normal vectors,  $N$ . For any point  $q$  on the boundary of the restricted region, the shortest distance between  $r(t)$  and  $q$  can be calculated by solving the equation (5.1.4).

$$\left( r(t) - q \right) \cdot \dot{r}(t) = 0 \quad (5.1.4)$$

where  $\dot{r}(t)$  is tangent. This method is computationally intensive for the paths of higher order polynomial like PH path which is of fifth order. Hence numerical approach is used to measure the distance. Once the threat is detected, the replanning

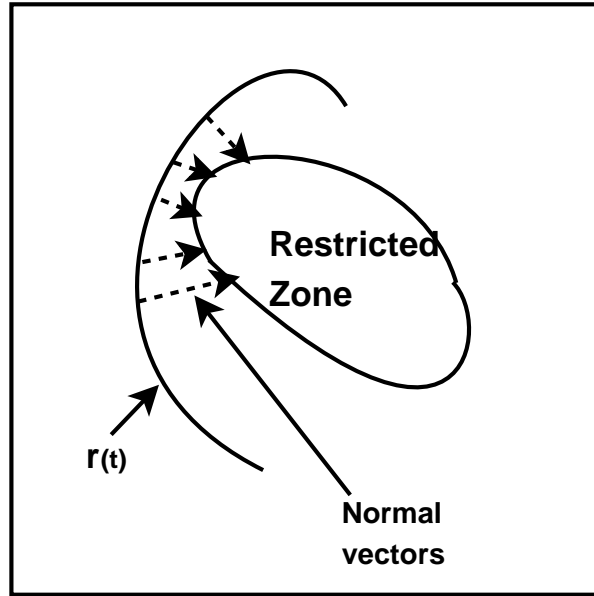


**Figure 5.2:** The threats are identified by intersection of path with threat boundary. This is applicable for the known threat region, because for a known region it is simple enough to know whether the path touches the boundary. If the intersection is non-empty, either the curvature of the path will be varied to avoid the collision or the flight-path will be replanned either by varying the curvature between two way-points or by creating an intermediate way-point, through which the path will be replanned

is carried out to avoid or destroy the threat. The threats are avoided either by: (i) increasing the curvature of the path between two way-points or (ii) creating an intermediate way-point and connecting the final way-point through the intermediate way-point.

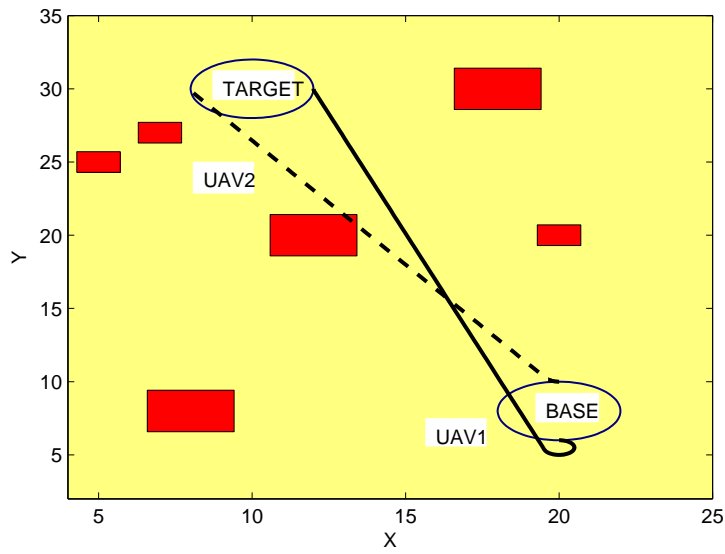
## 5.2 Replanning the path

Once the threat region is detected it is necessary to replan the path either to avoid the threat or to destroy the threat. Two approaches are used to replan the path: (i) Increasing the curvature of the path, (ii) Creating an intermediate way-point. The first approach is based on the fact that the curvatures determine the path. In the case of 2D Dubins and Clothoid composite paths, the start and finish radii are increased to meet the safety conditions (refer equation 2.1.6c). A safe 2D PH path is produced by increasing the boundary curvatures (refer equations (2.4.16a & 3.7.19a) and (2.4.16b & 3.7.19b)). The offset PH path is used to represent the safety boundary on either side of the flight path. While the 3D paths are tested for intersection with the safety tubes. The intersection points are calculated numerically.

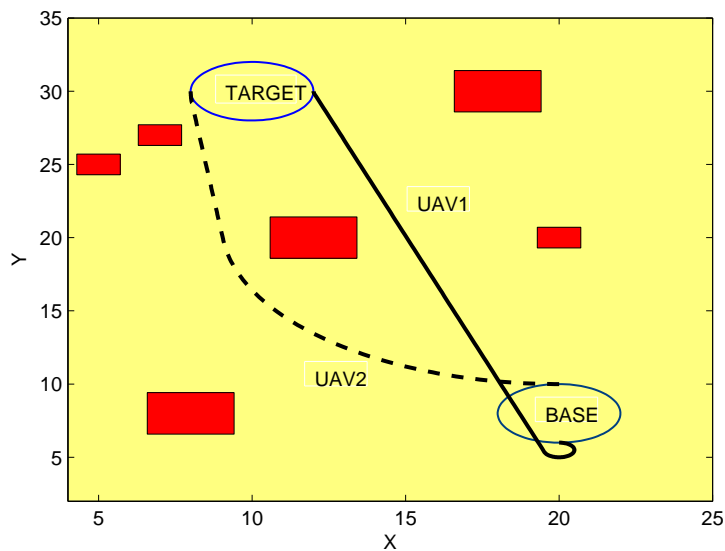


**Figure 5.3:** Calculation of safety margin using normal vectors. The distance between a point on the threat region and the flyable path is measure using the equation (5.1.4)

Figure (5.2) shows the paths of two UAVs in a cluttered environment. The flights path are intersecting at a point. Also the path of UAV2 is identified that it intersects with the threat region. This is identified by using the equation (5.1.3). The boundary curvatures are varied till the flight path avoids the threat region. Figure (5.2) shows the new safe, and flyable path after increasing the curvatures. The increase of curvature till the path avoids the threat is applicable for space not densely populated with threats. In the case of space of densely populated threats and pop-up threats it is necessary to create an intermediate way-point to take an evasive action. A safe flyable path is created by replanning the flyable path through the intermediate way-point. An intermediate point is created by simply calculating the area of triangle it forms with the UAV's current location and next way-point or target. In the case of constant altitude flights (considered in this thesis), intermediate new way-point can either be on the left or on the right side of the flyable path. Thus, two triangles can be formed with these points. The intermediate point with minimum are of triangle is chosen for replanning. The distance of the intermediate way-point is selected such that it must be at least at a distance of radius of safety-circle from the maximum edge of the boundary of the threat region. In the case of unknown threat it will be decided based on the sensor output. In this thesis, path planning with known threats are studied. Figure (5.2) shows the details, where the UAV2 takes an evasive action by creating an intermediate way-point, through which the flyable path is replanned.

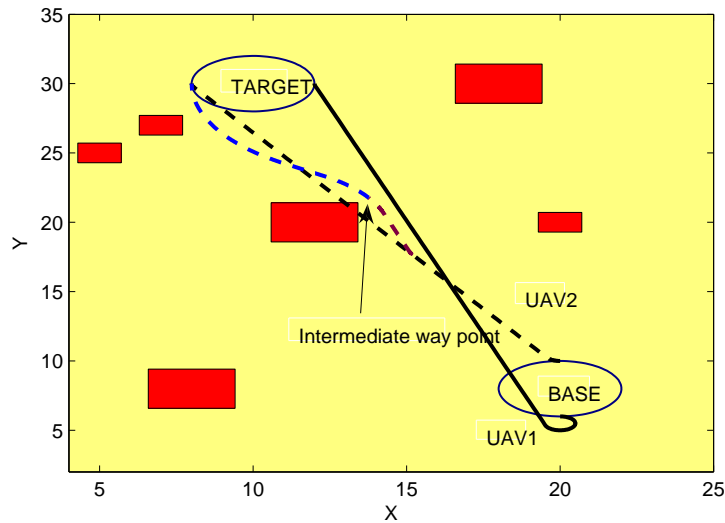


**Figure 5.4:** Two UAVs are flying from a base to a target in a cluttered environment. Two flight-paths intersect with each other and the UAV2 is passing through the threat or restricted region. The collision with the threat is detected by identifying the intersection of path with the threat region



**Figure 5.5:** The collision or threat avoidance is achieved by varying the curvature of the path. Here the turning radii of the Dubins path is varied to avoid the threats

## 5. PATH PLANNING IN CLUTTERED ENVIRONMENT



*Figure 5.6: The collision or threat avoidance is achieved by creating an intermediate way-point through which the path is replanned to avoid the threats*

### 5.3 Algorithm - Cluttered space

The previous chapter gives the algorithm for the simultaneous arrival in free space by increasing the radius of curvature. However, this approach can be used for cluttered environment, creating an intermediate way-point would produce an effective solution. Other methods like replanning from the point of failure of safety conditions and choose the next shortest path from the set of paths can also be implemented in the same way.

- (i) Produce flyable paths for each UAV.
- (ii) Change the course of the path to meet the safety constraints by creating intermediate way-points.
- (iii) Calculate the length of the paths.
- (iv) Find the reference path.
- (v) Increase the length of the shorter paths to the length of the reference path. This results in paths of equal length.
- (vi) Check again for the paths meeting the safety constraints.
- (vii) If not adjust the position of new way-point and increase the curvature to meet the safety conditions to produce the paths of equal lengths.

### 5.4 Summary

This chapter discusses the path planning in an environment of threats. Two methods are explained: detection by intersection and by measuring the distance. The threats are avoided by replanning the flyable path. The replanning is done either by increasing the curvature of the path or by creating an intermediate way-point. Both these approaches are illustrated with simulation results.

---

# Simulations and Results - 2D

**I**N this chapter, the simulations are conducted for constant speed UAVs flying at constant altitude. Therefore, only two dimensional maneuvers are considered. The proposed approach in solving simultaneous arrival problem is simulated using the Dubins, Clothoid and the PH paths. The initial and final poses are chosen randomly. The path planning using Dubins and Clothoid paths are simulated in free space while the PH paths are simulated both in free space and cluttered space. The path planning with the Dubins path in cluttered space is discussed in the previous chapter.

## 6.1 Simulations and Results - Dubins

Five UAVs are considered for simulation. The initial and final configurations are chosen randomly. The minimum turning radius is chosen as 1.2 units. The radius of safety circle is chosen as 2.5 units.

The shortest path of each UAV is calculated from the set of eight CLC paths. Thus, a set of five shortest paths formed for five UAVs. Figure (6.1) shows the shortest paths of the UAVs. The UAVs 1 to 5 follow the paths:  $\{LSR, LSL, RSR, LSL, RSL\}$  respectively. All arcs are of minimum turning radius. The lengths of paths are different from each other. The reference path is found out by using equation (4.3.2). The longest path from the set of shortest paths is the reference path. The path of UAV5 is the reference path as this is the longest in the set.



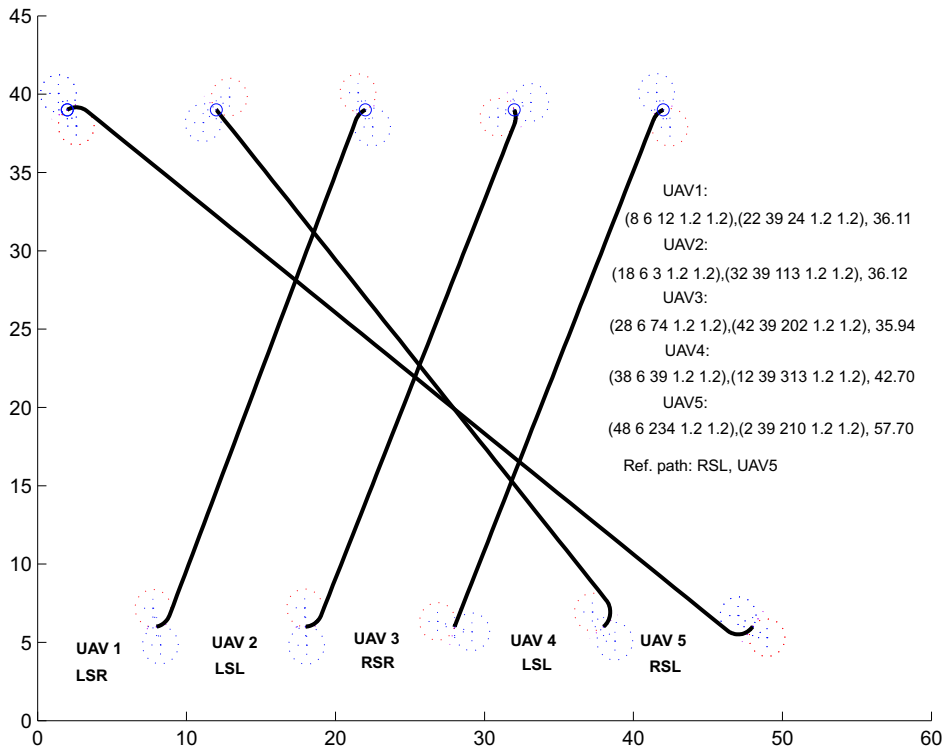
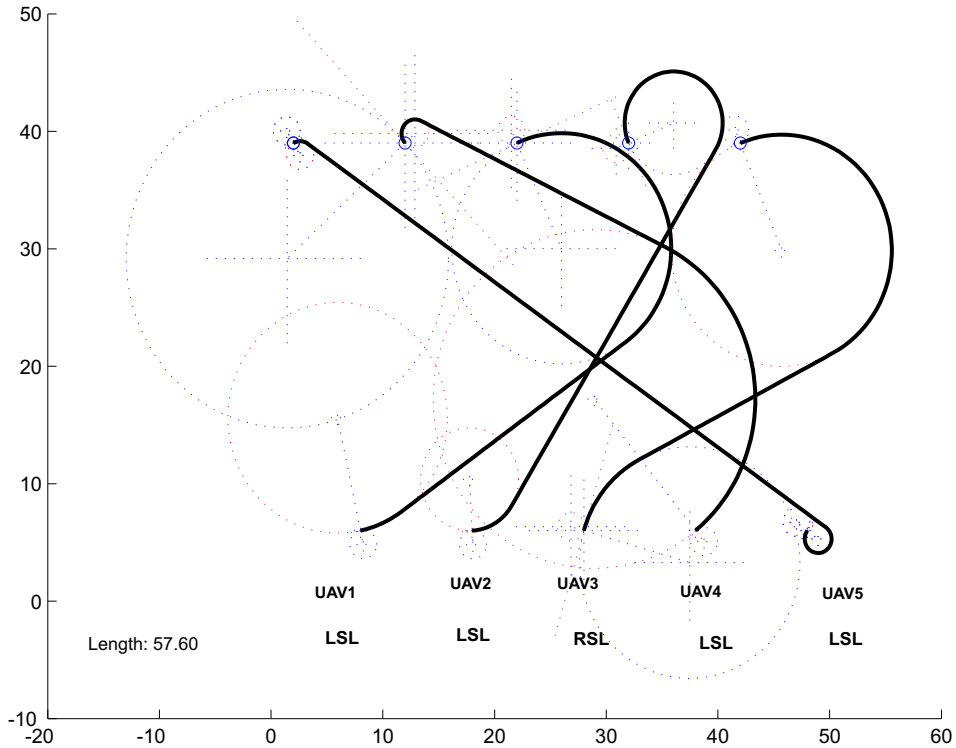


Figure 6.1: Shortest flyable paths of UAVs - Dubins 2D

The bisection method is used to calculate the optimal radius of curvature of the shorter paths. This is because the solution to the equation of path-length, (equation 2.4.1) or (equation 2.1.23) is not unique and also the solution may have complex roots. The optimal radii of paths of UAV1, UAV2, UAV3 and UAV4 are {9.82, 4.38, 9.86, 14.42} units respectively. Figure (6.1) shows the paths of equal length. From the figure, it can be observed that the route of paths are not the same as that of shortest CLC paths of UAVs. The routes are: {LSL LSL RSL LSL}. The route of UAV1 and UAV3 are changed from {LSR} to {LSL} and {RSR} to {RSL} respectively. This is because, the original routes designed with minimum turning radii did not meet the condition of existence of paths (equation 2.1.5a) and (equation 2.1.5b) with modified radius of curvature. So, the next shortest path from the set of CLC paths was selected for finding optimal radius to produce the path of length equal to the reference path. For any two paths, the safe flight-path is ensured as follows: By equation (4.2.2), the minimum separation of path should be greater than two times the radius of safety circle. The minimum separation distance is verified by calculating the Euclidean distance between the paths. From these values, the maximum and minimum separation is found out. The paths are safe to fly if they meet the condition. If the condition is not met at any point, the next condition (equation 4.2.6) is verified. i.e., non-intersection

## 6. SIMULATIONS AND RESULTS - 2D



**Figure 6.2:** Paths of equal length -Dubins 2D

of paths at equal distance. The length of each path to that point is calculated. If the difference between the lengths is greater than twice the radius of safety circle, the path is safe to fly. Otherwise, the path route must be changed with next shortest path from the set of CLC paths.

Figure (6.1) shows the coupling of paths for UAV1 with other UAVs. The first figure(top left corner) shows two intersection of paths of UAV1 with UAV2. The minimum distance between the paths is 5.1 units. This is just meeting the condition of minimum separation. As the path increment is uniform with constant speed of UAVs, the UAV1 and UAV2 are safe to fly in the paths: {LSL} and{LSL} respectively. The remaining paths UAV1 and UAV3, UAV1 and UAV4, UAV1 and UAV5 are providing safe flight paths as they are meeting the minimum separation distance at all points along the paths. Figure (6.1) shows the coupling of paths of UAV2 with UAV3, UAV4 and UAV5 and UAV3 with UAV4 (along row). The first two paths are meeting the minimum separation condition. Hence, they are safe paths. The path of UAV2 traverse the path of UAV5. Hence the UAV2 and UAV5 are not meeting the minimum separation. The minimum distance is 1 unit, four units less than minimum distance. The next condition, non-intersection at equal length is verified for these paths. The

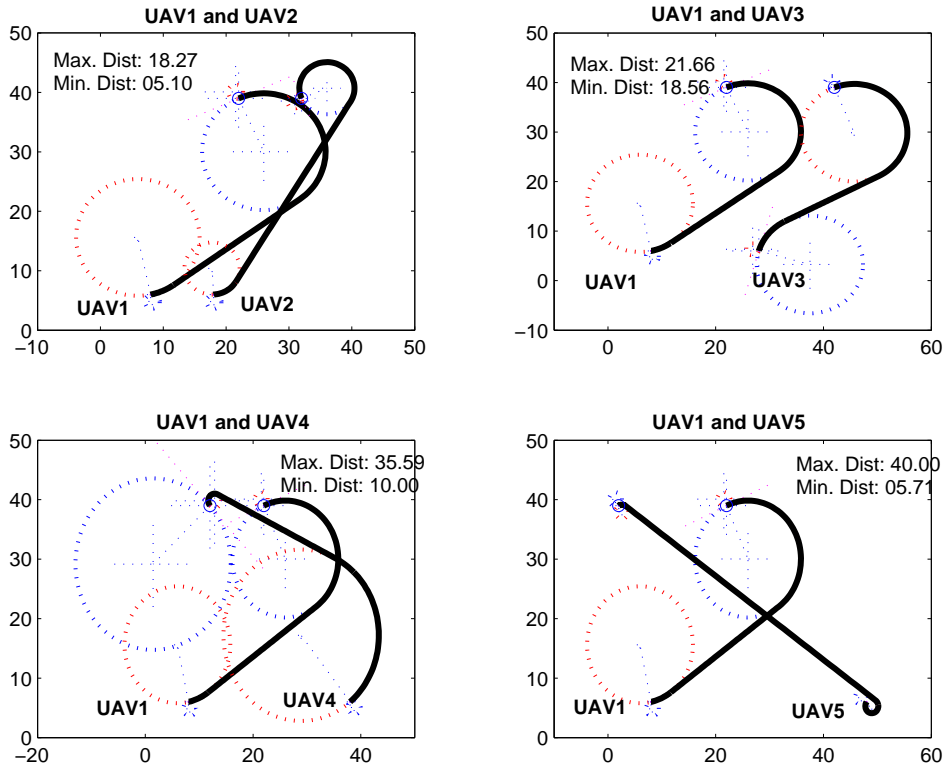


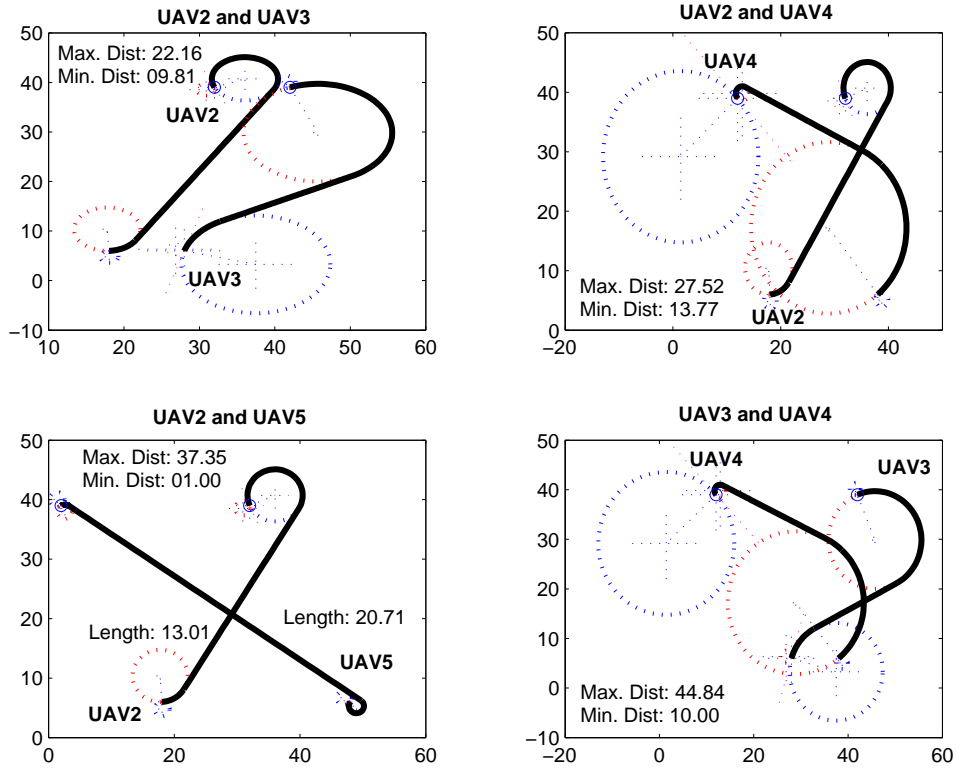
Figure 6.3: Separation distance for paths of first four combinations - Dubins 2D

length of path of UAV2 to the point of minimum distance is 13.1 units and that of UAV5 is 20.71 units. The difference between the lengths is greater than two times the radius of safety circle. This meets the condition for non-collision of UAVs. Hence the paths are safe to fly. The coupling paths of UAV3 with UAV4 well separated at corresponding points on the paths. The minimum distance between them is 10 units. Hence providing the safe flight-path. The figure (6.1) shows the coupling paths of UAV3 with UAV5 and UAV4 with UAV5. The paths are well separated meeting the minimum separation distance along the path. The paths are providing safe flight-paths to UAV4 and UAV5. Thus paths are flyable, safe and of equal in length, thus provides the simultaneous arrival to the UAVs.

## 6.2 Simulations and Results - Clothoid

The proposed solution to the path planning is simulated with a group of three UAVs, flying at a constant speed and at constant altitude. The UAVs are named as UAV1, UAV2, and UAV3. The initial and final configurations  $P_s, P_f$ , respectively, of the UAVs are pre-defined. All the UAVs are leaving the base at the same time. The maximum curvatures,  $\kappa_{max}$  of the UAVs are taken as  $\pm\frac{1}{4}$ . Hence the path shall have

## 6. SIMULATIONS AND RESULTS - 2D



**Figure 6.4:** Separation distance for paths of second four combinations - Dubins 2D

to have the curvature not exceeding this value. Figure (6.2) shows the initial paths generated with the maximum curvature bound of the UAVs. These are the flyable paths. The paths are verified for safe flights using the condition (4.2.2) for minimum separation distance. The paths failed at two intersecting points as shown in the figure. Hence the second condition (4.2.6) is used to verify the non-intersection of paths at equal length. In this particular case, the paths meet the safety constraints. Now, the length of the shorter paths are to be increased to that of the reference path using the equation (4.3.2). The path length of UAV1 longer than that of UAV2 and UAV3. Hence path of UAV1 is the reference path. The path-lengths of UAV2 and UAV3 are increased to that of UAV1 by decreasing the curvature of their Clothoid segments.

Figure (6.2) shows the paths of individual UAVs together. The safety conditions are tested for the paths using the equations (4.2.2) and (4.2.6). The point of intersections are found by the iterative search. The difference in the lengths of paths of UAV2 and UAV3 from their initial points differ by more than the two times the radius of safety circle. Thus generated the flyable and safe flight paths of equal lengths.

## 6.2 Simulations and Results - Clothoid

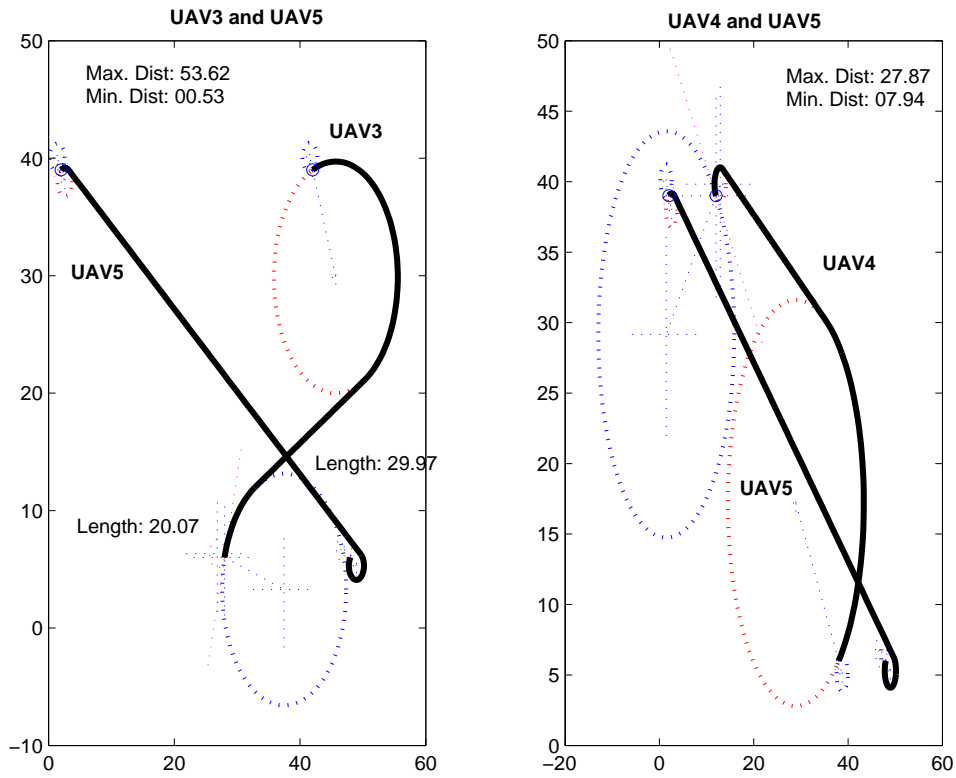


Figure 6.5: Separation distance for paths of last two combinations - Dubins 2D

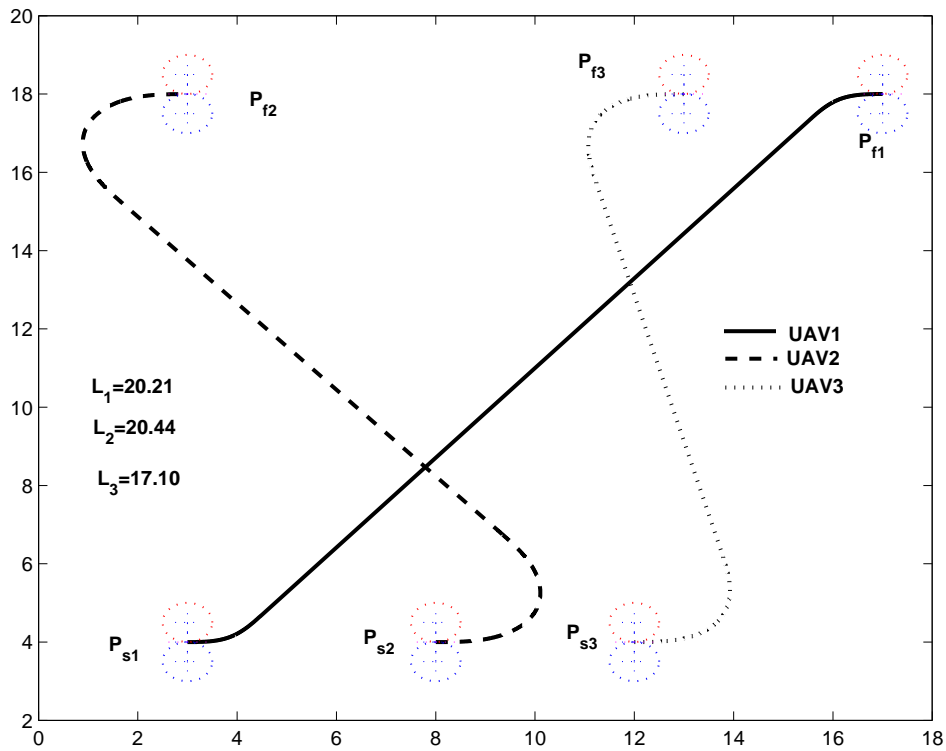


Figure 6.6: Initial paths of UAVs. All UAVs have different path lengths and the paths are intersecting with one another

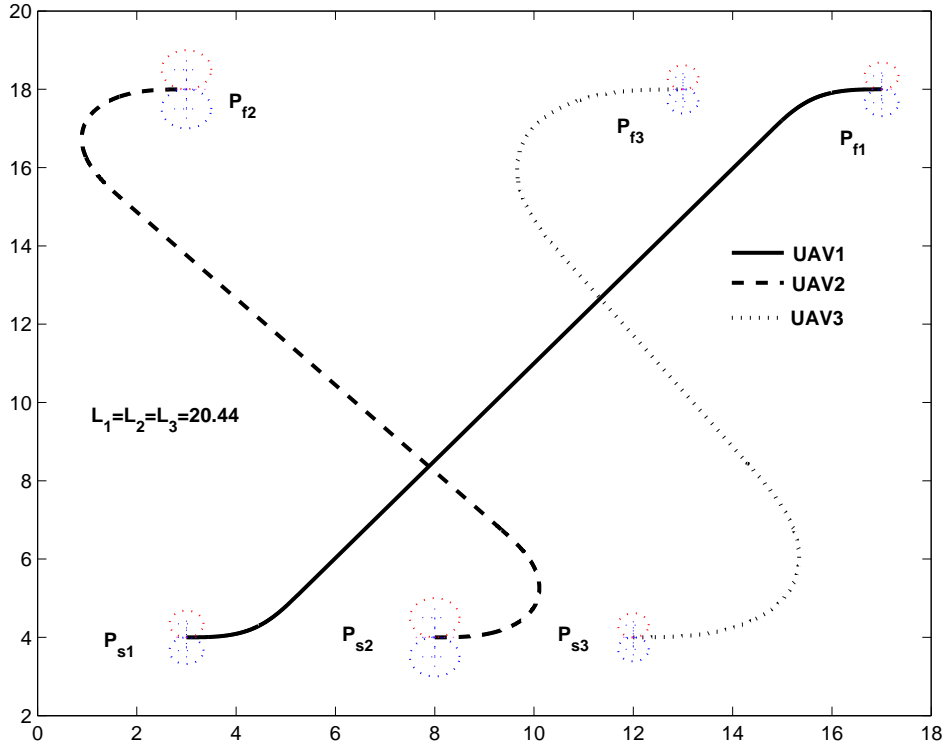


Figure 6.7: Paths of equal lengths - Clothoid 2D

### 6.3 Simulations and Results - PH

Three UAVs are considered in this case. The UAVs are named as UAV1, UAV2 and UAV3 and are flying at constant speed and at constant altitude. The initial and final configurations of the UAVs are well-defined. All the UAVs are leaving the base at the same time. The figures (6.3), (6.3) and (6.3) show the paths of individual UAVs. The central path (solid line) shows the flight path. The dashed path on either side of the flight-path shows offset paths with circular rings. The offset paths are generated at a distance of radius of the safety circle. Hence the circular rings have the diameter of the safety circle. The important points to be considered from the figures are: (a) The paths have curvature continuity, thus providing smoothness. (b) Each path has different route or trace. The maximum curvature of the UAV is  $\kappa_{max}$  is taken as  $\pm\frac{1}{3}$ . Hence the path shall have to have the curvature not exceeding this value. The maximum and minimum curvatures of the paths are  $(0.1142, -0.3000)$ ,  $(0.3263, 0.002)$  and  $(0.2718, 0.0055)$  respectively. The path of UAV2 is the reference path. The length of UAV1 and UAV3 are increased to that of UAV2 by the procedure using the equation (2.4.16). The figure (6.3) shows the paths of individual UAVs together. The start and finish points are shown with the tangent circles, which define the maximum curvature of the UAVs. The safety conditions are tested for the three paths using the

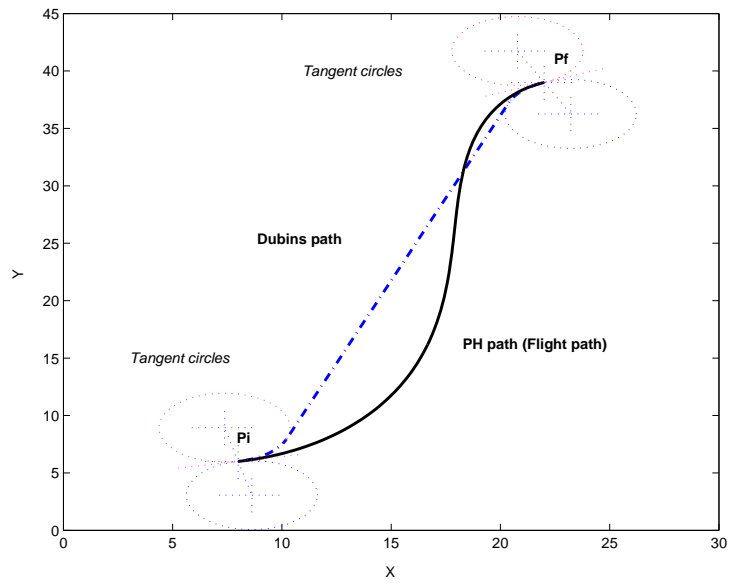


Figure 6.8: Dubins and PH Paths of UAV1

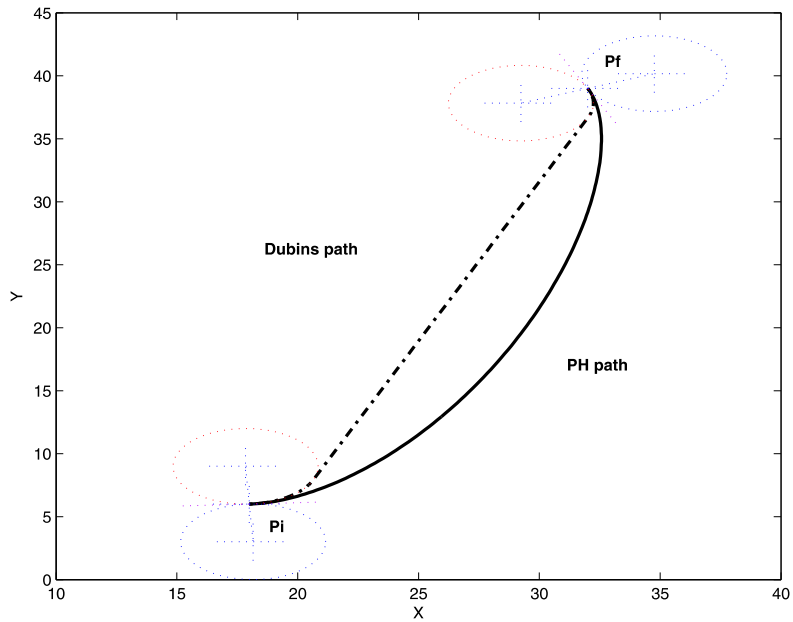
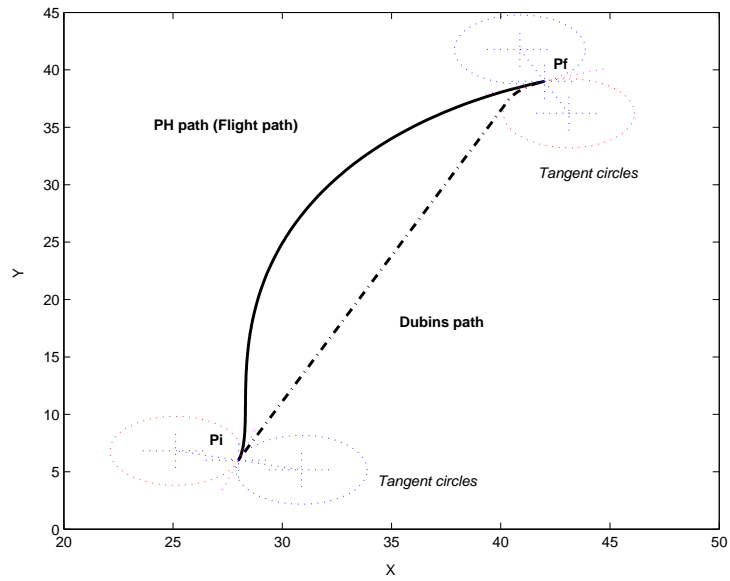
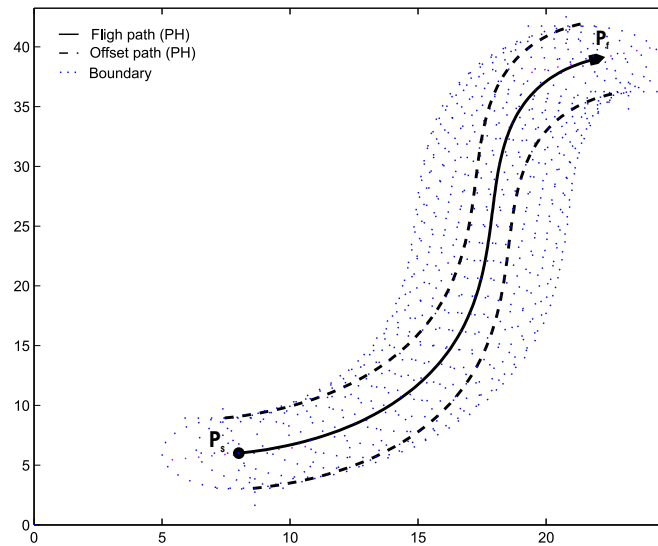


Figure 6.9: Dubins and PH Paths of UAV2

## 6. SIMULATIONS AND RESULTS - 2D

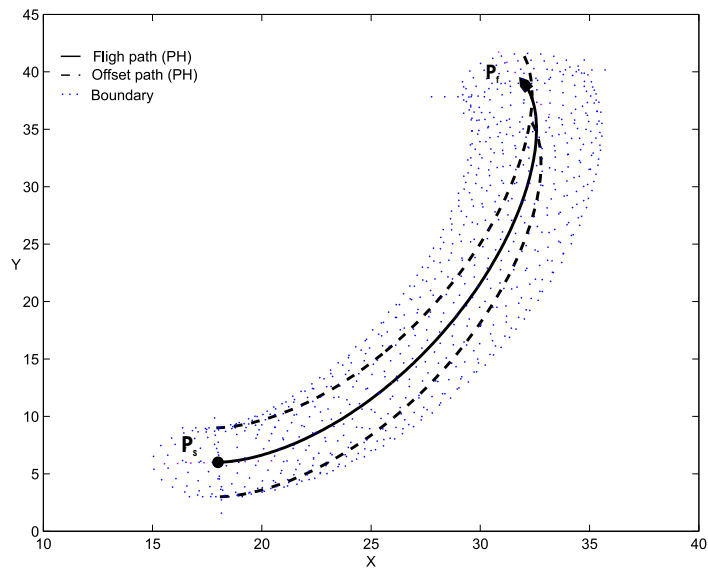


**Figure 6.10:** Dubins and PH Paths of UAV3

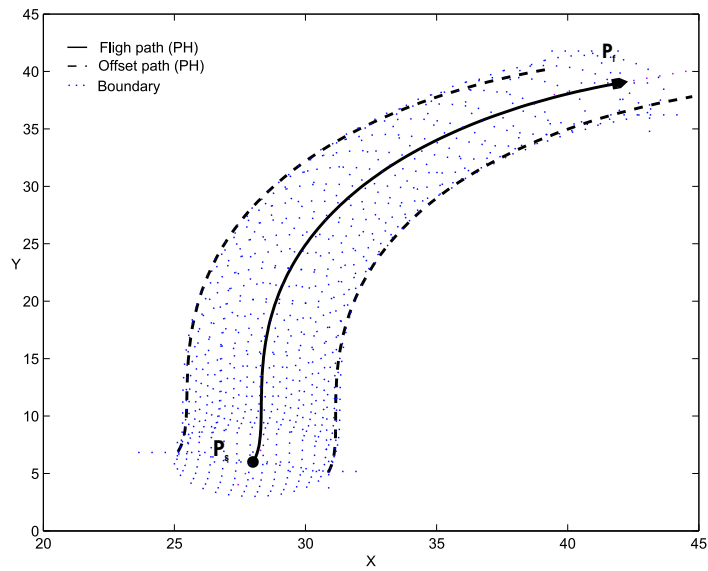


**Figure 6.11:** Flyable PH with offset paths of UAV1





**Figure 6.12:** Flyable PH with offset paths of UAV2

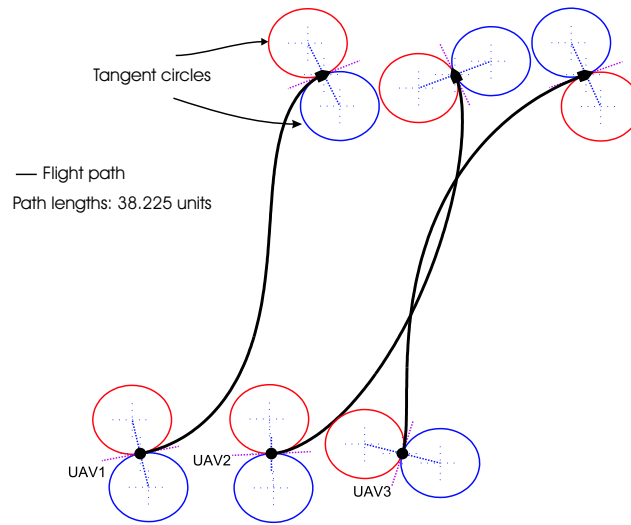


**Figure 6.13:** Flyable PH with offset paths of UAV3

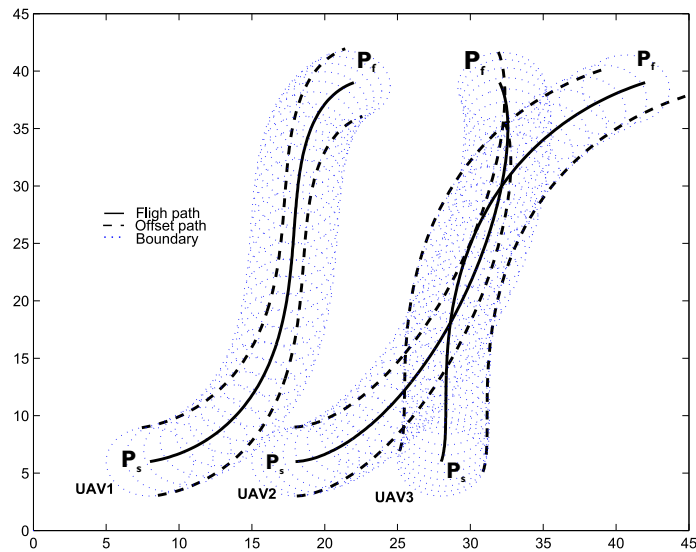
## 6. SIMULATIONS AND RESULTS - 2D

equation (4.2.8). Taking any two UAVs at time, the total number of times the safe flight paths are to be tested are six.

The figures (6.3) and (6.3) are the paths of all UAVs with their offset paths. The UAV1 is not intersecting with other paths. The paths of UAV2 and UAV3 are intersecting at two points. The point of intersections of paths of UAV2 and UAV3 are found by the iterative search. The difference in the lengths of paths of UAV2 and UAV3 from their initial points are 8.381 and 7.321 respectively. The values are greater than the diameter of the safety circle. This ensures the safe flight paths.



**Figure 6.14:** PH paths of equal lengths



**Figure 6.15:** PH Paths of UAVs, equal lengths with offset paths and safety rings

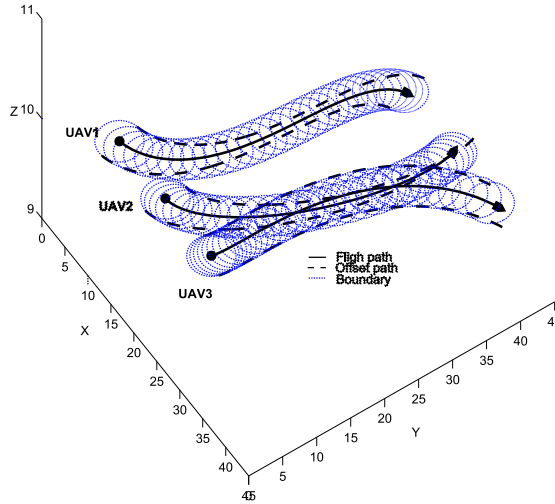
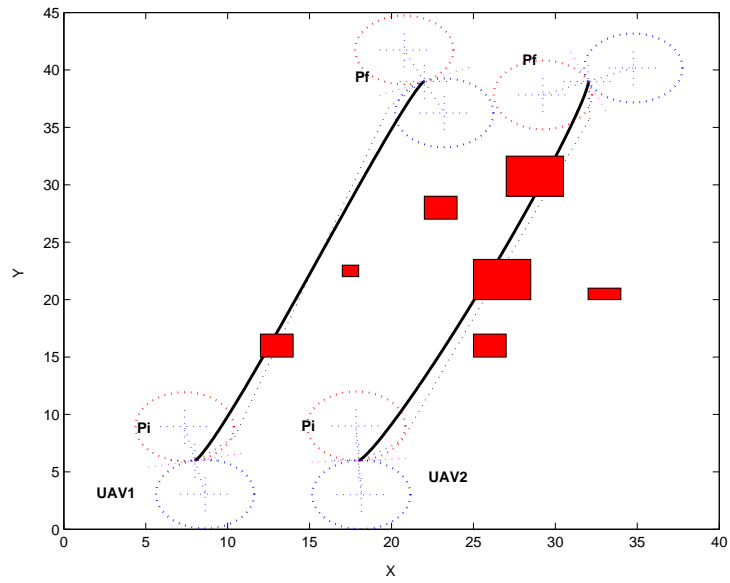


Figure 6.16: PH Paths of UAVs, equal lengths elevated at constant altitude

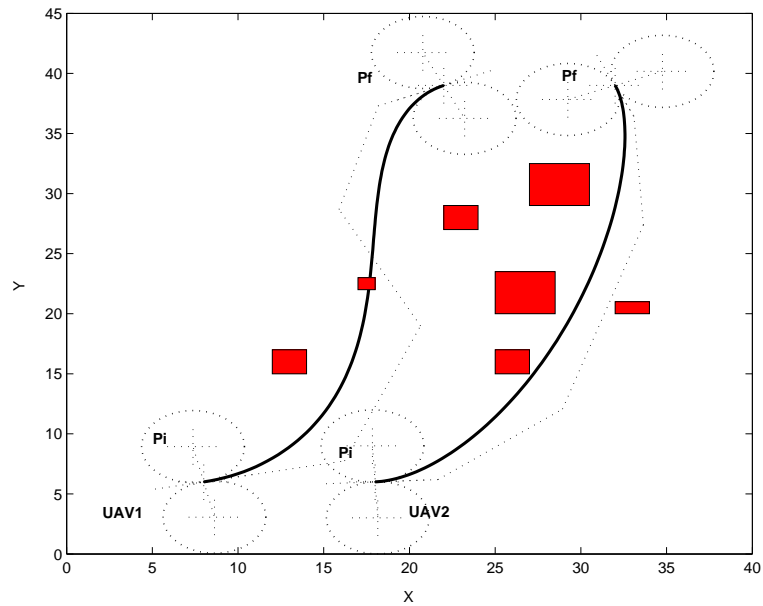
## 6.4 Path planning in cluttered space

Two UAVs are considered for simulation. The UAVs are assumed to be homogenous in their physical capabilities. The UAVs are flying at constant speeds at constant altitudes. Figure (6.17) shows the PH paths of the UAVs prior to curvature optimization.  $P_i$  and  $P_f$  are the initial and final configuration respectively. The patches observed on the figure are the restricted zones. It is seen that the UAVs are flying over the restricted zones. The crossing of the paths on the tangent circles (whose radius is taken as 3 units) shows that the paths are not meeting the constraint of the maximum curvature bound. Thus it is required that the paths are to be optimized for safe flight path following optimization for their curvature. The path lengths of UAV1 is: 35.96 units and that of UAV2 is 35.92 units. Figure (6.18) shows the UAV paths optimized for their curvatures. The tangent circles are not crossed by the paths. However, the paths do not satisfy the safe flight path with minimum safety margin greater than 3 units. The path of UAV1 is directly passing over the restricted zone and the path of UAV2 is not meeting the minimum safety margin. Both the paths need further change in their curvature and in turn their lengths. The path length are: UAV1 40.69 units and UAV2 37.13 units. Figure (6.19) shows that each UAV is provided with the safety margin. The offset curves (dashed lines) with a offset distance of  $\pm 3.01$  unit ensure the safety of UAVs. The path-length of UAV1 is 42.57 units and that of UAV2 is 41.12 units. The paths are not of equal lengths. The path-length of UAV1 is greater than that of UAV2. So, path of UAV1 is the reference path. The path length of UAV2

## 6. SIMULATIONS AND RESULTS - 2D



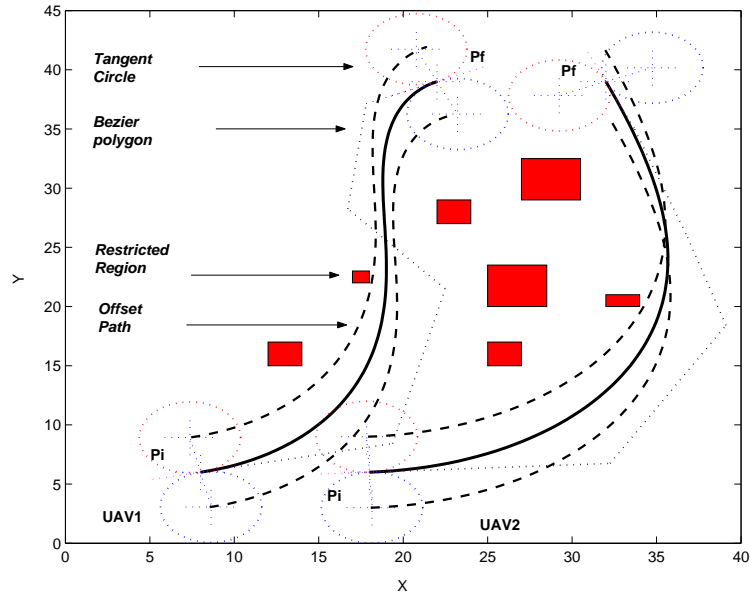
**Figure 6.17:** Initial paths (only tangent continuity) - PH 2D in cluttered space



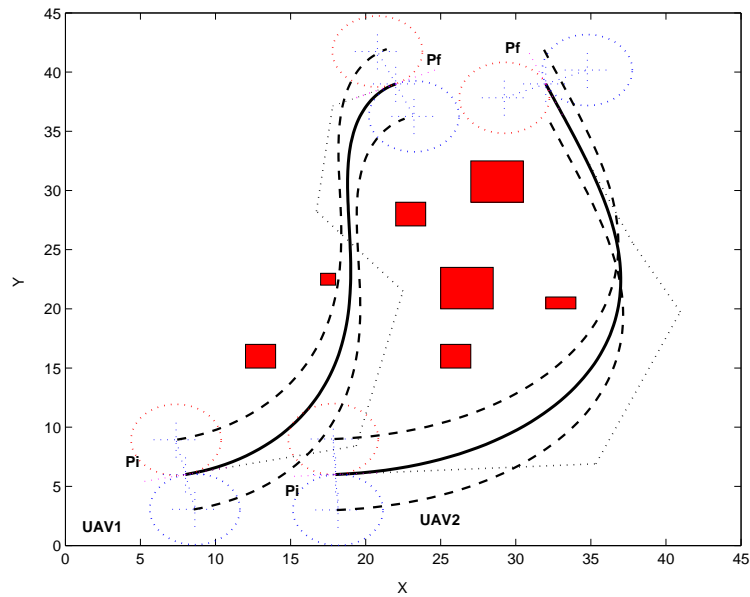
**Figure 6.18:** Flyable paths - PH 2D in cluttered space

## 6.4 Path planning in cluttered space

has to be increased to that of UAV1 for generating paths of equal length for simultaneous arrival. Figure (6.20) shows the paths of UAV1 and UAV2 having equal path length of 42.57 units. Thus, achieving the mission objective of simultaneous arrival to the target in an environment with restricted zones.



**Figure 6.19:** Feasible (safe and flyable) paths - PH 2D in cluttered space



**Figure 6.20:** Paths of equal lengths - PH 2D in cluttered space

### 6.5 Summary

This chapter concludes with the simulation results for two dimensional path planning in a free space. Simultaneous arrival on target is achieved by producing the path of equal lengths for all UAVs. Three types of path: Dubins, Clothoid - composite paths and the PH path - single path are used. The safety conditions, minimum separation distance and non-intersection at equal lengths are tested for all paths.

---

## Simulations and Results - 3D

**T**his chapter gives the simulation results of 3D path planning with the Dubins, and PH paths. The UAVs are assumed flying in free space and at constant speed and at constant altitude. The initial and final poses are chosen randomly. The safety constraints are satisfied by increasing the curvature of the paths. In 2D simulations, a safety circle of radius  $R_s$  is used for testing safety conditions. Here the safety circle becomes a safety sphere of radius  $R_s$ . All the UAVs are leaving the base at the same time.

### 7.1 Simulations and Results - Dubins

Three UAVs are considered for simulation. The minimum turning radius is chosen as 5. The initial and final poses are

$$\left( [0, 0, 0], [0, 10, 0] \ \& \ [51, 18, 51], [0, -10, 30] \right)$$

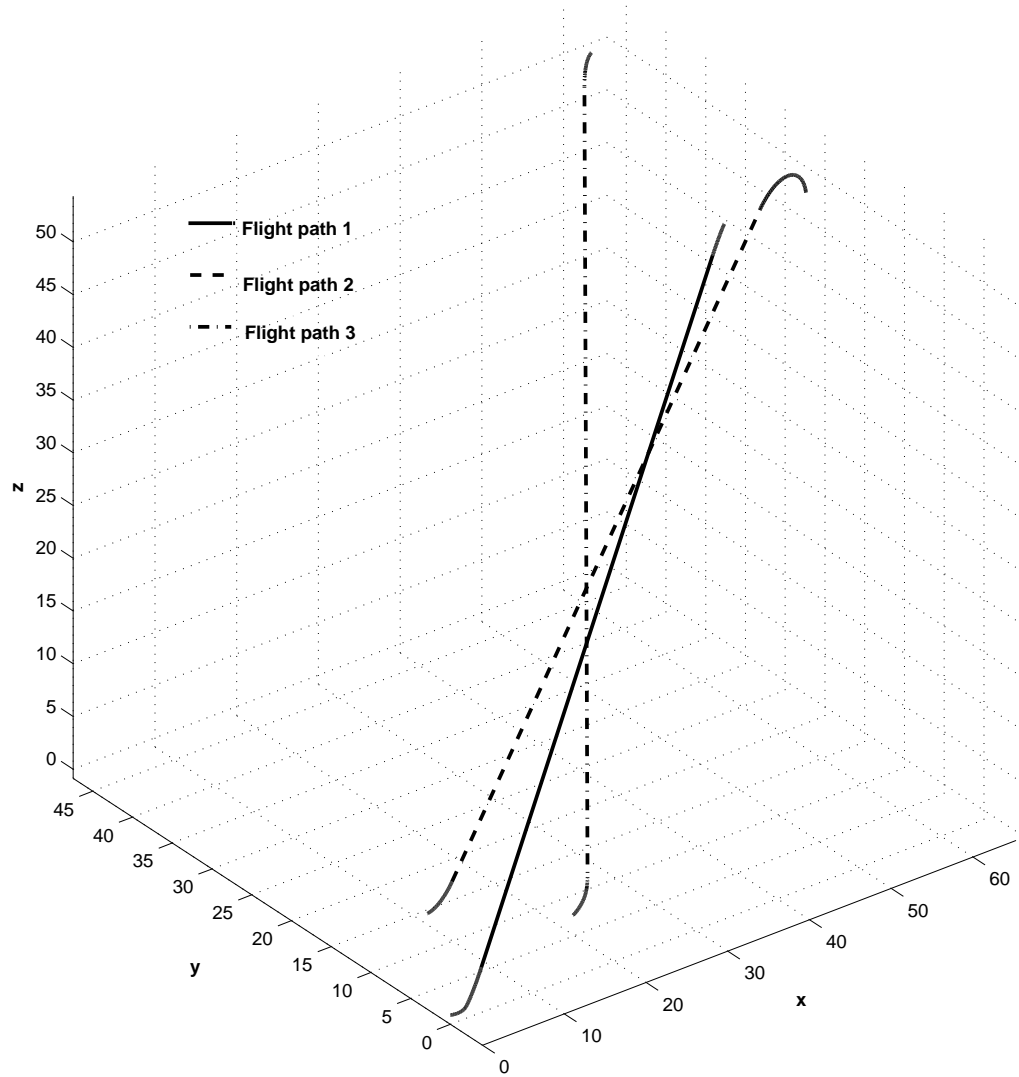
$$\left( [4, 7, 5], [0, -10, 0] \ \& \ [61, 18, 51], [0, 70, 30] \right)$$

$$\left( [15, 0, 5], [0, -28, 0] \ \& \ [61, 45, 51], [0, 10, 30] \right)$$

The radius of safety sphere  $R_s$  is 3. The minimum separation distance is 6 units. The flyable paths for each UAV is generated using the principle explained in the section 3.2. The length of the flight path of each UAV is calculated using the equation (3.5.1). Figure (7.1) shows the flyable paths of each UAV. The flight path 1, 2 and 3

## 7. SIMULATIONS AND RESULTS - 3D

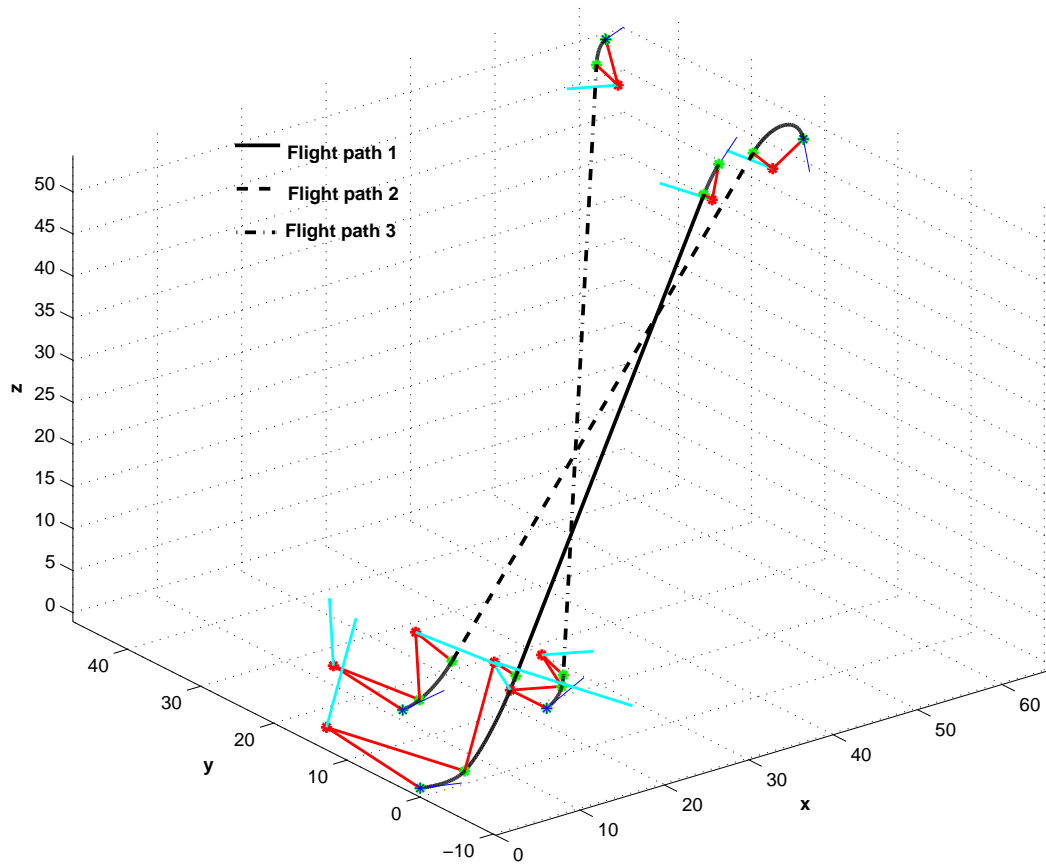
respectively corresponds to that of UAV1, UAV2 and UAV3. The length of each path is 76.27, 79.57, and 79.91 respectively. The length and trace of each path are different from one another. The path of UAV3 is the reference path found out by equation



**Figure 7.1:** Flyable Paths of UAVs - Dubins 3D

(4.3.2). Therefore, the path lengths of UAV1 and UAV2 have to be increased to that of UAV3. The turning radii of the UAV1 and UAV2 are increased to equalize their length with that of UAV3. As there is no direct relation exists between the path length and turning radius, iterative method is sought to find the optimal curvature of the paths. Also, the solution to the equation of path-length (3.5.1) is not unique and

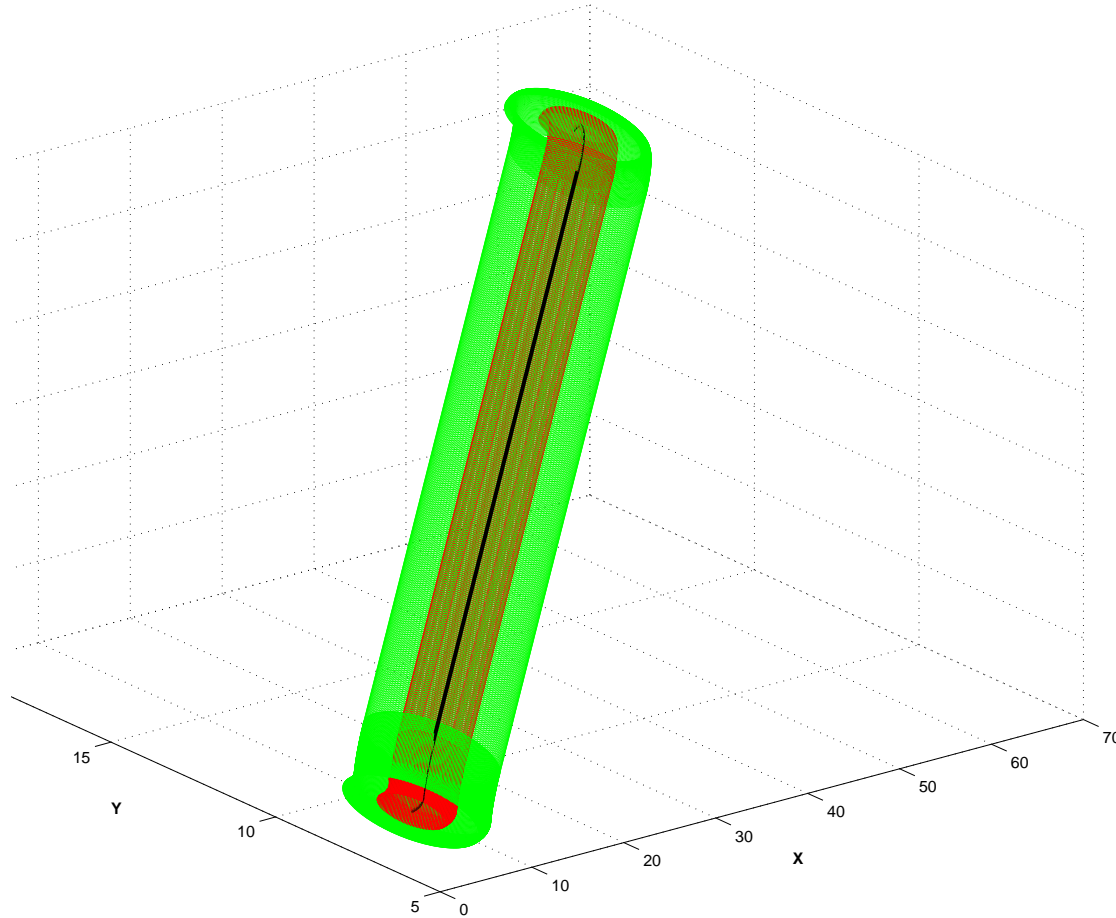




*Figure 7.2: Path of equal lengths - Dubins 3D*

## 7. SIMULATIONS AND RESULTS - 3D

also may result in complex roots. The new curvatures of UAV1 and UAV2 are 0.0759 and 0.1074 respectively. Figures (7.2) shows the paths of equal length. Figures (7.4), (7.6) and (7.5) show each two flight paths separately which are equal in length. Thus the flyable paths of equal lengths are produced. Now these paths are to be verified against the safety conditions. For any two paths, the safe flight-path is ensured as

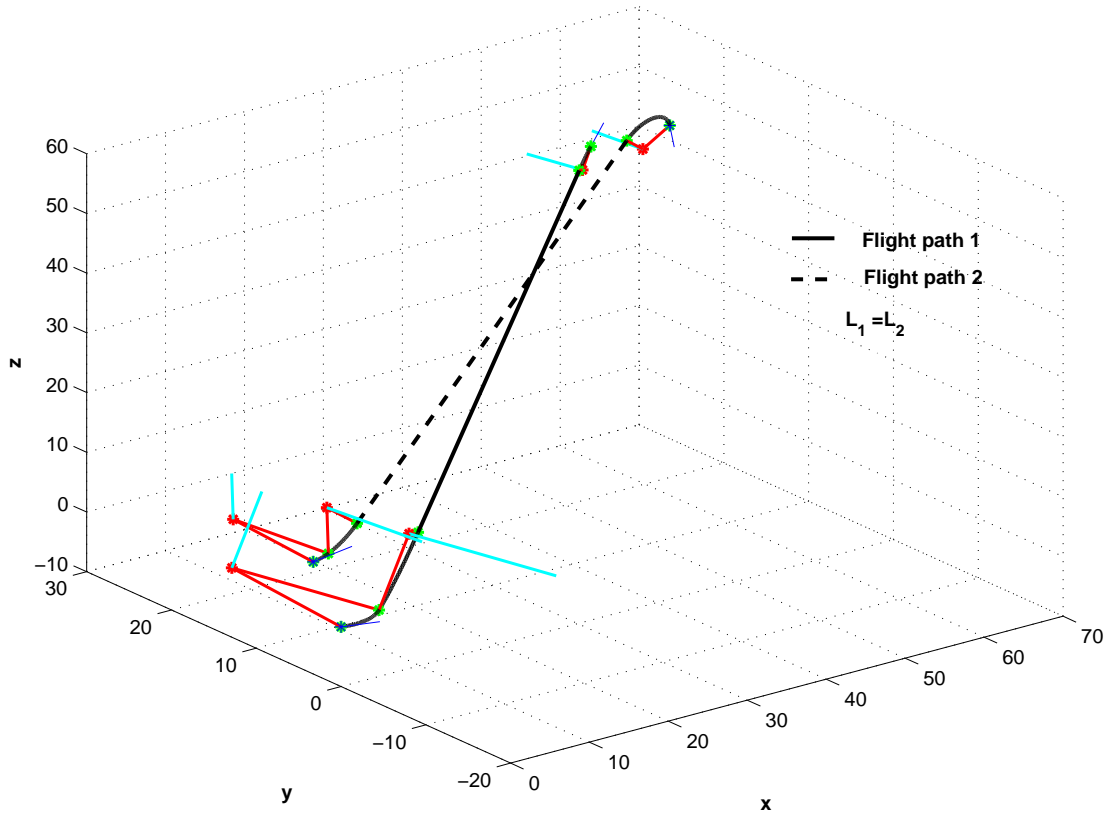


*Figure 7.3: Flight path with safety tubes - Dubins 3D*

follows: minimum separation of path should be greater than two times the radius of safety sphere (equation 4.2.2). The minimum separation distance is the Euclidean distance between the paths. This distance is measured numerically and validated against  $2R_s$ . The paths are safe to fly if they meet the condition. If the condition is not met at any point, the next condition, non-intersection of paths at equal length (equation 4.2.6) is to be validated. The length of each path to that point is calculated.

## 7.1 Simulations and Results - Dubins

If the difference between the lengths is greater than twice the radius of safety circle, the path is safe to fly. Otherwise, the path route must be changed either by further increasing the radius of turn or by replanning the path from the point of failure. Here the first method is adopted. The minimum separation distance between the paths

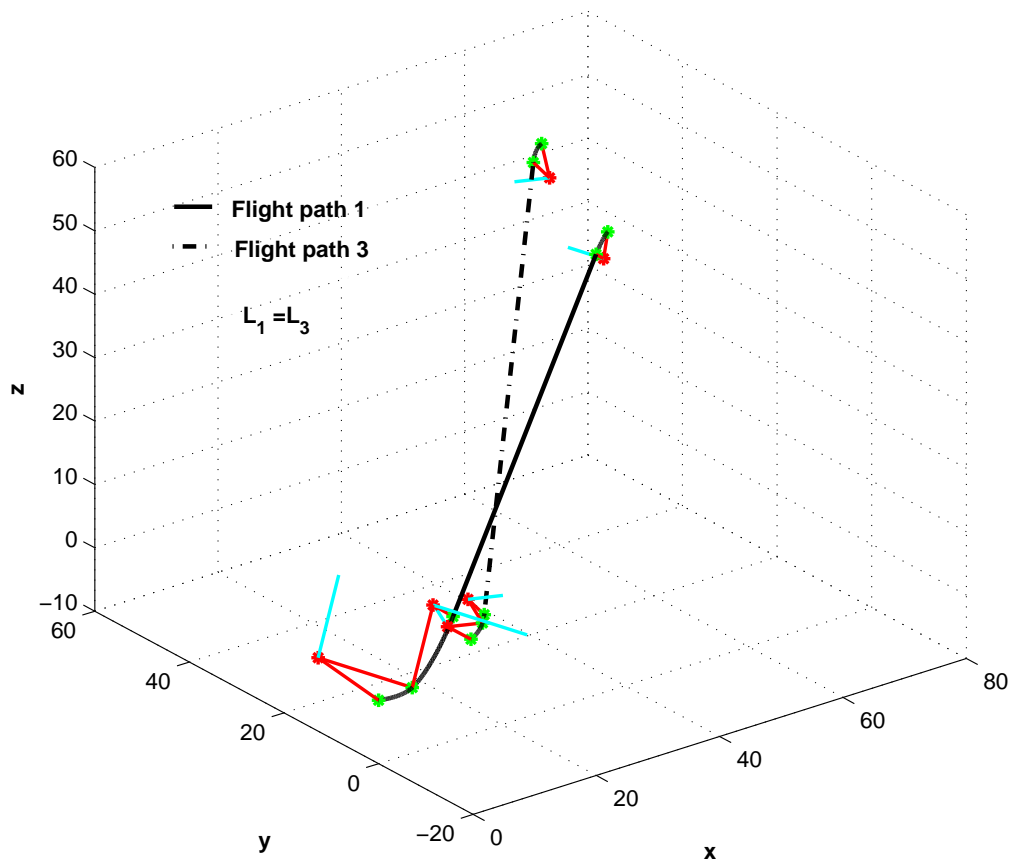


**Figure 7.4:** Paths of equal lengths - UAV1 & UAV2 - Dubins 3D

1 & 2, 2 & 3 and 1 & 3 respectively are: 5.47, 9.6, and 13.1. These values show that UAV1 fails to meet the minimum separation distance. Hence the flight path of UAV1 has to be tested for non-intersection at equal length. The path length of each path from the starting pose to the point of failure is calculated. The difference in path length of UAV1 & UAV2, UAV2 & UAV3 and UAV3 & UAV1 respectively from the initial pose to the point of intersection are 6.863, 6.541, and 15.48. The values are greater than the minimum separation distance ( $= 2R_s$ ). Hence the flight paths are safe to fly. The figures (7.4, 7.6 and 7.5) show the flight paths of sets of UAV1 & UAV2, UAV2 & UAV3 and UAV3 & UAV1 respectively. The intersection of paths can easily be visualized in the figure (7.7). All the paths with the safety tube around them is shown in the figure (7.3).

## 7. SIMULATIONS AND RESULTS - 3D

---



*Figure 7.5: Paths of equal lengths - UAV1 & UAV3 - Dubins 3D*

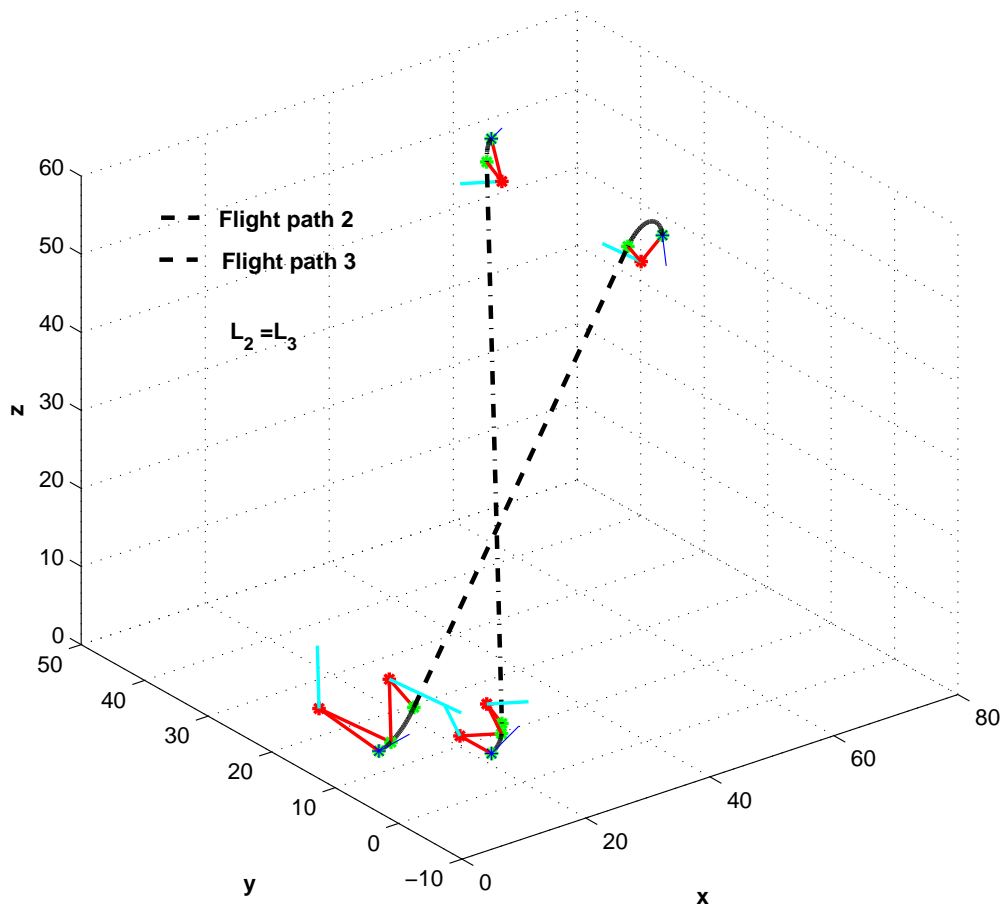
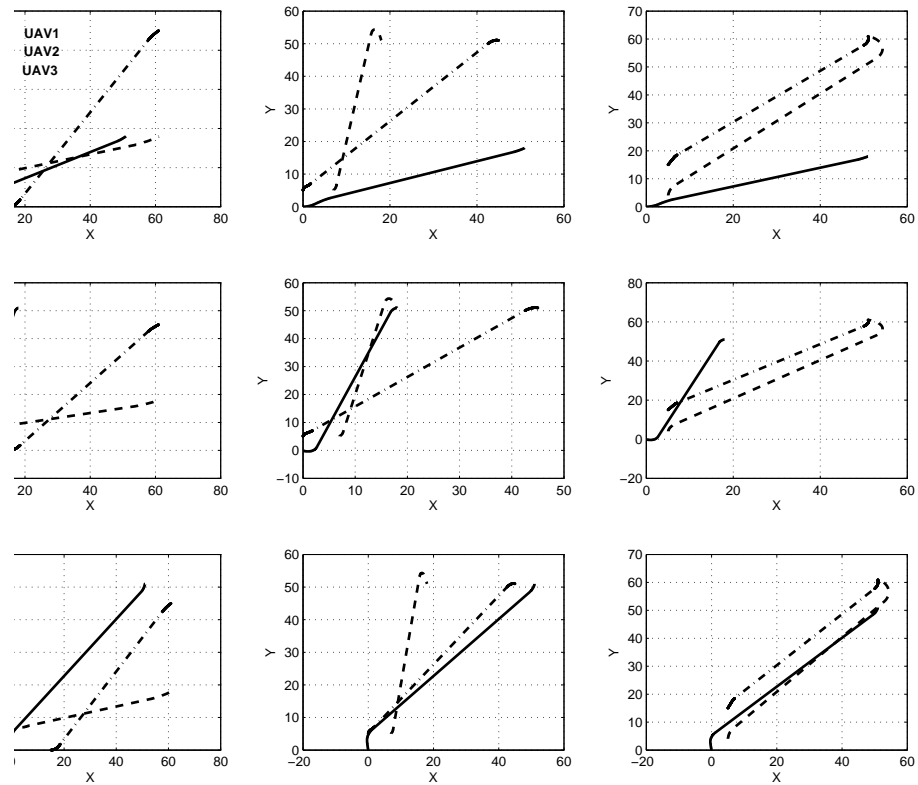


Figure 7.6: Paths of equal lengths - UAV2 & UAV3 - Dubins 3D

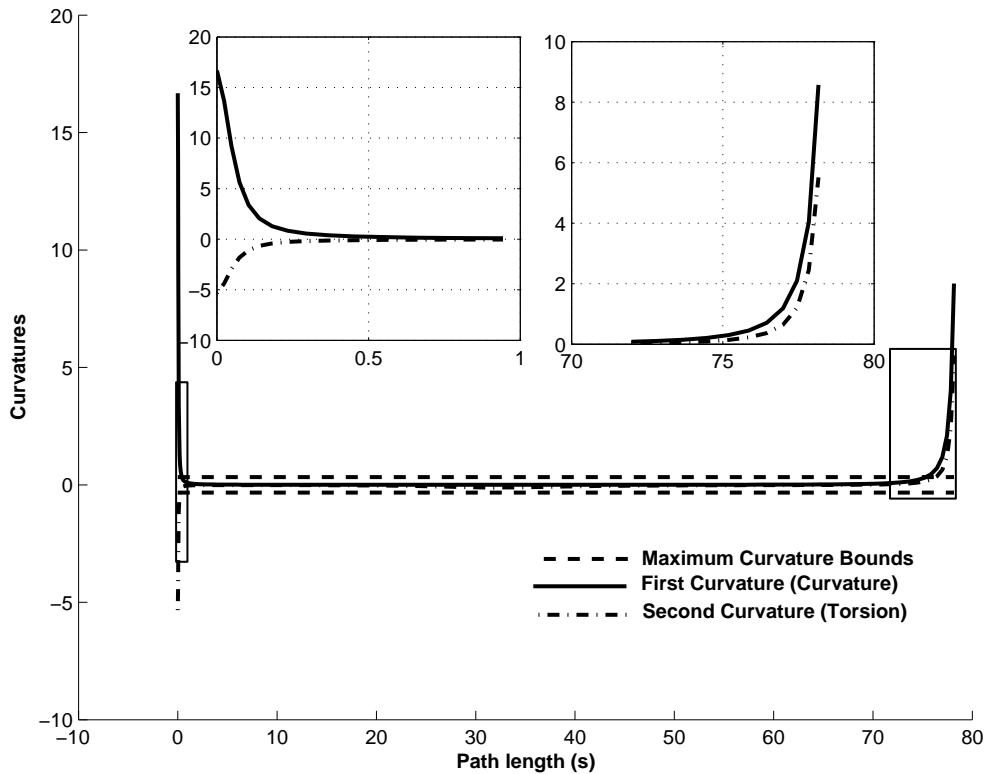
## 7. SIMULATIONS AND RESULTS - 3D



**Figure 7.7:** *Flight path intersections: The intersections are given in each plane for each UAV. The intersections are calculated numerically. This is to avoid the possibility of complex points during the intersection between lines and circles. The safety conditions: (i) Minimum separation distance and (ii) Non-intersection at equal lengths are tested individually on all planes*

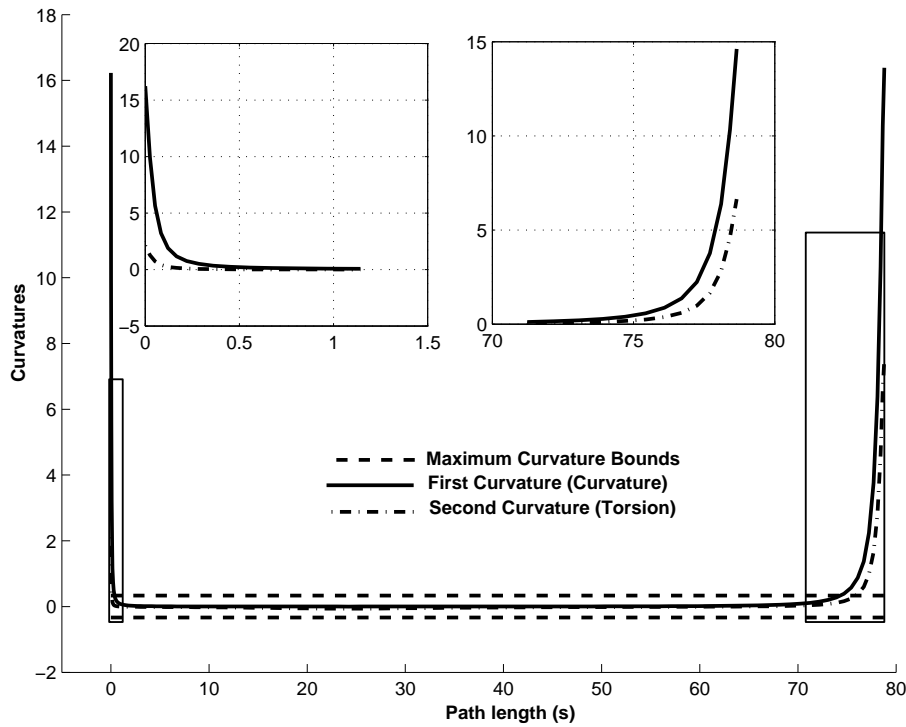
## 7.2 Simulations and Results - PH

Two UAVs are considered for the simulation. The UAVs are named as UAV1, and UAV2. The initial and final poses of the UAVs are pre-defined. The maximum curvatures,  $\kappa_{max}$  and  $\tau_{max}$  of the UAV are taken as  $\pm\frac{1}{3}$ . The output from the original PH solution provides only tangent continuous path (Refer section 3.6.1). The figures (7.2), and (7.2) show the curvature and torque variation of the paths initially generated using (3.6.6). These are tangent continuous paths, called as initial paths. The paths do not meet the maximum curvature bounds at the boundary points.



**Figure 7.8:** Curvature and Torsion variation with respect to path-length, Initial path -UAV1. Note down that the maximum curvature bounds do not meet at the boundary points

The paths are optimized for curvature bounds by increasing the positive constants in equations (3.7.2c) and (3.7.2d) till the conditions (3.7.15) and (3.7.18). The figures (7.2), and (7.2) show the paths which meet the curvatures bound. These paths are flyable by the UAVs. The central path (solid line) shows the flight path. The tube around the path designed by equation (4.2.4). The tubes have radius equal to that of safety-sphere. The important points to be considered from the figures are: (a) The paths have curvature continuity, thus providing smoothness. (b) Each path has different route or trace. The curvatures variations with path-length of flyable



**Figure 7.9:** Curvature and Torsion variation with respect to path-length, Initial path -UAV2. Note down that the maximum curvature bounds do not meet at the boundary points

paths are shown in figures (7.2) and (7.2). The path meets the maximum curvatures bound at all points on the path. The path length of UAV1 longer than that of UAV2. Hence path of UAV1 is the reference path. The length of UAV2 is increased to that of UAV1 by the procedure explained in section (3.7). The figure (7.2) shows the paths of individual UAVs together. The safety conditions are tested for the paths using the equations (4.2.2) and (4.2.6). The paths are shown with the tubes. The UAV1 is intersecting with that of UAV2 at two points. The point of intersections are found by the iterative search. The difference in the lengths of paths of UAV2 and UAV3 from their initial points differ by more than the two times the radius of safety sphere. This ensure the safe flight paths. Thus paths of equal lengths for a group of UAVs in achieving simultaneous arrival to a target is accomplished.

### 7.3 Summary

This chapter shows the simulation results of the 3D Dubins and PH paths. The clothoid paths are similar to that of the Dubins paths except that the circular arcs are replaced with clothoid arcs. The solution approach described in chapter 4 is applied to the path planning.



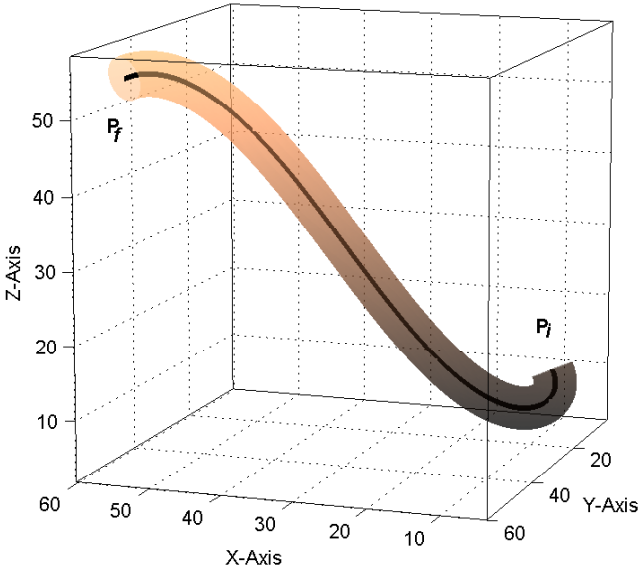


Figure 7.10: The path meets the maximum curvature bounds of the UAV

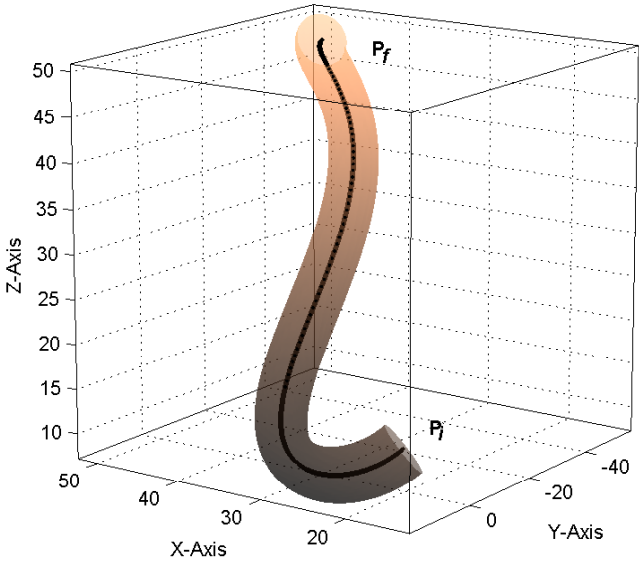


Figure 7.11: The path meets the maximum curvature bounds of the UAV

## 7. SIMULATIONS AND RESULTS - 3D

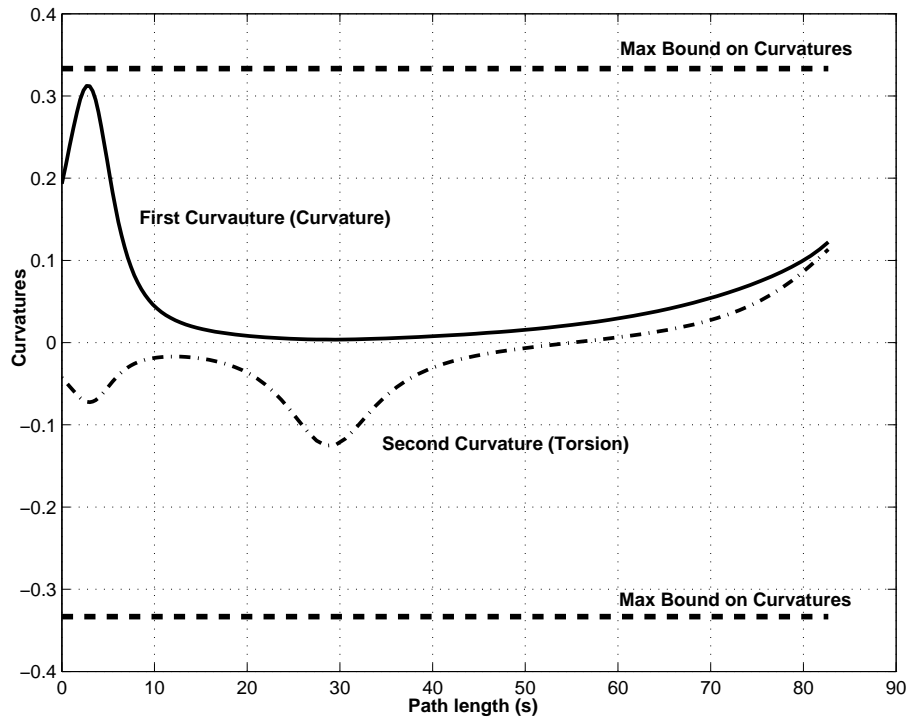


Figure 7.12: Curvatures variation: Flyable path of UAV1- PH 3D

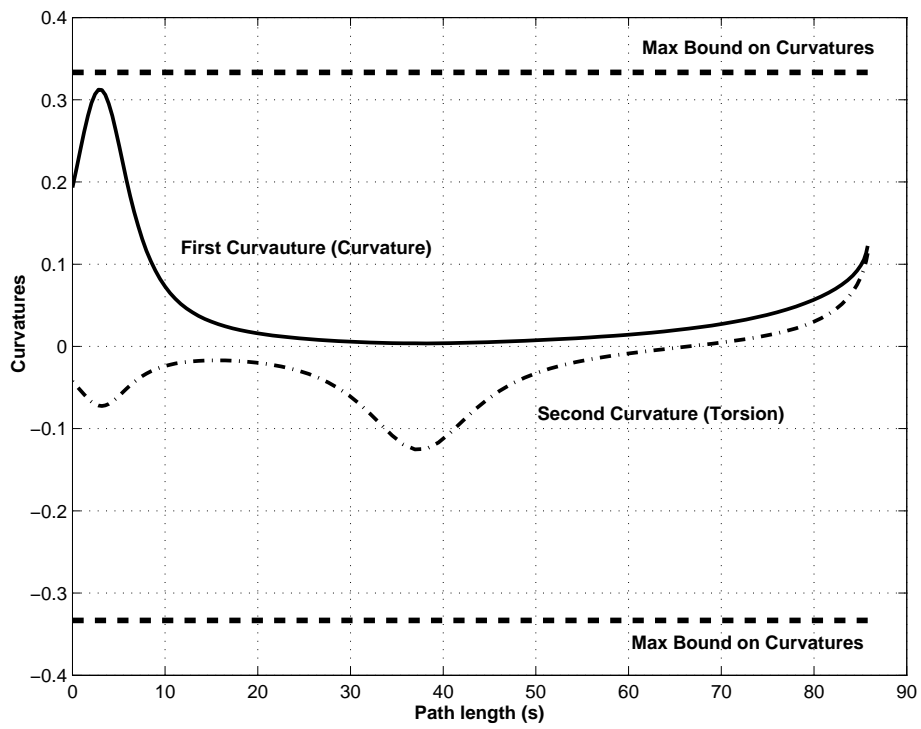
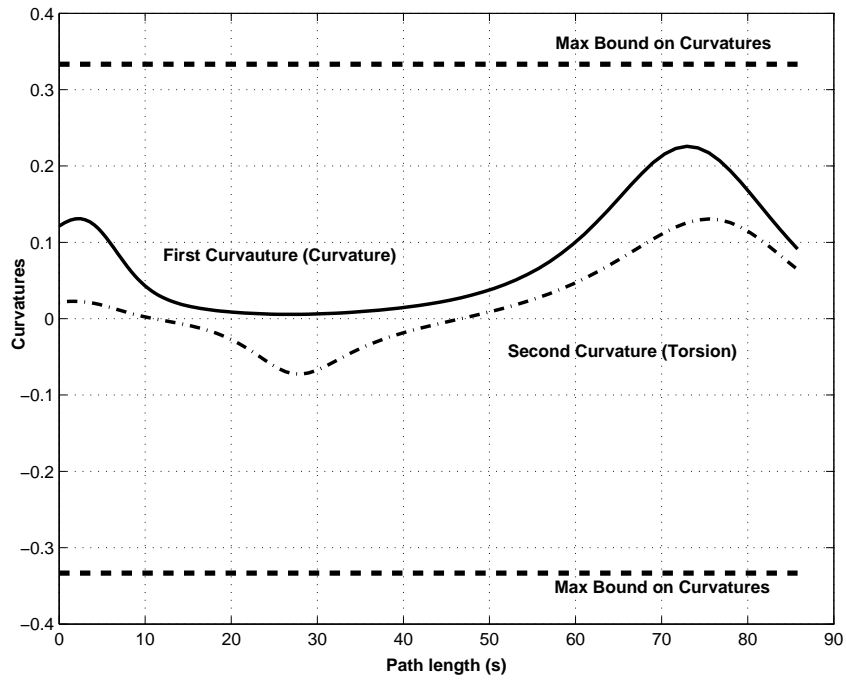
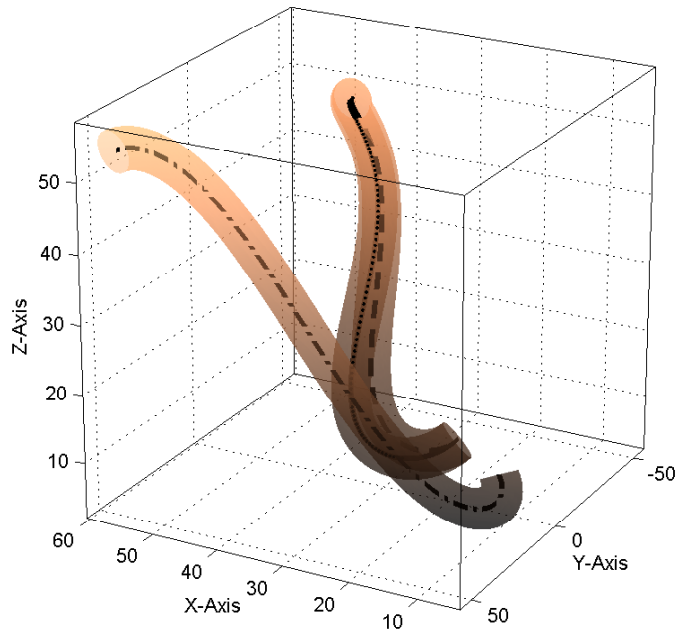


Figure 7.13: Curvatures variation: Flyable path of UAV2- PH 3D



**Figure 7.14:** Curvatures variation: Feasible path of UAV2 - PH 3D (Length is equal to that of UAV1)



**Figure 7.15:** Feasible (Flyable & Safe) paths - UAV1 & UAV2, '·' Path of UAV1, '·' - Path of UAV2-optimized for curvatures, '·' Path of UAV2-optimized for curvatures and path-length

---

# Conclusions and Future Work

## 8.1 Discussions and Conclusions

**A** safe and simultaneous arrival of UAVs are planned using the Dubins, Clothoid, and Pythagorean Hodograph paths. The safe paths are ensured by meeting three constraints: *(i)* curvature constraint, *(ii)* minimum separation distance and *(iii)* non-intersection of paths at equal lengths. The simultaneous arrival is ensured by design of paths of equal length. The boundary curvature of the paths are increased to meet the mission objective. The simulation results are shown between two way-points called base and target. However, the principles can be extended to any set of way-points.

The main theme of this thesis is to show that by simply varying the curvature of a path, it is possible to accomplish multiple objectives (e.g. shortest path, simultaneous arrival, safe flight). Three different type of paths are considered for the path planning: two composite paths: *(i)* Dubins and *(ii)* Clothoid paths and a single path: *(iii)* Pythagorean Hodograph path. The curvature discontinuity of the Dubins path limits its use to rotor-craft while the clothoid and PH paths provide curvature continuity can be used for both fixed-wing UAVs and rotor-craft.

The Dubins path are designed by principles of Euclidean and Differential geometries. It is shown that the existence and length of the Dubins path is simply a function of curvatures of turning circles. The advantage of differential geometry principles in

designing the Dubins and clothoid paths is demonstrated. A procedure is established to change the PH path of tangent continuity into the path of curvature continuity. In Dubins and PH cases, the curvature of the path is tuned to meet the path planning objectives.

The proposed methods are simulated in both two-dimensional and three-dimensional environments. The two dimensional case are shown in free-space and cluttered spaces. Three dimensional case is simulated with free-space.

The three main issues, convergence, computational time and scalability, in the multiple UAV path planning were discussed in this thesis. For the Dubins and Clothoid paths, there is always a solution as long as the turning circles obey the conditions (2.1.5a) and (2.1.5b) or (2.1.18). For the case of the PH path, the solution exists once the minimum energy path is available. The path length of the PH path is controlled by maintaining the path length close to that of the Dubins. This also can be achieved by using composite PH paths with curvature continuity at the end points. The simplicity of the proposed algorithms ensure that the computational time required for the coordinated guidance is implementable onboard. Finally as it is shown in equation (4.2.8), the proposed algorithms are easily scalable and thus implementable to swarms of large numbers of UAVs.

## 8.2 Future Work

There are two occasions, where iterative procedure is used in this thesis: (*i*) increasing the curvature to multiple constraints and (*ii*) finding the intersection points in threat avoidance. Elimination of iterative process can improve the efficiency of the path planner. This can be achieved by developing an analytical solution between the curvature and the regions of threat locations. The same idea is applied to intersection algorithm.

This thesis shows the path planning using the Dubins, Clothoid and PH paths separately. However, path planning can be achieved with the combination of all three paths. This would be an interesting area of further research.

The development of path planning with threat regions in three dimensional space is another area of future research work. This will be useful in path planning of spacecraft.

The interesting point of using the PH path is its rational properties. The offset path of these curves can be used to generate the safety bounds of the flight path and also

## **8. CONCLUSIONS AND FUTURE WORK**

---

used to represent the uncertainties. Also, these offset paths can be used for formation flying. These areas can further be explored for future research.

Currently an iterative method is employed to obtain the continuous curvature PH path from the tangent continuity PH path (refer section 2.4). There is a scope for future work to improve the efficiency of the PH path planning by finding an closed form solution to optimize for curvature continuity.

---

# Differential geometry

**D**ifferential geometry deals with geometry with the application of calculus to geometrical objects. In general, a curve  $r(t)$  is defined as a vector valued function in  $\mathfrak{R}^n$  space. The parameter  $t$  varies over a  $\mathfrak{R}$  number line. Mathematically, this is a continuous mapping  $r : I \rightarrow \mathfrak{R}^n$ , where  $I \in [a \ b]$  and  $t \in I$ . For example, a curve  $r(t)$  in  $3D$  is represented as  $r : I \rightarrow \mathfrak{R}^3$ , where  $r(t) = (x(t), y(t), z(t))$ . Thus a curve  $r(t)$  can be considered as a position vector in Euclidean space. If the parameter  $t$  is considered as time, the velocity and acceleration profiles can simply be found out by the application of calculus. The geometric properties of the curve/path per se can be studied by unit speed parametrization as follows:

The arc length  $s(t)$  of the curve  $r(t)$  is:

$$s(t) = \int_{s_1}^{s_2} \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2} dt \quad (\text{A.0.1})$$

The unit speed parametrization such that the parametric speed  $\dot{s} = \frac{ds}{dt}$ , of the path is unity. This is an ideal concept. This is explained as follows: Consider a vehicle starts moving at time  $t_1$  and stops at time  $t_2$ . The path length at time  $t_1$  is  $s_1$  and at time  $t_2$  is  $s_2$ . A path of unit speed parametrization have  $(t_2 - t_1 = s_2 - s_1)$ . This means that the time traveled is equal to the distance traveled. Mathematically,

$$\left| \frac{dr}{ds} \right| = \frac{\left| \frac{dr}{dt} \right|}{\left| \frac{ds}{dt} \right|} = 1 \quad (\text{A.0.2})$$

The physical significance of differential geometry of the curve is as follows. Taking  $t$  as time, the first derivative is the tangent vector and it defines velocity. The direction

## A. DIFFERENTIAL GEOMETRY

---

of (heading)velocity is specified by unit tangent vector,  $T$  and the speed is given by its modulus  $|T|$ . The second derivative is the acceleration vector and this has two components, one is along the tangent and other is normal to the tangent. The tangential acceleration is given by second derivative of velocity vector and its direction is along the direction of heading velocity. The direction of normal acceleration is given by a unit normal vector,  $N$  and its magnitude is equal to the centripetal acceleration given by  $\kappa|v|^2$ , where  $\kappa$  is the curvature and  $v$  is velocity. Thus the curvature is proportional to the lateral acceleration and hence the lateral force induced while the vehicle is turning. Taking the path-length as a parameter, the rate of change of tangent vector with respect to the arc length defines the tangent vector.

The cross product of the unit vectors,  $T$  and  $N$  produces a third unit vector, called Bi-normal  $B$  which is orthogonal to  $T$  and  $N$ . Thus the orthogonal triad  $(T, N, B)$  forms a moving frame on the curve. The plane spanned by the vectors,  $T$  and  $N$  is the Osculating plane. The vectors,  $N$  and  $B$  form the Normal plane and the vectors,  $B$  and  $T$  form the Rectifying plane. These three planes are orthogonal to each other. A continuous sequence of this triad represents orientation of the curve in space. The curvature and torsion ( $\kappa$ & $\tau$ ) completely specify a path in space.

$$\text{Unit Tangent Vector, } T = \frac{\dot{r}(t)}{|\dot{r}(t)|} \quad (\text{A.0.3})$$

$$\text{Unit Binormal Vector, } B = \frac{\dot{r}(t) \times \ddot{r}(t)}{|\dot{r}(t) \times \ddot{r}(t)|} \quad (\text{A.0.4})$$

$$\text{Unit Normal Vector, } N = B \times T \quad (\text{A.0.5})$$

The curvature profile at a point  $P$  is defined by the relation

$$\kappa = \frac{d\theta}{ds} \quad (\text{A.0.6})$$

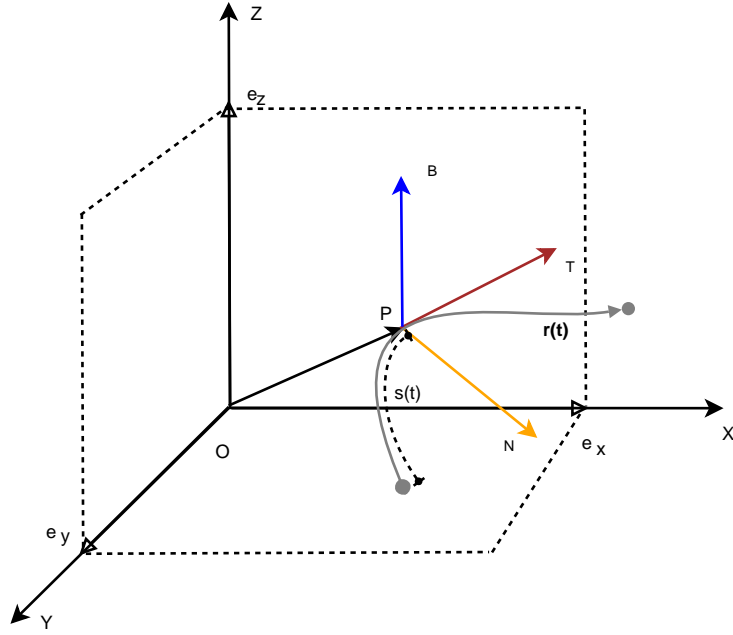
where  $s$  is the path length and  $\theta$  is the angle subtended by the tangent with the  $x$ -axis.

But,  $\frac{d\theta}{ds} = \frac{\frac{d\theta}{dt}}{\frac{ds}{dt}}$ . Hence, the equation (A.0.6) becomes

$$\omega = v\kappa \quad (\text{A.0.7})$$

where  $\omega(= \frac{d\theta}{dt})$  is the angular velocity and  $v(v = \frac{ds}{dt})$  is the linear velocity and  $t$  is the parameter, time.





**Figure A.1:** Curve with Frenet-Serret Frame  $T$  is unit tangent,  $N$  is unit normal and  $B$  unit binormal,  $\{T, N, B\}$  is the Frenet-Serret frame.  $r(t)$  is the equation of the path.  $P$  is the position vector of a point on the path.  $\{e_x, e_y, e_z\}$  is the unit vectors,  $s(t)$  is the path length

## A.1 Frenet Serret equations

The FS equations describe the rate of change of the curve with respect to the change of arc-length. The FS equations are:

$$T' = \kappa(s)N \quad (\text{A.1.1})$$

$$N' = -\kappa(s)T + \tau(s)B \quad (\text{A.1.2})$$

$$B' = -\tau(s)N \quad (\text{A.1.3})$$

In matrix form:

$$\begin{pmatrix} T' \\ N' \\ B' \end{pmatrix} = \begin{pmatrix} 0 & \kappa(s) & 0 \\ -\kappa(s) & 0 & \tau(s) \\ 0 & -\tau(s) & 0 \end{pmatrix} \begin{pmatrix} T \\ N \\ B \end{pmatrix} \quad (\text{A.1.4})$$

where the prime represents the derivative with respect to  $s$  and

$$\text{Curvature, } \kappa(s) = \frac{\|r'(s) \times r''(s)\|}{\|r'(s)\|^3} \quad (\text{A.1.5})$$

$$\text{Torsion, } \tau(s) = \frac{[r'(s) \cdot r''(s) \times r'''(s)]}{\|r'(s) \times r''(s)\|^2} \quad (\text{A.1.6})$$

The time-rate of change of the FS vectors in matrix form is:

$$\begin{pmatrix} \dot{T} \\ \dot{N} \\ \dot{B} \end{pmatrix} = \dot{s} \begin{pmatrix} 0 & \kappa(t) & 0 \\ -\kappa(t) & 0 & \tau(t) \\ 0 & -\tau(t) & 0 \end{pmatrix} \begin{pmatrix} T \\ N \\ B \end{pmatrix} \quad (\text{A.1.7})$$

where  $\dot{s} = \frac{ds}{dt}$  is the speed (parametric speed) and  $t$  is a parameter.

$$\text{Curvature, } \kappa(t) = \frac{\|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)\|}{\|\dot{\mathbf{r}}(t)\|^3} \quad (\text{A.1.8})$$

$$\text{Torsion, } \tau(t) = \frac{\dot{\mathbf{r}}(t) \cdot \ddot{\mathbf{r}}(t) \times \dddot{\mathbf{r}}(t)}{\|\dot{\mathbf{r}}(t) \times \ddot{\mathbf{r}}(t)\|^2} \quad (\text{A.1.9})$$

## A.2 Importance of Curvature and Torsion

Mathematically, a flyable path is a regular curve which captures both the geometric (locus of points) and kinematic (motion) aspects. A regular curve  $r$  is a mapping  $r : [a, b] \rightarrow \mathbf{R}^3$  at least twice continuously differentiable,  $r \in \mathbf{C}^3$  and satisfying the regularity condition  $\frac{dr}{dt} \neq 0$  for all  $t \in [a, b]$ . Regularity means that the point moving along the curve is not allowed to stop, a natural requirement for fixed-wing UAVs. However, considering the kinematic constraints, it is important for the path to have curvature continuity.

By the principles of differential geometry [58], [57], the curvature and torsion are fundamental properties of a path, by which a curve is completely determined in space. In two dimension, only curvature is enough. Apart from the geometric insights, these two properties play an important role in mechanics of a moving vehicle. The physical significance of these properties are that the curvature is proportional to the lateral acceleration and is measured by rate of change of tangent vector, while the torsion is proportional to the angular momentum and is measured by rate of change of tangent plane.

$$\kappa(t) = \frac{\dot{\mathbf{r}} \times \ddot{\mathbf{r}}}{\|\dot{\mathbf{r}}\|^3} \quad (\text{A.2.1})$$

$$\tau(t) = \frac{\det\{\dot{\mathbf{r}}, \ddot{\mathbf{r}}, \dddot{\mathbf{r}}\}}{\|\dot{\mathbf{r}} \times \ddot{\mathbf{r}}\|^3} \quad (\text{A.2.2})$$

From the equation (A.2.1), the curvature and torsion respectively are the function of first two and three derivatives of the path. Hence, it is necessary to have a path of minimal order enough to satisfy curvature constraints and additional flexibility to negotiate with safety constraints.

---

## Pythagorean Hodograph

**P**ythagorean Hodograph is known for its rational properties. Consider a path  $r(t) = (x(t), y(t))$  and its length is  $s(t)$ .

$$s(t) = \int_{t_1}^{t_2} \sqrt{\dot{x}^2 + \dot{y}^2} dt \quad (\text{B.0.1})$$

where  $\dot{x} = \frac{dx}{dt}$ ,  $\dot{y} = \frac{dy}{dt}$  and  $t \in [t_1 t_2]$

The parametric speed is  $\dot{s} = \frac{ds}{dt}$ . If  $\dot{s} = \dot{r}$ , this is called unit speed parametrization.

To calculate the path length exactly, that is without any approximation, the equation (B.0.1) should have closed form solution. But this is not easy to obtain even for simple polynomials except straight line [70].

The derivatives  $\dot{x}$  and  $\dot{y}$  are called hodographs. The path length is the function of the hodographs. To arrive at a simple solution without any approximation, the term inside the square-root term should be a square of some polynomial  $\sigma(t)$ . Or in other words, if a polynomial  $\sigma(t)$  is selected such that:

$$\sigma(t)^2 = x(t)^2 + y(t)^2 \quad (\text{B.0.2})$$

Such formulation eliminate the approximation in calculation of path length. Also, it results in rational properties of the path. Using the basic algebraic formulae  $(a+b)^2 = a^2 + b^2 + 2ab$ , and  $(a^2 - b^2) = (a+b)(a-b)$ , assign the hodographs with these polynomials  $u(t)$ ,  $v(t)$ , and  $w(t)$  such that:

$$\dot{x}(t) = [u(t)^2 - v(t)^2]w(t) \quad (\text{B.0.3a})$$

$$\dot{y}(t) = 2u(t)v(t)w(t) \quad (\text{B.0.3b})$$

## B. PYTHAGOREAN HODOGRAPH

---

This implies:

$$\sigma(t) = [u(t)^2 + v(t)^2]w(t) \quad (\text{B.0.4})$$

Taking  $w(t) = 1$  and  $\text{GCD}(u(t), v(t)) = 1$  gives a regular PH path, that is  $\text{GCD}(\dot{x}, \dot{y}) = 1$  of odd degree.

Such a formulation results that the path length and parametric speed is just a polynomial. And also the offset path, curvature and torsion all become rational.

$$s(t) = \int_0^1 w(t)[u(t)^2 + v(t)^2]dt, \quad t \in [0, 1] \quad (\text{B.0.5})$$

$$T = \frac{u(t)^2 - v(t)^2, 2u(t)v(t)}{u(t)^2 + v(t)^2} \quad (\text{B.0.6})$$

$$N = \frac{2u(t)v(t), v(t)^2 - u(t)^2}{u(t)^2 + v(t)^2} \quad (\text{B.0.7})$$

$$\kappa = \frac{2(u(t)\dot{v}(t) - v(t)\dot{u}(t))}{w(t)(u(t)^2 + v(t)^2)^2} \quad (\text{B.0.8})$$

---

# Quaternion

Quaternion is introduced by Hamilton in 1843, can be considered as hyper-complex number or rank 4. A quaternion  $q$  is a combination of a scalar part  $q_s$  and a vector part  $\vec{q}_v$  such that  $q = q_s + \vec{q}_v$ , where  $\vec{q}_v = iq_1 + jq_2 + kq_3$  and the triad  $\{i, j, k\}$  a standard orthonormal basis in  $\mathfrak{R}^3$ . The triad follows the rule:  $i^2 = j^2 = k^2 = ijk = -1$  and the other rules are similar to that of complex number.

## C.1 Properties of quaternion

Consider two quaternions:  $p = p_s + \vec{p}_v$  and  $q = q_s + \vec{q}_v$ .

Addition, subtraction, conjugate and magnitude respectively are:

$$p + q = (p_s + q_s) + (\vec{p}_v + \vec{q}_v) \quad (\text{C.1.1})$$

$$p - q = (p_s - q_s) + (\vec{p}_v - \vec{q}_v) \quad (\text{C.1.2})$$

$$q^* = q_s - \vec{q}_v \quad (\text{C.1.3})$$

$$|q|^2 = q^* q = qq^* = q_s^2 + |\vec{q}_v|^2 \quad (\text{C.1.4})$$

$$(pq)^* = q^* p^* \quad (\text{C.1.5})$$

Unit quaternion is a quaternion with  $|q| = 1$  and norm of a quaternion is:

$$N(q) = q_s - \vec{q}_v \quad (\text{C.1.6})$$

## C. QUATERNION

---

Multiplication: Non-commutative

$$pq = p_s q_s - \vec{p}_v \cdot \vec{q}_v + p_0 \vec{q}_v + q_0 \vec{p}_v + \vec{p}_v \times \vec{q}_v \quad (\text{C.1.7})$$

where the first term in RHS is a scalar and the remaining terms form a vector.



---

# Bibliography

- [1] Structures Committee on Materials, Commission on Engineering Aeronautics for Advanced Uninhabited Air Vehicles, and US Technical Systems, National Research Council. *Uninhabited Air Vehicles: Enabling Science for Military Systems*. Number ISBN: 978-0-309-06983-0. National Academy Press, 2000.
- [2] US Office of the secretary of defense. Unmanned Aircraft Systems Roadmap 2005-2030. August 2005.
- [3] J. G. Bender. An overview of systems studies of automated highway systems. *IEEE Transactions on Vehicular Technology*, 40(1):82–99, 1991.
- [4] S.E.Shladover. Automatic vehicle control developments in the PATH program. *IEEE Transactions on Vehicular Technology*, 40(1):114–130, 1991.
- [5] Y.K.Chan and M.Foddy. Real time optimal flight path generation by storage of massive data bases. In *IEEE NEACON, Institute of Electrical and Electronics Engineers, New York*, pages 516–521, 1985.
- [6] J.Hebert, D.Jacques, M.Novy, and M.Pachter. Cooperative control of uavs, aiaa-2001-4240. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, Aug. 6-9 2001*.
- [7] J.L.Vian and J.R.More. Trajectory optimization with risk minimization for military aircraft. *AIAA Journal of Guidance, Control and Dynamics*, 12(3):311–317, 1989.

## Bibliography

---

- [8] M.Zabarankin, S.Uryasev, and P.Pardalos. *Optimal Risk Path Algorithms, Cooperative Control and Optimization*, volume 66, pages 271–303. Kluwer Academic Publishers, Dordrecht, 2002.
- [9] J. Yuh. Underwater robotics. In *IEEE Conference on Robotics and Automation*, pages 932–937, 2000.
- [10] J. Yuh. Developments in underwater robotics. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1862–1867, 1995.
- [11] T. R. Smith, H. Hanßmann, and N. E. Leonard. Orientation control of multiple underwater vehicles. In *40th IEEE Conference on Decision and Control*, pages 4598–4603, 2001.
- [12] P.Oliveira, A. Pascoal, V. Silva, and C. Silvestre. Mission control of the marius auv: System design, implementation, and sea trials. *International Journal of Systems Science:special issue on Underwater Robotics*, 29(10): 1065–1080, 1998.
- [13] P.Tompkins, A.Stentz, and D.Wettergreen. Global path planning for mars rover exploration. In *IEEE Aerospace Conference*, March 2004.
- [14] D.B.Gennery. Traversability analysis and path planning for a planetary rover. *Autonomous Robots*, pages 131–146, October 28 2004.
- [15] Steven M. LaValle. *Planning algorithm*. Cambridge University Press, 2006.
- [16] J.C.Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.
- [17] L.E.Dubins. On curves of minimal length with a constraint on average curvature and with perscribed initial and terminal positions and tangent. *American Journal of Mathematics*, 79:497–516, 1957.
- [18] X. N. Bui, J. D. Boissonnat, P. Soueres, and J. P. Laumond. Shortest path synthesis for dubins non-holonomic robots. In *IEEE International Conference on Robotics and Automation*, pages 2–4, 1994.
- [19] A. Shkel and V. Lumelsky. On calculation of optimal paths with constrained curvature: The case of long paths. In *IEEE International Conference on Robotics and Automation*, page 35783583, 1996.



- [20] T.G.McGee, S.Spry, and J.K.Hedrick. Optimal path planning in a constant wind with a bounded turning rate. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California*, August, 15-18 2005.
- [21] M. Massink and N. Francesco. Modelling free flight with collision avoidance. In *IEEE International Conference on Engineering of Complex Computer Systems*, page 270279, 2001.
- [22] M.Shanmugavel, A.Tsourdos, R. Żbikowski, and B.A.White. 3d dubins sets based coordinated path planning for swarm of uavs, aiaa-2006-6211. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, Colorado*, August 21-24 2006.
- [23] A. Bicchi and L. Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(4):3578–3583, 2000.
- [24] M.Robb, B.A.White, A.Tsourdos, and D.Rulloda. Reachability guidance: a novel concept to improve mid-course guidance. In *American Control Conference*, volume 1, pages 339– 345, 8-10 June 2005.
- [25] J.A.Reeds and R.A.Shepp. Optimal paths for a car that goes both forward and backward. *Pacific Journal of Mathematics*, 145(2):367393, 1990.
- [26] J.P.Laumond. Feaible trajectories for mobile robots with kinematic and environment constraints. In *Proceedings of the Int. Conf. on intelligent Autonomous Systems, Amsterdam (NL)*, pages 346–354, December 1986.
- [27] K.Komoriya and K.Tanie. Trajectory design and control of a wheel-type mobile robot using b-spline curve. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, Tsukuba (JP)*, pages 398–405, 1989.
- [28] T.Hongo A.Takahashi and Y.Ninomiya. Local path planning and control for agv in positioning. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, Tsukuba (JP)*, pages 392–397, September 1989.
- [29] W.L.Nelson. Continuous curvature paths for autonomous vehicles. In *IEEE Int. Conf. on Robotics and Automation, Scottsdale, AZ (US)*, volume 3, pages 1260–1264, May 1989.

## Bibliography

---

- [30] R.Liscano and D.Green. Design and implementation of a trajectory generator for n indoor mobile robot. In *IEEE-RSJ Int. Conf. on Intelligent Robots and Systems, Tsukuba (JP)*, pages 380–385, September .
- [31] Y.Kanayama and B.I.Hartman. Smooth local path planning for autonomous vehicles. In *IEEE Int. Conf. on Robotics and Automation, Scottsdale, AZ (US)*, pages 1265–1270, May .
- [32] A.Piazzi and C.Guarino Lo Bianco. Quintic  $G^2$  splines for trajectory planning of autonomous vehicles. In *IEEE Intelligent Vehicles Symp., Dearborn, MI (US)*, pages 193–200, 2000.
- [33] S.Iyengar, C.Jorgensen, N.Rao, and R.Weisbin. Robot navigation algorithm using learned spatial graphs. *Robotica*, 4(2):93100, 1986.
- [34] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transaction on Robotics and Automation*, 12(4):566580, 1996.
- [35] Per Olof Pettersson and Patrick Doherty. Probabilistic roadmap based path planning for an Autonomous Unmanned Aerial Vehicle. In *Workshop on Connecting Planning and Theory with Practice, , 14th International Conference on Automated Planning and Scheduling, ICAPS*, 2004.
- [36] Z. Shen P. Cheng and S. M. LaValle. Rrt-based trajectory design for autonomous automobiles and spacecraft. *Archives of Control Sciences*, 11 (3-4):167–194, 2001.
- [37] S. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, October 1998.
- [38] Yeonju Eun and Hyochoong Bang. Cooperative control of multiple unmanned aerial vehicles using the potential field theory. *Journal of Aircraft*, 43(6):1805–1814, November 2006.
- [39] L.C.Polymenakos, D.P.Bertsekas, and J.N.Tsitsiklis. Implementation of efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 43(2):278–283, 1998.
- [40] J.N.Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.

- [41] Myungsoo Jun and Raffaello D'Andrea. *Path planning for unmanned aerial vehicles in uncertain and adversarial environments*, chapter 6, pages 95–110. Kluwer Academic Publishers, 2003.
- [42] A. Davari Ademoye. T.A. and W. Cao. Three dimensional obstacle avoidance maneuver planning using mixed integer linear programming. *Robotics and Applications*, 537, 2006.
- [43] T.Schouwenaars, B.De Moor, E.Feron, and J.How. Mixed integer programming for multi-vehicle path planning. In *European control conference, Porto, Portugal*, September 2001.
- [44] A.Richards, J.How, T.Schouwenaars, and E.Feron. Plume avoidance maneuver planning using mixed integer. In *AIAA Guidance, Navigation and Control Conference*, August 2001.
- [45] Jonathan P. How Arthur Richards. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *American Control Conference*, 2002.
- [46] I.M.Mitcheel and S. Sastry. Continuous path planning with multiple constraints. In *IEEE Conference on Decision and Control*, pages 5502–5507, 2003.
- [47] Guang Yang and Vikram Kapila. Optimal path planning for unmanned air vehicles with kinematic and tactical constraints. In *IEEE Conference on Decision and Control*, pages 1301–1306, 2002.
- [48] E.Gagnon C.A.Rabbath and M.Lauzon. On the cooperative control of multiple unmanned aerial vehicles. *IEEE Canadian Review*, pages 15 – 18, 2004.
- [49] P.Chandler, S.Rasmussen, and M. Pachter. Uav cooperative control. In *AIAA GNC, Denver*, August 2000.
- [50] S.Bortoff. Path planning for uavs. In *American Control Conference, Chicago, IL*, 2000.
- [51] T.McLain. Cooperative rendezvous of multiple unmanned air vehicles. In *AIAA GNC, Denver*, August 2000.

## Bibliography

---

- [52] Kevin B. Judd and Timothy W. McLain. Spline based path planning for unmanned air vehicles, aiaa-2001-4238. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada*, Aug. 6-9 2001.
- [53] Madhavan Shanmugavel, Antonios Tsourdos, Rafał Żbikowski, and Brian A. White. Path planning of multiple uavs using dubins sets. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, San Francisco, California*, Aug. 15-18 2005.
- [54] Madhavan Shanmugavel, Antonios Tsourdos, Rafał Żbikowski, and Brian A. White. A solution to simultaneous arrival of multiple uavs using pythagorean hodograph curves. In *American Control Conference, Mineapolis*, June 14-16 2006.
- [55] Madhavan Shanmugavel, Antonios Tsourdos, Rafał Żbikowski, and Brian A. White. 3d path planning for multiple UAVs using Pythagorean Hodograph curves. In *AIAA Guidance, Navigation, and Control Conference and Exhibit, Hilton Head, South Carolina*, 20 - 23 Aug 2007.
- [56] Madhavan Shanmugavel, Antonios Tsourdos, Rafał Żbikowski, and Brian A. White. Path planning of multiple UAVs with clothoid curves. In *17<sup>th</sup> IFAC Symposium on Automatic Control in Aerospace, Toulouse, France*, 25 - 29 June 2007.
- [57] Erwin Kreyszig. *Differential geometry*. Dover Publications, Inc., New York, 1991.
- [58] Martin M Lipschutz. *Schaum's Outline of Differential Geometry*. Number ISBN-10: 0070379858. McGraw-Hill, New York, 1969.
- [59] R.T.Farouki and T.Sakkalis. Pythagorean Hodographs. *IBM Journal of Research and Development*, 34(5):736–752, 1990.
- [60] Takashi Maekawa. An overview of offset curves and surfaces. *Computer Aided Design*, (31):165–173, 1999.
- [61] R.T.Farouki and C.A.Neff. Hermite interpolation by pythagorean hodograph quintics. *Mathematics of Computation*, 64(212):1589–1609, 1995.
- [62] R.T.Farouki. The elastic bending energy of pythagorean-hodograph curves. *Computer Aided Geometric Design*, 13:227–241, 1996.

- [63] H.Bruyninckx and D. Reynaerts. Path planning for mobile and hyper-redundant robots using pythagorean hodograph curves. In *8th International Conference on Advanced Robotics, ICAR '97*, pages 595–600, 1997.
- [64] R.T.Farouki and T.Sakkalis. Pythagorean hodograph space curves. *Advanced in Computational Mathematics*, 2:41–66, 1994.
- [65] R.T.Farouki, M.Kandari, and T.Sakkalis. Hermite interpolaiton by rotation-invariant spatial pythagorean hodograph curves. *Advances in Computational Mathematics*, 17:369–383, 2002.
- [66] J.O.Kim and P.K.Khosla. Real-time obstacle avoidance using harmonic potential functions. *IEEE Transactions on Robotics and Automation*, 8(3):338–349, 1992.
- [67] S.M.Li, J.D.Boskovic, S.Seereeram, R.Prasanth, J.Amin, R.K.Mehra, R.W. Beard, and T.W.McLain. Autonomous hierarchical control of multiple unmanned combat air vehicles (UCAVs). In *American Control Conference, Anchorage, AK*, volume 1, pages 274 – 279, 2002.
- [68] R.Beard, T.McLain, M.Goodrich, and E.Anderson. Coordinated target assignment and intercept for for unmanned air vehicles. *IEEE Transactions of Robotics Automation*, 18(6):911–922, 2002.
- [69] H.I.Yang and Y.J.Zhao. Trajectory planning for autonomous aerospace vehicles amid obstacles and conflicts. *Journal of Guidance, Control and Dynamics*, 27(6):997–1008, Novermber-December 2004.
- [70] R.T.Farouki and T.Sakkalis. Real rational curves are not unit speed. *Computer Aided Geometric Design*, 8:181–187, 1991.