CRANFIELD UNIVERSITY

Mohammed Afzal Shah

Cooperative Path Planning and Cooperative Perception for UAVs Swarm

DEFENCE COLLEGE OF MANAGEMENT AND TECHNOLOGY
DEPARTMENT OF INFORMATICS AND SYSTEMS ENGINEERING
CRANFIELD UNIVERSITY
SHRIVENHAM

PhD Thesis

CRANFIELD UNIVERSITY

DEFENCE COLLEGE OF MANAGEMENT AND TECHNOLOGY
DEPARTMENT OF INFORMATICS AND SYSTEMS ENGINEERING

PhD Thesis
Academic Year 2008-2011

Mohammed Afzal Shah

Cooperative Path Planning and Cooperative Perception for UAVs Swarm

Supervisor
Dr. Antonios Tsourdos
7 January 2011

Glossary

$p(x, y)$	Coordinates of a point in 2D
θ	Heading angle of UAV
$Pose(x, y, \theta)$	General pose of UAV in 2D
$Pose_{si}(x_{si}, y_{si}, \theta_{si})$	Starting pose or configuration of *ith* UAV in 2D
$Pose_{fi}(x_{fi}, y_{fi}, \theta_{fi})$	Final pose or configuration of *ith* UAV in 2D
$p(x, y, z)$	Coordinates of a point in 3D
(θ_{si}, ϕ_{si})	Yaw, Pitch
$Pose(x, y, z, \theta, \phi)$	General pose of UAV in 3D
$Pose_{si}(x_{si}, y_{si}, z_{si}, \theta_{si}, \phi_{si})$	Starting pose or configuration of *ith* UAV in 3D
$Pose_{fi}(x_{fi}, y_{fi}, z_{fi}, \theta_{fi}, \phi_{fi})$	Final pose or configuration of *ith* UAV in 3D
$r(t)$	Pythagorean Hodograph path
$r_i(t) = [x(t), y(t), z(t)]$	Path of *ith* UAV in parametric form
κ_i	Curvature of the *ith* path
τ_i	Torsion of the *ith* path
$\kappa_{\max}$	Maximum curvature of the path
$\tau_{\max}$	Maximum torsion of the path
R_{si}	Radius of safety circle or sphere around *ith* UAV
R_{sj}	Radius of safety circle or sphere around *jth* UAV
d	Distance between two neighbouring UAVs
$x(t)$	X-component of Path $r(t)$
$y(t)$	Y-component of Path $r(t)$
$x'(t)$	Derivative of $x(t)$
$y'(t)$	Derivative of $y(t)$
σ	Speed of UAV
$P_k = (x_k, y_k)$	General control point of the PH path in2D
P_0, P_1, P_2, P_3	First, second, third, and fourth control points of PH cubic
$r'(t)$	Derivative of PH curve

$dp_{si} = (dx_{si},\ dy_{si})$	Derivative at starting pose of UAV in 2D
$dp_{fi} = (dx_{fi},\ dy_{fi})$	Derivative at final pose of UAV in 2D
$P_k = (x_k, y_k, z_k)$	General control point of the PH path in3D
$dp_{si} = (dx_{si},\ dy_{si},\ dz_{si})$	Derivative at starting pose of UAV in 3D
$dp_{fi} = (dx_{fi},\ dy_{fi},\ dz_{fi})$	Derivative at final pose of UAV in 3D
$(\lambda_{fi},\ \mu_{fi},\ \nu_{fi})$	Direction cosine of dp_{fi}
$(\lambda_{si},\ \mu_{si},\ \nu_{si})$	Direction cosine of dp_{si}
E	Elastic Bending Energy
$A(t)$	Quaternion
$A^*(t)$	Quaternion conjugate
$\vec{i},\ \vec{j},\ \vec{k}$	Unit vectors in directions of X, Y and Z axis
$(\vec{t}, \vec{n}, \vec{b})$	Basis vectors of Frenet Frame
$\vec{t}$	Unit Tangent vector
$\vec{n}$	Unit normal vector
$\vec{b}$	Unit binormal vector
$(\vec{e}_1, \vec{e}_2, \vec{e}_3)$	Basis vectors of Rotation Minimising Frame
$({}^cX, {}^cY, {}^cZ)$	Coordinates of the scene point in camera frame
(x, y, z)	Coordinates of the image point in camera frame
$({}^wX, {}^wY, {}^wZ)$	Coordinates of a point in world frame
R	Rotation Matrix
T	Translation vector
f	Focal- length
O_x, O_y	Coordinates in pixel of image centre
$s_x,\ s_y$	Effective size of the pixel (in millimetres) in the horizontal and vertical direction respectively
${}^{im}x,\ {}^{im}y$	Coordinates of image point in image frame
w_k, v_k	Process and measurements noise
$\hat{x}_k^-$	Priori state estimate at step k
$\hat{x}_k$	Posteriori state estimate at step k

Z_k	Measurement
P_k	*Posteriori* estimate error covariance
$X = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T$	State vector of target
A, C	The system and measurement matrices
$\vec{x}_i$	Position vector of target
$\hat{\vec{x}}_i$	Position estimate of the target
$\delta \vec{x}_i$	Uncertainty in position estimate of target
$\vec{d}_i$	Unit direction vector
$\hat{\vec{d}}_i$	Unit direction vector estimate
$\delta\vec{d}_i$	Uncertainty in unit direction estimate of target
$\vec{q}$	True position of target
$\hat{q}^*$	Optimal target position estimate used as a measurement in the Kalman filter
δq^*	Error in estimates $\hat{q}^*$
(x_{cen}, y_{cen})	Centre of the square shaped obstacle
(x_{iwp}, y_{iwp})	Location of the intermediate waypoint
θ_{iwp}	Direction or heading angle at the intermediate waypoint
(x_{uav}, y_{uav})	Current position of the UAV given by onboard position sensors
R_c	Radius of communication sphere of UAV
R_{fov_i}, R_{fovj}	Radii of the fields of view of the *ith* and *jth* UAVs cameras
x_{obj}	Object of interest in combined fields of view
$(\vec{t}_{cam}, \vec{n}_{cam}, \vec{b}_{cam})$	Camera reference frame with the origin at UAVs' centre of Mass and obtained by rotating $(\vec{t}, \vec{n}, \vec{b})$ by a tilt angle ζ about the optical axis of the camera

Abstract

In this research Pythagorean Hodograph based path planning and camera based cooperative perception are investigated separately and then these two entirely separate areas (Path Planning and Perception) are integrated for the application in online pop-up obstacle locating & avoidance and moving target tracking & surveillance in dynamic environments. The path planning is integrated with the cooperative perception to deal with the challenges posed by the dynamic environment. The aim of this integration is to achieve maximum autonomy required to execute a mission autonomously by multiple fixed wings UAVs in a dynamic environment.

During the mission execution, the cooperating UAVs start from some initial location in the operating environment and finish at some final location while trying to achieve the mission's objectives in a cooperative way. Naturally planning a feasible (safe and flyable) path for each participating UAV from initial position to a final location becomes a compulsory task of mission planning. For fixed wing UAVs flyable paths mean, paths which have tangential and curvature continuity and which obey the kinematic and dynamic constraint of the UAVs. In this research an algorithm based on Pythagorean hodograph curves is developed and used for planning feasible (safe and flyable) paths. The Pythagorean hodograph (PH) yields paths of exact length having tangential and curvature continuity. These continuous paths are made flyable for the UAVs by imposing the kinematic constraints of the UAVs. These constraints are imposed by the curvature and torsion manipulation of the planned paths. The safety of these paths is ensured by making it free of inter collisions between the vehicles and collisions with the known obstacles. These feasible paths are known as the initial paths or reference trajectories.

In this research the operating environment is assumed to be dynamic in which changes are taking place at all times. Each UAV taking part in the mission is equipped with a vision sensor to perceive these changes continuously in a cooperative way. As the mission is assumed to be executed in day light, therefore light intensity video camera is used as a vision sensor. A perception algorithm for locating an object cooperatively in 3D is developed in this research. This algorithm is based on the optimization of errors in target position acquired by the on board camera. The algorithm is used by the cooperative perception system for optimal position estimation of the object in the scene. The target position information between the participating UAVs is exchanged through wireless communication for data fusion purposes.

After developing efficient algorithms for path planning and cooperative perception, the two algorithms are integrated to be used in reactive obstacle avoidance and target tracking.

During the mission, when the UAVs start their flight on the reference trajectories generated by the path planning algorithm, the perception algorithm comes into action.

During the travel on these paths if the perception system of any of the UAVs detects an interrupting obstacle which was not known a *priori* in the map, then the exact location of this obstacle is determined with the help of the perception algorithm in a cooperative way. Using the location of the interrupting obstacle determined by the perception algorithm the path planning algorithm plans an evasive manoeuvre for the corresponding UAV to avoid it. After avoiding the obstacle the UAV comes back to its reference trajectory as soon as possible.

In the operation of surveillance and tracking during the mission, the onboard perception algorithm locates an object of interest dynamically and the Pythagorean hodograph (PH) path planner uses this location to generate the paths for the cooperating UAVs to keep in close proximity of the target. In this case the close proximity of the target means to follow the moving target in such a way that it remains in the fields of views of the UAVs cameras at any time. By this integration of path planning and cooperative perception the continuous surveillance and tracking of the target was made possible even when the individual UAV experiences failure.

During this research the mid flight obstacle locating & avoidance, and target surveillance & tracking have been successfully achieved by the integration of the path planning and cooperative perception. The purpose of this integration is to achieve an enhanced autonomy for the cooperating group of UAVs to increase the probability of their survival in mission being executed in dynamic environments.

Acknowledgments

First of all I humbly thank the Almighty for providing me with such a glorious opportunity to undertake my PhD in such a reputed institution.

I express sincere gratitude to my honourable supervisor Professor Antonios Tsourdos whose guidance and encouragement supported me throughout my research.

I cordially thank my thesis committee, Dr. Mark Richardson, Dr. David James and Dr. Peter Silson for their valuable suggestion and timely advice.

My special thanks go to all my colleagues and friends at Heaviside lab, Wallington Hall and Marlborough Hall whose support was always there in hard times.

Last but not least I will express my gratitude and obligations to ESPRC and my industrial sponsor BAE Systems for their generous grant.

Mohammed Afzal Shah

UK

December 2010.

Contents

List of Figures

List of Table

Chapter 1

Introduction

The numerous advantages of the use of an Unmanned Aerial Vehicle (UAV) in today's hi-tech world, and its superiority over its manned counterpart in the Dull, Dirty and Dangerous missions has made it a suitable option in many civil and military applications. The use of UAVs has excluded human casualties in dangerous missions like, airborne combat, military surveillance, borders patrol and search & rescue operations. They have extensively reduced the health risks in dirty tasks such as operating in chemically and biologically contaminated environments. They also have lessened the boredom, dullness and cost of the lengthy and repetitive tasks including road traffic monitoring, weather forecast, power cable inspection for damages and forest fire detection. These benefits of the use of the Autonomous Ariel Vehicles in civil and military missions have made the UAVs an established research area in academia.

With the availability of computationally powerful processors, the miniature positioning sensors (GPS, INS), perception sensors (camera, ultrasounds, range, radar) and the compact wireless communication kits, the replacement of remotely man operated UAVs with fully autonomous UAVs does not seem to be a distant reality. Now a days there is an extensive research to use various sensors on the UAVs to make them an autonomous platforms suitable for deployment in dynamic environments. The main aim of using multiple sensors on the UAV is to maximise the autonomy of the vehicle which guarantees its safety in a dynamic environment.

After the success of single UAV missions, there is an extensive research on the possibility of deploying a group of cooperating autonomous UAVs in a single mission. The purpose of deployment of multiple cooperative autonomous UAVs in a single mission is to provide a mean to minimise the mission's execution time, maximise robustness to the adversities of operating environment, and to avert catastrophic failure of the mission to graceful degradation in case of UAVs loss. Therefore the future holds the vision of the deployment of a swarm of more autonomous UAVs working cooperatively to achieve the objectives of the mission in an optimal way. This vision opens a new research chapter where the main focus is on the cooperation to achieve optimality and integration of diverse technologies to achieve maximum autonomy. To realize this future vision of the use of UAVs there are hectic research activities in academia and R&D throughout the world. This research is intended to be a step forward in the direction of realization of this vision.

In this research the mission is assumed to start from some initial point and concludes at some goal point, therefore the path planning from start to goal point for a group of cooperating UAVs deployed in the mission becomes an essential part of this research. Since the mission is executed in a dynamic environment which is changing all the times, therefore there is a need for some sensors to perceive these changes. The camera sensor is used in this research for locating obstacles in 3D, and surveillance & tracking of an object of interest in the operating environment. Since onboard camera sensors are used in a cooperative way to perceive the events taking place dynamically in the operating environment, therefore, the theme of this research is "Path planning and cooperative perception for swarm of the UAVs".

This research adopts the approach of integration of path planning and cooperative perception for successful mission execution in uncertain environments. In literature there are many references to the path planning and cooperative perception as two separate areas in the UAVs mission context, but their integration such that they assist each other in achieving the mission objectives cooperatively in an optimal way, is a novel approach and motivation of this research.

1.1 Schematic of Integration of Path Planning and Cooperative Perception

In practice the mission or task is defined by human operator who generates a set of waypoints to achieve the objectives of the mission. These waypoints (dictated by the mission and generated by human operator on the basis of partial knowledge of the environment) are input to the Path Planner. The Path Planner connects the waypoints through a number of feasible trajectories called reference trajectories. The number of trajectories corresponds to the number of UAVs taking part in the mission such that for each UAV there is one reference path.

During the mission execution the UAVs travel on these reference trajectories. During these journeys if any of the sensors on board UAVs detects an obstacle which was not known a *priori* and which is in conflict with the reference trajectory of any of the vehicles in the group, the path planner is informed. Based on the local information obtained from the sensor the path planner makes amendments to the reference trajectory of the corresponding UAV to avoid the obstacle. Thus the obstacle is located by perception system and avoided by path planning by generating a path clear of the obstacle. By the integration of path planning and cooperative perception the obstacle is successfully avoided.

During the mission execution if there is a need for cooperation (in perception) between the UAVs to track the motion of an object of interest in the operating environment, then the path planner is requested, which generates paths for the corresponding UAVs to ensure that the target is kept in sight all the time. The path planning provides the guidance for the UAVs to keep the target in sight and the cooperative perception system

locates the target in 3D as it is being chased. By means of the integration of path planning and cooperative perception the target is tracked effectively.

In this research the changes to the reference trajectories are made because of the following two reasons.

- Obstacles locating and avoidance
- Target tracking and surveillance.

To make changes to the reference path because of the above reasons during the mission execution, the path planning and perception system must work together. The reason being, the path planner needs waypoints to amend the reference path online (mid course mission execution) and the perception system provides these waypoints. This means for autonomous operation both the path planning and cooperative perception needs each other.

Figure 1.1 summarises the process of integration of path planning and cooperative perception.

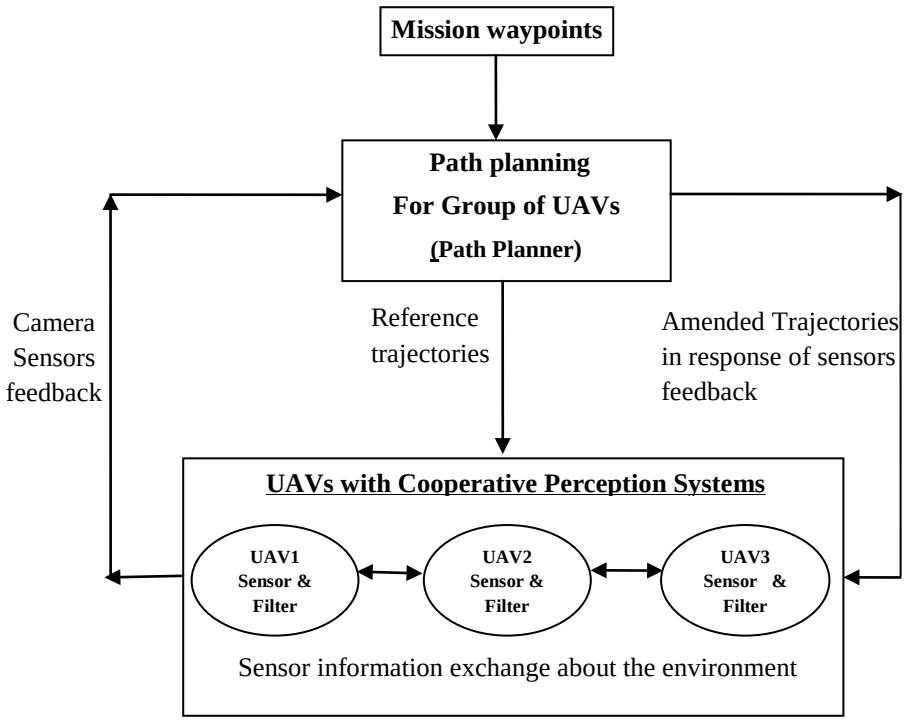

Figure 1.1: Schematic of interaction between path planning and cooperative perception.

1.2 Scenario

The following scenario was considered for this research.

A mission is planned to fly a group of UAVs safely from base B to target T as shown in figure 1.2. All the vehicles start from the base at the same time. During the flight the vehicles will avoid inter collision and collision with the obstacles. The UAVs may need to track a moving target during the mission.

The map of the operating environment is partially known. The absence of the complete knowledge about the environment necessitates perception. Therefore each vehicle in the group is provided with a camera to get the 3D position estimations of unknown obstacles or the motion estimates of moving target under surveillance cooperatively.

Initially the paths are planned for the UAVs offline by the path planning algorithm (Path Planner) for the whole of the journey on the basis of the available knowledge about the environment. The known obstacles are avoided during offline path planning.

After the UAVs are set off on these paths, the unknown obstacles are detected and located by the perception system. Once the obstacles are located they are avoided by making changes to initial paths by the path planner only if they are in conflict with the initial paths.

During these journeys there are occasions when the UAVs need to cooperate to track an object of interest. In such a case the path planner produce paths for cooperating UAVs to facilitate the cooperative tracking of the target.

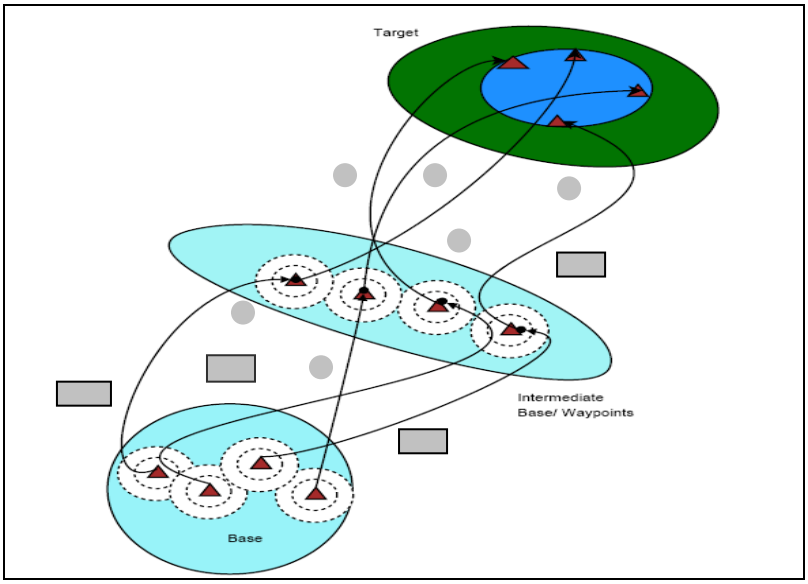

Figure 1.2: Reference trajectories of the swarm of UAVs in the environment with known obstacles (rectangles), unknown obstacles (circles).

1.3 Problem Definition

The problem definition of path planning for Swarm of the UAVs and cooperative perception is divided into the following sub problems:

1.3.1 Path planning.

1.3.2 Cooperative perception.

1.3.3 Integration of path planning and cooperative perception.

1.3.3.1 Obstacle voidance.

1.3.3.2 Target tracking.

Each sub problem is addressed separately in the following sections.

1.3.1 Problem Definition of Path Planning

In the context of mission planning, the path planning for Swarm of UAVs is defined as planning paths to guide a group of cooperating UAVs, during the mission, from some starting point to a finishing point such that the following constraints are fulfilled:

1. The paths are flyable. (The kinematics and dynamic constraints of the vehicles are observed).
2. All The paths are safe. (Keeping minimum safe distance between the neighbouring vehicles to avoiding inter-collision, and Keeping minimum safe distance from the obstacles to avoid collision with the obstacles)
3. The lengths of the paths are kept minimal.
4. The maximum objectives of the mission are achieved. (Each waypoint is visited).

These constraints must be observed by the path of every UAV in the group during the entire mission. The objective of this research is to design an algorithm (Path Planner), which could produce a path obeying all the above constraints. The details of the mathematical derivation of the path planning algorithm follow in chapter 2 and 3.

1.3.2 Problem Definition for Cooperative Perception

In this research the environment is partially known, therefore each UAV is assumed to be equipped with a perception system to detect the unknown obstacles within the certain range, during its travel. The perception system comprise of vision sensors (camera) and an estimation algorithm (Kalman Filter).

In the context of this research the cooperative perception is defined as acquiring location or motion information about some object (obstacle or object of interest to be tracked) in the scene from different viewpoints cooperatively through vision sensor onboard each vehicle, exchanging and fuse this information using filter mechanism, to get the 3D optimal estimates of location and motion of an object. The onboard camera is used as a sensor in this research to get the information about the location of an obstacle or state of motion of an object under surveillance.

The detail treatment of the problem the cooperative Perception follows in chapter 4 and 5.

1.3.3 Integration of Path Planning and Cooperative Perception

In this research the path planning and cooperative perception is integrated for application in reactive obstacles avoidance and target tracking and surveillance. Therefore it is further divided in to two sub problem:

1.3.3.1 Integration of Path Planning and Cooperative Perception for Obstacle Avoidance.

1.3.3.2 Integration of Path Planning and Cooperative Perception for Target tracking.

1.3.3.1 Integration of Path Planning and Cooperative Perception for Reactive Obstacle Avoidance

The known obstacles are avoided during the initial path planning offline while the unknown obstacles are detected, located and avoided during the mission execution on line. The perception system detects and locates the obstacle, and the path planner avoids it by planning a path around it using the information about the obstacle location obtained from the perception system. This is carried out systematically by:

1. Establishing safety region around the obstacle location (given by perception system) by an enclosing circle of radius R_s.
2. The UAV flight path is amended such that the intersection of the points of the enclosing circle and that of the UAV path (generated by path planner on line) is empty.

Chapter 6 is dedicated to the detail description of the problem of reactive obstacles avoidance and its solution.

1.3.3.2 Integration of Path Planning and Cooperative Perception for Target Tracking

When an object of interest is spotted by the onboard camera of any UAVs, a new waypoint is created on the basis of the current location of the target. The path planner plans two paths for the two nearest UAVs to guide them to locations suitable for tracking the object of interest. The concerned UAVs are positioned to these suitable geographical locations and orientations for cooperative tracking and surveillance by following these paths. After positioning the UAVs to suitable locations, the path planner continues to plans successive paths for the cooperating UAVs as the target moves, such that the following constraints are imposed to ensure the cooperative tracking.

1. The object of interest must belong to the union of the fields of view of onboard cameras (of cooperating UAVs) all the times while moving on their

respective paths. In other words the target is chased as it moves such that it remains in the field of view of any of the two UAVs at any time.

2. The cooperating UAVs must stay in communication range to ensure information exchange about the motion of the target.

The path planning provides the guidance for the UAVs to maintain close target proximity and the cooperative perception system locate the target in 3D. Therefore by this integration of path planning and cooperative perception the target is tracked.

Chapter 7 describes the problem of target tracking and surveillance in detail.

1.4 Solution Approach

The solution approach to the problem of path planning, the problem of cooperative perception and the problem of the integration of path planning and cooperative perception is discussed here briefly. The details of the solution will follow in separate chapters later.

Fixed wing UAVs with certain dynamic constraints are assumed to be employed in a mission in this research. These UAVs have some limitations, for example they can not stop mid flight or cut sharp turns. They can not reverse their directions of flight and they can not fly below certain speed. Therefore the paths to be flyable by these UAVs must be continuous curvature and tangent wise. These paths must also obey the kinematic and dynamic constraints of the UAVs. Furthermore the flyable paths must be safe i.e. collision free. The flyable and safe paths (feasible path) for the UAVs are produced by the Pythagorean Hodograph (PH) curves in this research. The PH curves are parametric polynomial whose first derivatives obey the Pythagorean principle. The PH curves produce paths of exact length with tangential and curvature continuity. These paths are made to obey the kinematic constraints of the UAVs by manipulating their curvatures and torsions. Parametric polynomials of degree three are the least degree polynomials which can have non trivial Pythagorean hodographs. These polynomials are called Cubic PH. The cubic PH curves are very simple to calculate but very inflexible to manipulate to achieve various configurations. PH curves of degree five (quintic) are the least degree PH curves which have the flexibility required to achieve any configuration. PH quintic is a good compromise between computational complexity and the manipulating flexibility. In this research PH quintic are used to solve the problem of path planning. Chapter 2 and 3 have the detail of the solution.

The problem of cooperative perception is to deal with erroneous target position acquisition due to the noises entailed with combined use of the camera to perceive the events taking place in a dynamic environment and GPS sensors to acquire the positions of UAVs. The combination of the two sensors produces erroneous positions due to the associated noise while locating the target in 3D. This error in target position is reduced by optimization. A target was tracked using at least two cameras on two different UAVs. Using the observed bearings of the target from each UAV, an estimate of the

absolute target position was obtained by minimizing the errors in distance from the estimate to each measurement. The optimal target position estimate thus obtained is then used as a measurement input to a simple linear Kalman filter that uses the target location as the state variables $[x, y, z]$. Chapter 5 have the detail of the solution.

The integration of path planning and cooperative perception is the prerequisite of the operation of locating and avoiding an obstacle, and tracking an object of interest.

In case of obstacle avoidance, the cooperative perception was used to locate an obstacle and the path planning is used to produce an evasive path on line to avoid it.

In case of surveillance and tracking of ground target (in a known terrain) a continuous unbroken vigilance is needed. This is not possible with single UAV because there is a speed difference between the UAV and the ground target, which results in target being lost from the field of view of camera. Thus the UAV needed to be manoeuvred such that it keeps up with the target. In this research the continuous unbroken vigilance is provided by two manoeuvring UAVs such that at least one of the UAVs has the target in its field of view at any time. The information is made available to the other UAVs through wireless communication. The PH path planning is used to plan the guidance manoeuvres to follow the target. And the cooperative perception is used to locate the target in 3D as it is chased. In other words the target is chased by the path planning in such a way that the continuous coverage of the target is made available to the perception system for the purpose of tracking.

1.5 Thesis Contribution

- The Pythagorean Hodograph curves are calculated in 2D as a starting point and then extended to 3D to suit the real mission. The quaternions were used to calculate the PH paths in 3D.
- The iterations in the process of imposing the kinematic constraints is eliminated by taking the derivatives of the UAV path at the initial and final poses comparable to maximum curvature and torsion attainable by the UAV. This reduces the feasible path calculation time by a considerable amount.
- Frenet frame and Rotation Minimising Frames were calculated for PH paths. These frames are used as camera reference frame in perception system to locate an object in the scene.
- A vision based algorithm for dynamic target localisation by two cooperative UAVs travelling on their respective paths is obtained, which is then used in various UAV applications (reactive obstacle avoidance, target surveillance and tracking) later in the research. The algorithm uses optimisation technique to get the optimal position estimates of the target.

- A new classification of PH paths has been specified. All the possible Pythagorean hodograph paths that can be calculated from the interpolation of initial and final poses are divided into two classes of paths. One class includes all the paths that undergo sign change of curvature, they are given the name of 'S' shape PH paths. The other class includes all the paths that do not undergo sign change of curvature; they are given the name of 'C' shape PH paths. The 'C' shape paths are used efficiently in many applications later in this research.
- A novel approach of integrating the path planning and the cooperative perception system of the swarm has been used for the first time for obstacle locating and avoidance, target surveillance and tracking.
- An efficient algorithm based on PH paths for reactive obstacle avoidance in dynamic environments is derived. This algorithm uses the 'C' shape PH paths. The algorithm is very successful in avoiding obstacle of any size interrupting with the path of UAV.
- The obstacle avoidance algorithm is able to avoid a cluster of small obstacles, if there is no free corridor among them, by assuming the aggregation to be one big obstacle, by virtue of its ability to deal with the obstacle of any size. The search for free corridors among the obstacles however needs some further work.
- Successful target surveillance and tracking is achieved by the integration of path planning and perception. The PH path planner is used to plan the chasing path for the UAV to keep the moving ground target under surveillance. The 'C' shaped PH paths are used skilfully to achieve close target proximity and vision based tracking algorithm is used to track the ground target.

1.6 Disseminations, Presentations from the Thesis

Conference Papers

1. M. A. Shah, N. Aouf. *Dynamic Cooperative Perception and Path Planning for Collision Avoidance,* 6th International Symposium on Mechatronics and its Application ISMA' 09 March 2009 Sharjah Dubai.
2. M. Shah, and N. Aouf. *3D Cooperative Pythagorean Hodograph Path Planning and Obstacle Avoidance for Multiple UAVs,* IEEE International Conference on Cybernetic Intelligent Systems, September 09 Birmingham, UK.
3. M. Shah, A. Tsourdos, D. James, N. Aouf. *Collision Avoidance in the Framework of Pythagorean Hodograph Curves,* 7th International Conference on Informatics in Control Automation and Robots, ICINCO' June 2010 Funchal Madeira-Portugal.

4. M. Shah, A. Tsourdos, D. James, N. Aouf. *On the Definition of Curve's Frame at the Points of inflection* 18th Symposium on Automatic Control in Aerospace ACA 2010, Sept 2010 Nara, Japan.
5. M. Shah, A. Tsourdos, D. James, N. Aouf. *Target Tracking Using Cooperative Perception and Multi-Sensor Fusion*, 18th Symposium on Cognitive Sensor COGIS 2010, Nov 2010 Crawley UK.

1.7 Organisation of Thesis

The thesis can be divided in three major parts. The first part is about path planning, this part comprises chapter 2 and chapter 3. The second part is about cooperative perception. This part comprises chapter 4 and chapter 5. The Third part is about the integration of path planning and cooperative perception. This part comprises chapter 6 and chapter 7. Chapter 1 is the general introduction and chapter 8 is about the conclusion and the possible future work.

Chapter 1 is the general introduction of the research project been undertaken. It gives the problem statement, the need for integration of path planning and cooperative perception. It also describes in brief the solution approaches that will be followed in later chapters and summarises the contribution of the thesis.

Chapter 2 gives the detail literature review of the path planning, the problem statement of path planning and the derivation of Pythagorean Hodograph path in 2D. Pythagorean Hodograph has been used to produce the feasible paths for the UAVs in the group in 2D. Pythagorean Hodograph paths are produced by the Hermit interpolation of the initial and final configurations. First the expression for the PH cubic is derived and then the PH quintic is derived to accommodate the flexibility of pose manipulation. The simulation results for 2D PH paths are given at the end of the chapter.

Chapter 3 extends the 2D Pythagorean Hodograph into 3D to emulate the real mission scenario. Pythagorean Hodograph has been used to produce feasible paths for the group of UAV in 3D. Pythagorean Hodograph paths are produced by the Hermit interpolation of the initial and final configurations, but this time Quaternions algebra is used for derivation of PH paths in order to overcome the burden of mathematical derivation by normal algebra. First the expression for the PH cubic is derived and then the PH quintic is derived to accommodate the flexibility of pose manipulation. The simulation results for 3D PH paths are given at the end of the chapter.

Chapter 4 is about the definition of the reference frame for onboard camera sensors of the perception system. The camera reference frame is derived from the UAVs' paths. First the Frenet frame is calculated at each point on the UAV path and then the Rotation Minimizing Frame (RMF) is calculated by minimizing the rotation of these Frenet frames. The RMF is then used as local camera reference frame.

Chapter 5 describes the perception system of the swarm. The components of the vision based perception system have been described, and the problem at hand is mathematically formulated. The solution to the problem is approached in two ways. First a simple solution is tried as a starting point to prove the principal. In this simple solution the problem of locating the target dynamically by two UAVs travelling on their respective PH paths is formulated as locating the target from multi-view geometric reconstruction ignoring all the associated sensors noise. After arriving at a satisfactory solution in the first case the problem is redefined by taking into account all the associated sensors noise to emulate the real situation. This time the problem of locating the object is formulated as an optimisation problem in which the optimized position estimates are obtained by minimizing the uncertainties in the observed position of the target through the individual sensors on board each vehicle. The simulation results are given at the end of the chapter to show the validity of the technique.

Chapter 6 describes the mid course reactive collision avoidance with the pop up obstacles. In this chapter all the possible PH paths given by Hermit interpolation of any two given poses are analysed and classified in to 'S' shape and 'C' shape categories for obstacle avoidance purposes. A method has been proposed which successfully avoid an obstacle lying at a point on the path where the traditional method of curvature manipulation of obstacle avoidance fails. The proposed method consists of four steps: First the obstacle is located by the camera and then described by an enclosing circle/ellipse. Secondly the obstacle is avoided by introducing a new waypoint (intermediate pose) in between the initial and final pose. Third, the three poses are connected by two Pythagorean hodographs curves of minimum bending energy such that each curve connecting the two consecutive poses join each other smoothly. Fourth, if the path is still not clear of the obstacle, the curvature of the part of the path which is not clear of the obstacle is manipulated until the obstacle is avoided. The simulation results at the end of the chapter shows the efficiency of the proposed method.

Chapter 7 describes cooperative target surveillance and tracking. In this chapter Pythagorean Hodograph paths (PH) are adopted for surveillance of ground target. The PH paths are used to guide the UAVs to chase a moving ground target. Two 'C' shape PH curves are combined to cover each instant of position estimation when the target is detected. In succeeding instance of the target motion (change of position) two more 'C' shaped curves are used to cover the motion such that it joins the previous path smoothly. This process is repeated until the surveillance operation is completed. The two 'C' shape curves which cover the same motion segment share the intermediate pose, such that the intermediate pose is the final pose for the first 'C' shape curve and the initial pose for the second 'C' shape curve. The intermediate poses at various instances is the target position estimated in time. The proposed method can accommodate the relative moving speed differences of the target and the UAVs. Several paths are used for moving target to test the proposed methods of target surveillance. The simulation results show the efficiency of the proposed method.

Chapter 8 describes the conclusion and further work which can be an extension of the current work and which can play vital role in making the PH paths a full fledge framework for path planning required for various autonomous UAVs operation.

Chapter 2

Cooperative Path Planning for Multiple UAVs Using Pythagorean Hodographs in 2D

The path planning for the UAV is the most challenging part of the UAV mission. In the UAV context the path planning is the process of planning a flyable and safe optimal path from start to goal point. The fixed wing UAVs can not fly below certain speed. It can not stop or reverses during its flight neither can cut sharp corners. All these constitute dynamics and kinematics constraints, which should be respected while planning a path for the UAVs. A flyable path means a path with continuous tangent and curvature profiles obeying the kinematic and dynamic constraints of the vehicle. A safe path means a path which is free from collisions. A path which is flyable and safe is called a feasible path. A feasible path could be easy or difficult (if not impossible) to trace by the UAVs and could either be short or long. The bending energy of a path is the measure of ease with which a path could be traced by the UAV. The time during which the UAV traverses a complete path with certain constant speed is the measure of length of a path. Together, these attributes of a path are the measure of its optimality. Therefore optimality of a path is a trade off or a finer balance of these attributes. In path planning these tradeoffs are addressed using different techniques.

In literature the optimal path planning is tried in a variety of ways depending upon the type of constraints imposed during the mission by vehicle dynamics, by the operating environment, by task being undertaken and above all by the objectives of the mission. Techniques like Voronoi Diagram, Visibility Graph, Fussy Logic, Genetic Algorithms, Potential Field, Linear Programming, and Cubic Splines etc are used for path planning. Differential Geometry techniques like Dubins path and Clothoids arcs are also used for path planning. Recently Pythagorean hodograph has been used for path planning in few papers. In the subsequent section the literature review of the path planning is summarised.

2.1 Review of Path Planning

In literature different approaches for Path Planning have been proposed. Some of these approaches were studied during the literature review with their advantages and disadvantages. The summary of this review is presented in the following paragraphs.

In [1, 2] Voronoi Diagrams are used for path planning. In Voronoi diagrams the threats and obstacles are represented by constructing cells around them. The safe paths are

around the boundaries of these cells. The shortest path is figured out by the use of Dijikstra algorithm [3]. Generally the distance is the only factor considered in choosing the optimal path, therefore the optimal paths been found by classical Voronoi techniques are unrealistic for UAVs [2]. These could be the shortest and safest path for ground robots, but for UAVs these are unlikely to be considered as feasible paths because they do not consider the kinematics and dynamics constraints of the vehicles. Again, as we are considering the dynamic environment, so modifying the path in real time according to the changing environment would be an essential part of the path planning. In this case the Voronoi diagram method becomes an expensive option computationally. Therefore it is dropped as a candidate method for path planning of UAVs in dynamic environments.

The second approach is the visibility graph method [4, 5] in which a visibility graph is constructed around polyhedral obstacles. The vertices of the polygon are the nodes of the map. Two nodes of the visibility graph share an edge if their corresponding vertices are with in line of sight of each other. These edges are combined to find numbers of paths in visibility graph method. The optimal path is selected among these paths using shortest path algorithms. The resulting optimal path is a polylines consisting of multiple angles and sharp corners. This path is then smoothened to make it flyable. This method is not suitable for dynamic environment, because when the start or goal points are changed the visibility graph has to be recomputed, which is very expensive computationally [6]. So the visibility graph method for path planning in case of the UAVs is dropped because of its computational inefficiency for dynamic environment.

The third worth mentioning approach is optimization techniques (like Linear Programming, Mixed Integer Programming and Non Linear Programming) [7, 8, 9, 10, 11, 12] in which an objective function is synthesised based on certain parameters and the goal is to optimize the objective function under certain constraints. This method can produce flyable paths if the path length is optimized under the curvature and safety constraints. But these are not suitable options for online path planning in changing environments because of their multitude nature at every decision point, and the heavy computation entailed with it. Also their tendency to produce local optimal solution makes them an unsuitable option.

Genetic Algorithms (GAs) are useful algorithms for optimising solutions to problems; especially those that are analytically intractable. They are inspired, as the name suggests, by the biological concepts of genetics and evolution. GA uses the principle of evolution to refine and Optimise designs whose parameters interact in a complex manner. Individuals representing different potential solutions are preferentially selected according to their fitness and pass on their “genes”, i.e, characteristics, to future generations. Mating takes place between these individuals with the hope that by sharing the characteristics of successful individuals, even fitter individuals can be created. Mutation also occurs to inject new genes into a population. As in nature, most mutations are harmful but the occasional beneficial one can help

improve the fitness of the individuals it affects, i.e. find better solutions. These algorithms use some of the concepts of evolutionary theory, and provide an effective way of searching a large and complex solution space to give close to optimal solutions in much faster times than random trial-and-error. They are also generally more effective at avoiding local minima than differentiation-based approaches. A significant number of researchers have applied Genetic Algorithms to the task of optimising some objective function in path planning. Many different approaches to this task of optimization have been taken [13, 14, 15, 16]. But Optimization based on these techniques (Evolutionary Algorithm, Random Search and Heuristics) have probabilistic nature, and can not guarantee a precise optimal solution. They can only achieve a close approximation to optimal solution. Precision and certainty are crucial in critical missions involving close proximity flights, because decision based on uncertain information could lead to disastrous consequences.

Dubins sets are used for path planning for car like robots. Dubins sets [17, 18, 19] are composite paths consisting of two circular arcs joined by a line, or three consecutive tangential circular arcs, or a subset of either of these two. The line is the shortest distance between the two points in linear motion case and the circular arc is the shortest distance in angular motion case. The circular arcs are described in terms of radius which could be constrained to obey the dynamics of the UAV (curvature bound). So the Dubins sets gives us an overall shortest path obeying the minimum turning radius to start with, even without considering the other path planning techniques. Dubins paths give the shortest path, but it suffers from the problem of curvature continuity, i.e. the curvature jumps from a non-zero value to zero curvature at the point where the circular arc joins the tangent line. This will result in a jerky path, which is very undesirable in dynamic environment and close proximity flight. The curvature continuity is the requirement of smooth flyable path.

Clothoids [20, 21, 22] are used to achieve paths with curvature continuity. The Clothoids are the curves whose curvature varies linearly along its arc length. The problem with Clothoids is the evaluation of curve length which does not yield a closed form solution.

Cubic splines and Dubin sets are used for collision avoidance [95-97]. A method is presented for solving path planning problem by bending smooth cubic splines to avoid obstacle collision [95]. The use of cubic splines guarantees continuous vehicle heading and turning angle which are compulsory for the UAVs trajectories. Monte Carlo simulation is used to validate the performance of the algorithm. A smooth path planning algorithm for UAVs in cluttered natural environment is presented [96]. A Rapid Exploring Random Tree (RRT) is used to produce piece wise collision free linear path amongst the obstacles, and cubic Bezier spiral curves are used to generate a continuous curvature path which satisfies the minimum curvature constraint. The inverse kinematics and Dubins set are used for path planning and obstacle avoidance for the case of steerable needle used in medical applications [97]. In 3D the needle can be

interpreted as airplane with constant speed and pitch, zero yaw and controllable roll angle.

Polynomials are used for path planning to avoid the obstacle [98-100]. A planning methodology based on polynomials for non holonomic mobile manipulator in the presence of obstacle is developed [98]. To find collision free path with smaller length, two techniques are used. The first uses the intermediate path points and the second exploits the periodicity of the trigonometric function involved. Polynomials are used for path planning and obstacle on the path is avoided by increasing the order of the polynomial that is used in planning the initial trajectories [99]. An efficient Bezier curve based approach for the path planning in a multi agent robot soccer system is presented [100]. An obstacle avoidance scheme is incorporated for dealing with the stationary and moving obstacle. When the robot is approaching a moving obstacle in the field, it is decelerated and deviated to another Bezier path leading to the estimated target position. The radius of the curvature of the path at its end point is determined from the known terminal velocity constraint of the robot.

In [23, 24, 25, 26] Pythagorean Hodograph polynomials are used for path planning.

This research also uses planner and spatial Pythagorean Hodograph curves of degree five (PH quintic) in Bezier form for optimal path planning. The motivation of using the Pythagorean Hodograph is because of their computational suitability, which makes them very well suited for path planning in dynamic environments. Their ability to yield a guaranteed curvature and tangential continuity is making them a favourite option for UAVs which require curvature and tangential continuity for their paths to be flyable. Moreover their ability to yield paths of exact length is seen to be very useful aspect to ensure the safety and coordination of the UAVs. Also for PH curves there exists a systematic procedure which yields a path with minimum bending energy. The advantage of the minimum bending energy path is that it can be followed by the UAVs very conveniently. The PH curves also have the flexibility of achieving any pose if higher than order three PH curve is used. Their analytic and predictive nature makes them a suitable option in dynamic environments. In this research PH curves are used successfully for reactive obstacles avoidance and target tracking. Thus a full fledge PH curves based frame work has been established covering a full spectrum of UAVs operations which are using path planning (reactive collision avoidance, target tracking, continuous unbroken surveillance of certain region or a target, convoy escorting, automatic navigation, simultaneous localisation and map building etc).

In this chapter, Pythagorean Hodograph in 2D is used as the starting point for path planning. Later the work is extended to 3D Pythagorean hodograph. The Pythagorean Hodograph based path planning and cooperative perception is integrated to apply effectively in various UAVs applications like sense and avoid, and target tracking and surveillance cooperatively by multiple UAVs. During the course of research the Pythagorean hodograph curve are classified and composite Pythagorean hodograph is

introduced to suit the application of mid course obstacle avoidance and continuous target surveillance.

2.2 Definition of Path Planning

In the context of mission planning and execution, the path planning for Swarm of UAVs is defined as planning paths to guide a group of cooperating UAVs, during the mission, from some starting point to a finishing point such that the following constraints are fulfilled:

1. The paths are flyable. (The kinematics and dynamic constraints of the vehicles are observed).
2. All the paths are safe (keeping minimum safe distance between the neighbouring vehicles to avoid inter-collision, and keeping minimum safe distance from the obstacles to avoid collision with the obstacles)
3. The bending energies and lengths of the paths are kept minimal.
4. The maximum objectives of the mission are achieved (each waypoint is visited).

These constraints must be observed by the path of every UAV in the group during the entire mission. The subject of this chapter is to design a path planning algorithm (Path Planner) based on PH curves, which could produce a path obeying all the above constraints.

2.3 Mathematical Statement of the Problem of 2D Path Planning

Consider *n* homogenous UAVS deployed in a mission. All the UAVs leave the base at the same time. The base and the goal can be connected through a set of waypoints. The problem is simplified by taking the base and the goal as the two successive waypoints. Each UAV is considered to be at the centre of a safety circle with a radius R_s . It should be noted that in this work all the obstacles are not known a *priori*, therefore each UAV is equipped with a vision sensor, which detects the unknown pop-up obstacles within the sensor range, as it proceeds along the reference trajectory in the environment.

To guarantee the safety of the UAVs in the group, the safety circles for all UAVs must be disjoint. For any two UAVs travelling on their respective paths, if the intersection of their safety circles is empty then the paths are safe to fly, otherwise the paths have to be re-planned to avoid inter collision between the vehicles. This safety criterion is also valid for the obstacles avoidance.

The start and goal points for the UAVs are specified by their initial and final poses. The pose is a combination of position and orientation of the UAV. In 2D the position of the

UAV is denoted by position coordinates $p(x, y)$ of centre of mass of UAV while tracing the reference trajectory. In this chapter, since path planning at constant altitude is considered therefore the orientation of UAV is specified by its heading angle θ only. Thus the general pose of the UAV in 2D is $Pose(x, y, \theta)$.

Let the starting pose of the *ith* UAV is $Pose_{si}(x_{si}, y_{si}, \theta_{si})$ and the final pose of the same UAV is $Pose_{fi}(x_{fi}, y_{fi}, \theta_{fi})$ then the path of the UAV is defined as the parametric curve $r_i(t) = [x(t), y(t)]$ with curvature κ_i connecting the initial and final pose:

$$Pose_{si}(x_{si}, y_{si}, \theta_{si}) \xrightarrow{r_i(t)} Pose_{fi}(x_{fi}, y_{fi}, \theta_{fi}) \tag{2.1}$$

$r(t)$ in this case is Pythagorean Hodograph curve in parametric form subjected to the following constraints:

1. $$|\kappa_i(t)| \le \kappa_{\max} \tag{2.2}$$

   Where $\kappa_{\max}$ is the maximum curvature attainable by the UAVs.

2. The vehicles must keep safe distance to avoid inter collision while tracing their corresponding trajectories. This can be written in mathematical form as follow:

   $$\boldsymbol{R}_{si} \cap \boldsymbol{R}_{sj} = \varphi \tag{2.3}$$

   That is, the intersection of the safety circles of radius R_{si} and R_{sj} corresponding to *ith* and *jth* UAVs must be empty.

   Or equivalently the minimum distance d between the two vehicles must be:

   $$d \ge 2R_s$$

   Diagrammatically this can be represented as follow:

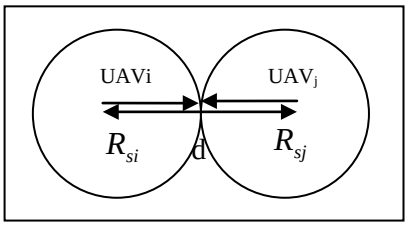

Figure 2.1: Non intersecting safety circles of two neighbouring UAVs

3. The vehicles must keep safe distance to avoid collisions with known obstacles while tracing their corresponding trajectories. This can be written in mathematical form as follow:

   $$d(obstacle, UAV) \ge R_{obstacle} + R_s$$

Where $d(obstacle, UAV)$ is the distance between the centre of the circle enclosing the obstacle and the centre of the safety circle of the UAV. $R_{obstacle}$ is the radius of the circle enclosing the square obstacle. Diagrammatically this can be represented as follow:

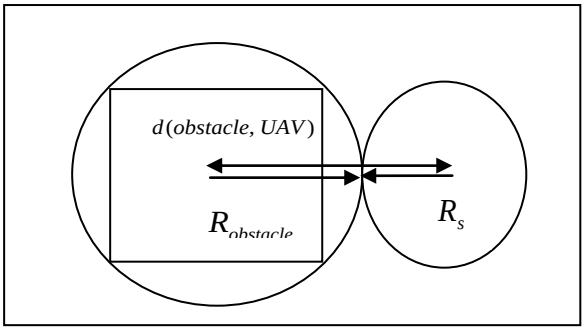

Figure 2.2: Non intersecting enclosing circles of UAVs and square obstacle

The solution to the problem of 2D path planning is approached through the application of planer Pythagorean Hodograph curves which is discussed in the following section in detail.

2.4 Pythagorean Hodograph in 2D

Path planning using Pythagorean Hodograph in 2D is the starting point of this research. The Pythagorean Hodograph [27, 28, 29, 32] offers a perfect solution to the path planning problem. They are the first derivatives of parametric polynomials, which satisfy the Pythagorean condition. The first derivative in time domain is the velocity vector which is parallel to the tangent to the curve. If $r(t)$ represents the curve and $s(t)$ represents the length of the curve from instant t_1 to instant t_2 then:

$$s(t) = \int_{t1}^{t2} |r'(t)|\, dt \tag{2.4}$$

$$= \int_{t1}^{t2} \sqrt{x'(t)^2 + y'(t)^2}\, dt \tag{2.5}$$

Where $x(t)$ and $y(t)$ are the parametric components of the curve, t is the parameter of the curves and $x'(t)$, $y'(t)$ are the derivatives (hodographs) of $x(t)$ and $y(t)$. For calculating the path length equation (2.5) needed to be solved. The presence of the square root in the integrand may not result in closed form solution, and a numerical quadrature is needed which yields a numerical approximation to the true solution. To avoid this situation we need to get rid of the square root in equation (2.5). The radical sign can be eliminated from equation (2.5) if:

$$\sigma^2(t) = x'^2(t) + y'^2(t) \tag{2.6}$$

This is Pythagorean law, in which the polynomial σ is the hypotenuse and the hodographs $x'(t)$ and $y'(t)$ are the two sides of the right angle triangle. Since $x'(t)$ and $y'(t)$ are Pythagorean hodographs. Therefore equation (2.5) can be written as:

$$s(t) = \int_{t1}^{t2} |\sigma(t)|\, dt \tag{2.7}$$

This yields a closed form solution.

We can produce the Pythagorean Hodograph in parametric form by selecting two polynomials $u(t), v(t)$ and $w(t) = 1$ [27] such that:

$$x'(t) = w(t)\left[u^2(t) - v^2(t)\right] \tag{2.8}$$

And
$$y'(t) = 2\,w(t)\,u(t)\,v(t) \tag{2.9}$$

Therefore:

$$\sqrt{x'^2(t) + y'^2(t)} = \sqrt{[w(t)\{u^2(t) - v^2(t)\}]^2 + [2w(t)u(t)v(t)]^2}$$

$$= \sqrt{[w(t)\{u^2(t) + v^2(t)\}]^2}$$

$$\sigma(t) = w(t)[u^2(t) + v^2(t)] \tag{2.10}$$

The commonly used non trivial polynomials in literature giving Pythagorean hodographs, are polynomial of degree three (Cubic) and polynomial of degree five (Quintic). The cubic is simple to compute but less flexible to manipulate, because it has only one shape freedom, while the quintic is less simple to compute but more flexible to manipulate. Quintic is best suited for the purpose of path planning because more flexibility is needed for poses manipulation.

As a starting point of this chapter, expressions for PH cubic is derived first, and then proceeded to derive the expression for PH quintic to accommodate the challenges of pose manipulation. The following sections give the detail of the process of deriving the expressions for PH cubic and quintic.

2.4.1 Derivation of Cubic Pythagorean Hodograph in 2D

A cubic Pythagorean Hodograph is produced by selecting two linear polynomials $u(t), v(t)$ and $w(t) = 1$ such that conditions (2.8), (2.9) and (2.10) are satisfied [27]:

The first linear polynomials $u(t)$ in Bernstein form:

$$u(t) = u_0 t + u_1(1 - t) \tag{2.11}$$

Similarly the second linear polynomial $v(t)$ in Bernstein form

$$v(t) = v_0 t + v_1(1 - t) \tag{2.12}$$

Squaring equation (2.11) and (2.12)

$$u^2(t) = u_0^2 t^2 + u_1^2 (1-t)^2 + 2u_0 u_1 t(1-t) \tag{2.13}$$

$$v^2(t) = v_0^2 t^2 + v_1^2 (1-t)^2 + 2v_0 v_1 t(1-t) \tag{2.14}$$

Putting equation (2.13) and (2.14) in equation (2.8):

$$x'(t) = \left[u^2(t) - v^2(t)\right] \tag{2.15}$$

$$x'(t) = \left[u_0 t + u_1(1-t)\right]^2 - \left[v_0 t + v_1(1-t)\right]^2$$

$$x'(t) = (u_0^2 - v_0^2)t^2 + (u_1^2 - v_1^2)(1-t)^2 + 2(u_0 u_1 - v_0 v_1)t(1-t) \tag{2.16}$$

Putting equation (2.11) and (2.12) in equation (2.9):

$$y'(t) = 2u(t)v(t)$$

$$y'(t) = 2\left[u_0 t + u_1(1-t)\right]\left[v_0 t + v_1(1-t)\right]$$

$$y'(t) = u_0 v_0 t^2 + u_1 v_1 (1-t)^2 + (u_0 v_1 + u_1 v_0)t(1-t) \tag{2.17}$$

Taking Integrals of both sides of equation (2.16), (2.17):

$$x(t) = \int_0^t \left[(u_0^2 - v_0^2)t^2 + (u_1^2 - v_1^2)(1-t)^2 + 2(u_0 u_1 - v_0 v_1)t(1-t)\right] dt \tag{2.18}$$

$$y(t) = \int_0^t \left[u_0 v_0 t^2 + u_1 v_1 (1-t)^2 + (u_0 v_1 + u_1 v_0)t(1-t)\right] dt \tag{2.19}$$

Now since the indefinite integrals of the Bernstein is given by [27, 28]:

$$\int \left[\binom{n-1}{k}(1-t)^{n-1-k} t^k\right] dt = \frac{1}{n}\sum_{j=k+1}^{n}\binom{n}{j}(1-t)^{n-j} t^j$$

$$= \frac{1}{n}\sum_{j=k+1}^{n}\binom{n}{j}(1-t)^{n-(k+1)} t^{k+1}$$

Where $k = 0,.....n-1$, $\binom{n}{j} = \frac{n!}{(n-j)!\, j!}$ and $t \in [0\ \ 1]$.

Therefore equations (2.18) and (2.19) respectively yield [27]:

$$x(t) = \sum_{k=0}^{3} x_k \binom{3}{k}(1-t)^{3-k} t^k \tag{2.19}$$

$$y(t) = \sum_{k=0}^{3} y_k \binom{3}{k}(1-t)^{3-k} t^k \tag{2.20}$$

Equation (2.19) and (2.20) give the expression for Pythagorean Hodograph cubic in 2D $r(t) = [x(t),\ y(t)]^T$ with control points $P_k = (x_k, y_k)$ in compact form as follow:

$$r(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \sum_{k=0}^{3} P_k \binom{3}{k} (1-t)^{3-k} t^k \tag{2.21}$$

Where $k = 0,.....3$, $\binom{3}{k} = \frac{3!}{(3-k)!k!}$ and $t \in [0\ \ 1]$.

Equation (2.21) on expansion gives:

$$r(t) = P_0(1-t)^3 + 3P_1(1-t)^2 t + 3P_2(1-t)t^2 + P_3 t^3 \tag{2.22}$$

Where the control points $P_k(x_k, y_k)$, $k = 0,1,2,3$ for PH cubic in 2D [27] are given below:

$$P_1 = P_0 + \frac{1}{3}(u_0^2 - v_0^2, \quad 2u_0 v_0) \tag{2.23}$$

$$P_2 = P_1 + \frac{1}{3}(u_0 u_1 - v_0 v_1, \quad u_0 v_1 + u_1 v_0) \tag{2.24}$$

$$P_3 = P_2 + \frac{1}{3}(u_1^2 - v_1^2, \quad 2u_1 v_1) \tag{2.25}$$

Where P_0 is constant of integration.

Equation (2.22) in matrix form can be represented as follow:

$$r(t) = \begin{bmatrix} (1-t)^3 & 3t(1-t)^2 & 3t^2 & 3(1-t)t^3 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Implies:

$$r(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix} \tag{2.26}$$

This is the matrix form of PH cubic in 2D. If the control points of the curve are known then equation (2.26) can be used to calculate the planer PH curve.

2.4.1.1 Specifying the Control Points in Case of 2D PH Cubic

Control points P_0, P_1, P_2, P_3 for PH curve can be specified from the known initial and final poses of the UAV. For this purpose the expression for the derivative of PH curve is needed.

The derivative of the PH curve $r'(t)$ in compact form is given by the following equation as described in [27, 28]:

$$r'(t) = n\sum_{k=0}^{n-1} \Delta P_k \binom{n-1}{k} (1-t)^{n-1-k} t^k \tag{2.27}$$

Where $\binom{n-1}{k} = \frac{(n-1)!}{(n-1-k)!k!}$, $t \in [0\ 1]$ and ΔP_k denotes the k[th] forward path difference $(P_{k+1} - P_k)$ for $k = 0,....,n-1$.

For cubic PH curve, $n = 3$ in equation (2.27) therefore:

$$r'(t) = 3\sum_{k=0}^{2} \Delta P_k \binom{2}{k} (1-t)^{2-k} t^k \tag{2.28}$$

Equation (2.28) on expansion gives:

$$r'(t) = 3(P_1 - P_0)(1-t)^2 + 3(P_2 - P_1)(1-t)t + 3(P_3 - P_2)t^2 \tag{2.29}$$

Putting $t = 0$ in equation (2.22) and (2.29) gives:

$$r(0) = P_0 \tag{2.30}$$

$$r'(0) = 3(P_1 - P_0) \tag{2.31}$$

Similarly putting $t = 1$ in equation (2.22) and (2.29) gives:

$$r(1) = P_3 \tag{2.32}$$

$$r'(1) = 3(P_3 - P_2) \tag{2.33}$$

Since the initial and final configurations for *ith* UAV are known therefore the starting point $p_{si} = (x_{si}, y_{si})$ and the derivative at the starting point $dp_{si} = (dx_{si}, dy_{si})$ is known. Also the final point $p_{fi} = (x_{fi}, y_{fi})$ and the derivative at the final point $dp_{fi} = (dx_{fi}, dy_{fi})$ are known:

$$r(0) = p_{si} = (x_{si}, y_{si}) \tag{2.34}$$

$$r'(0) = 3(P_1 - P_0) = (dx_{si}, dy_{si}) \tag{2.35}$$

$$r(1) = P_3 = (x_{fi}, y_{fi}) \tag{2.36}$$

$$r'(1) = 3(P_3 - P_2) = (dx_{fi}, dy_{fi}) \tag{2.37}$$

Equations (2.35) and (2.37) can be rearranged as:

$$P_1 = P_0 + \frac{1}{3}(dx_{si}, dy_{si}) \tag{2.38}$$

$$P_2 = P_3 - \frac{1}{3}(dx_{fi}, dy_{fi}) \tag{2.39}$$

Equations (2.34), (2.36), (2.38) and (2.39) give all the control points.

Now if the equations (2.34), (2.36), (2.38) and (2.39) are subsituted in equations (2.23)-(2.25), the coefficients u_0, u_1 and v_0, v_1 of the constituents' curves represented by equations (2.11) and (2.12) which form the planer PH curve can also be determined.

In the next section the expression for 2D Pythagorean Hodograph Quintic is derived which is the subject of path planning part of this research.

2.4.2. Derivation of Quintic Pythagorean Hodograph in 2D

The advantage of cubic PH is its computational simplicity and the disadvantage is its poses manipulation inflexibility. The polynomials of degree five (quintic) are the polynomial of least degree which has the flexibility good enough to be manoeuvred to achieve various configurations. Quintic PH curve can be obtained by selecting two polynomials $u(t)$ and $v(t)$ to be of second degree [28]. Thus the first quadratic polynomials $u(t)$ in Bernstein form:

$$u(t) = \sum_{k=0}^{2} u_k \binom{2}{k} t^k (1-t)^{(2-k)}, t \in [0 \;\; 1]$$

$$u(t) = u_0(1-t)^2 + 2u_1(1-t)t + u_2 t^2 \tag{2.40}$$

The second quadratic polynomials $v(t)$ in Bernstein form:

$$v(t) = \sum_{k=0}^{2} v_k \binom{2}{k} t^k (1-t)^{(2-k)}, \quad t \in [0 \;\; 1]$$

$$v(t) = v_0(1-t)^2 + 2v_1(1-t)t + v_2 t^2 \tag{2.41}$$

Squaring equation (2.40)

$$u^2(t) = {u_0}^2(1-t)^4 + 4u_0u_1(1-t)^3t + 2(u_0u_2 + {u_1}^2)(1-t)^2t^2 + 4u_1u_2(1-t)t^3 + {u_2}^2t^4 \tag{2.42}$$

Squaring equation (2.41)

$$v^2(t) = {v_0}^2(1-t)^4 + 4v_0v_1(1-t)^3t + 2(v_0v_2 + {u_1}^2)(1-t)^2t^2 + 4v_1v_2(1-t)t^3 + {v_2}^2t^4 \tag{2.43}$$

Putting equation (2.42) and (2.43) in equation (2.8) gives:

$$x'(t) = ({u_0}^2 - {v_0}^2)(1-t)^4 + 4(u_0u_1 - v_0v_1)(1-t)^3t + 2(u_0u_2 + {u_1}^2 - v_0v_2 - {v_1}^2)(1-t)^2t^2 + 4(u_1u_2 - v_1v_2)(1-t)t^3 + ({u_2}^2 - {v_2}^2)t^4 \tag{2.44}$$

Similarly putting equation (2.40) and (2.41) in equation (2.9) gives:

$$y'(t) = 2\left[u_0(1-t)^2 + 2u_1(1-t)t + u_2t^2\right]\left[v_0(1-t)^2 + 2v_1(1-t)t + v_2t^2\right]$$

$$= 2u_0v_0(1-t)^4 + 4(u_0v_1 + u_1v_0)(1-t)^3t + 2(u_0v_2 + 4u_1v_1 + u_2v_2)(1-t)^2t^2 + 4(u_1v_2 - u_2v_1)(1-t)t^3 + (u_2v_2)t^4 \tag{2.45}$$

Taking Integral of both sides of equation (2.44) and (2.45):

$$x(t)=\int\left[\begin{array}{l}(u_0{}^2-v_0{}^2)(1-t)^4+4(u_0u_1-v_0v_1)(1-t)^3t+\\2(u_0u_2+u_1{}^2-v_0v_2-v_1{}^2)(1-t)^2t^2+4(u_1u_2-v_1v_2)(1-t)t^3+(u_2{}^2-v_2{}^2)t^4\end{array}\right]dt \tag{2.46}$$

$$y(t)=\int\left[\begin{array}{l}2u_0v_0(1-t)^4+4(u_0v_1+u_1v_0)(1-t)^3t\\+2(u_0v_2+4u_1v_1+u_2v_2)(1-t)^2t^2+4(u_1v_2-u_2v_1)(1-t)t^3+(u_2v_2)t^4\end{array}\right]dt \tag{2.47}$$

Now since the indefinite integrals of the Bernstein is given by [27, 28]:

$$\int\left[\binom{n-1}{k}(1-t)^{n-1-k}t^k\right]dt=\frac{1}{n}\sum_{j=k+1}^{n}\binom{n}{j}(1-t)^{n-j}t^j$$

$$=\frac{1}{n}\sum_{j=k+1}^{n}\binom{n}{j}(1-t)^{n-(k+1)}t^{k+1}$$

Where $k=0,.....n-1,\ \binom{n}{j}=\frac{n!}{(n-j)!\,j!}$ and $t\in[0\ 1]$.

Therefore equation (2.46) and (2.47) respectively yield:

$$x(t)=\sum_{k=0}^{5}x_k\binom{5}{k}(1-t)^{5-k}t^k \tag{2.48}$$

$$y(t)=\sum_{k=0}^{5}y_k\binom{5}{k}(1-t)^{5-k}t^k \tag{2.50}$$

Equation (2.48) and (2.49) give the expression for 2D Pythagorean Hodograph quintic $r(t)=[x(t),\ y(t)]^T$ with control points $P_k=(x_k,y_k)$ in compact form as follow:

$$r(t)=\left[\begin{array}{l}x(t)\\y(t)\end{array}\right]=\sum_{k=0}^{5}P_k\binom{5}{k}(1-t)^{5-k}t^k \tag{2.51}$$

Equation (2.51) on expansion yields:

$$r(t)=P_0(1-t)^5+5P_1t(1-t)^4+10P_2t^2(1-t)^3+10P_3t^3(1-t)^2+5P_4t^4(1-t)+P_5t^5 \tag{2.52}$$

Where $P_k(x_k,y_k),\ k=0,1,2,3,4,5$ are the the control points for PH quintic in 2D. The control points are very important because it determine the shape of the curve [29]. For quintic in 2D the control points [28] are given as follow:

$$P_1=P_0+\frac{1}{5}(u_0^2-v_0^2,\ 2u_0v_0) \tag{2.53}$$

$$P_2=P_1+\frac{1}{5}(u_0u_1-v_0v_1,\ u_ov_1+u_1v_0) \tag{2.54}$$

$$P_3 = P_2 + \frac{1}{5}(u_1^2 - v_1^2,\ 2u_1v_1) + \frac{1}{15}(u_0u_2 - v_0v_2,\ u_0v_2 + u_2v_0) \tag{2.55}$$

$$P_4 = P_3 + \frac{1}{5}(u_1u_2 - v_1v_2,\ u_1v_2 + u_2v_1) \tag{2.56}$$

$$P_5 = P_4 + \frac{1}{5}(u_2^2 - v_2^2,\ 2u_2v_2) \tag{2.57}$$

Equation (2.52) in matrix form can be represented as follow:

$$r(t) = \begin{bmatrix}(1-t)^5 & 5t(1-t)^4 & 10t^2(1-t)^3 & 10t^3(1-t)^2 & 5t^4(1-t) & t^5\end{bmatrix} \begin{bmatrix}P_0\\P_1\\P_2\\P_3\\P_4\\P_5\end{bmatrix}$$

Implies:

$$r(t) = \begin{bmatrix}1 & t & t^2 & t^3 & t^4 & t^5\end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0\\ -5 & 5 & 0 & 0 & 0 & 0\\ 10 & -20 & 10 & 0 & 0 & 0\\ -10 & 30 & -30 & 10 & 0 & 0\\ 5 & 20 & 30 & -20 & 5 & 0\\ -1 & 5 & -10 & 10 & -5 & 1\end{bmatrix} \begin{bmatrix}P_0\\P_1\\P_2\\P_3\\P_4\\P_5\end{bmatrix} \tag{2.58}$$

This is the matrix form of PH quintic in 2D. If the control points of 2D PH quintic are known then equation (2.58) can be used to calculate the planer PH quintic curve.

2.4.2.1 Determining the Unknown Control Points in Case of PH Quintic

Since the initial and final configurations for each UAV are known therefore four out of the six control points P_0, P_1, P_4, P_5 can be specified from equation (2.52) and its derivative.

The derivative of the PH curve $r'(t)$ in compact form is given by [27, 28]:

$$r'(t) = n\sum_{k=0}^{n-1} \Delta P_k \binom{n-1}{k} (1-t)^{n-1-k} t^k \tag{2.59}$$

Where $\binom{n-1}{k} = \frac{(n-1)!}{(n-1-k)!k!}$, $t \in [0\ 1]$ and ΔP_k denotes the k[th] forward path difference $(P_{k+1} - P_k)$ for $k = 0,....,n-1$

For quintic PH curve $n = 5$ in equation (2.59) therefore:

$$r'(t) = 5\sum_{k=0}^{4} \Delta P_k \binom{4}{k} (1-t)^{4-k} t^k \tag{2.60}$$

Equation (2.60) on expansion gives:

$$r'(t) = 5(P_1 - P_0)(1-t)^4 + 20(P_2 - P_1)(1-t)^3 t + 30(P_3 - P_2)(1-t)^2 t^2 + 20(P_4 - P_3)(1-t)t^3 + 5(P_5 - P_4)t^4 \tag{2.61}$$

Substituting $t = 0$ in equations (2.52) and (2.61):.

$$r(0) = P_0 = (x_{si},\ y_{si}) \tag{2.62}$$

$$r'(0) = 5(P_1 - P_0) = [dx_{si},\ dy_{si}] \tag{2.63}$$

Similarly substituting $t = 1$ in equations (2.51) and (2.61):

$$r(1) = P_5 = (x_{fi},\ y_{fi}) \tag{2.64}$$

$$r'(1) = 5(P_5 - P_4) = [dx_{fi},\ dy_{fi}] \tag{2.65}$$

Therefore equation (2.62), (2.63), (2.64) and (2.65) implies:

$$P_0 = (x_{si},\ y_{si}) \tag{2.66}$$

$$P_5 = (x_{fi},\ y_{fi}) \tag{2.67}$$

$$P_1 = P_0 + \frac{1}{5}[dx_{si},\ dy_{si}] \tag{2.68}$$

$$P_4 = P_5 - \frac{1}{5}[dx_{fi},\ dy_{fi}] \tag{2.69}$$

From equation (2.66)-(2.69) four control points P_0, P_1, P_4, P_5 are known because P_0, P_1 and P_4, P_5 are specified by the initial and final poses of the UAV. Only two control points P_2, P_3 needed to be determined.

The procedure to find out the unknown control points is given in [28]. This procedure can be applied to x and y components of the equation (2.53) and (2.57) with $(a,b) = (5\Delta x_0,\ 5\Delta y_0)$ and $(a,b) = (5\Delta x_4,\ 5\Delta y_4)$ respectively, to obtain the expression for $(u_0,\ v_0)$ and $(u_2,\ v_2)$. Also from the procedure we know that:

$$|\Delta P_k| = \sqrt{\Delta x_k^2 + \Delta y_k^2} \text{ and } c = \sqrt{a^2 + b^2}$$

Equation (2.53):

$$P_1 = P_0 + \frac{1}{5}(u_0^2 - v_0^2,\ 2u_0 v_0)$$

$$5(P_1 - P_0) = (u_0^2 - v_0^2,\ 2u_0 v_0)$$

In component form:

$$5(x_1 - x_0) = u_0^2 - v_0^2$$

$$5\Delta x_0 = u_0^2 - v_0^2$$

$$5(y_1 - y_0) = 2u_0 v_0$$

$$5\Delta y_0 = 2u_0 v_0$$

The mentioned procedure [28], for (u_0, v_0) with $a = 5\Delta x_0$, and $b = 5\Delta y_0$ gives:

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \pm \begin{bmatrix} \sqrt{\dfrac{(c-a)}{2}} \\ sign(b)\sqrt{\dfrac{(c-a)}{2}} \end{bmatrix}$$

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \pm \begin{bmatrix} \sqrt{\dfrac{\sqrt{a^2+b^2}+a}{2}} \\ sign(5\Delta y_0)\sqrt{\dfrac{\sqrt{a^2+b^2}-a}{2}} \end{bmatrix}$$

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \pm \begin{bmatrix} \sqrt{\dfrac{\sqrt{25(\Delta x_0^2 + \Delta y_0^2)}+5\Delta x_0}{2}} \\ sign(5\Delta y_0)\sqrt{\dfrac{\sqrt{25(\Delta x_0^2 + \Delta y_0^2)}-5\Delta x_0}{2}} \end{bmatrix}$$

$$\begin{bmatrix} u_0 \\ v_0 \end{bmatrix} = \pm\sqrt{\frac{5}{2}} \begin{bmatrix} \sqrt{\|\Delta P_0\| + \Delta x_0} \\ sign(\Delta y_0)\sqrt{\|\Delta P_0\| - \Delta x_0} \end{bmatrix} \qquad (2.70)$$

Similarly equation (2.57) yields:

$$P_5 = P_4 + \frac{1}{5}(u_2^2 - v_2^2,\ 2u_2 v_2)$$

$$5(P_5 - P_4) = (u_2^2 - v_2^2,\ 2u_2 v_2)$$

In component form:

$$5(x_5 - x_4) = u_2^2 - v_2^2$$

$$5\Delta x_4 = u_2^2 - v_2^2$$

$$5(y_5 - y_4) = 2u_2 v_2$$

$$5\Delta y_4 = 2u_2 v_2$$

Similarly for (u_2, v_2), with $a = 5\Delta x_4$, $b = 5\Delta y_4$:

$$\begin{bmatrix} u_2 \\ v_2 \end{bmatrix} = \pm\sqrt{\frac{5}{2}} \begin{bmatrix} \sqrt{\|\Delta P_4\| + \Delta x_4} \\ sign(\Delta y_4)\sqrt{\|\Delta P_4\| - \Delta x_4} \end{bmatrix} \tag{2.71}$$

Now in order to calculate (u_1, v_1), taking control points P_1 and P_4, thus

$$P_4 - P_1 = (P_4 - P_3) + (P_3 - P_2) + (P_2 - P_1) \tag{2.72}$$

Putting the values of control points P_1, P_2, P_3, P_4 component wise in equation (2.72):

$$x_4 - x_1 = \frac{1}{5}(u_1u_2 - v_1v_2) + \frac{2}{15}(u_1^2 - v_1^2) + \frac{1}{15}(u_0u_2 - v_0v_2) + \frac{1}{5}(u_0u_1 - v_0v_1)$$

$$\frac{15}{2}(x_4 - x_1) = \frac{3}{2}(u_1u_2 - v_1v_2) + u_1^2 - v_1^2 + \frac{1}{2}(u_0u_2 - v_0v_2) + \frac{3}{2}(u_0u_1 - v_0v_1)$$

$$\frac{15}{2}(x_4 - x_1) = u_1^2 - v_1^2 + \frac{3}{2}(u_2 + u_0)u_1 - \frac{3}{2}(v_2 + v_0)v_1 + \frac{1}{2}(u_0u_2 - v_0v_2) \tag{2.73}$$

Similarly for y component:

$$\frac{15}{2}(y_4 - y_1) = 2u_1v_1 + \frac{3}{2}(v_0 + v_2)u_1 + \frac{3}{2}(u_0 + u_2)v_1 + \frac{1}{2}(u_0v_2 + u_2v_0) \tag{2.74}$$

Changing variables:

$$\tilde{u}_1 = u_1 + \frac{3}{4}(u_0 + u_2) \tag{2.75}$$

$$\tilde{v}_1 = v_1 + \frac{3}{4}(v_0 + v_2) \tag{2.76}$$

Equation (2.75) and (2.76) implies:

$$u_1 = \tilde{u}_1 - \frac{3}{4}(u_0 + u_2) \tag{2.77}$$

$$v_1 = \tilde{v}_1 - \frac{3}{4}(v_0 + v_2) \tag{2.78}$$

Putting equations (2.77) and (2.78) in (2.73) yields:

$$\frac{15}{2}(x_4 - x_1) = \left[\tilde{u}_1 - \frac{3}{4}(u_0 + u_2)\right]^2 - \left[\tilde{v}_1 - \frac{3}{4}(v_0 + v_2)\right]^2 + \frac{3}{2}(u_2 + v_0)\left[\tilde{u}_1 - \frac{3}{4}(u_0 + u_2)\right] - \frac{3}{2}(v_0 + v_2)\left[\tilde{v}_1 - \frac{3}{4}(v_0 + v_2)\right] + \frac{1}{2}(u_0u_2 - v_0v_2)$$

$$\frac{15}{2}(x_4 - x_1) = \tilde{u}_1^2 + \frac{9}{16}(u_0 + u_2)^2 - \frac{3}{2}\tilde{u}_1(u_0 + u_2) - \tilde{v}_1^2 - \frac{9}{16}(v_0 + v_2)^2 + \frac{3}{2}\tilde{v}_1(v_0 + v_2)$$
$$-\frac{9}{8}(u_0 + u_2)^2 + \frac{3}{2}\tilde{u}_1(u_0 + u_2) + \frac{9}{8}(v_0 + v_2)^2 - \frac{3}{2}\tilde{v}_1(v_0 + v_2) + \frac{1}{2}(u_0u_2 - v_0v_2) \quad (2.79)$$

Cancelling the like terms with opposite signs yields:

$$\frac{15}{2}(x_4 - x_1) = \tilde{u}_1^2 - \tilde{v}_1^2 - \frac{9}{16}(u_0 + u_2)^2 + \frac{9}{16}(v_0 + v_2)^2 + \frac{1}{2}(u_0u_2 - v_0v_2)$$

Making $\tilde{u}_1^2 - \tilde{v}_1^2$ the subject of equation:

$$\tilde{u}_1^2 - \tilde{v}_1^2 = \frac{9}{16}(u_0^2 + u_2^2 + 2u_0u_2) - \frac{9}{16}(v_0^2 + v_2^2 + 2v_0v_1) - \frac{1}{2}(u_0u_2 - v_0u_2 - v_0v_2) +$$
$$+\frac{15}{2}(x_4 - x_1)$$

$$= +\frac{9}{16}(u_2^2 + u_2^2) + \frac{9}{8}u_0u_2 - \frac{9}{16}(v_0^2 + v_2^2) - \frac{9}{8}v_0v_1 - \frac{1}{2}u_0u_2 + \frac{1}{2}v_0v_2 + \frac{15}{2}(x_4 - x_1)$$

$$\tilde{u}_1^2 - \tilde{v}_1^2 = \frac{9}{16}(u_0^2 - v_0^2 + u_2^2 - v_2^2) - \frac{5}{8}u_0u_2 + \frac{5}{8}v_0v_2 + \frac{15}{2}(x_4 - x_1) \quad (2.80)$$

$$\text{Let} \quad a = \tilde{u}_1^2 - \tilde{v}_1^2 \quad (2.81)$$

Putting equation (2.81) in equation (2.80):

$$a = \frac{9}{16}(u_0^2 - v_0^2 + u_2^2 - v_2^2) - \frac{5}{8}(u_0u_2 + v_0v_2) + \frac{15}{2}(x_4 - x_1) \quad (2.82)$$

Similarly the expression for y component is calculated to be:

$$2\tilde{u}_1\tilde{v}_1 = \frac{9}{8}(u_0v_0 - u_2v_2) + \frac{5}{8}(u_0v_2 - u_2v_0) + \frac{15}{2}(y_4 - y_1) \quad (2.83)$$

$$\text{Let} \quad b = 2\tilde{u}_1\tilde{v}_1 \quad (2.84)$$

Putting equation (2.84) in equation (2.83):

$$b = \frac{9}{8}(u_0v_0 - u_2v_2) + \frac{5}{8}(u_0v_2 - u_2v_0) + \frac{15}{2}(y_4 - y_1) \quad (2.85)$$

Now $u_0, v_0, u_2, v_2, x_1, x_4, y_1, y_4$ in equations (2.82) and (2.85) are known. Therefore a and b can be calculated and the procedure in [28] can be applied to the equations (2.81) and (2.84) to obtain the expressions for $\tilde{u}_1$ and $\tilde{v}_1$. These values of $\tilde{u}_1$ and $\tilde{v}_1$ are then substituted in equations (2.77) and (2.78) to get the expressions for u_1 and v_1 as follow:

$$\begin{bmatrix} u_1 \\ v_1 \end{bmatrix} = -\frac{3}{4}\begin{bmatrix} u_0 + u_2 \\ v_0 + v_2 \end{bmatrix} \pm \sqrt{\frac{1}{2}}\begin{bmatrix} \sqrt{c + a} \\ sign(b)\sqrt{c - a} \end{bmatrix} \quad (2.86)$$

Once the values of the coefficients u_0, u_1, u_2 and v_0, v_1, v_2 of the constituents' curves of the Bernstein polynomials represented by equations (2.40) and (2.41) are calculated, all the unknown control points of PH quintic can be calculated using equations (2.53)-(2.57).

2.4.3 Path of Minimum Bending Energy

The shape of the Pythagorean hodograph curve depends upon the values of the coefficients of the constituents' polynomials chosen to yield the PH curve. The sign ambiguity in equations (2.70), (2.71) and (2.86) results in different values of the coefficient of constituents' polynomials yielding multiple solutions for the same initial and final configurations. If the signs of u_1, v_1 are assumed to be fixed in equation (2.86) then four PH quintic curves can be obtained for the sign choices of $++, +-, -+, --$ for (u_0, v_0) and (u_2, v_2) in equation (2.70), and (2.71). From among these four solutions the best one is selected on the basis of minimum bending energy [30]. Thus the bending energy for each of the four curves is calculated by the following expression:

$$E = \int \kappa^2(s) ds$$

$$= \int \kappa^2(s) \frac{ds}{dt} dt$$

$$E = \int \kappa^2(t) \sigma(t) dt \tag{2.87}$$

Where κ (kappa) is the curvature of the UAV path.

The bending energies are then compared for all the resultant curves, and the curve with least bending energy is chosen to be the UAV path. The least bending energy path has the unique attribute of being traced by the UAV with minimum effort.

In order to make the selected least bending energy curve, a feasible UAV path, the kinematic and safety constraints of the UAV are imposed.

2.4.4 Kinematic Constraints

After generating the least energy path, the path is to be made flyable for the vehicles by imposing the kinematic constraints of the vehicles. In this chapter, since the planer PH curves are considered therefore the curvature constraint is taken to be the kinematic constraints of the vehicle only. The way in which the curvature constraint is imposed is discussed in the following subsection.

2.4.4.1 Imposing Curvature Constraints

The resulting minimum bending energy UAVs trajectory has tangential continuity and curvature continuity, but it should have a curvature small enough such that it is possible

for the UAV to trace it. The UAV trajectory is made to obey curvature constraint of the UAVs by increasing the radius of the curve, [31], such that:

$$|\kappa(t)| \leq \kappa_{\max} \tag{2.88}$$

$\kappa_{\max}$ is the maximum bound on the curvature and $\kappa(t)$ is the trajectory's curvature at time t, given by:

$$\kappa(t) = \frac{|r'(t) \times r''(t)|}{|r'(t)|^3} \tag{2.89}$$

In this research the curvature of the path is changed by manipulating the control points P_1, P_2, P_3 and P_4 iteratively such that the condition (2.88) is satisfied. The first and the last control points P_0, P_5 are kept unchanged. The rest of the control points are changed by increasing the values of constants k_1 and k_2 in equations (2.90a), (2.90b) and equations (2.92a), (2.92b) as described in [31].

$$dx_{si} = k_1(\varepsilon_{si} \cos\theta_{si}) \tag{2.90a}$$

$$dy_{si} = k_2(\varepsilon_{si} \sin\theta_{si}) \tag{2.90b}$$

Where:

$$\varepsilon_{si} = \sqrt{dx_{si}^2 + dy_{si}^2} \tag{2.91a}$$

$$dx_{si} = x_1 - x_0 \tag{2.91b}$$

$$dy_{si} = y_1 - y_0 \tag{2.91c}$$

(x_0, y_0) and (x_1, y_1) being the coordinates of the first and second control points P_0, P_1. Similarly:

$$dx_{fi} = k_1(\varepsilon_{fi} \cos\theta_{fi}) \tag{2.92a}$$

$$dy_{fi} = k_2(\varepsilon_{fi} \sin\theta_{fi}) \tag{2.92b}$$

Where:

$$\varepsilon_{fi} = \sqrt{dx_{fi}^2 + dy_{fi}^2} \tag{2.93a}$$

$$dx_{fi} = x_5 - x_4 \tag{2.93b}$$

$$dy_{fi} = y_5 - y_4 \tag{2.93c}$$

(x_4, y_4) and (x_5, y_5) being the coordinates of the fourth and fifth control points P_4, P_5.

2.4.5 Safety Constraints

After achieving flyable UAV paths, safety of the generated paths is addressed. A flyable path is safe if it is free of inter collisions and free of collision with the obstacles. Such paths are called feasible paths and can be traced by the UAVs safely. The way the safety

constraint is imposed to avoid collisions with the known static obstacles and inter-collisions is discussed in the following subsections.

2.4.5.1 Avoiding Collisions with Known Obstacles

In the operating environment there are two kinds of obstacles: i.e. obstacles which are known a *priori* and obstacles which are not known in advance but discovered by the perception system mid course mission. The known obstacles are considered in this section for avoidance. For the unknown obstacles avoidance a separate chapter is dedicated later on. If any of the planned trajectories happen to collide with any of the known obstacles in the map then the path is not safe and it should be amended. The obstacle can be avoided by changing the curvature of the trajectory until the it is clear of obstacles. The obstacle is enclosed in a circle of certain radius. Using the control points, the curvature of the trajectory is changed such that the set of points of intersection of the UAV path and the enclosing circle is null.

2.4.5.1 Avoiding Inter Collision

UAV paths are checked for inter-collision of vehicles. Paths are searched for the possible inter-collision between them. The paths are free of inter-collisions, if the minimum distance between the UAV travelling on the paths at any time is equal to or greater than twice the safety radius R_s:

$$d_{\min} \geq 2R_s \tag{2.94}$$

Where $d_{\min}$ is the minimum distance between the UAVs travelling on the path at any time. R_s is the safety radius of each UAV. If the above condition is not satisfied then the curvature is changed further until the condition gets satisfied.

2.5 Simulation Results and Discussion

In this simulation a swarm of three UAVs is considered for mission deployment. All the UAVs will start from certain starting point at the same time and will reach the goal point at the same time. During their flights from starting position to finishing position all the UAVs will avoid inter-collisions and collisions with known obstacles. The proposed path planning algorithm will plan safe and flyable flight paths for all the UAVs in the group.

For this simulation, the initial and final poses, the curvature constraints and the safety radii of all the UAVs are summarised in table 2.1.

No	UAVs in the Swarm	Initial Pose		Final Pose		Curvature Limits $\kappa_{\max}$ (metre)	Safety Radius R_s (metre)
		Position (x_{si}, y_{si}) (metre)	**Heading Angle** θ_{si} **(**Degree)	**Position** (x_{fi}, y_{fi}) (metre)	**Heading Angle** θ_{fi} (Degree)		
1	UAV1	(8, 6)	130	(22, 40)	24	-1/3, +1/13	1
2	UAV2	(14, 6)	124	(17, 40)	120	-1/3, +1/13	1
3	UAV3	(18, 6)	3	(26, 40)	113	-1/3, +1/13	1

Table 2.1: Table of initial conditions, maximum curvature and safety radius.

All the known static obstacles are given in the terrain database. For simplicity, all the obstacles are assumed to be of square/rectangular shape in this simulation. The locations of the obstacle are described by the coordinates of their vertices. The vertices of the obstacle1, obstacle2 and obstacle 3 are given table 2.2.

No	Obstacles	Vertex 1	Vertex 2	Vertex 3	Vertex 4
1	Obstacle 1	(5, 20)	(5,22)	(8.5, 22)	(8.5, 20)
2	Obstacle 2	(12.5, 13)	(12.5, 15)	(17, 15)	(17, 13)
3	Obstacle 3	(24, 11)	(24, 13)	(27, 13)	(27, 11)

Table 2.2: Table of Locations of the known obstacles.

From the Hermit interpolation of known initial and final poses of each UAV, the solution for the UAV path results in four possible paths as shown in figure 2.3. In the figure the small circles represents the control points of the curve, the dotted lines represents the control polygon and the solid curve represents the UAVs paths.

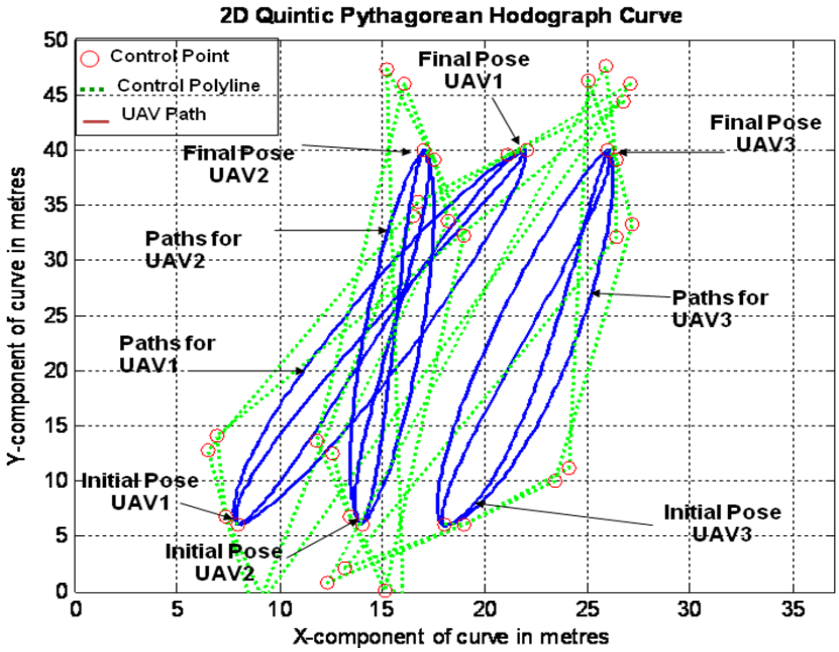

Figure 2.3: Four possible PH paths for each UAV1, UAV2 and UAV3

For each UAV, the best path among these four paths is the one with minimum bending energy. The bending energy of each path is determined in case of each UAV, and the path of least bending energy is selected. Thus the best path for each UAV is sorted from groups of four paths. The minimum bending energy path for each of the three UAVs is shown in figure 2.4.

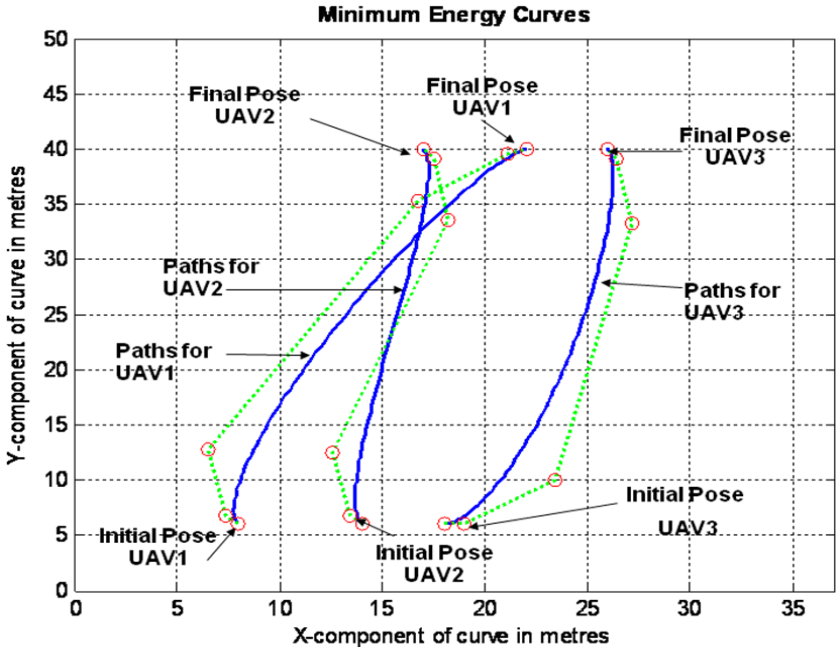

Figure 2.4: Minimum energy curves for UAV1, UAV2 and UAV3

The minimum energy path is flyable for the UAVs if it obeys the curvature constraint of the vehicle. The paths shown in figure 2.4 are made to obey the curvature constraints by curvature manipulation. This results in three flyable paths as shown in figure 2.5.

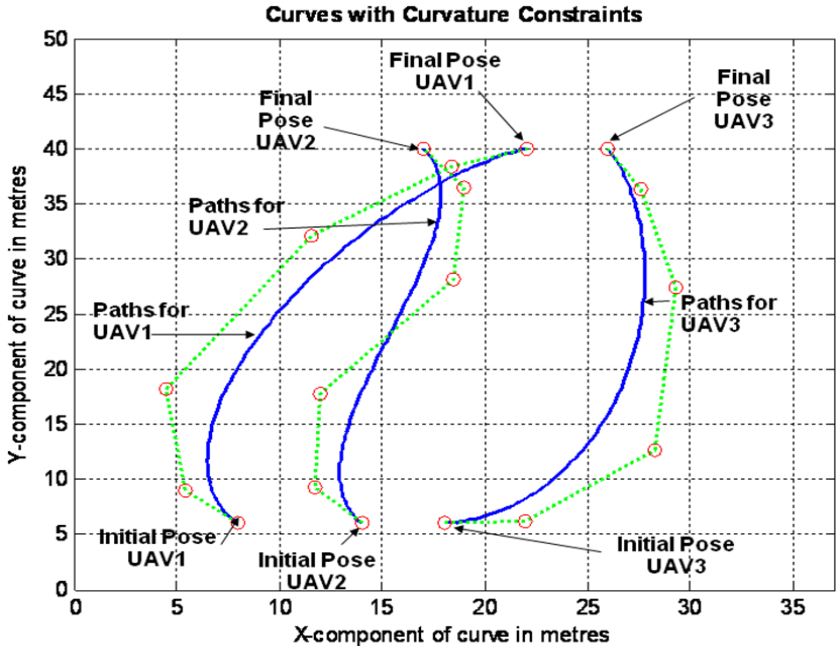

Figure 2.5: Curves obeying curvature constraints

Now all the known obstacles in the map are taken into account. The positions of the obstacles relative to the paths are shown in figure 2.6.

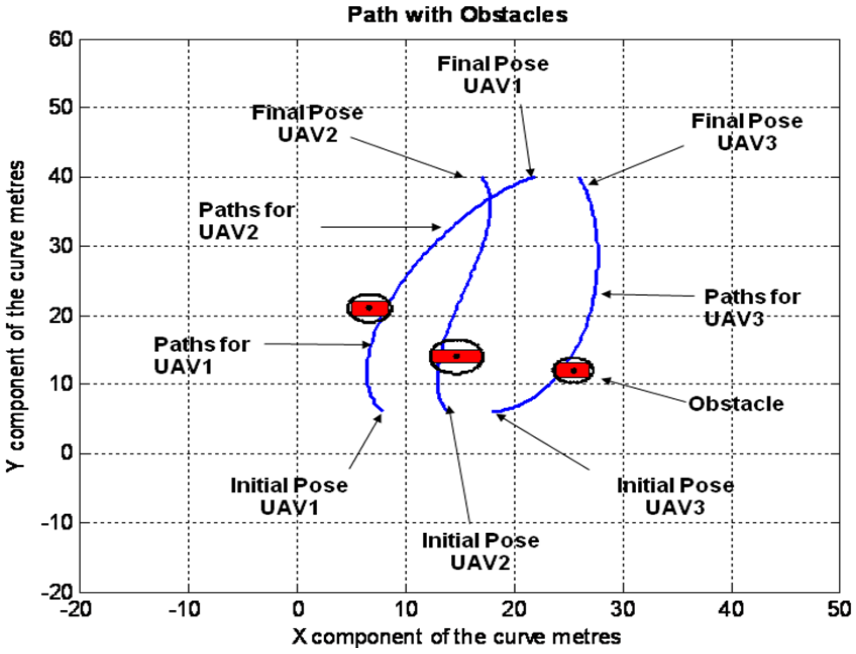

Figure 2.6: Paths with obstacles

These obstacles are avoided by varying the curvature. Figure 2.7 shows the obstacle been avoided in case of each path. The paths in figure 2.7 are feasible paths.

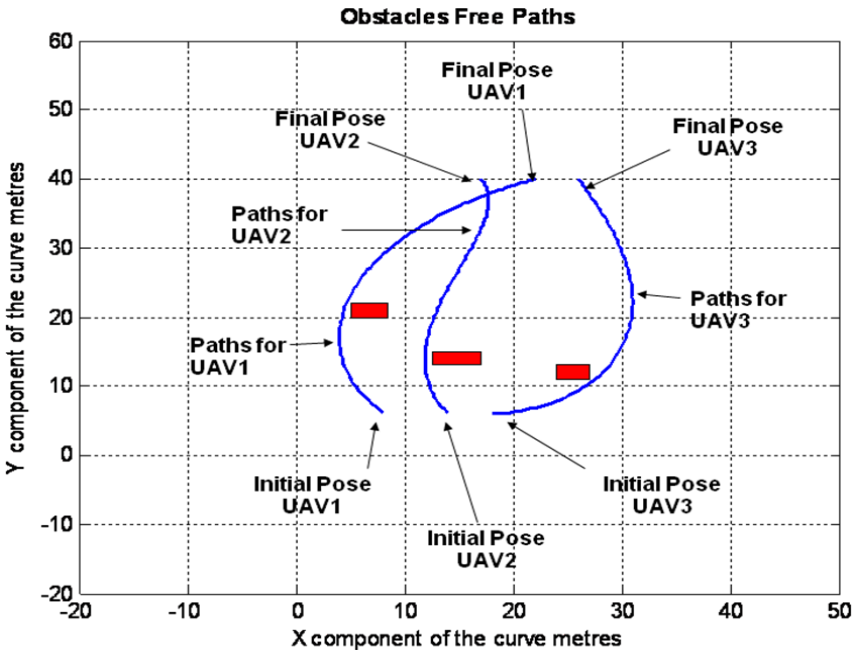

Figure 2.7: Obstacles free paths

At this stage the length of all the paths are made equal to the longest path for the purpose of checking inter collision and simultaneous arrival at the destination. In this simulation the length of each path comes out to be 42.5 metres as shown in figure 2.8.

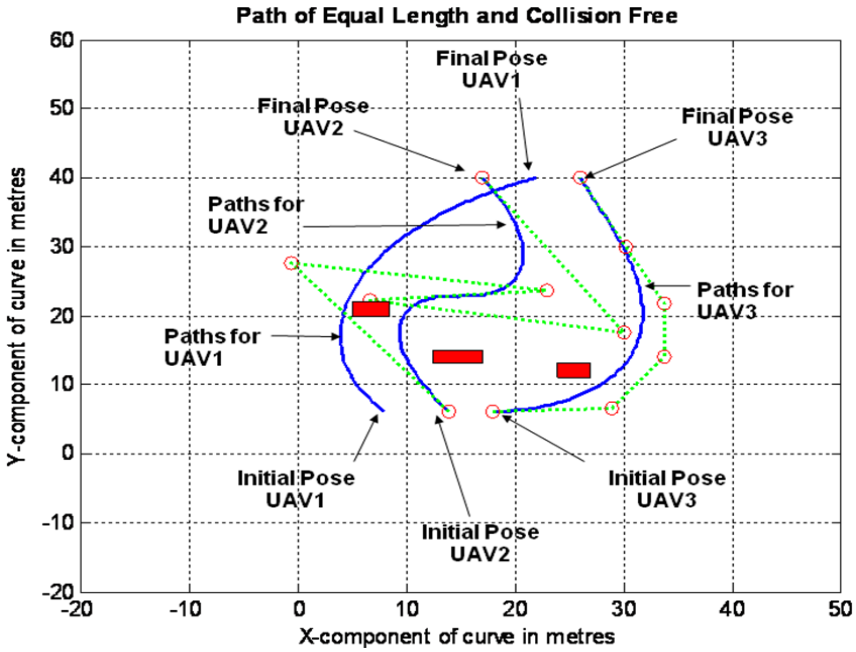

Figure 2.8: Obstacles free paths with equal lengths

After getting flyable paths of equal lengths a number of checks are made to see if any two of the three paths undergoes inter collision. For three UAVs, taking two paths at one time, the number of the checks are $\binom{3}{2} = 3$. The two paths are free from inter collision if the minimum distance between UAVs travelling on these paths at any time is equal to or greater than twice the safety radius. In this simulation the radius of safety circle of each UAV is taken as 1 metre, therefore the minimum safe distance between the two UAVs is 2 metres. The following checks are made for the inter collision:

- Between path1 and path2 (figure 2.9, 2.10)
- Between path2 and path3 (figure 2.11, 2.12)
- Between path1 and path3 (figure 2.13, 2.14)

The criterion for safety is:

If the distance between two UAVs remains equal to or more than two metres while travelling on their paths then they are safe from inter collision.

Figure 2.9 shows two paths, one for UAV1 and the other for UAV2. The distance profile between UAV1 and UAV2 travelling on path1 and path2 respectively is shown in figure 2.10. The dotted line in figure 2.10 shows the safety boundary, which, in other words represents the aforementioned safety criterion. It is obvious that the UAVs never cross the safety boundary while travelling on path1 and path2, therefore they are safe. Figure 2.10 shows that the two UAVs never getting closer than four metres during their travel.

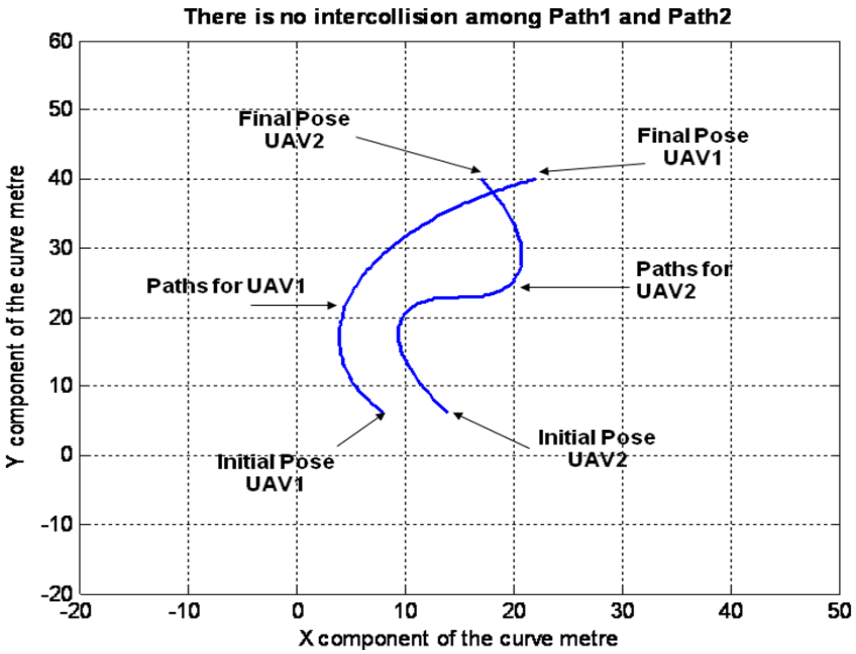

Figure 2.9: Path1 & path2

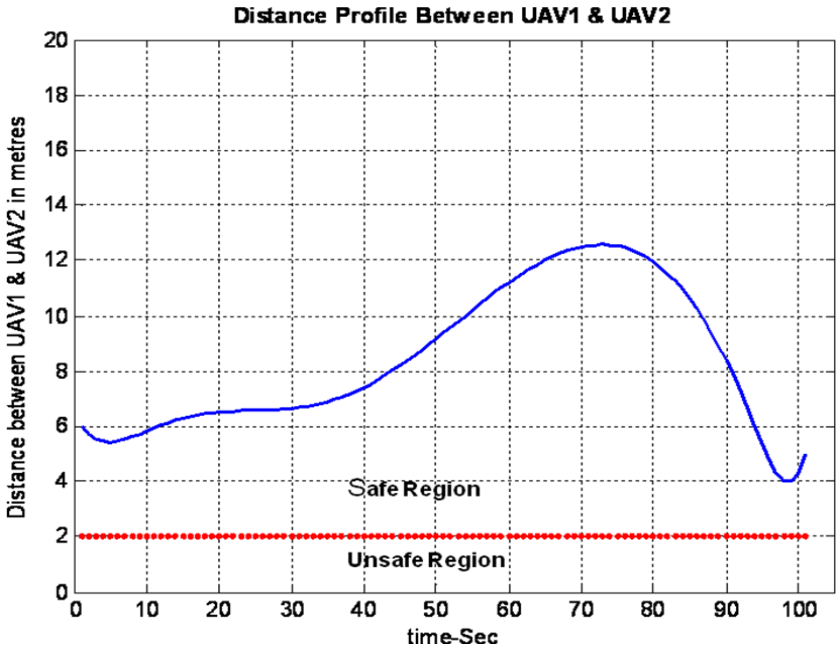

Figure 2.10: No collision between UAV1 and UAV2.

Figure 2.11 shows two paths, path2 and path3. UAV2 travels on path2 while UAV3 travels on path3. The distance profile between UAV2 and UAV3 travelling on path2 and path3 respectively is shown in figure 2.12. It is obvious that the UAVs never cross the safety boundary while travelling on path2 and path3, therefore they are safe.

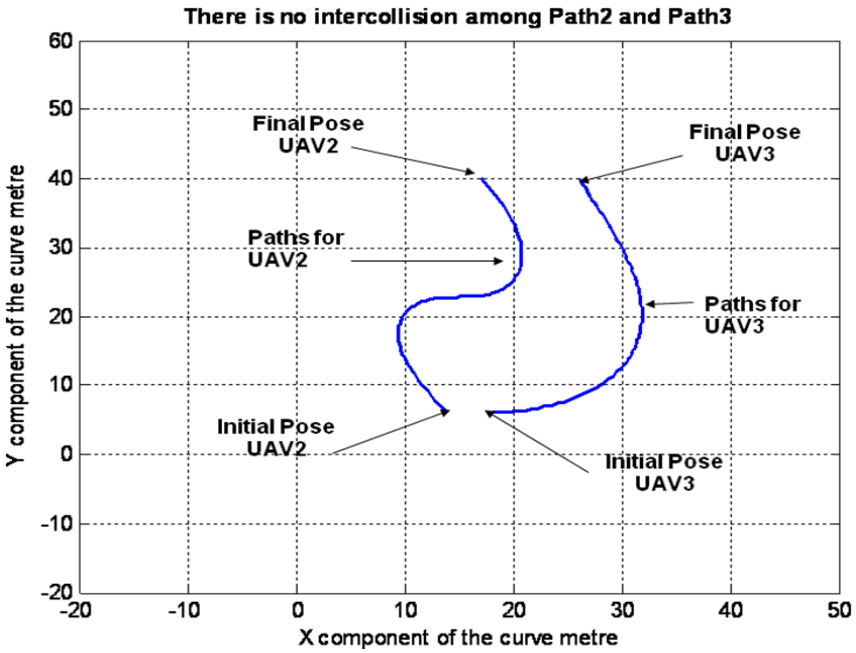

Figure 2.11: Path2 & path3

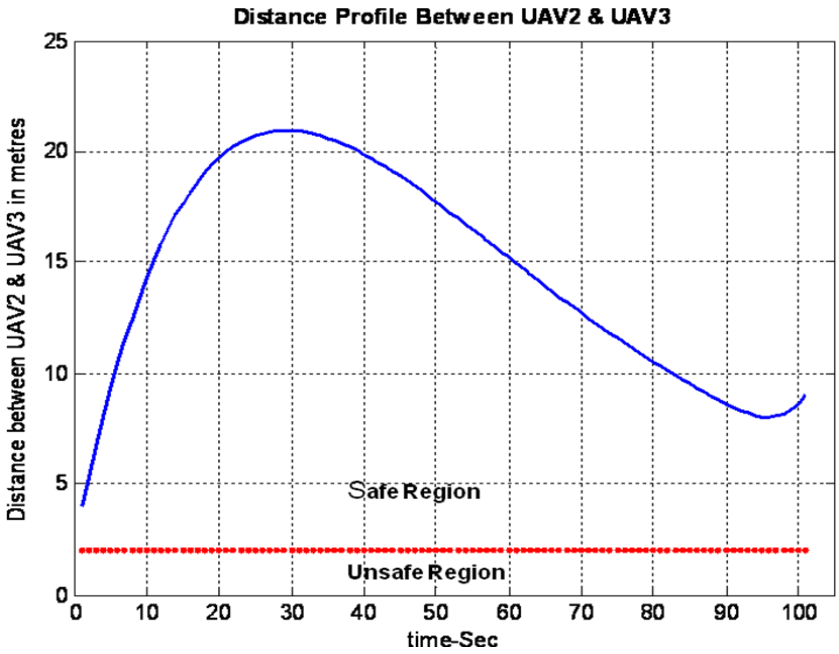

Figure 2.12: No collision between UAV2 & UAV3

Figure 2.13 shows two paths, path1 and path3. UAV1 travels on path1 while UAV3 travels on path3. The distance profile between UAV1 and UAV3 travelling on path1 and path3 respectively is shown in figure 2.14. It is obvious that the UAVs never cross the safety boundary while travelling on path1 and path3, therefore they are safe.

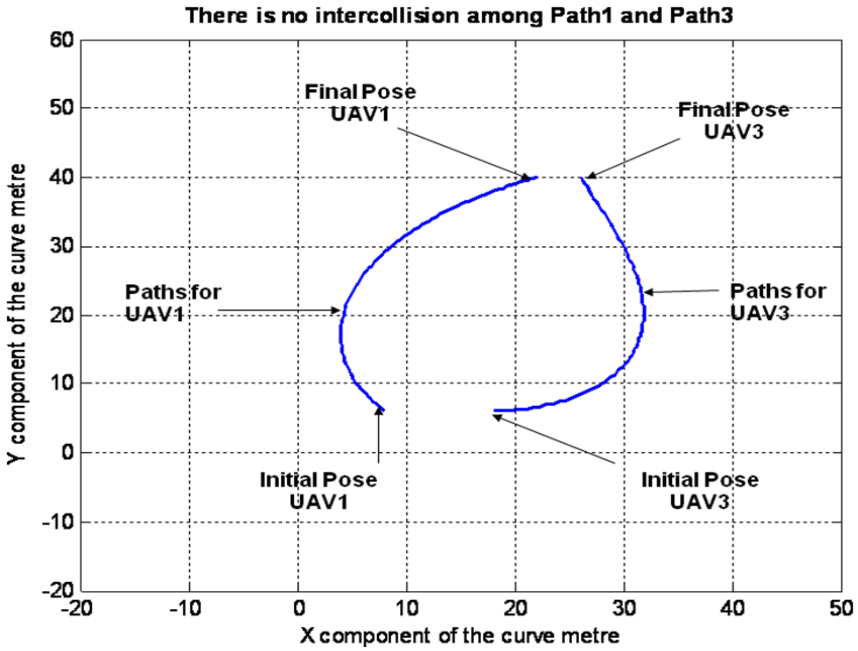

Figure 2.13: Path1 and path3.

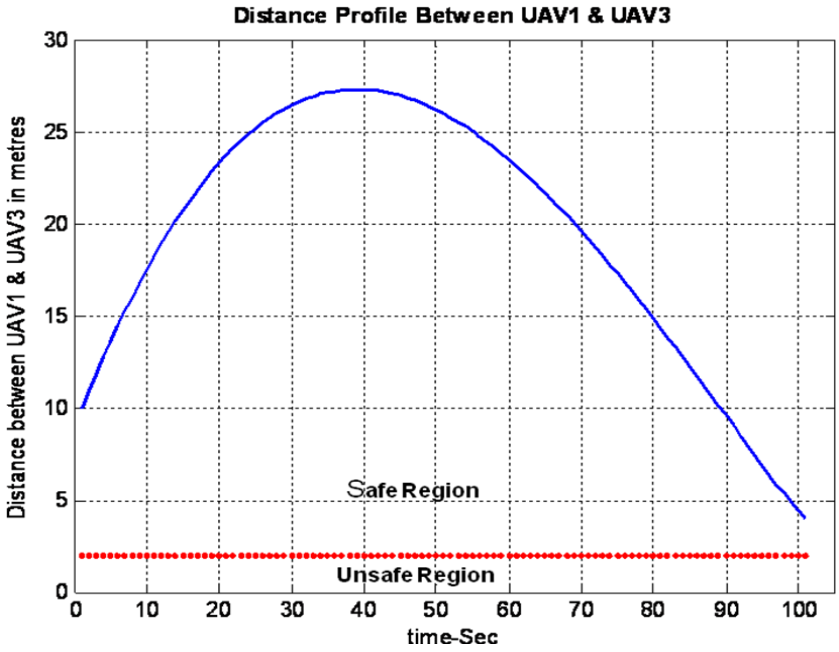

Figure 2.14: No collision between UAV1 and UAV3.

The condition for inter collision was checked and all paths were found to be free of inter collision. If any of the above condition is not satisfied then the whole process is repeated unless feasible paths are obtained.

The curvature plot against the path length of each of path1, path2, and path3 are shown in the figures 2.15, 2.16 and 2.17. In order to show that all the paths obey the curvature constraint, the curvature plots along with maximum and minimum curvature bounds against the path length are also shown in these figures. The maximum and minimum curvature bounds used for this simulation are:

$$\kappa_{\min} = -\frac{1}{3} = -0.333 \text{ metres}$$

$$\kappa_{\max} = +\frac{1}{3} = +0.333 \text{ metres}$$

and each path is 42.5 metres long.

Figures 2.15, 2.16 and 2.17 show that all the paths obey the curvature constraints.

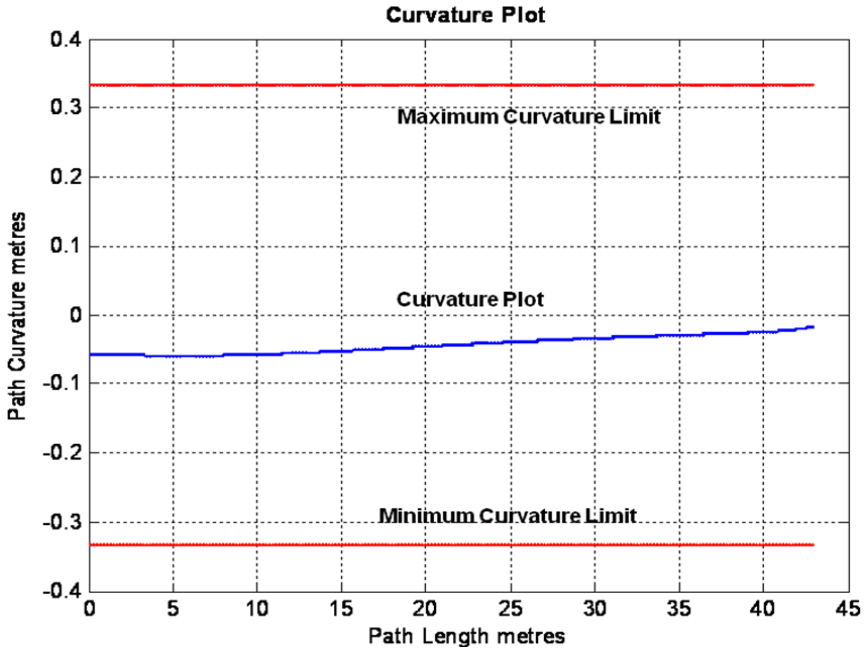

Figure 2.15: Curvature plots of path1 along with maximum and minimum curvature bound against path length

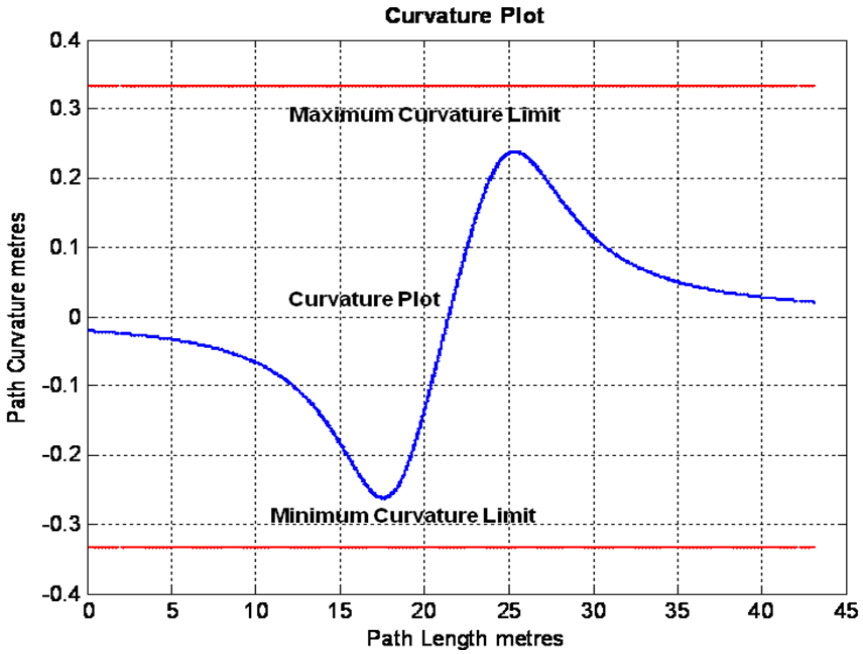

Figure 2.16: Curvature plots of path2 along with maximum and minimum curvature bounds against the path length

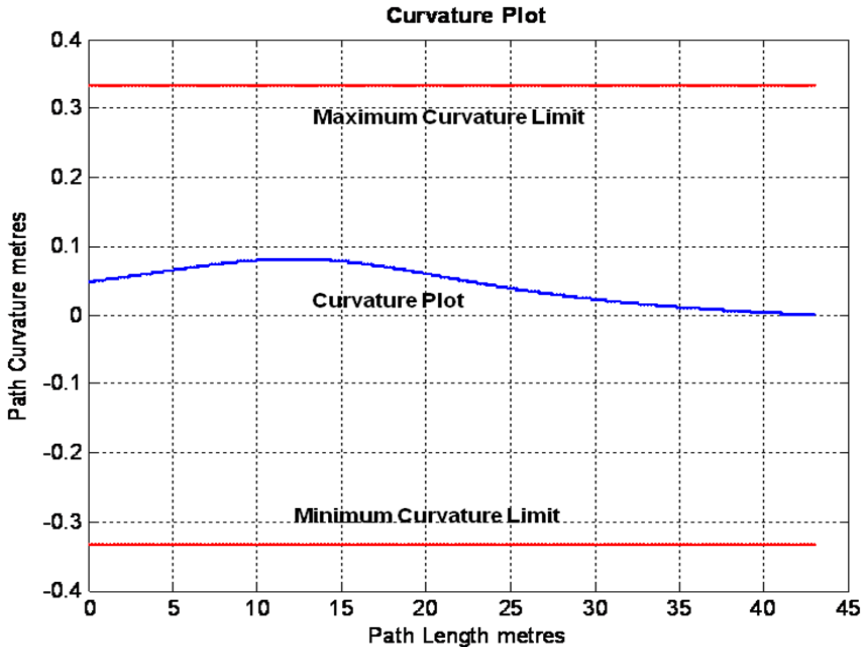

Figure 2.17: Curvature plots of path3 along with maximum and minimum curvature bounds against the path length

2.6 Conclusion

In this chapter, 2D path planning algorithm (2D Path Planner) based on Pythagorean Hodograph quintic has been derived and simulated for UAVs swarm. The algorithm successfully calculated safe and flyable paths (feasible paths) for all the UAVs in the group. The paths calculated by the path planning algorithm are optimal with respect to bending energy. These paths have the attractive feature of being traceable with least effort. This feature makes them fuel optimal as well as time optimal because hard demands for extreme acceleration are avoided by keeping the bending energy minimum. However the paths calculated by this algorithm are 2D paths, which are unrealistic for real missions. To accommodate real mission challenges and demands a path planner in 3D is derived in the next chapter.

Chapter 3

Cooperative Path Planning for Multiple UAVs Using Pythagorean Hodographs in 3D

In previous chapter path planning in 2D has been dealt with as a starting point of this research. The problem was simplified by imposing a constant altitude constraint. In real life missions the UAVs flies in 3D space. Therefore in this chapter, 3D cooperative path planning of UAVs in an environment with known stationary obstacles is considered. The constraint of the constant altitude has been relaxed. 3D Pythagorean Hodograph (PH) is used for UAVs path planning. In this path planning inter-collisions and collisions with the known obstacles are avoided. UAVs paths are first generated using 3D PH curves assuming obstacle free environment. Then, these paths are made flyable by imposing the kinematic constraints of the vehicles. The flyable paths are then amended according to the known obstacles in the map to avoid collisions with these obstacles. These paths are further checked for inter collision. In the end safe and flyable paths are generated. Simulation results based on a swarm of UAVs show the efficiency of the proposed 3D path planning algorithm.

3.1 Problem Formulation

A mission is planned to fly a group of UAVs safely from base B to target T. All the vehicles start from the base at the same time. The UAVs flight is assumed to be in known environment containing static obstacles only. During the flight, the vehicles will avoid inter-collision and collision with the known stationary obstacles.

The base and the goal points for each UAV are specified by initial and final poses. By Hermit interpolation of these poses the paths of UAVs are calculate. If $p_{si}(x_{si}, y_{si}, z_{si})$ represents the starting position and (θ_{si}, ϕ_{si}) represents the starting orientation of the *ith* UAV then the starting pose of the *ith* UAV is $Pose_{si}(x_{si}, y_{si}, z_{si}, \theta_{si}, \phi_{si})$. Similarly if $p_{fi}(x_{fi}, y_{fi}, z_{fi})$ represent the final position and (θ_{fi}, ϕ_{fi}) represents the final orientation of the *ith* UAV then the final pose of the same UAV is $Pose_{fi}(x_{fi}, y_{fi}, z_{fi}, \theta_{fi}, \phi_{fi})$. Where θ is the angle measured in XY plane with respect to X-axis and ϕ is the angle measured in XZ plane with respect to X-axis. Now the path of the UAV is defined as a parametric curve $r(t) = [x(t), y(t), z(t)]$ with curvature κ_i and torsion, τ_i connecting the initial and final pose. Mathematically this can be written as:

$$Pose_{si}(x_{si}, y_{si}, z_{si}, \theta_{si}, \phi_{si}) \xrightarrow{ri_i(t)} Pose_{fi}(x_{fi}, y_{fi}, z_{fi}, \theta_{fi}, \phi_{fi})$$

Where $r_i(t)$ is the Pythagorean Hodograph path of *ith* UAV in parametric form subjected to the following constraints:

1. $$|\kappa_i(t)| \le \kappa_{\max}$$
$$|\tau_i(t)| \le \tau_{\max}$$

   Where $\kappa_{\max}$ and $\tau_{\max}$ are the maximum curvature and torsion attainable by the UAVs.

2. The vehicles must keep safe distance to avoid inter collision while tracing their corresponding trajectories. This can be written in mathematical form as follows:

$$\boldsymbol{R}_{si} \cap \boldsymbol{R}_{sj} = \varphi$$

   That is, the intersection of the safety spheres of radius R_{si} and R_{sj} corresponding to ith and jth UAVs must be empty.

   Or equivalently the minimum distance between the two vehicles must be:

$$d \ge 2R_s$$

3. The vehicles must keep safe distance to avoid collisions with known obstacles while tracing their corresponding trajectories. This can be written in mathematical form as follow:

$$d(obstacle, UAV) \ge R_{obstacle} + R_s$$

   Where $d(obstacle, UAV)$ is the distance between the centre of the sphere enclosing the obstacle and the centre of the safety sphere of the UAV. $R_{obstacle}$ is the radius of the sphere enclosing the square obstacle.

The solution to the problem of path planning is approached through the application of 3D Pythagorean Hodograph curves which is discussed in the following section in detail.

3.2 Pythagorean Hodograph in 3D (Spatial Pythagorean Hodograph)

In the previous chapter a simple case of UAVs path planning in 2D (constant altitudes) was considered as a starting point. But the real missions are planned in 3D space therefore to deal with a real life problem of path planning polynomial and their hodographs in three dimensions [32, 33, 34 35] are needed. Although the extension of ordinary polynomial curve from R^2 to R^3 is simple but it becomes more complicated when dealing with Pythagorean hodograph curves. This is because of complexity in finding an appropriate characterization for polynomial solution to satisfy the Pythagorean hodograph condition in three dimensions i.e.:

$$x'^{2}(t) + y'^{2}(t) + z'^{2}(t) = \sigma^{2}(t) \tag{3.1}$$

Where $x'(t)$, $y'(t)$ and $z'(t)$ are the hodographs of the parametric curve $r(t) = [x(t), y(t), z(t)]$ and $\sigma(t)$ is the speed. For a parametric curve $r(t) = [x(t), y(t), z(t)]$ in three dimensions the hodograph $r'(t) = [x'(t), y'(t), z'(t)]$ is of the following form, if it is Pythagorean [27, 28]:

$$x'(t) = [u^{2}(t) + v^{2}(t) - p^{2}(t) - q^{2}(t)] \tag{3.2}$$

$$y'(t) = 2[u(t)q(t) - v(t)p(t)] \tag{3.3}$$

$$z'(t) = 2[v(t)q(t) - u(t)p(t)] \tag{3.4}$$

The polynomials $u(t), v(t), p(t)$ and $q(t)$ are chosen such that the parametric speed is:

$$\sigma(t) = u^{2}(t) + v^{2}(t) + p^{2}(t) + q^{2}(t) \tag{3.5}$$

If the polynomial $u(t), v(t), p(t)$ and $q(t)$ are specified in term of Bernstein coefficient on $t \in [0, 1]$, then the control points of spatial PH curve they define, can be expressed in terms of these coefficients in a manner analogous to that of planner PH curves.

3.2.1 Spatial Pythagorean Hodograph (PH) cubic

For simplicity, first the expression for spatial PH cubic curve is derived as a starting point. In order to derive an expression for spatial PH cubic four linear polynomials $u(t), v(t), p(t)$ and $q(t)$ are chosen in Bernstein form as follow:

$$u(t) = u_0(1-t) + u_1(t) \tag{3.6}$$

$$v(t) = v_0(1-t) + v_1(t) \tag{3.7}$$

$$p(t) = p_0(1-t) + p_1(t) \tag{3.8}$$

$$q(t) = q_0(1-t) + q_1(t) \tag{3.9}$$

Such that the Pythagorean condition:

$$\sigma(t) = u^{2}(t) + v^{2}(t) + p^{2}(t) + q^{2}(t) \tag{3.10}$$

is satisfied.

Now putting these values in equations (3.2), (3.3) and (3.4) gives:

$$\begin{aligned} x'(t) &= [u_0(1-t) + u_1(t)]^2 + [v_0(1-t) + v_1(t)]^2 - [p_0(1-t) + p_1(t)]^2 - [q_0(1-t) + q_1(t)]^2 \\ &= [u_0^2 + v_0^2 - p_0^2 - q_0^2](1-t)^2 + 2[u_0u_1 + v_0v_1 - p_0p_1 + q_0q_1](1-t)t + [u_1^2 + v_1^2 - p_1^2 - q_1^2]t^2 \end{aligned} \tag{3.11}$$

$$y'(t) = 2[\{u_0(1-t) + u_1(t)\}\{q_0(1-t) + q_1(t)\} - \{v_0(1-t) + v_1(t)\}\{p_0(1-t) + p_1(t)\}]$$

$$= 2[\{u_0q_0 + v_0p_0\}(1-t)^2 + \{u_1q_0 + u_0q_1 + v_1p_0 - v_0p_1\}(1-t)t + \{u_1q_1 + v_1p_1\}t^2 \tag{3.12}$$

$$z'(t) = 2[\{v_0(1-t) + v_1(t)\}\{q_0(1-t) + q_1(t)\} - \{u_0(1-t) + u_1(t)\}\{p_0(1-t) + p_1(t)\}]$$

$$= 2[\{v_0q_0 + u_0p_0\}(1-t)^2 + \{v_1q_0 + v_0q_1 - u_1p_0 - u_0p_1\}(1-t)t + \{v_1q_1 - u_1p_1\}t^2 \tag{3.13}$$

Taking Integral of both sides of equations (3.11), (3.12) and (3.13):

$$x(t) = \int [u_0^2 + v_0^2 - p_0^2 - q_0^2](1-t)^2 + 2[u_0u_1 + v_0v_1 - p_0p_1 + q_0q_1](1-t)t + [u_1^2 + v_1^2 - p_1^2 - q_1^2]t^2 dt \tag{3.14}$$

$$y(t) = \int 2[\{u_0q_0 + v_0p_0\}(1-t)^2 + \{u_1q_0 + u_0q_1 + v_1p_0 - v_0p_1\}(1-t)t + \{u_1q_1 + v_1p_1\}t^2 dt \tag{3.15}$$

$$z(t) = \int 2[\{v_0q_0 + u_0p_0\}(1-t)^2 + \{v_1q_0 + v_0q_1 - u_1p_0 - u_0p_1\}(1-t)t + \{v_1q_1 - u_1p_1\}t^2 dt \tag{3.16}$$

Now since the indefinite integrals of the Bernstein is given by [28]:

$$\int \left[\binom{n-1}{k} (1-t)^{n-1-k} t^k \right] dt = \frac{1}{n} \sum_{j=k+1}^{n} \binom{n}{j} (1-t)^{n-j} t^j$$

$$= \frac{1}{n} \sum_{j=k+1}^{n} \binom{n}{j} (1-t)^{n-(k+1)} t^{k+1}$$

Where $k = 0,.....n-1$, $\binom{n}{j} = \frac{n!}{(n-j)!\, j!}$ and $t \in [0\ 1]$.

Therefore equations (3.14), (3.15) and (3.16) yield:

$$x(t) = \sum_{k=0}^{3} x_k \binom{3}{k} (1-t)^{3-k} t^k \tag{3.17}$$

$$y(t) = \sum_{k=0}^{3} y_k \binom{3}{k} (1-t)^{3-k} t^k \tag{3.18}$$

$$z(t) = \sum_{k=0}^{3} z_k \binom{3}{k} (1-t)^{3-k} t^k \tag{3.19}$$

From equations (3.17), (3.18) and (3.19) the general equation of 3D Pythagorean Hodograph cubic $r(t) = [x(t), y(t), z(t)]$ with control points $P_k = (x_k, y_k, z_k)$ in compact form can be written as follow:

$$r(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \sum_{k=0}^{3} P_k \binom{3}{k} (1-t)^{3-k} t^k \tag{3.20}$$

Equation (3.20) on expansion gives:

$$r(t) = P_0(1-t)^3 + 3P_1(1-t)^2 t + 3P_2(1-t)t^2 + P_3 t^3 \tag{3.21}$$

The control points $P_k = (x_k, y_k, z_k)$, $k = 0,1,2,3$ for PH cubic in 3D are given below [27]:

$$P_1 = P_0 + \frac{1}{3}(u_0^2 + v_0^2 - p_0^2 - q_0^2,\ \ 2(u_0 q_0 + v_0 p_0),\ \ \ 2(v_0 q_0 - u_0 p_0)) \tag{3.22}$$

$$P_2 = P_1 + \frac{1}{3}(u_0 u_1 + v_0 v_1 - p_0 p_1 - q_0 q_1,\ \ u_0 q_1 + u_1 q_0 + v_0 p_1 + v_1 p_0,\ \ v_0 q_1 + v_1 q_0 - u_0 p_1 - u_1 p_0) \tag{3.23}$$

$$P_3 = P_2 + \frac{1}{3}(u_1^2 + v_1^2 - p_1^2 - q_1^2,\ \ 2(u_1 q_1 + v_1 p_1),\ \ 2(v_1 q_1 - u_1 p_1)) \tag{3.24}$$

Where P_0 is the integration constant being freely chosen.

Equation (3.21) in matrix form can be represented as follow:

$$r(t) = \begin{bmatrix}(1-t)^3 & 3t(1-t)^2 & 3t^2(1-t) & t^3\end{bmatrix} \begin{bmatrix}P_0 \\ P_1 \\ P_2 \\ P_3\end{bmatrix} \tag{3.25}$$

Implies:

$$r(t) = \begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix} \begin{bmatrix}1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1\end{bmatrix} \begin{bmatrix}P_0 \\ P_1 \\ P_2 \\ P_3\end{bmatrix} \tag{3.26}$$

This is the matrix form of PH cubic in 3D. If the control points of the PH curve are known then equation (3.26) can be used to calculate the spatial PH cubic curve $r(t)$.

3.2.1 Specifying Control Points in Case of Spatial PH Curve

Control points P_0, P_1, P_2, P_3 in case of 3D PH curve can be specified from the known initial and final poses of the UAV. For this purpose the expression for the derivative of PH curve is needed.

The derivative of the PH curve $r'(t)$ in compact form is given by [27, 28]:

$$r'(t) = n\sum_{k=0}^{n-1} \Delta P_k \binom{n-1}{k} (1-t)^{n-1-k} t^k \tag{3.27}$$

Where $\binom{n-1}{k} = \frac{(n-1)!}{(n-1-k)!k!}$, $t \in [0\ 1]$ and ΔP_k denotes the k[th] forward path difference $(P_{k+1} - P_k)$ for $k = 0,....,n-1$.

For cubic PH curve, $n = 3$ in equation (3.27) therefore:

$$r'(t) = 3\sum_{k=0}^{2} \Delta P_k \binom{2}{k} (1-t)^{2-k} t^k \tag{3.28}$$

Equation (3.28) on expansion gives:

$$r'(t) = 3(P_1 - P_0)(1-t)^2 + 3(P_2 - P_1)(1-t)t + 3(P_3 - P_2)t^2 \tag{3.29}$$

Putting $t = 0$ in equation (3.22) and (3.29) gives:

$$r(0) = P_0 \tag{3.30}$$

$$r'(0) = 3(P_1 - P_0) \tag{3.31}$$

Similarly putting $t = 1$ in equation (3.22) and (3.29) gives:

$$r(1) = P_3 \tag{3.32}$$

$$r'(1) = 3(P_3 - P_2) \tag{3.33}$$

Since the initial and final configurations for *ith* UAV are known therefore the starting position $p_{si} = (x_{si},\ y_{si},\ z_{si})$ and the derivative at the starting point $dp_{si} = (dx_{si},\ dy_{si},\ dz_{si})$ is known. Also the final point $p_{fi} = (x_{fi},\ y_{fi},\ z_{fi})$ and the derivative at this point $dp_{fi} = (dx_{fi},\ dy_{fi},\ dz_{fi})$ are known:

$$r(0) = P_0 = (x_{si},\ y_{si},\ z_{si}) \tag{3.34}$$

$$r'(0) = 3(P_1 - P_0) = (dx_{si},\ dy_{si},\ dz_{si}) \tag{3.35}$$

$$r(1) = P_3 = (x_{fi},\ y_{fi},\ z_{fi}) \tag{3.36}$$

$$r'(1) = 3(P_3 - P_2) = (dx_{fi},\ dy_{fi}, dz_{fi}) \tag{3.37}$$

Equations (3.35) and (3.37) can be rearranged as:

$$P_1 = P_0 + \frac{1}{3}(dx_{si},\ dy_{si},\ dz_{si}) \tag{3.38}$$

$$P_2 = P_3 - \frac{1}{3}(dx_{fi},\ dy_{fi},\ dz_{fi}) \tag{3.39}$$

Equations (3.34), (3.36), (3.38) and (3.39) give all the control points.

Now if equations (3.34), (3.36), (3.38) and (3.39) are substituted in equations (3.22)-(3.24), the coefficients u_0, u_1, u_2, v_0, v_1, v_2, p_0, p_1, p_2 and q_0, q_1, q_2 of the constituents' curves represented by equations (3.6), (3.7), (3.8) and (3.9) which form the PH curve can be determined.

In the next section the expression for 3D Pythagorean Hodograph Quintic is derived which is the subject of this chapter.

3.2.2 Spatial Pythagorean Hodograph (PH) Quintic

As in this research Pythagorean Hodograph quintic is used for path planning for its desirable flexibility to achieve the required configuration, therefore the case of the spatial Pythgorean Hodograph cubic is extended to spatial Pythagorean Hodograph quintic. To derive the expression for PH Quintic, four quadratic polynomials $u(t), v(t), p(t)$ and $q(t)$ in Bernstein form [32, 33] are chosen:

$$u(t) = u_0(1-t)^2 + u_1 2(1-t)t + u_2 t^2 \tag{3.40}$$

$$v(t) = v_0(1-t)^2 + v_1 2(1-t)t + v_2 t^2 \tag{3.41}$$

$$p(t) = p_0(1-t)^2 + p_1 2(1-t)t + p_2 t^2 \tag{3.42}$$

$$q(t) = q_0(1-t)^2 + q_1 2(1-t)t + q_2 t^2 \tag{3.43}$$

Such that the Pythagorean condition:

$$\sigma(t) = u^2(t) + v^2(t) + p^2(t) + q^2(t) \tag{3.44}$$

is satisfied.

Now putting these values of quadratic polynomials $u(t), v(t), p(t)$ and $q(t)$ in equations (3.2), (3.3) and (3.4) expressions for hodographs $x'(t)$, $y'(t)$ and $z'(t)$ are derived in the same manner as that of planar PH quintic. Integrating the resulting equations in the manner similar to that of planar PH quintic the following expressions are obtained for the control points $P_k = (x_k, y_k, z_k)$ for $k = 0,1,2,3,4,5$ of spatial PH quintic [33], with $P_0 = (x_0, y_0, z_0)$ being an arbitrary integration constant.

$$x_1 = x_0 + \frac{1}{5}(u_0^2 + v_0^2 - p_0^2 - q_0^2) \tag{3.45}$$

$$x_2 = x_1 + \frac{1}{5}(u_0 u_1 + v_0 v_1 - p_0 p_1 - q_0 q_1) \tag{3.46}$$

$$x_3 = x_2 + \frac{1}{15}(2u_1^2 + u_0 u_2 + 2v_1^2 + v_0 v_2 - 2p_1^2 - p_0 p_2 - 2q_1^2 - q_0 q_2) \tag{3.47}$$

$$x_4 = x_3 + \frac{1}{5}(u_2 u_1 + v_2 v_1 - p_2 p_1 - q_2 q_1) \tag{3.48}$$

$$x_5 = x_4 + \frac{1}{5}(u_2^2 + v_2^2 - p_2^2 - q_2^2) \tag{3.49}$$

$$y_1 = y_0 + \frac{2}{5}(u_0 q_0 + v_0 p_0) \tag{3.50}$$

$$y_2 = y_1 + \frac{1}{5}(u_0 q_1 + u_1 q_0 + v_0 p_1 + v_1 p_0), \tag{3.51}$$

$$y_3 = y_2 + \frac{1}{15}(u_0 q_2 + 4u_1 q_1 + u_2 q_0 + v_0 p_2 + 4v_1 p_1 + v_2 p_0), \tag{3.29}$$

$$y_4 = y_3 + \frac{1}{5}u_1 q_2 = u_2 q_1 + v_1 p_2 + v_2 p_1), \tag{3.52}$$

$$y_5 = y_4 + \frac{2}{5}(u_2 q_2 + v_2 p_2), \tag{3.53}$$

$$z_1 = z_0 + \frac{2}{5}(v_0 q_0 - u_0 p_0), \tag{3.54}$$

$$z_2 = z_1 + \frac{1}{5}(v_0 q_1 + v_1 q_0 - u_1 p_0), \tag{3.55}$$

$$z_3 = z_2 + \frac{1}{15}(v_0 q_2 + 4v_1 q_1 + v_2 q_0 - u_0 p_2 - 4u_1 p_1 - u_2 p_0), \tag{3.56}$$

$$z_4 = z_3 + \frac{1}{5}(v_1 q_2 + v_2 q_1 - u_1 p_2 - u_2 p_1), \tag{3.57}$$

$$z_5 = z_4 + \frac{2}{5}(v_2 q_2 - u_2 p_2), \tag{3.58}$$

The control points of spatial PH quintic defined by equations (3.45)-(3.58) can satisfy 15 interpolation conditions, since it is defined by 15 variables-The coefficients of $u(t), v(t), p(t)$ and $q(t)$ and the initial condition $P_0(x_0, y_0, z_0)$. The Hermit interpolation of data of the form:

$$(x_{si},\, y_{si},\, z_{si}) = r(0),\ (dx_{si},\, dy_{si},\, dz_{si}) = r'(0),$$

and

$$(x_{fi},\, y_{fi},\, z_{fi}) = r(1),\ (dx_{fi},\, dy_{fi},\, dz_{fi}) = r'(1)$$

gives the following systems of equations:

$$u_0^2 + v_0^2 - p_0^2 - q_0^2 = dx_{si} \tag{3.59}$$

$$2(u_0 q_0 + v_0 p_0) = dy_{si} \tag{3.60}$$

$$2(v_0 q_0 - u_0 p_0) = dz_{si} \tag{3.61}$$

$$u_2^2 + v_2^2 - p_2^2 - q_2^2 = dx_{fi}, \tag{3.62}$$

$$2(u_2 q_2 + v_2 p_2) = dy_{fi}, \tag{3.63}$$

$$2(v_2 q_2 - u_2 p_2) = dz_{fi}, \tag{3.64}$$

$$\begin{aligned} 3(u_o u_1 + v_0 v_1 - p_0 p_1 - q_0 q_1) + (2u_1^2 + 2v_1^2 - 2p_1^2 - 2q_1^2 + u_0 u_2 + v_0 v_2 - q_0 q_2) \\ + 3(u_1 u_2 + v_1 v_2 - p_1 p_2 - q_1 q_2 = 15(x_{fi} - x_{si}) - 3(dx_{si} + dx_{fi}), \end{aligned} \tag{3.65}$$

$$3(u_0q_1+u_1q_0+v_0p_1+v_1p_0)+(u_0q_2+4u_1q_1+u_2q_0+v_0p_2+4v_1p_1+v_2p_0)$$
$$+3(u_1q_2+u_2q_1+v_1p_2+v_2p_1)=15(y_{fi}-y_{si})-3(dy_{si}+dy_{fi}), \quad (3.66)$$

$$3(v_0q_1+v_1q_0-u_op_1-u_1p_0)+(v_0q_2+4v_1q_1+v_2q_0-u_0p_2-4u_1p_1-u_2p_0)$$
$$+3(v_1q_2+v_2q_1-u_1p_2-u_2p_1)=15(z_{fi}-z_{si})-3(dz_{fi}+dz_{fi}) \quad (3.67)$$

Conditions (2.59)-(3.67) amount to a system of nine coupled quadratic equations for twelve unknowns.

Since calculating the control point for PH quintic results in nine coupled quadratic equation in twelve real unknowns [33], therefore it becomes very cumbersome and complicated. The Quaternions [36, 37, 38] produces the control points of PH quintic in an elegant manner and therefore used to calculate the control points for spatial PH quintic.

3.3 Quaternions Based Control points Calculation for Spatial Pythagorean Hodograph

Before going into the process of control points calculation based on Quaternion for spatial Pythagorean Hodograph curves, two lemmas are derived first, which will provide the milestones for the job ahead.

Lemma:1 For any vector say v and quaternion u the form uvu^* is always equivalent to the system of three quadratic scalar equations (u^* being the conjugate of u):

$$u_o^2+u_x^2-u_y^2-u_z^2=\lambda$$

$$2(u_ou_z+u_xu_y)=\mu$$

$$2(u_xu_z-u_ou_y)=\nu$$

Proof*:*

Writing:

$$u=u_o+u_xi+u_yj+u_zk \quad (3.68)$$

And $$v=\lambda i+\mu j+\nu k \quad (3.69)$$

Since $v=uiu^*$

Therefore $\lambda i+\mu j+\nu k=(u_o+u_xi+u_yj+u_zk)i(u_o+u_xi+u_yj+u_zk)^*$

$$=(u_o+u_xi+u_yj+u_zk)i(u_o-u_xi-u_yj-u_zk)$$

$$=(-u_x+u_oi+u_zj-u_yk)(u_o-u_xi-u_yj-u_zk)$$

$$= (-u_x + u_o i + u_z j - u_y k)(u_o - u_x i - u_y j - u_z k)$$

$$\lambda i + \mu j + \nu k = (-u_o u_x + u_o u_x - u_y u_z + u_y u_z) + (u_0^2 + u_x^2 + u_y^2 + u_z^2)i +$$
$$(u_x u_y + u_o u_z + u_x u_y - u_o u_z) j + (u_x u_z - u_o u_y + u_o u_y + u_x u_z)k$$

After the terms cancellation:

$$\lambda i + \mu j + \nu k = (u_o^{\,2} + u_x^{\,2} - u_y^{\,2} - u_z^{\,2})i + 2(u_o u_z + u_x u_y) j + 2(u_x u_z + u_x u_y)k \qquad (3.70)$$

Comparing both sides of the equation (3.70) gives:

$$u_o^{\,2} + u_x^{\,2} - u_y^{\,2} - u_z^{\,2} = \lambda \qquad (3.71)$$

$$2(u_o u_z + u_x u_y) = \mu \qquad (3.72)$$

$$2(u_x u_z - u_o u_y) = \nu \qquad (3.73)$$

Lemma2 : The Solution of $u\,i\,u^*$ is $u = \sqrt{\frac{1}{2}(1+\lambda)}\left[i + \frac{u}{1+\lambda} j + \frac{\lambda}{1+\lambda} k \right]$

Proof:

Suppose for simplicity, co-ordinates are chosen so that the first unit vector u coincides with $\boldsymbol{i}$. The second unit vector v has a general orientation. The quaternion solution u to the equation:

$$u\,i\,u^* = v \qquad (3.74)$$

specifies a spatial rotation of i into v.

For any vector say v and quaternion u the form uvu^* always yields a pure vector. Writing u as $u = u_o + u_x i + u_y j + u_z k$ and $v = \lambda i + \mu j + \nu k$, $u\,i\,u^*$ is equivalent to the system of three quadratic scalar equations by *lemma 1*:

$$u_o^{\,2} + u_x^{\,2} - u_y - u_z^{\,2} = \lambda \qquad (3.75)$$

$$2(u_o u_z + u_x u_y) = \mu \qquad (3.76)$$

$$2(u_x u_z - u_o u_y) = \nu \qquad (3.77)$$

in four unknowns (u_o, u_x, u_y, u_z).

Squaring equations (3.75), (3.76) and (3.77) and adding gives after simplification:

$$(u_o^{\,2} + u_x^{\,2} + u_y^{\,2} + u_z^{\,2})^2 = \lambda^2 + \mu^2 + \gamma^2$$

$$(u_o^{\,2} + u_x^{\,2} + u_y^{\,2} + u_z^{\,2})^2 = 1 \qquad (3.78)$$

The unit quaternion is automatically satisfied.

Putting $u_o = 0$, a particular solution to the above system of equation can be obtained as:

$$u_x^{\,2} - u_y^{\,2} - u_z^{\,2} = \lambda \tag{3.79}$$

$$2u_x u_y = \mu \tag{3.80}$$

$$2u_x u_z = \gamma \tag{3.81}$$

Implies:

$$u_y = \frac{\mu}{2u_x} \tag{3.82}$$

$$u_z = \frac{\nu}{2u_x} \tag{3.83}$$

Putting in equation (3.79):

$$u_x^{\,2} - (\frac{\mu}{2u_x})^2 - (\frac{\nu}{2u_x})^2 = \lambda$$

$$u_x^{\,2} - \frac{\mu^2}{4u_x^{\,2}} - \frac{\nu^2}{4u_x^{\,2}} = \lambda$$

$$4u_x^{\,4} - 4\lambda\, u_x^{\,2} - (\mu^2 + \nu^2) = 0$$

$$u_x^{\,2} = \frac{\lambda + \sqrt{\lambda^2 + u^2 + \gamma^2}}{2}$$

$$4u_x^{\,4} - 4\lambda\, u_x^{\,2} - (\mu^2 + \nu^2) = 0$$

$$u_x^{\,2} = \frac{4\lambda + \sqrt{16\lambda^2 + 4\times 4(\mu^2 + \nu^2)}}{8}$$

$$u_x^{\,2} = \frac{\lambda + \sqrt{\lambda^2 + u^2 + \gamma^2}}{2}$$

Since $\lambda^2 + \mu^2 + \gamma^2 = 1$

Therefore $u_x^{\,2} = \frac{\lambda + 1}{2}$ (3.84)

$$u_x = \sqrt{\frac{1}{2}(\lambda + 1)} \tag{3.85}$$

$$u_y = \frac{\mu}{\sqrt{\frac{1}{2}(\lambda+1)}} \tag{3.86}$$

Similarly: $$u_z = \frac{\nu}{\sqrt{\frac{1}{2}(\lambda+1)}} \tag{3.87}$$

Now putting these results of u_x, u_y and u_z in equation (3.68) i.e. $u = u_x i + u_y j + u_z k$:

$$u = \sqrt{\frac{1}{2}(\lambda+1)}\,(i + \frac{\mu}{1+\lambda} j + \frac{\nu}{1+\lambda} k) \tag{3.88}$$

The most general solution to equation (3.74):

$$\boldsymbol{u\, i\, u^*} = \boldsymbol{\nu}$$

Can thus be parameterized in term of the variable ζ as given in [33]:

$$u(\zeta) = \sqrt{\frac{1}{2}(1+\lambda)}\left(-\sin\zeta + \cos\phi i + \frac{\mu\cos\zeta + \nu\sin\zeta}{1+\lambda} j + \frac{\nu\cos\zeta - \mu\sin\zeta}{1+\lambda} k\right) \tag{3.89}$$

In the subsequent sections the expression for the control points of spatial Pythagorean hodographs will be derived with the help of these lemmas and then the expression already derived for PH curves in 2D will be used to calculate the spatial PH curve. The only difference is: the control points in 2D were comprised of x and y components only, while the control points in 3D are composed of three components i.e. x, y and z.

3.3.1 Quaternion Based Control Points for Spatial Cubic Pythagorean Hodograph

Although this research is mainly concerned with the spatial Pythagorean hodograph quintic but as a starting point, first the control points for cubic is calculated using Quaternions. This makes the whole process of Quaternion based control points calculation easy to understand and also serve the purpose of completeness. Therefore in this section the Quaternion based control points for spatial PH cubic are calculated using *lemma 1*, and *Lemma 2*.

For any Quaternion polynomial written as:

$$A(t) = u(t) + v(t)i + p(t)\,j + q(t)k \tag{3.90}$$

The appropriate quaternion form for the spatial Pythagorean hodograph (derivative of quaternion polynomial) is produced by [39]:

$$r'(t) = A(t)\; i\; A^*(t)$$

$$
\begin{aligned}
&= \left[u^2(t) + v^2(t) - p^2(t) - q^2(t)\right] i \\
&\quad + 2\left[u(t)q(t) - v(t)\,p(t)\right] j \\
&\quad + 2\left[v(t)q(t) - u(t)\,p(t)\right] k
\end{aligned} \tag{3.91}
$$

Where A^* is the conjugate of A.

To define spatial PH cubic, consider a linear Quaternion polynomial

$$A(t) = A_0(1-t) + A_1 t \tag{3.92}$$

This gives the control points in term of Quaternion co-efficient A_0, A_1 [33]

$$P_1 = P_0 + \frac{1}{3} A_0\, i\, A_0^* \tag{3.93}$$

$$P_2 = P_1 + \frac{1}{6}(A_0\, i\, A_1^* + A_1\, i\, A_0^*) \tag{3.94}$$

$$P_3 = P_2 + \frac{1}{3} A_1\, i\, A_1^* \tag{3.95}$$

Where P_0 corresponds to the integration constant.

Since the points P_0 and P_3 and the derivatives at the points P_0 and P_3 are known from the known initial and final poses, therefore A_0 and A_1 can be determined from equation (3.93) and (3.95) using equation (3.89) and *Lemma 2.* These values of A_0 and A_1 can be substituted in equation (3.94) to calculate the values of control points P_1, P_2. Once the control points P_0, P_1, P_2 and P_3 are known it can be substituted in the equation (3.26) i.e.

$$r(t) = \begin{bmatrix} 1 & t & t^2 & t^3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -3 & 3 & 0 & 0 \\ 3 & -6 & 3 & 0 \\ -1 & 3 & -3 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

to calculate the spatial PH cubic curve $r(t)$.

Now Comparing the Quaternion way of calculating the unknown control points and the algebraic way to do the same, for cubic PH curve, it is clear that the Quaternion form offer much cleaner and elegant solution.

3.3.2 Qaternion Based Control Points for Spatial Quintic Pythagorean Hodograph

After deriving control points for PH cubic using Quaternions, control points of spatial PH quintic in Quaternion form are tried to calculate using *lemma 1* and *lemma 2* in this section.

To define spatial PH quintic, consider a quadratic polynomial:

$$A(t) = A_0(1-t)^2 + A_1.2(1-t)t + A_2t^2 \tag{3.96}$$

With Quaternion coefficients:

$$\begin{aligned} A_r &= u_r + v_r i + p_r j + q_r k \\ for \quad r &= 0,1,2. \end{aligned} \tag{3.97}$$

In Quaternion form the spatial Pythagorean hodograph

$$r'(t) = [u^2(t) + v^2(t) - p^2(t) - q^2(t)]i + 2[u(t)q(t) + v(t)p(t)]j + 2[v(t)q(t) - u(t)p(t)]k \tag{3.98}$$

Can be expressed in term of $A(t)$ as follow:

$$r'(t) = A(t)iA^*(t) \tag{3.99}$$

This gives the control points of the PH quintic curve in quaternion form, $P_r = x_r, +y_r j + z_r k$ by the following formulae as given in [33]:

$$P_1 = P_0 + \frac{1}{5}A_0 iA_0^* \tag{3.100}$$

$$P_2 = P_1 + \frac{1}{10}(A_0 iA_1^* + A_1 iA_0^*) \tag{3.101}$$

$$P_3 = P_2 + \frac{1}{30}(A_0 iA_2^* + 4A_1 iA_1^* + A_2 iA_0^*) \tag{3.102}$$

$$P_4 = P_3 + \frac{1}{10}(A_1 iA_2^* + A_2^* iA_0^*) \tag{3.103}$$

$$P_5 = P_4 + \frac{1}{5}A_2 iA_2^* \tag{3.104}$$

Where $P_0 = p_{si} = (x_{si},\ y_{si},\ z_{si})$ is the initial point, and $P_5 = p_{fi} = (x_{fi},\ y_{fi},\ z_{fi})$ is the final point. The coefficients in equation (3.96) are to be determined by solving Hermit interpolation problem. Once A_0, A_1 and A_2 are known, the control points of the PH quintic curve can be easily calculated from equations (3.100)-(3.104).

To solve the Hermit interpolation problem, the starting position of the *ith* UAV in the group, $p_{si} = (x_{si},\ y_{si},\ z_{si})$, and the derivative at the starting position $dp_{si} = (dx_{si},\ dy_{si},\ dz_{si})$, are expressed as pure vector Quaternion as follow:

$$(x_{si},\ y_{si},\ z_{si}) = x_{si}i + y_{si}j + z_{si}k \tag{3.105}$$

$$(dx_{si},\ dy_{si},\ dz_{si}) = dx_{si}i + dy_{si}j + dz_{si}k \tag{3.106}$$

Similarly expressing the final position of the *ith* UAV, $p_{fi} = (x_{fi},\ y_{fi},\ z_{fi})$ and the derivative at the final position $dp_{fi} = (dx_{fi},\ dy_{fi},\ dz_{fi})$, as pure vector Quaternion:

$$(x_{fi},\, y_{fi},\, z_{fi}) = x_{fi}i + y_{fi}j + z_{fi}k \tag{3.107}$$

$$(dx_{fi},\, dy_{fi},\, dz_{fi}) = dx_{fi}i + dy_{fi}j + dz_{fi}k \tag{3.108}$$

By equation (3.99), the end derivatives dp_{si} and dp_{fi} can be written as:

$$A_0 i A_0^* = dp_{si} \tag{3.109}$$

$$A_2 i A_2^* = dp_{fi} \tag{3.110}$$

By *lemma 2,* from equation (3.109) and (3.110), A_0, A_2 can be written as:

$$A_o = \sqrt{\frac{1}{2}(1+\lambda_{si})\left|dp_{si}\right|}\ (-\sin\zeta_0 + \cos\zeta_0 i + \frac{\mu_{si}\cos\zeta_0 + \nu_{si}\sin\zeta_0}{1+\lambda_{si}}\, j + \frac{\nu_{si}\cos\zeta_0 - \mu_{si}\sin\zeta_0}{1+\lambda_{si}}\, k) \tag{3.111}$$

$$A_2 = \sqrt{\frac{1}{2}(1+\lambda_{fi})\left|dp_{fi}\right|}\ (-\sin\zeta_2 + \cos\zeta_2 i + \frac{\mu_{fi}\cos\zeta_2 + \nu_{fi}\sin\zeta_2}{1+\lambda_{fi}}\, j + \frac{\nu_{fi}\cos\zeta_2 - \mu_{fi}\sin\zeta_2}{1+\lambda_{fi}}) \tag{3.112}$$

Where $(\lambda_{si},\, \mu_{si},\, \nu_{si})$ and $(\lambda_{fi},\, \mu_{fi},\, \nu_{fi})$ are the direction cosine of dp_{si}, dp_{fi} respectively and ζ_0, ζ_2 are the free angular variables.

For determining A_1, the interpolation of the end point with $p_{si} = (x_{si},\, y_{si},\, z_{si})$ as integration constant gives the condition:

$$\int_0^1 A(t) i A^*(t) dt = p_{fi} - p_{si} \tag{3.113}$$

The position vector $(p_{fi} - p_{si})$ can be written as the summation of the vectors from initial point $p_{si} = (x_{si},\, y_{si},\, z_{si})$ to final point $p_{fi} = (x_{fi},\, y_{fi},\, z_{fi})$ along the lines connecting the control points $P_o, P_1, P_2, P_3, P_4, P_5$ as follow:

$$p_{fi} - p_{si} = \overrightarrow{P_0P_1} + \overrightarrow{P_1P_2} + \overrightarrow{P_2P_3} + \overrightarrow{P_3P_4} + \overrightarrow{P_4P_5} \tag{3.114}$$

Alternatively each vector can be written as the difference of the initial and final point, therefore equation (3.114) can be written as follows:

$$p_{fi} - p_{si} = P_1 - P_0 + P_2 - P_1 + P_3 - P_2 + P_4 - P_3 + P_3 + P_5 - P_4 \tag{3.115}$$

Putting the values of control points $P_0, P_1, P_2, P_3, P_4, P_5$ in equation (3.115) gives:

$$\begin{aligned} p_{fi} - p_{si} = {} & P_0 + \frac{1}{5}A_0 i A_0^* - P_0 + P_1 + \frac{1}{10}(A_0 i A_1^* + A_1 i A_0^*) - P_1 \\ & + P_2 + \frac{1}{30}(A_0 i A_2^* + 4A_1 i A_1^* + A_2 i A_0^*) - P_2 + P_3 + \frac{1}{10}(A_1 i A_2^* + A_2 i A_1^*) - P_3 \\ & + P_4 + \frac{1}{5}A_2 i A_2^* - P_4 \end{aligned} \tag{3.116}$$

After cancellation of the terms equation (3.116) yields:

$$p_{fi} - p_{si} = \frac{1}{5} A_0 iA_0^* + \frac{1}{10}(A_0 iA_1^* + A_1 iA_0^*) + \frac{1}{30}(A_0 iA_2^* + 4A_1 iA_1^* + A_2 iA_0^*) + \frac{1}{10}(A_1 iA_2^* + A_2 iA_1^*) + \frac{1}{5} A_2 iA_2^* \tag{3.117}$$

$$= \frac{24A_0 iA_0^* + 12(A_0 iA_1^* + A_1 iA_0^*) + 4(A_0 iA_2^* + 4A_1 iA_1^* + A_2 iA_0^*) + 12(A_1 iA_2^* + A_2 iA_1^*) + 24A^2 iA_2^*}{120}$$

$$120(p_{fi} - p_{si}) = 24A_0 iA_0^* + 12(A_0 iA_1^* + A_1 iA_0^*) + 4(A_0 iA_2^* + 4A_1 iA_1^* + A_2 iA_0^*) + 12(A_1 iA_2^* + A_2 iA_1^*) + 24A^2 iA_2^* \tag{3.118}$$

Subtracting $15(A_0 iA_0^* + A_2 iA_2^*)$ from both sides of equation (3.118):

$$120(p_{fi} - p_{si}) - 15(A_0 iA_0^* + A_2 iA_2^*) = 9A_0 iA_0^* + 12(A_0 iA_1^* + A_1 iA_0^*) + 4(A_0 iA_2^* + 4A_1 iA_1^* + A_2 iA_0^*) + 12(A_1 iA_2^* + A_2 iA_1^*) + 9A^2 iA_2^* \tag{3.119}$$

Adding $5(A_0 iA_2^* + A_2 iA_0^*)$ to both sides of equation (3.119)

$$120(p_{fi} - p_{si}) - 15(dp_{si} + dp_{fi}) + 5(A_0 iA_2^* + A_2 iA_0^*) = 9A_0 iA_0^* + 12(A_0 iA_1^* + A_1 iA_0^*) + 4(A_0 iA_2^* + 4A_1 iA_1^* + A_2 iA_0^*) + 12(A_1 iA_2^* + A_2 iA_1^*) + 9A^2 iA_2^* + 5(A_0 iA_2^* + A_2 iA_0^*) \tag{3.120}$$

Expanding (3.120) gives:

$$120(p_{fi} - p_{si}) - 15(dp_{si} + dp_{fi}) + 5(A_0 iA_2^* + A_2 iA_0^*) = 9A_0 iA_0^* + 12A_0 iA_1^* + 12A_1 iA_0^* + 4A_0 iA_2^* + 16A_1 iA_1^* + 4A_2 iA_0^* + 12A_1 iA_2^* + 12A_2 iA_1^* + 9A^2 iA_2^* + 5A_0 iA_2^* + 5A_2 iA_0^* \tag{3.121}$$

Grouping the terms and taking common (3.121):

$$120(p_{fi} - p_{si}) - 15(dp_{si} + dp_{fi}) + 5(A_0 iA_2^* + A_2 iA_0^*) = 3A_0(3iA_0^* + 4iA_1^* + 3iA_2^*) + 4A_1(3iA_0^* + 4iA_1^* + 3iA_2^*) + 3A_2(3iA_0^* + 4iA_1^* + 3iA_2^*) \tag{3.122}$$

$$120(p_{fi} - p_{si}) - 15(dp_{si} + dp_{fi}) + 5(A_0 iA_2^* + A_2 iA_0^*) = (3A_0 + 4A_1 + 3A_2)(3iA_0^* + 4iA_1^* + 3iA_2^*)$$

$$120(p_{fi} - p_{si}) - 15(dp_{si} + dp_{fi}) + 5(A_0 iA_2^* + A_2 iA_0^*) = (3A_0 + 4A_1 + 3A_2)\, i\, (3A_0 + 4A_1 + 3A_2)^* \tag{3.123}$$

Equation (3.123) has the form $A\, i\, A^*$ if $A = (3A_0 + 4A_1 + 3A_2)$

In equation (3.123) considering the term $(A_0 iA_2^* + A_2 iA_0^*)$ in the expression $5(A_0 iA_2^* + A_2 iA_0^*)$:

Since $(A_2 iA_0^*) = (A_0 iA_2^*)^*$

Therefore $(A_0^* \, i \, A_2^* + A_2^* \, i \, A_0^*) = (A_0 iA_2^*) + (A_0 iA_2^*)^*$ = twice the vector part of $A_0 iA_2^*$

Hence:

$$(A_0 \, i \, A_2^* + A_2 \, i \, A_0^*) = 2 \times A_0 iA_2^* \tag{3.124}$$

Putting equation (3.111) and (3.112) in expression $2 \times A_0 \, i \, A_2^*$ yields:

$$2 \times A_0 \, i \, A_2^* = 2 \times \sqrt{\frac{1}{2}(1+\lambda_{si})|dp_{si}|}\left[(-\sin\zeta_o + \cos\zeta_o i + \frac{\mu_{si}\cos\zeta_o + \nu_{si}\sin\zeta_o}{1+\lambda_{si}} j + \frac{\nu_{si}\cos\zeta_o - \mu_{si}\sin\zeta_o}{1+\lambda_{si}} k)\right] i$$
$$\sqrt{\frac{1}{2}(1+\lambda_{fi})|dp_{fi}|}\left[(-\sin\zeta_2 + \cos\zeta_2 i + \frac{\mu_{fi}\cos\zeta_2 + \nu_{fi}\sin\zeta_2}{1+\lambda_{fi}} j + \frac{\nu_{fi}\cos\zeta_2 - \mu_{fi}\sin\zeta_2}{1+\lambda_{fi}})\right]$$

$$2 \times A_0 \, i \, A_2^* = \sqrt{(1+\lambda_{si})|dp_{si}|(1+\lambda_{fi})|dp_{fi}|}$$
$$\left[\begin{array}{l}\left[\cos(\zeta_2-\zeta_0) - \frac{(\mu_{si}\mu_{fi} - \nu_{si}\nu_{fi})\cos(\zeta_2-\zeta_0) + (\mu_{si}\nu_{fi} - \mu_{si}\nu_{fi})\sin(\zeta_2-\zeta_0)}{(1+\lambda_{si})(1+\lambda_{fi})}\right] i \\ + \left[\frac{\mu_{si}\cos(\zeta_2-\zeta_0) - \nu_{si}\sin(\zeta_2-\zeta_0)}{(1+\lambda_{si})} + \frac{\mu_{fi}\cos(\zeta_2-\zeta_0) + \nu_{fi}\sin(\zeta_2-\zeta_0)}{(1+\lambda_{fi})}\right] j \\ + \left[\frac{\nu_{si}\cos(\zeta_2-\zeta_0) + \mu_{si}\sin(\zeta_2-\zeta_0)}{(1+\lambda_{si})} + \frac{\nu_{fi}\cos(\zeta_2-\zeta_0) - \mu_{fi}\sin(\zeta_2-\zeta_0)}{(1+\lambda_{fi})}\right] k\end{array}\right]$$

Therefore by equation (3.124):

$$A_0 \, i \, A_2^* + A_2 \, i \, A_0^* = \sqrt{(1+\lambda_{si})|dp_{si}|(1+\lambda_{fi})|dp_{fi}|}$$
$$\left[\begin{array}{l}\left[\cos(\zeta_2-\zeta_0) - \frac{(\mu_{si}\mu_{fi} - \nu_{si}\nu_{fi})\cos(\zeta_2-\zeta_0) + (\mu_{si}\nu_{fi} - \mu_{si}\nu_{fi})\sin(\zeta_2-\zeta_0)}{(1+\lambda_{si})(1+\lambda_{fi})}\right] i \\ + \left[\frac{\mu_{si}\cos(\zeta_2-\zeta_0) - \nu_{si}\sin(\zeta_2-\zeta_0)}{(1+\lambda_{si})} + \frac{\mu_{fi}\cos(\zeta_2-\zeta_0) + \nu_{fi}\sin(\zeta_2-\zeta_0)}{(1+\lambda_{fi})}\right] j \\ + \left[\frac{\nu_{si}\cos(\zeta_2-\zeta_0) + \mu_{si}\sin(\zeta_2-\zeta_0)}{(1+\lambda_{si})} + \frac{\nu_{fi}\cos(\zeta_2-\zeta_0) - \mu_{fi}\sin(\zeta_2-\zeta_0)}{(1+\lambda_{fi})}\right] k\end{array}\right] \tag{3.125}$$

Putting equation (3.125) in equation (3.123) yields:

$$(3A_0 + 4A_1 + 3A_2)\,i\,(3A_0 + 4A_1 + 3A_2)^* =$$
$$= 120(P_{fi} - P_{si}) - 15(d_{ii} + d_{fi}) + 5\sqrt{(1+\lambda_{si})|dp_{si}|(1+\lambda_{fi})|dp_{fi}|}$$
$$\left[\begin{array}{l} \left[\cos(\zeta_2 - \zeta_0) - \dfrac{(\mu_{si}\mu_{fi} - \nu_{si}\nu_{fi})\cos(\zeta_2 - \zeta_0) + (\mu_{si}\nu_{fi} - \mu_{si}\nu_{fi})\sin(\zeta_2 - \zeta_0)}{(1+\lambda_{si})(1+\lambda_{fi})}\right] i \\ + \left[\dfrac{\mu_{si}\cos(\zeta_2 - \zeta_0) - \nu_{si}\sin(\zeta_2 - \zeta_0)}{(1+\lambda_{si})} + \dfrac{\mu_{fi}\cos(\zeta_2 - \zeta_0) + \nu_{fi}\sin(\zeta_2 - \zeta_0)}{(1+\lambda_{fi})}\right] j \\ + \left[\dfrac{\nu_{si}\cos(\zeta_2 - \zeta_0) + \mu_{si}\sin(\zeta_2 - \zeta_0)}{(1+\lambda_{si})} + \dfrac{\nu_{fi}\cos(\zeta_2 - \zeta_0) - \mu_{fi}\sin(\zeta_2 - \zeta_0)}{(1+\lambda_{fi})}\right] k \end{array}\right] \quad (3.126)$$

Writing the RHS of equation (3.126) as:

$$c = c_x i + c_y j + c_z k$$

yields:

$$(3A_0 + 4A_1 + 3A_2)\,i\,(3A_0 + 4A_1 + 3A_2)^* = c_x i + c_y j + c_z k \quad (3.127)$$

By *lemma 2* it is deduced that the solution of $(3A_0 + 4A_1 + 3A_2)\,i\,(3iA_0 + 4iA_1 + 3iA_2)^*$ is:

$$(3A_0 + 4A_1 + 3A_2) = \sqrt{\frac{1}{2}(1+\lambda)|c|}\left(-\sin\zeta_1 + \cos\zeta_1 i + \frac{\mu\cos\zeta_1 + \nu\sin\zeta_1}{1+\lambda}\,j + \frac{\nu\cos\zeta_1 - \mu\sin\zeta_1}{1+\lambda}\,k\right)$$

$$A_1 = -\frac{3}{4}(A_0 + A_2) + \frac{1}{4}\sqrt{\frac{1}{2}(1+\lambda)|c|}\left(-\sin\zeta_1 + \cos\zeta_1 i + \frac{\mu\cos\zeta_1 + \nu\sin\zeta_1}{1+\lambda}\,j + \frac{\nu\cos\zeta_1 - \mu\sin\zeta_1}{1+\lambda}\,k\right) \quad (3.128)$$

Where (λ, μ, ν) are the direction cosines of c and ζ_1 is another free angular variable. The values of A_0, A_1 and A_2 given by equations (3.111), (3.112) and (3.128) are substituted in equations (3.100)-(3.104) to find the values of control points P_1, P_2, P_3, P_4 for given values of P_0 and P_5. Once the control points are known they can be substituted in the following equation:

$$r(t) = \begin{bmatrix} 1 & t & t^2 & t^3 & t^4 & t^5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ -5 & 5 & 0 & 0 & 0 & 0 \\ 10 & -20 & 10 & 0 & 0 & 0 \\ -10 & 30 & -30 & 10 & 0 & 0 \\ 5 & 20 & 30 & -20 & 5 & 0 \\ -1 & 5 & -10 & 10 & -5 & 1 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \\ P_4 \\ P_5 \end{bmatrix}$$

to calculate the spatial PH quintic curve $r(t)$.

3.3.2.1 Least Bending Energy

In [33] it has been proposed that the 3D curve will have a least bending energy if:

$$\zeta_1 = \zeta_2 = \zeta_3 = -\frac{\pi}{2} \tag{3.129}$$

is substituted in equations (3.111), (3.112) and (3.128). In this research minimum bending energy paths for UAVs are produced by obeying equation (3.129). The least bending energy path has the unique attribute of being able to be traced by the UAV with minimum effort.

In order to make the calculated least bending energy curve, a feasible UAV path, the kinematic and safety constraints of the UAV are imposed.

3.3.2.2 Kinematic Constraints

After generating the least energy paths, the paths are made flyable for UAVs by imposing the kinematic constraints of the vehicle. In 3D case, the curvature and torsion constraints are considered to be the kinematic constraints of the vehicle.

3.3.2.2.1 Curvature Constraints

The resulting UAV trajectory has tangential continuity but it should have a curvature small enough such that it is possible for the UAV to trace it. The UAV trajectory is made to obey curvature constraints by increasing the radius of the curve, [27] such that:

$$|\kappa(t)| \le \kappa_{\max} \tag{3.130}$$

$\kappa_{\max}$ is the maximum bound on the curvature the UAV can achieve and $\kappa(t)$ is the trajectory's curvature at time t, given by:

$$\kappa(t) = \frac{|r'(t) \times r''(t)|}{|r'(t)|^3} \tag{3.131}$$

In this research the curvature of the path is changed by manipulating the control points P_1, P_2, P_3 and P_4 iteratively such that the condition (3.130) is satisfied. The first and the last control points P_0, P_5 are kept unchanged. The rest of the control points are changed by increasing the values of constants k_1, k_2 and k_3 iteratively equations (3.132) and equations (3.134) as described in [31].

$$\begin{aligned} dx_{si} &= k_1(\varepsilon_{si} \sin\phi_{si} \cos\theta_{si}) \\ dy_{si} &= k_2(\varepsilon_{si} \sin\phi_{si} \sin\theta_{si}) \\ dz_{si} &= k_3(\varepsilon_{si} \cos\phi_{si}) \end{aligned} \tag{3.132}$$

Where :

$$\varepsilon_{si} = \sqrt{dx_{si}^2 + dy_{si}^2 + dz_{si}^2}$$
$$dx_{si} = x_1 - x_0$$
$$dy_{si} = y_1 - y_0$$
$$dz_{si} = z_1 - z_0 \tag{3.133}$$

(x_0, y_0, z_0) and (x_1, y_1, z_1) being the coordinates of the first and second control points P_0, P_1.

Similarly:

$$dx_{fi} = k_1(\varepsilon_{fi} \sin\phi_{fi} \cos\theta_{fi})$$
$$dy_{fi} = k_2(\varepsilon_{fi} \sin\phi_{fi} \sin\theta_{fi})$$
$$dz_{fi} = k_3(\varepsilon_{fi} \cos\phi_{fi}) \tag{3.134}$$

Where:

$$\varepsilon_{fi} = \sqrt{dx_{fi}^2 + dy_{fi}^2 + dz_{fi}^2}$$
$$dx_{fi} = x_5 - x_4$$
$$dy_{fi} = y_5 - y_4$$
$$dz_{fi} = z_5 - z_4 \tag{3.135}$$

(x_4, y_4, z_4) and (x_5, y_5, z_5) being the coordinates of the fourth and fifth control points P_4, P_5.

3.3.2.2.2 Torsion Constraint

In 3D, the curvature constraint is accompanied by torsion constraint as well. The rate, at which the osculating plane rotates about the tangent vector, is known as torsion. The torsion measures the speed of rotation of the binormal vector at the given point. It is a number that depends on the speed of the particle. The torsion τ is positive for a right-handed curve, and negative for a left-handed curve. A curve with curvature $\kappa \neq 0$ is planar iff $\tau = 0$.

Mathematically the torsion $\tau(t)$ of the curve $r(t)$ at any time t is given by:

$$\tau(t) = \frac{[r'(t) \times r''(t)] \bullet r'''(t)}{|r'(t) \times r''(t)|^2} \tag{3.136}$$

For a path to be flyable, the torsion $\tau(t)$ must obey the constraint:

$$|\tau(t)| \leq \tau_{max} \tag{3.137}$$

In this research it has been achieved in the same way as that of curvature constraint by changing iteratively the control points of the planned path using equation (3.132) and (3.134).

3.3.2.3 Safety Constraints

After achieving flyable UAV paths, safety of the generated paths is addressed. A flyable path is safe if it is free of inter collisions and free of collisions with the obstacles. Such paths are called feasible paths and can be traced by the UAVs safely. The safety constraints are explained in the following Subsections.

3.3.2.3.1 Avoiding collisions with static obstacles

If any of the planned trajectories happens to collide with any of the known obstacles in the map then the path is not safe and it should be amended. The obstacle can be avoided by changing the curvature of the trajectory using equations (3.132) and (3.134) until it is clear of obstacles.

The obstacle is enclosed in a sphere of certain radius. The trajectory is amended by changing the control points such that the set of points of intersection between the path and the sphere is null. Thus, obstacles avoidance of the UAV's path is achieved.

3.3.2.3.2 Avoiding inter collision

UAV paths are checked for inter-collision of vehicles. Paths are searched for the possible points of inter-collision between them. The paths are safe, if the minimum distance between any two UAVs at any time during their travel on these paths is equal to or greater than twice the safety radius R_s :

$$d_{\min} \geq 2R_s \tag{3.138}$$

Where $d_{\min}$ is the minimum distance between the UAVs travelling on the path at any time. R_s is the safety radius of each UAV.

If the above condition is not satisfied then the curvature is changed until that condition gets satisfied.

3.4 Simulation Results

In this simulation a swarm of three UAVs is considered for mission deployment in the environment with known obstacles. All the UAVs will start from certain starting point at the same time and will reach the goal point at the same time. During their flights from starting position to a finishing position all the UAVs will avoid inter-collisions and collisions with the known obstacles. The proposed path planning algorithm will plan safe and flyable flight paths for all the UAVs in the group.

For this simulation, the initial and final positions of the UAVs, the derivatives at these positions, the kinematic constraints and the safety spheres radii of all the UAVs are summarised in table 3.1.

No	UAVs in Swarm	Initial Pose		Final Pose		Kinematic Constraints $\kappa_{\max}, \tau_{\max}$ (metre)	Safety Radii R_s (metre)
		Positions (x_{si}, y_{si}, z_{si}) (metres)	**Derivatives** $dx_{si}, dy_{si}, dz_{si}$	**Positions** (x_{fi}, y_{fi}, z_{fi}) (metres)	**Derivatives** $dx_{fi}, dy_{fi}, dz_{fi}$		
1	UAV1	(0, 0, 0)	0.9, -5, 2	(51, 18,50)	3, -9, -18	$\pm\frac{1}{3}, \pm\frac{1}{3}$	1
2	UAV2	(4, 7, 5)	2, -7, 4	(61,18, 51)	2, 14, 4	$\pm\frac{1}{3}, \pm\frac{1}{3}$	1
3	UAV3	(15, 0, 5)	3, -2, 1	(61, 45,51)	3, 10, 3	$\pm\frac{1}{3}, \pm\frac{1}{3}$	1

Table 3.1: Table of initial conditions, maximum curvature and safety radii.

All the known static obstacles are given in the terrain database. For simplicity, all the obstacles are assumed to be of spherical shape with their centres at some points on UAVs paths in this simulation. The locations of the obstacle are described by the coordinates of their centres and their respective radii. The centres and radii of the obstacle1, obstacle2 and obstacle 3 are given in table 3.2.

No	Obstacles	Centre of the sphere (metres)	Radius of the sphere (metres)
1	Obstacle 1	(9.8, -1.5, 13.3)	0.5
2	Obstacle 2	(15, 2.5, 15.2)	0.5
3	Obstacle 3	(27.3, 3.75, 13.89)	0.5

Table 3.2: Locations of the known obstacles.

The simulation results of generating flyable paths by the proposed path planning algorithm for a group of three cooperating UAVs in three dimensions are shown in the following series of figures.

Figure 3.1 shows three paths (PH quintic in 3D) for three UAVs generated by the proposed path planning algorithm (3D Pythagorean Hodograph path planner). No curvature or torsion constraint is yet imposed. The small circles show the control points, the dotted lines represent the control polygon and the curves represent the three minimum bending energy PH paths for the three UAVs linking their respective initial and final poses.

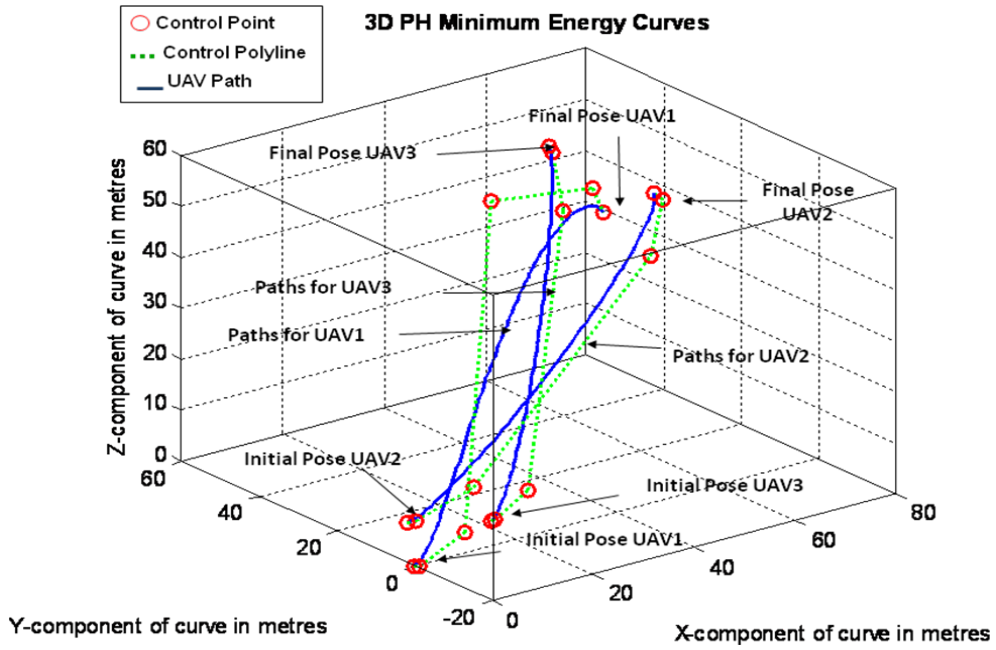

Figure 3.1: Minimum energy curves for UAV1, UAV2 and UAV3 in 3D

The curvature and torsion profiles of the above three paths before imposing the curvature and torsion constraints are shown in the figures 3.2 and 3.3 respectively, which clearly do not obey the curvature and torsion constraints of the vehicles.

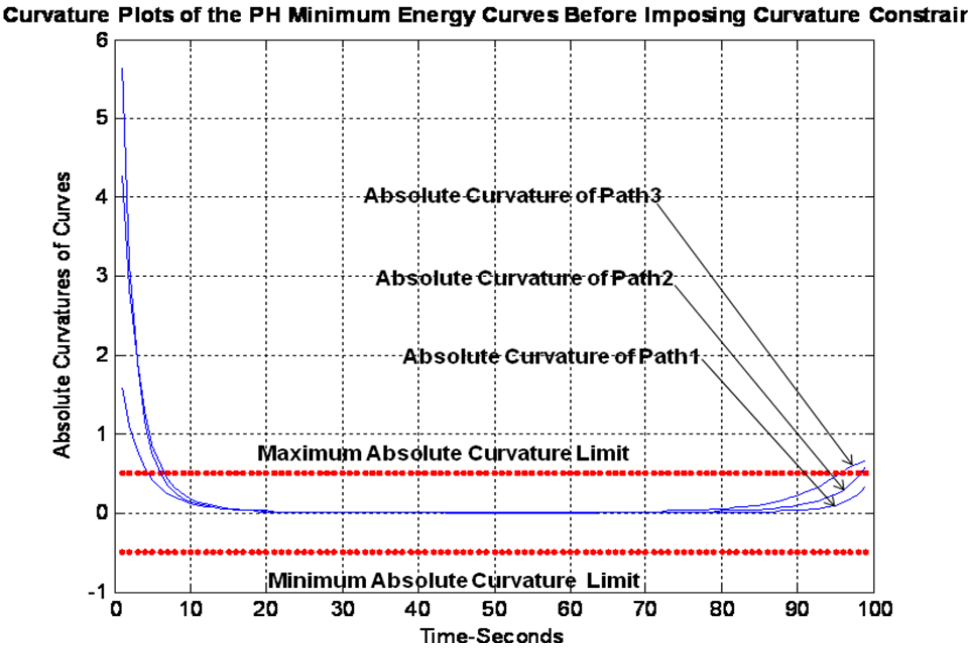

Figure 3.2: Curvature profiles of the paths for UAV1, UAV2 and UAV3 in 3D

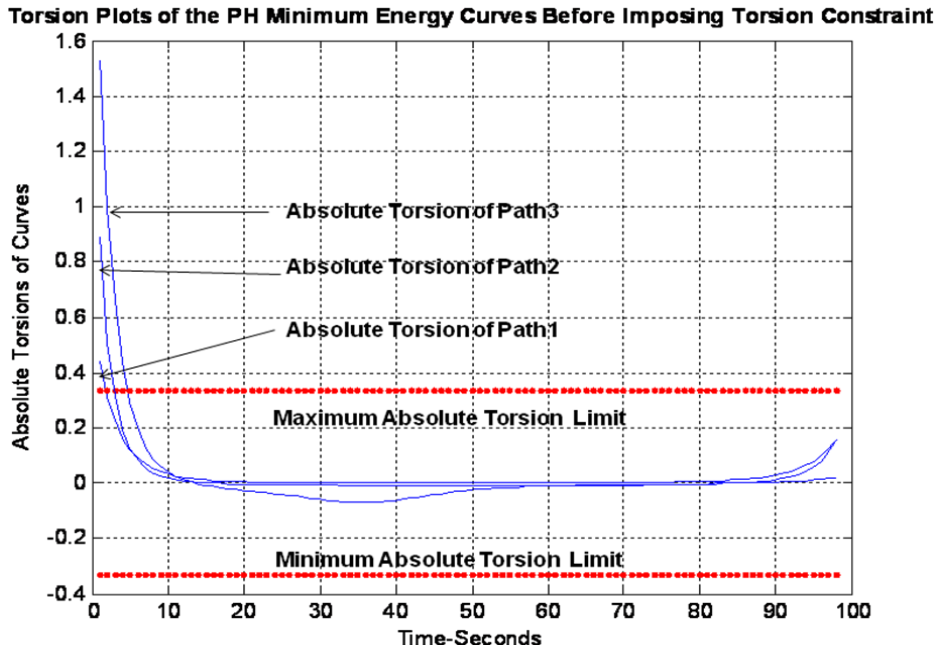

Figure 3.3: Torsion profiles of the paths for UAV1, UAV2 and UAV3 in 3D

Next paths shown in the figure 3.1 are made to obey the curvature and torsion constraints. This results in the paths shown in figure 3.4 and 3.5.

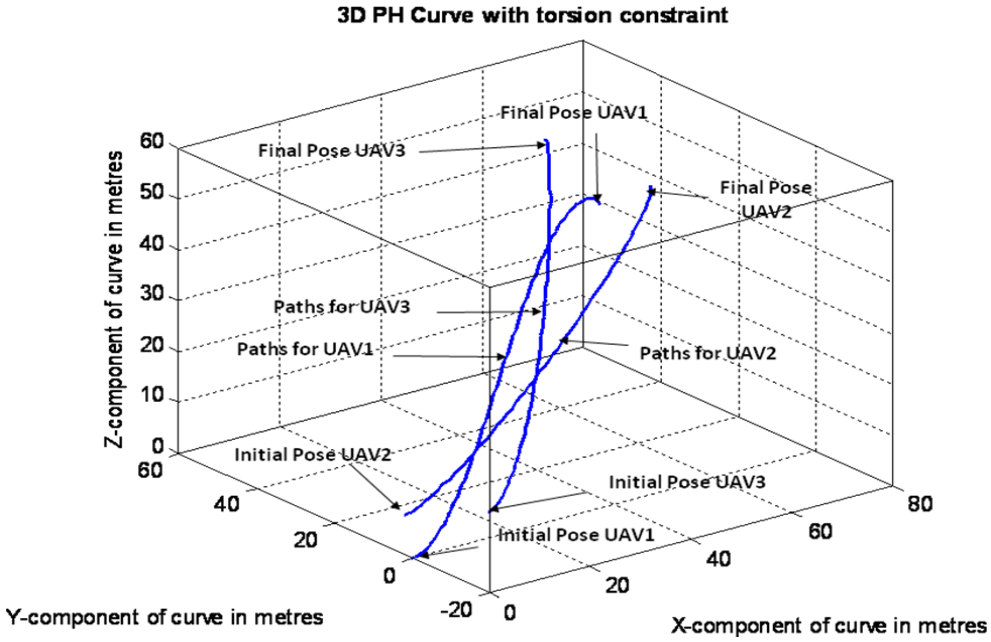

Figure 3.4: Curves obeying torsion constraints.

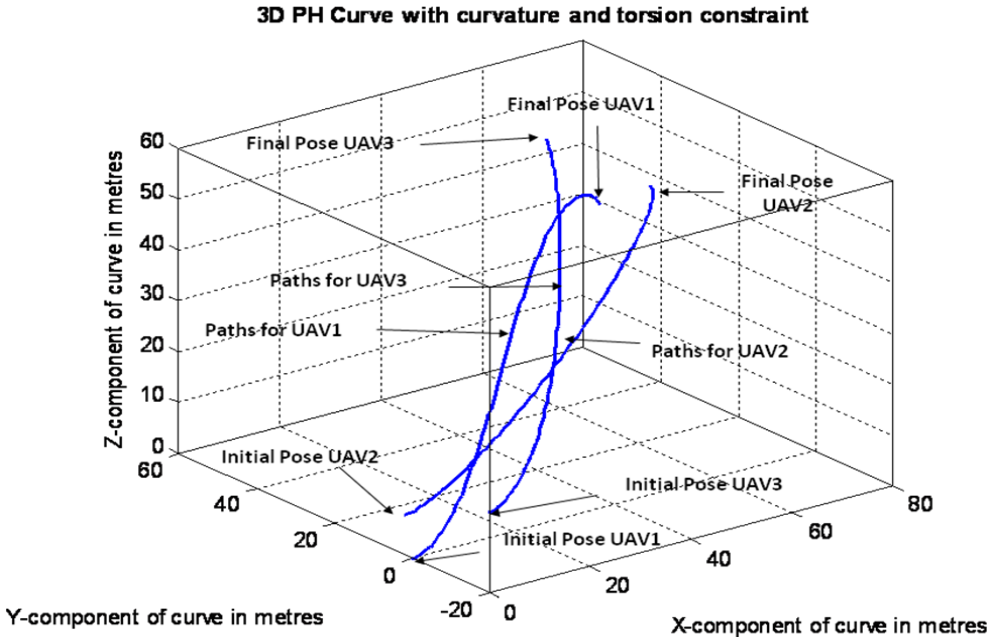

Figure 3.5: Curves obeying torsion and curvature constraints.

The curvature and torsion profiles of the three paths after imposing the torsion and curvature constraints are shown in the figures 3.6 and 3.7, which clearly obey the Kinematic constraints of the vehicles. The maximum and minimum curvature and torsion bounds used for this simulation are:

$$\kappa_{\min} = -\frac{1}{3} = -0.333 \text{ metre}, \ \tau_{\min} = -\frac{1}{3} = -0.333 \text{ metre}$$

$$\kappa_{\max} = +\frac{1}{3} = +0.333 \text{ metre}, \ \tau_{\max} = +\frac{1}{3} = +0.333 \text{ metre}$$

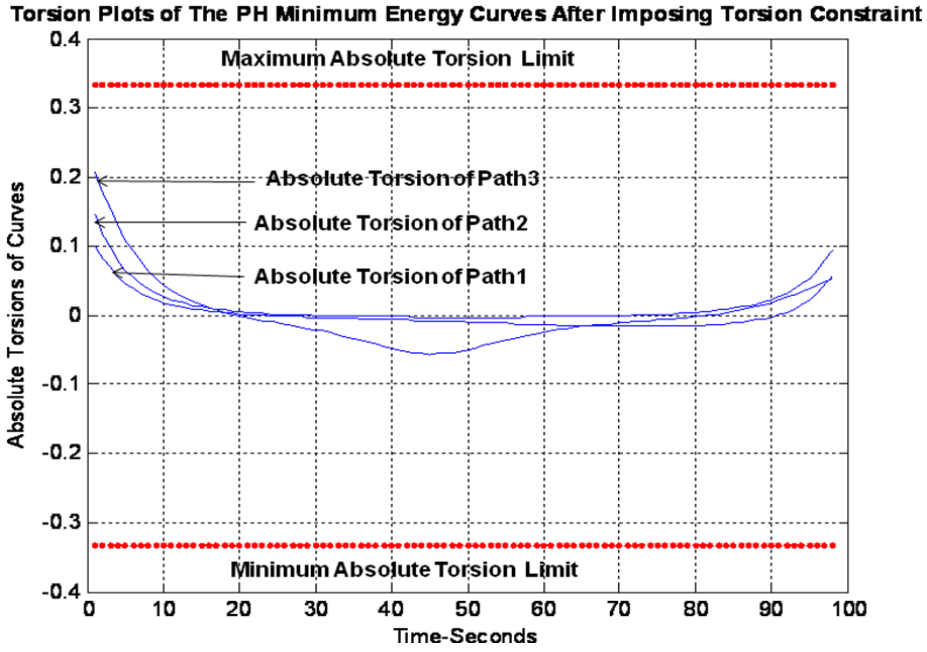

Figure 3.6: Torsion profiles of the paths for UAV1, UAV2 and UAV3 in 3D

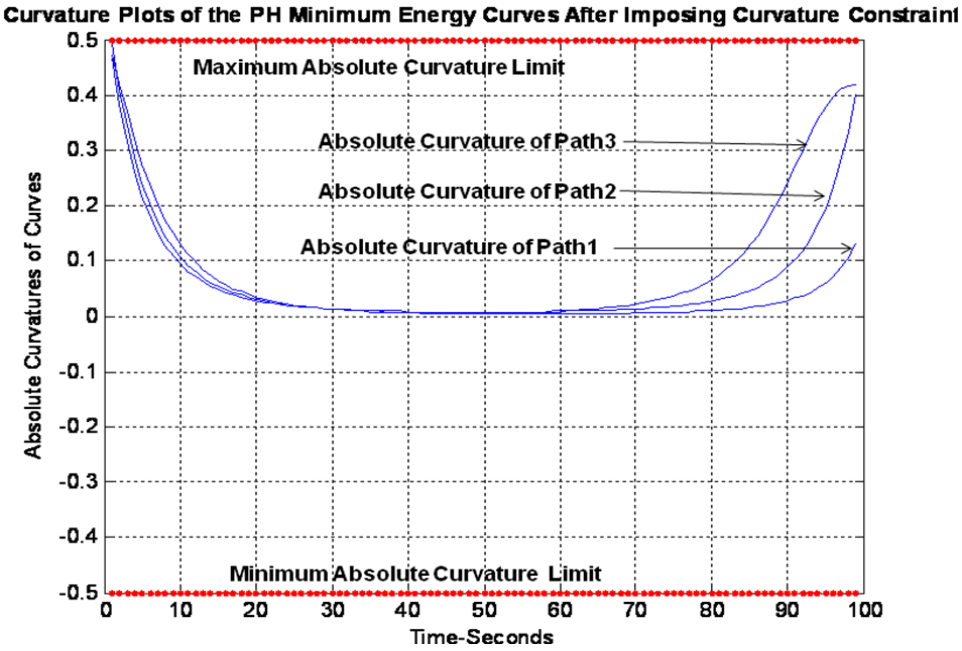

Figure 3.7: Curvature profiles of the paths for UAV1, UAV2 and UAV3 in 3D

Now all the know obstacles in the UAVs map are taken into account and checked for the possibility of collision with any of the UAV paths. The positions of the obstacles relative to the paths are shown in the figure 3.8.

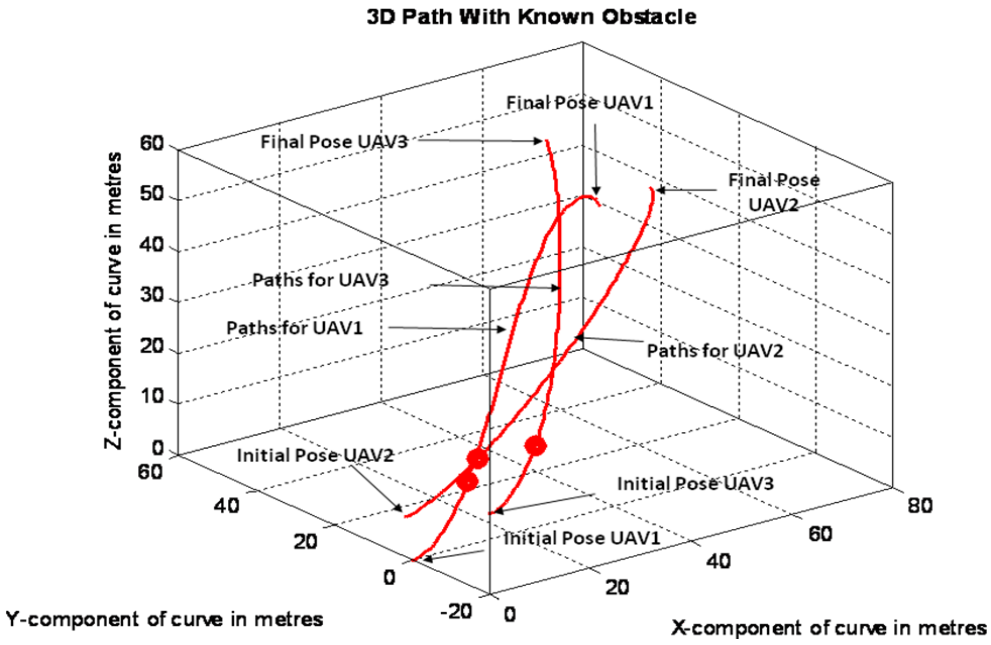

Figure 3.8: Paths with obstacles

These obstacles are avoided by varying the curvature of the paths such that kinematic constraints are not contradicted. Figure 3.9 shows that the obstacles have been avoided in case of each path. Figure 3.10 shows the same figure from different viewing angle just to show the paths clearance. The paths in figure 3.9 are feasible paths, i.e. safe and flyable.

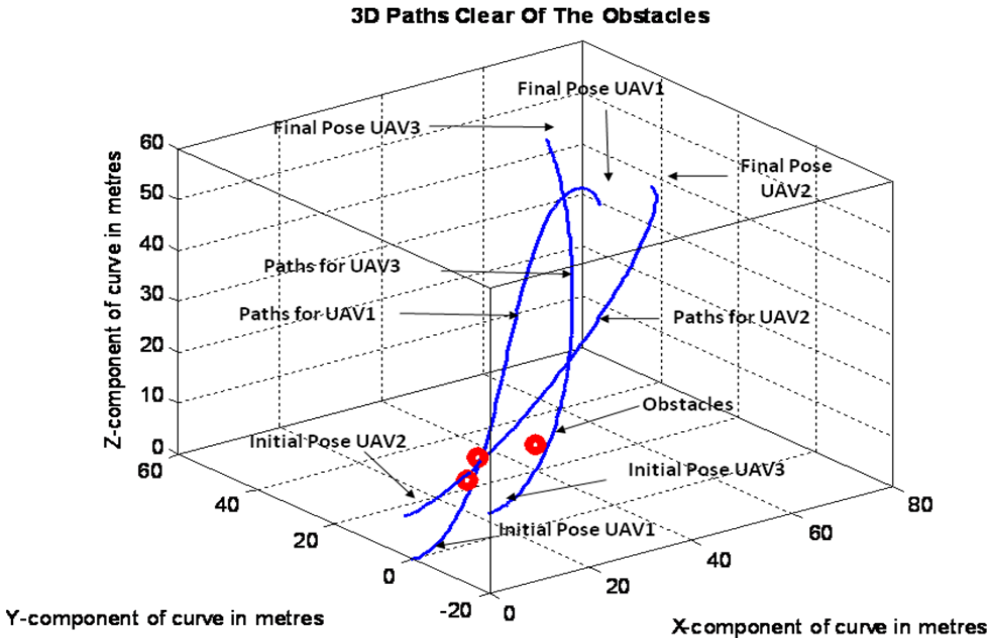

Figure 3.9: Obstacles free paths

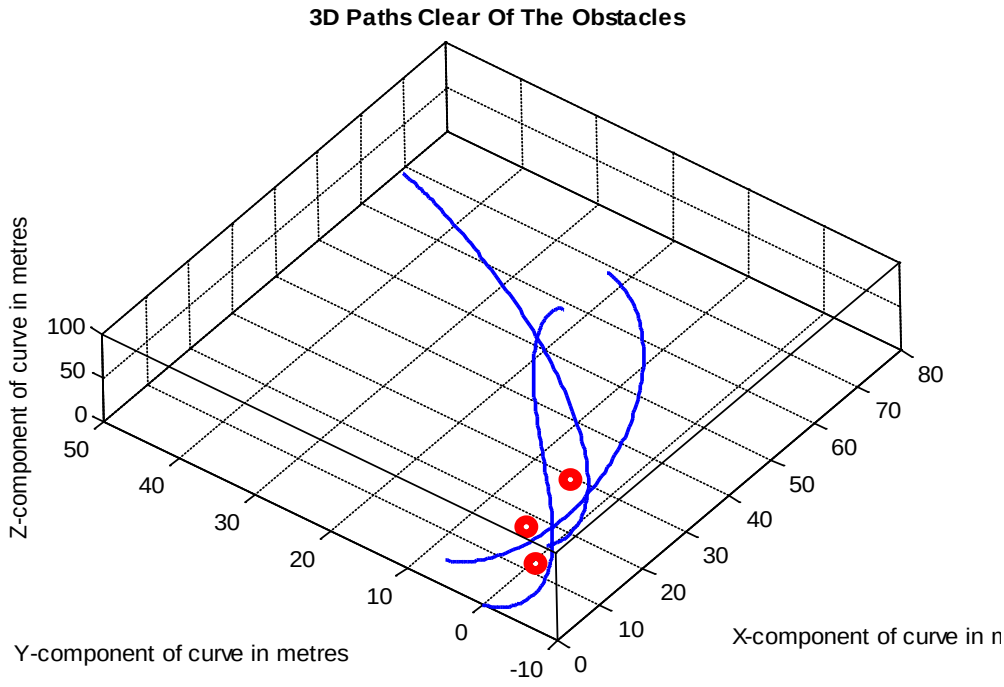

Figure 3.10: Obstacles free paths (different view)

At this stage, the lengths of all the paths are made equal to the longest path. This is done for the purpose of checking the inter collision and simultaneous arrival. In this simulation the length of each path comes out to be 89.743 units. The paths of equal length are shown in figure 3.11.

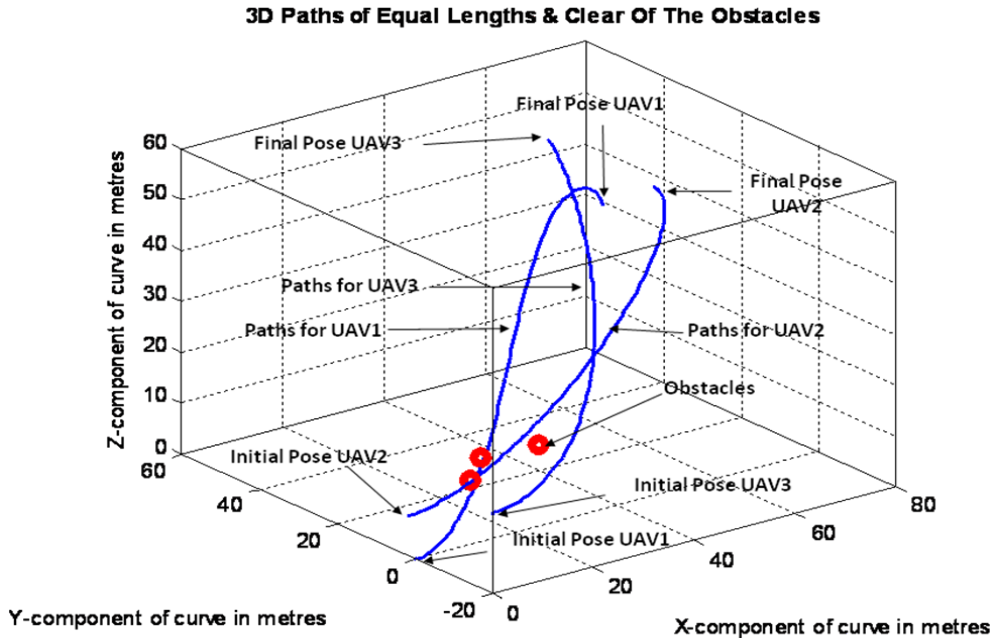

Figure 3.11: Obstacles free paths with equal lengths

After getting flyable paths of equal lengths, a number of checks are made to see if any two of the three paths lead to an inter-collision situation. For three UAVs, taking two at one time, the numbers of safety checks to be made is three. Any two paths are free from inter-collision if the minimum distance between the two UAVs travelling on their respective paths at any time is equal to or greater than twice the safety radius. In this simulation, the radius of the safety sphere of each UAV is taken as one metre. Therefore the minimum safe distance between the two UAVs is two metres. The following checks are made for inter-collision:

- Between path1 and path2 (figure 3.12, 3.13)
- Between path2 and path3 (figure 3.14, 3.15)
- Between path1 and path3 (figure 3.16, 3.17)

The criterion for safety is:

If the minimum distance between two UAVs at any time remains equal to or more than two metres while travelling on their paths then they are safe from inter collision.

Figure 3.12 shows two paths, one for UAV1 and the other for UAV2. The distance plot between UAV1 and UAV2 travelling on path1 and path2 respectively is shown in figure 3.13. The dotted line in figure 3.13 shows the safety boundary, which, in other words represents the aforementioned safety criterion. It is obvious that the two UAVs never cross the safety boundary while travelling on path1 and path2, therefore they are safe. Figure 3.13 shows that the two UAVs never getting closer than six metres to each other during their travel.

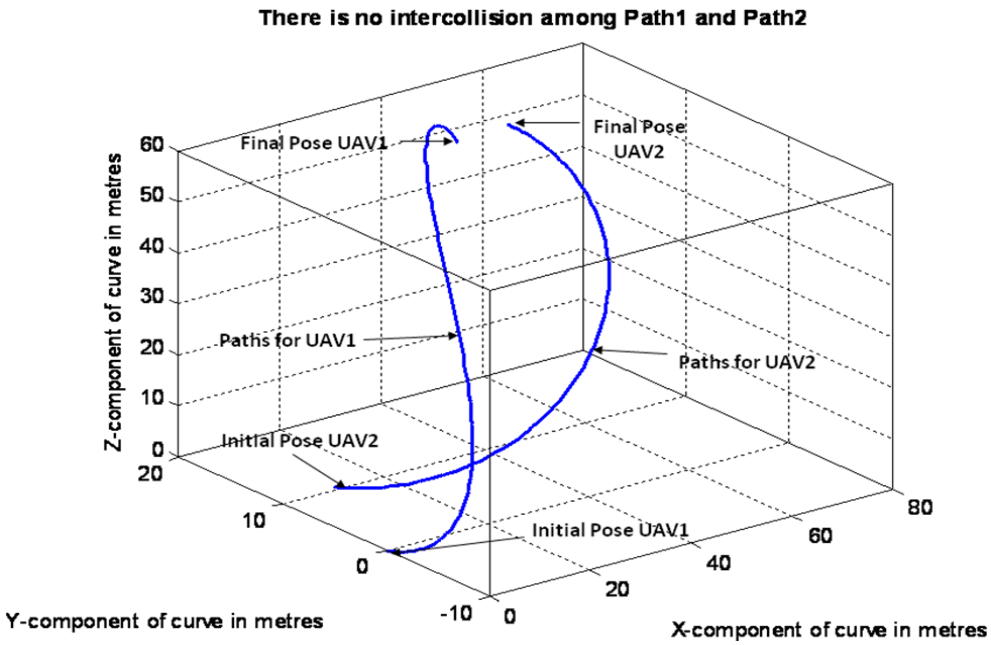

Figure 3.12: Path1 & path2

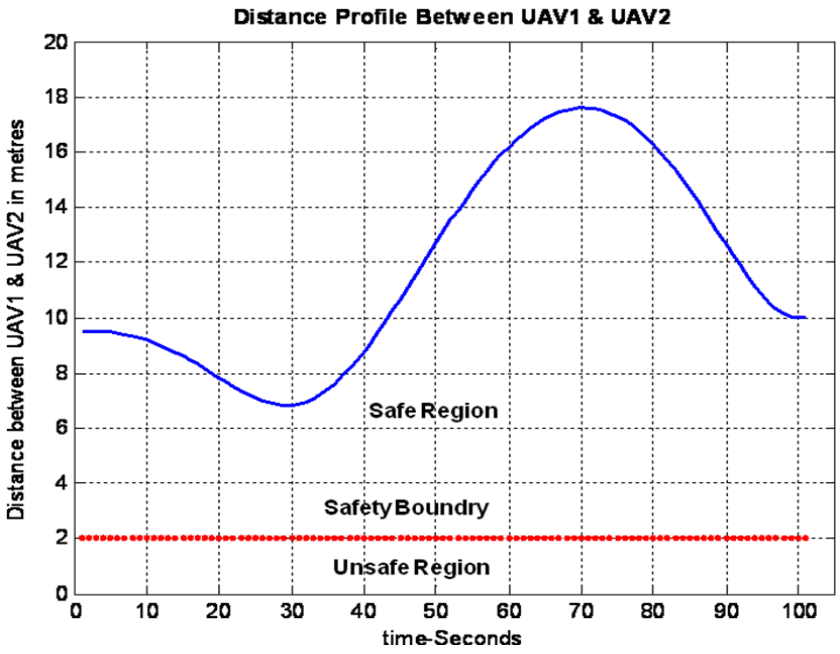

Figure 3.13: No Collision between UAV1 & UAV2

Figure 3.14 shows two paths, path2 and path3. UAV2 travels on path2 while UAV3 travels on path3. The distance profile between UAV2 and UAV3 travelling on path2 and path3 respectively is shown in figure 3.15. It is obvious that the UAVs never cross the safety boundary while travelling on path2 and path3, therefore they are safe.

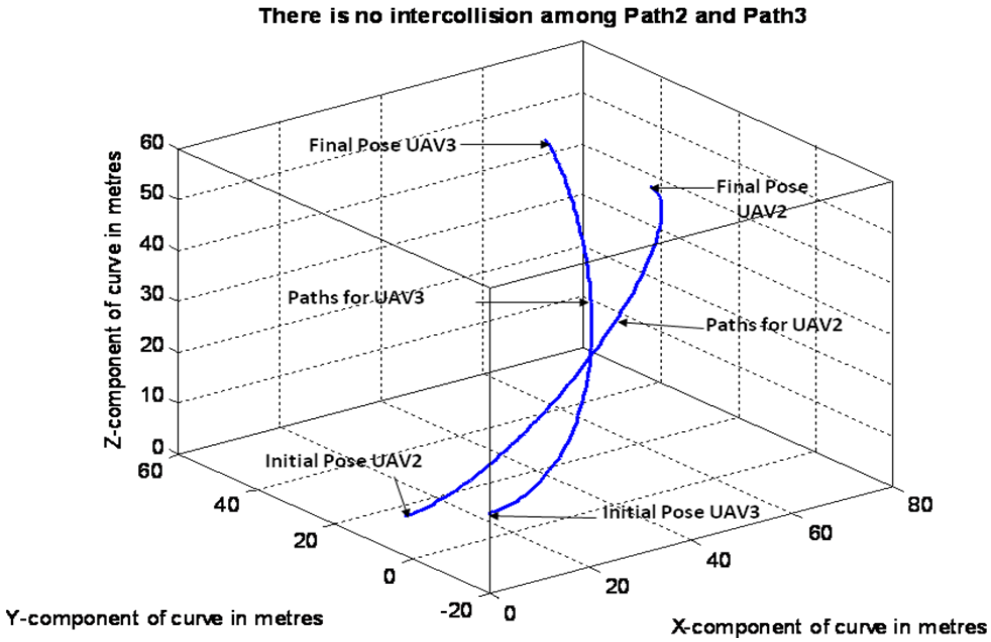

Figure 3.14: Path2 & path3

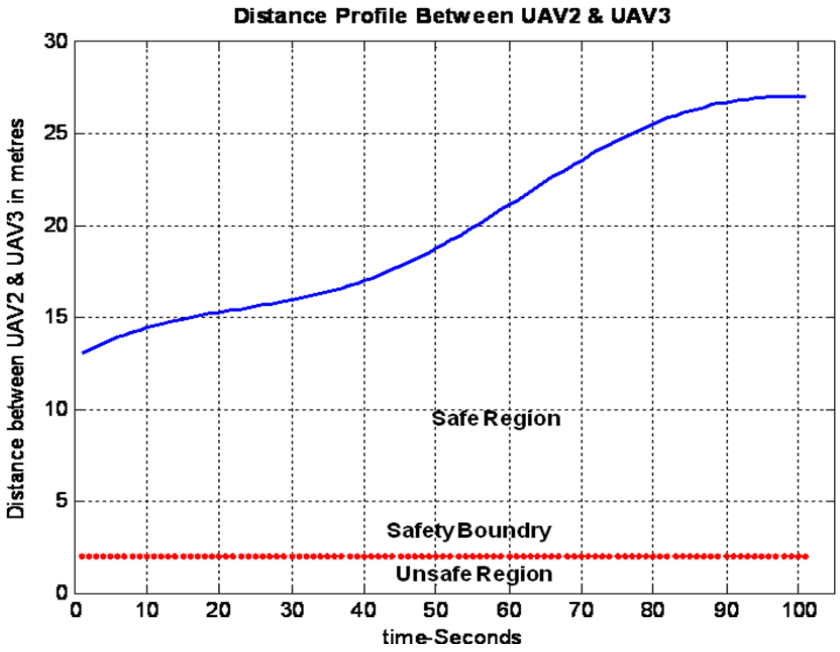

Figure 3.15: No Collision between path2 & path3

Figure 3.16 shows two paths, path1 and path3. UAV1 travels on path1 while UAV3 travels on path3. The distance profile between UAV1 and UAV3 travelling on path1 and path3 is shown in figure 3.17. It is obvious that the UAVs never cross the safety boundary while travelling on path1 and path3, therefore they are safe.

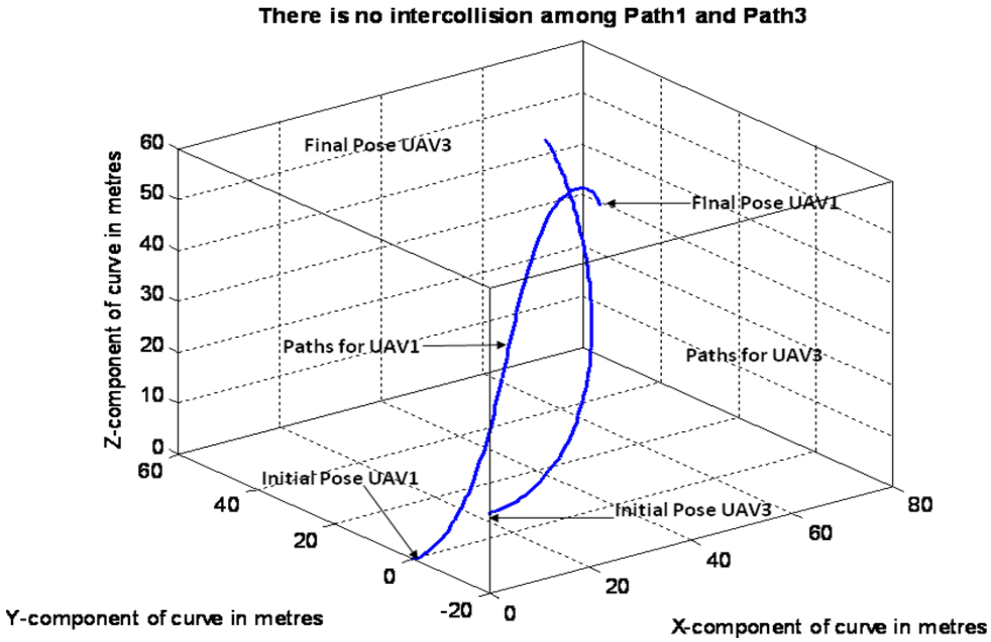

Figure 3.16: Path1 & path3

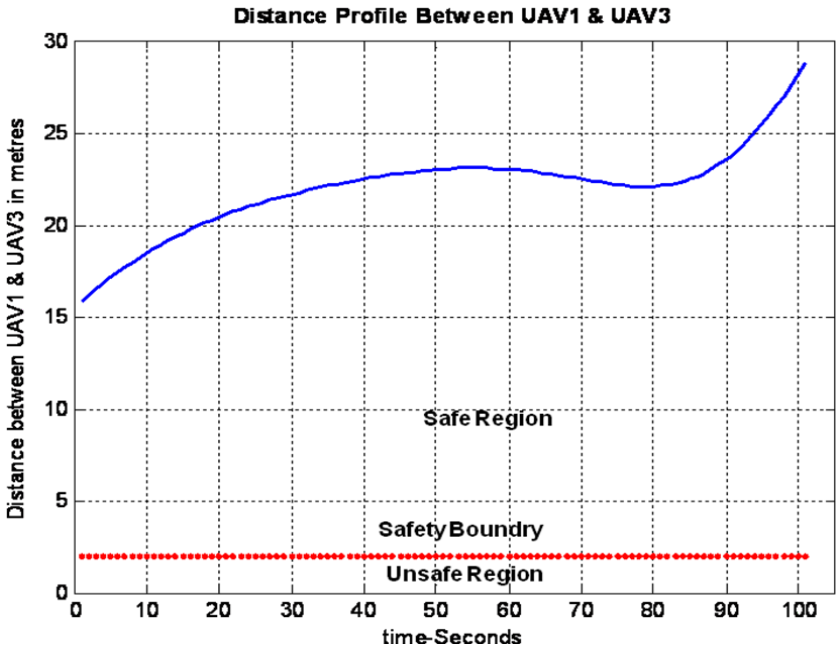

Figure 3.17: No collision between path1 & path3

The condition for inter collision was checked and all paths were found to be free of inter collision. If the condition of inter-collision is not satisfied by any pair of the planned paths then the process of curvature change is repeated until safe paths are obtained.

3.5 Conclusion

In this chapter 3D path planning algorithm based on Pythagorean Hodograph quintic has been derived and simulated for UAVs swarm. The algorithm successfully calculated safe and flyable paths in 3D for all the UAVs in the group using Quaternions mathematics. The paths calculated by the 3D path planning algorithm are optimal with respect to bending energy. These paths have the attractive feature of being able to be traced with least effort. This feature makes them fuel optimal as well as time optimal because hard demands for extreme acceleration are avoided by keeping the bending energy minimum. The paths calculated by this algorithm are spatial paths, which can accommodate real mission challenges and demands.

Chapter 4

Derivation of the Camera Reference Frame from the UAVs Pythagorean Hodograph Paths

In chapter 2 and 3, 2D and 3D trajectories were generated for a swarm of UAVs. All this was done offline and all the known obstacles were considered for avoidance. The UAVs will fly on these trajectories during their planned mission in dynamic environment. Since the flights of the UAVs are in dynamic environment therefore camera sensor is attached to each UAV to observe the changes occurring in the environment. The inclusion of the camera in the UAV platform enable the UAVs to get useful information about the dynamic environment and make autonomous decisions necessary for the safe execution of the mission. The intended use of camera in this research is to watch for pop- up obstacles interrupting the flight paths so that it can be avoided. The camera will also be used for tracking and surveillance of the object of interest which could appear in the scene during the mission. In order to get the position information about the pop-up obstacles or the target being tracked a frame of reference is needed to be defined for the camera in which all the calculations will be made. The transformations between the defined camera reference frame and the universal frame are made through translation vector and rotation matrix in order to get the position information of the target in Universal frame.

In this chapter the camera reference frame is calculated dynamically at every point of its motion. Since the camera is rigidly attached to the UAV, therefore the path of the camera is the same as that of the UAV. Thus the camera frames are derived from the path of the UAV itself. This frame is used as a local reference frame for position calculation of the target in the scene using camera sensor. In the subsequent section the detail definition and derivation of this frame is given.

4.1 Frenet Serret Frame

The starting point in the definition of the camera reference frame is the Frenet Serret Frame of the UAV path. The Frenet Serret frame at each point of a regular curve $r(t)$ is defined as an orthonormal basis $(\vec{t},\vec{n},\vec{b})$ in R^3 aligned with local intrinsic curve geometry as described in [40]. Figure 4.1 shows the Frenet frame of the curve at multiple points of the curve.

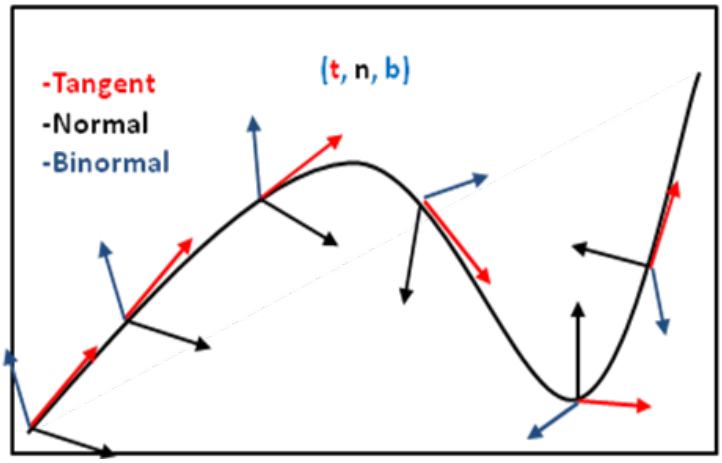

Figure 4.1: Frenet frame of the curve

The elements of this basis are the curve unit tangent vector $\vec{t}$, unit normal vector $\vec{n}$ and a unit binormal vector $\vec{b}$ given by:

$$\vec{t} = \frac{r'}{|r'|} \tag{4.1}$$

$$\vec{n} = \frac{r' \times r''}{|r' \times r''|} \tag{4.2}$$

$$\vec{b} = \vec{t} \times \vec{n} \tag{4.3}$$

Where r is the PH curve, r' and r'' are the first and second derivative of the PH curve.

The unit tangent vector $\vec{t}$, unit normal vector $\vec{n}$ and a unit binormal vector $\vec{b}$ are related to their derivatives by the following set of equations:

$$\begin{bmatrix} \vec{t}' \\ \vec{n}' \\ \vec{b}' \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} \vec{t} \\ \vec{n} \\ \vec{b} \end{bmatrix} \tag{4.4}$$

Where $\vec{t}'$, $\vec{n}'$ and $\vec{b}'$ are the derivatives of the unit tangent vector $\vec{t}$, unit normal vector $\vec{n}$ and a unit binormal vector $\vec{b}$ respectively. κ is the curvature and τ is the torsion of the curve given by the following equations:

$$\kappa(t) = \frac{|r'(t) \times r''(t)|}{|r'(t)|^3} \tag{4.5}$$

$$\tau = \frac{(r' \times r'') \bullet r'''}{|r' \times r''|^2} \tag{4.6}$$

Equation (4.4) gives the orientation and relative rotation of the UAV and hence the camera at every point of the curve.

In this research the unit tangent vector is taken along the direction of motion of the UAV, the unit normal vector is taken along the direction of lateral acceleration and perpendicular to the unit tangent vector and the unit binormal vector is perpendicular to the plane determined by the unit tangent vector and unit normal vector. This is shown in figure 4.2.

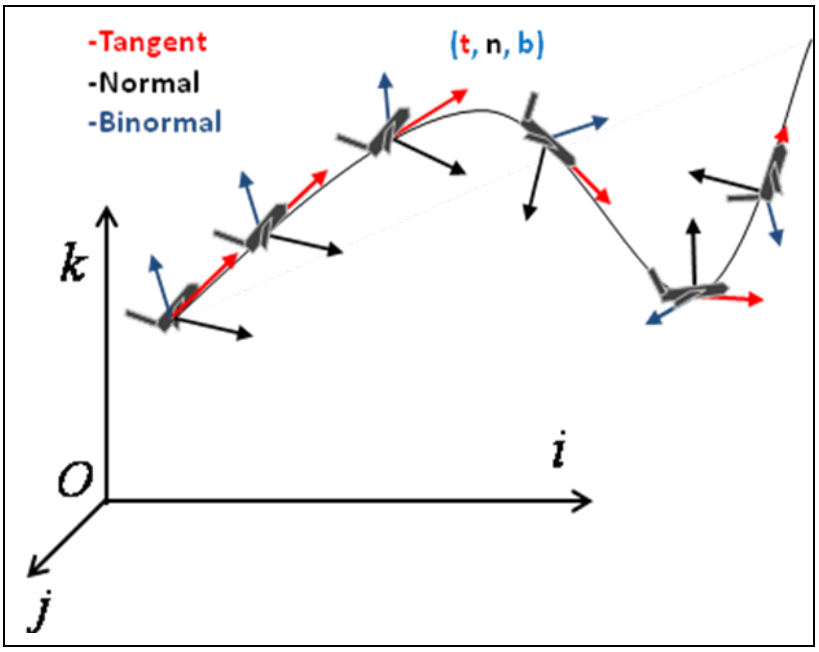

Figure 4.2: Frenet frame of a curve defining the orientation of the UAV and camera.

Frenet frame is considered to be the camera reference frame, with origin at the centre of mass of the UAV. The point of attachment of the camera with the body of the UAV is the centre of mass of UAV. The transformations between this local frame and the universal frame are achieved by the use of the translation vector and rotation matrix [41].

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{4.7}$$

Where r_{mn} for $m,n = 1,2,3$ are the unit vectors in the principal directions $(\hat{i}, \hat{j}, \hat{k})$ of Frenet frame expressed in universal frame i.e.

$$R = \begin{bmatrix} t'.i & n'.i & n'.i \\ t'.j & n'.j & n'.j \\ t'.k & n'.k & n'.k \end{bmatrix}$$

The translation vector is the UAV position vector expressed in universal frame and is given by GPS positioning sensors.

On the regular curve ($r(t) \neq 0 \forall t \in R$) the unit tangent vector $\vec{t}$ is defined at every point of the curve but the unit normal vector $\vec{n}$ and the unit binormal vector $\vec{b}$ are not defined at the point of inflection. The point of inflection is the point on the curve at which the unit normal reverses its direction as shown in figure 4.3.

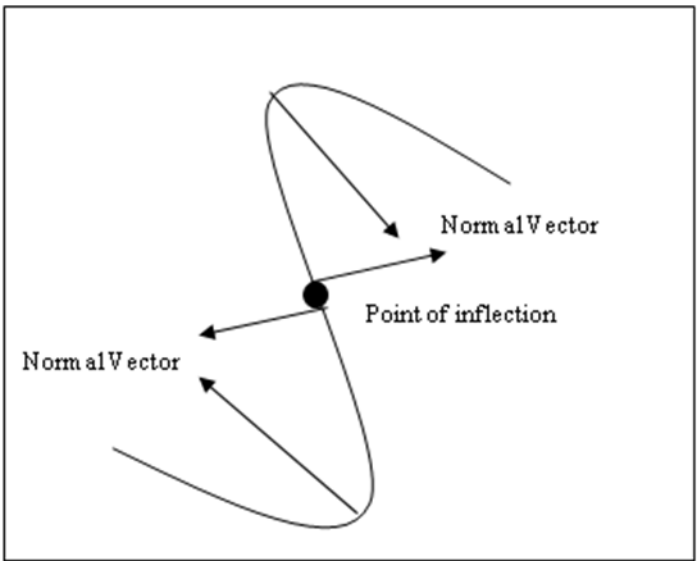

Figure 4.3: Direction of unit normal vector reversal at the point of inflection.

At the point of inflection the vector $r'(t)$ and $r''(t)$ become parallel to each other and their cross product becomes zero: i.e.

$$r' \times r'' = 0$$

This makes the unit normal $\vec{n}$ and the unit binormal vector $\vec{b}$ to be undefined, as implied by equations (4.2) and (4.3). Therefore the Frenet frame becomes undefined at the point of inflection. In fact $\vec{n}$ and $\vec{b}$ experience a sudden reversal upon passing through point of inflection.

For a curve having no inflection point, using the Frenet frame as camera reference frame is a perfect idea. However, for a curve having such point, there is problem of indeterminacy of frame at that point. The Frenet frame is not defined at the inflection point, and hence the camera reference frame is not defined at this point. This problem of the indeterminacy of Frenet frame at the point of inflection can be solved by defining a frame which is defined at every point of the path. Rotation Minimizing Frame [42] is used to overcome this problem.

4.2 Rotation Minimizing Frame (RMF)

Let the Frenet frame of space curve $r(t)$ be $(\vec{t},\vec{n},\vec{b})$ with $\vec{t}$ as unit tangent vector, $\vec{n}$ as unit normal vector and $\vec{b}$ as unit binormal vector respectively. The Rotation Minimizing Frames with basis $(\vec{e}_1,\vec{e}_2,\vec{e}_3)$ can be defined along a space curve $r(t)$ if we choose $\vec{e}_1=\vec{t}$ and rotate $\vec{n}$, $\vec{b}$ in the normal plane through an angle θ such that:

$$\begin{bmatrix}\vec{e}_2\\ \vec{e}_3\end{bmatrix}=\begin{bmatrix}\cos\theta & \sin\theta\\ -\sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}\vec{n}\\ \vec{b}\end{bmatrix} \tag{4.8}$$

This allows us to define a frame $(\vec{e}_1,\vec{e}_2,\vec{e}_3)$ at every point on the curve and solve the problem of indeterminacy of Frenet frame at the inflection point.

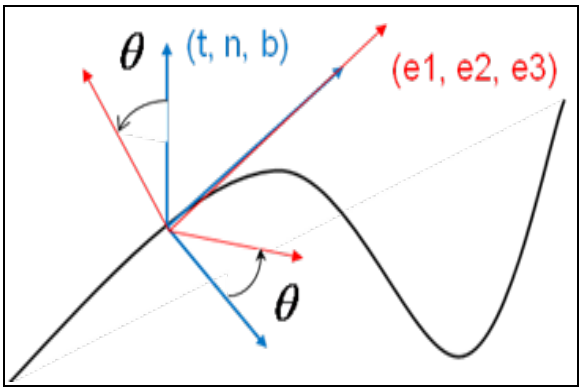

Figure 4.4: Frenet frame (blue) and Rotation Minimizing frame (red).

From equation (4.4) we know that:

$$\begin{bmatrix}\vec{t}\,'\\ \vec{n}'\\ \vec{b}'\end{bmatrix}=\begin{bmatrix}0 & \kappa & 0\\ -\kappa & 0 & \tau\\ 0 & \tau & 0\end{bmatrix}\begin{bmatrix}\vec{t}\\ \vec{n}\\ \vec{b}\end{bmatrix}$$

This system of equations shows that $\vec{t}$ changes at instantaneous rate κ in the direction of $\vec{n}$. $\vec{n}$ changes at a rate of $-\kappa$ in the direction of $\vec{t}$ and at a rate of τ in the direction of $\vec{b}$. $\vec{b}$ changes at a rate of τ in the direction of $\vec{n}$. The change in $\vec{t}$ is unavoidable because we have chosen $\vec{e}_1=\vec{t}$, however the unnecessary rotation of $\vec{n}$ in direction of $\vec{b}$ and vice versa can be minimized by a suitable choice for variation of θ in equation (4.8).

The expression for $\theta(t)$ as noted by Guggemheimer [43], is given by equation (4.9) below:

$$\theta(t)=\theta_0-\int\tau(u)|\,r'(u)|\,du \tag{4.9}$$

Where θ_0 is the constant of integration, and represents the initial orientation of RMF with respect to the Frenet Frame.

4.3 Rotation Minimizing Frame (RMF) for Pythagorean Hodograph Quintic Curves

Since the paths of the UAVs are Pythagorean hodograph type therefore Rotation Minimizing Frame for Pythagorean hodograph is needed to be calculated. After calculating the Rotation Minimizing Frame for PH quintic, the Frenet Serret Frame is replaced with Rotation Minimising Frame to overcome the problem of indeterminacy of frame at the point of inflection.

For this purpose $\theta(t)$ in equation (4.9) is needed to be calculated and substituted in equation (4.8). $r'(t)$ in equation (4.9) must be Pythagorean Hodograph quintic as this research considers the paths of UAVs to be PH.

Differentiating equation (4.9) gives:

$$\frac{d\theta}{dt} = -\left|r'(t)\right|\tau \tag{4.10}$$

But from equation (4.6):

$$\tau = \frac{(r' \times r'') \bullet r'''}{\left|r' \times r''\right|^2}$$

Therefore:

$$\frac{d\theta}{dt} = -\left|r'(t)\right| \frac{(r' \times r'') \bullet r'''}{\left|r' \times r''\right|^2} \tag{4.11}$$

Now

$$\left|r' \times r''\right|^2 = (y'z'' - y''z')^2 + (z'x'' - z''x')^2 + (x'y'' - x''y')^2 \tag{4.12}$$

Putting the values of:

$$x'(t) = [u^2(t) + v^2(t) - p^2(t) - q^2(t)]$$

$$y'(t) = 2[u(t)q(t) - v(t)p(t)]$$

$$z'(t) = 2[v(t)q(t) - u(t)p(t)]$$

And:

$$x''(t) = [u^2(t) + v^2(t) - p^2(t) - q^2(t)]'$$

$$y''(t) = 2[u(t)q(t) - v(t)p(t)]'$$

$$z''(t) = 2[v(t)q(t) - u(t)p(t)]'$$

in equation (4.12) and simplifying yields:

$$\left|r' \times r''\right|^2 = \sigma^2 \rho$$

Where:

$$\sigma = u^2 + v^2 + p^2 + q^2$$

And the polynomial ρ is:

$$\rho = 4[(u'p - up')^2 + (u'q - uq')^2 + (v'p - vp')^2 + (v'q - vq')^2 + 2(uv' - u'v)(pq' - p'q)]$$

After cancellation terms in numerator and denominator equation (4.11) becomes:

$$\frac{d\theta}{dt} = -\left|r'(t)\right| \frac{(r' \times r'') \bullet r'''}{\sigma^2 \rho}$$

$$= -\frac{(r' \times r'') \bullet r'''}{\sigma \rho} \tag{4.13}$$

By partial fraction the right hand side becomes:

$$\frac{(r' \times r'') \bullet r'''}{\sigma \rho} = \frac{a(t)}{\sigma(t)} + \frac{b(t)}{\rho(t)} \tag{4.14}$$

Where $a(t)$ and $b(t)$ are quartic polynomials such that:

$$(r' \times r'') \bullet r''' = a(t)\rho(t) + b(t)\sigma(t)$$

$a(t)$ and $b(t)$ can be obtained by equating the co-efficient of like terms on both sides of the above equation and then solving the resulting equations.

Integrating equation (4.14) yields:

$$\theta(t) = \theta_0 - \int_0^t \frac{a(t)}{\sigma(t)} dt - \int_0^t \frac{b(t)}{\rho(t)} dt \tag{4.15}$$

Now if the roots of σ and ρ are denoted by z_1, z_2, z_3, z_4 and w_1, w_2, w_3, w_4 respectively. Then the partial fraction coefficients of the second and third terms in equation (4.15) are c_1, c_2, c_3, c_4 and d_1, d_2, d_3, d_4 respectively. Therefore:

$$\frac{a(t)}{\sigma(t)} = \sum_{k=1}^{4} \frac{c_k}{t - z_k} \text{ and } \frac{b(t)}{\rho(t)} = \sum_{k=1}^{4} \frac{d_k}{t - w_k}$$

$$\theta(t) = \theta_0 - \int_0^t \sum_{k=1}^{4} \frac{c_k}{t - z_k} dt - \int_0^t \sum_{k=1}^{4} \frac{d_k}{t - w_k} dt$$

$$\theta(t) = \theta_0 - \sum_{k=1}^{4} \left[c_k \log(t - z_k) + d_k \log(t - w_k)\right] \tag{4.16}$$

Since σ and ρ are real polynomials, their complex roots must occur in conjugate form and the corresponding partial fraction coefficients are also complex conjugates. If $z, \bar{z}$ and $c, \bar{c}$ are conjugate roots and coefficient respectively then:

$$c\log(t-z)+\bar{c}\log(t-\bar{z})=2[\mathrm{Re}(c)\log|t-z|-\mathrm{Im}(c)\arg(t-z)]$$

Similarly for $w, \bar{w}$ and $d, \bar{d}$ to be conjugate roots and coefficients:

$$d\log(t-w)+\bar{d}\log(t-\bar{w})=2[\mathrm{Re}(d)\log|t-w|-\mathrm{Im}(d)\arg(t-w)]$$

Therefore the expression for $\theta(t)$ becomes:

$$\theta(t)=\theta_0-2\sum_{k=1}^{2}[\mathrm{Re}(c_k)\log|t-z_k|-\mathrm{Im}(c_k)\arg(t-z_k)]$$

$$-2\sum_{k=1}^{2}[\mathrm{Re}(d_k)\log|t-w_k|-\mathrm{Im}(d_k)\arg(t-w_k)] \qquad (4.17)$$

Equation (4.17) is the expression for $\theta(t)$ calculated for PH quintic. θ_0 is the constant of integration, and can be evaluated by substituting $\theta(0) = 0$ in equation (4.17).

By the use of the Rotation Minimizing Frame $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$ the problem of indeterminacy of Frenet frame at the point of inflection is overcome. Thus the Rotation Minimising Frames of the UAV path can be used as a camera reference frame in perception system for locating pop-up obstacle and target tracking and surveillance.

In this research the Frenet Frame will be replaced by the Rotation Minimizing Frame as camera reference frame for onboard UAV camera whenever there is a point of inflection in the path of UAV.

4.4 Simulation Results

Figure 4.5 shows the path of a UAV with inflection point. The path was calculated by PH path planner using initial pose of the UAV to be (0, 0, 0, 45^o, 70.5^o) and the final pose of the UAV to be (0.06, 2.53, 3.4, 53.1^o, 94.45^o).

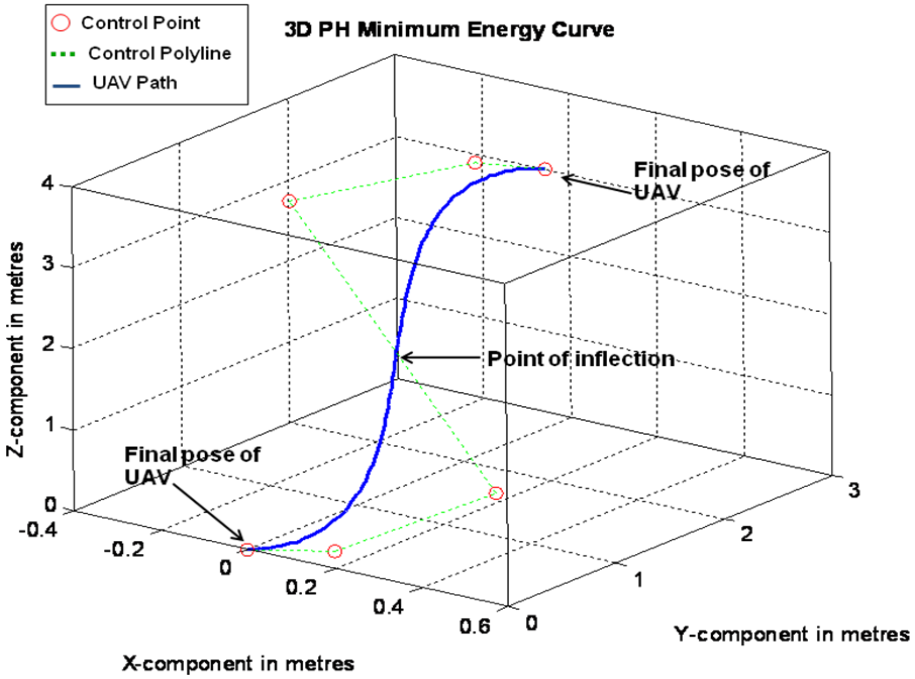

Figure 4.5: Path with inflection point

The curvature profile (absolute value of curvature are taken) of the path is shown in the figure 4.6. The absolute curvature of the path is shown along the vertical axis and the time is shown along the horizontal axis. Figure 4.6 shows that the curvature is starting from a bigger value of 1.4 and gradually decreases until it reaches zero at t=50 and then start increasing again. The point of the curve at t=50 is a point of inflection.

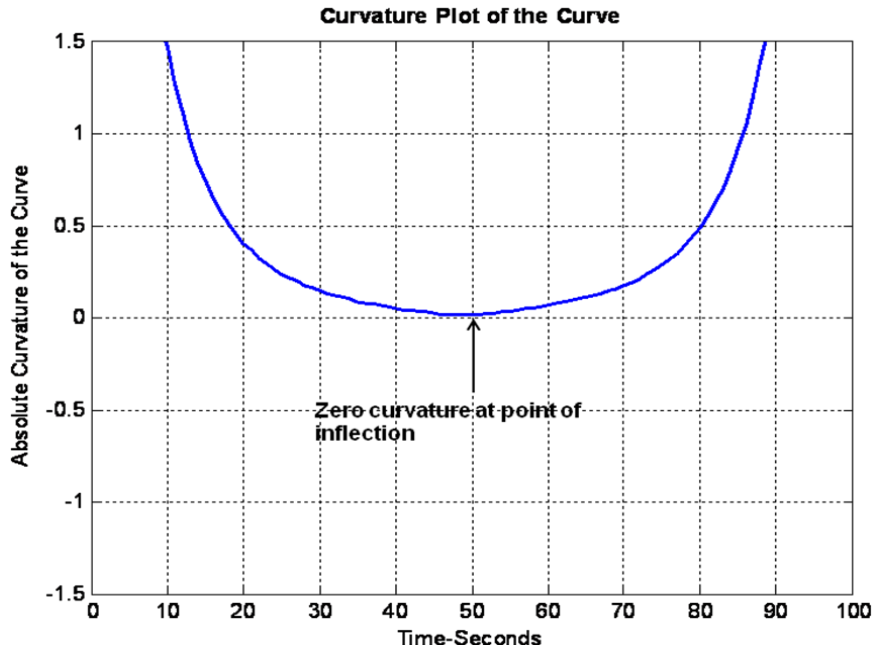

Figure 4.6: Curvature profile of the path

The point of inflection is the point at which the direction of motion stops changing. The direction of motion at certain point is specified by the tangent to the curve at that point. i.e. the first derivative of the curve at that point. The change in the direction of motion is the derivative of tangent or the derivative of the first derivative of the curve or double derivative of the curve i.e $r''(t)$. Since there is no change in direction of motion at the point of inflection, therefore:

$$r''(t)=0$$

Implies:

$$r'(t)\times r''(t)=0$$

This will make:

$$\kappa(s)=\frac{\left|r'(t)\times r''(t)\right|}{\left\|r'(t)\right\|^3}=0$$

Consequently the radius of curvature becomes infinite and therefore the normal vector becomes undefined (the radius of curvature being represented by normal vector). For an undefined unit normal vector, the unit binormal vector:

$$\vec{b}=\vec{t}\times\vec{n}$$

Also become undefined. And therefore the Frenet Frame cannot be defined at point of inflection. Figure 4.7 shows the point of inflection where the unit normal vector reverses its direction.

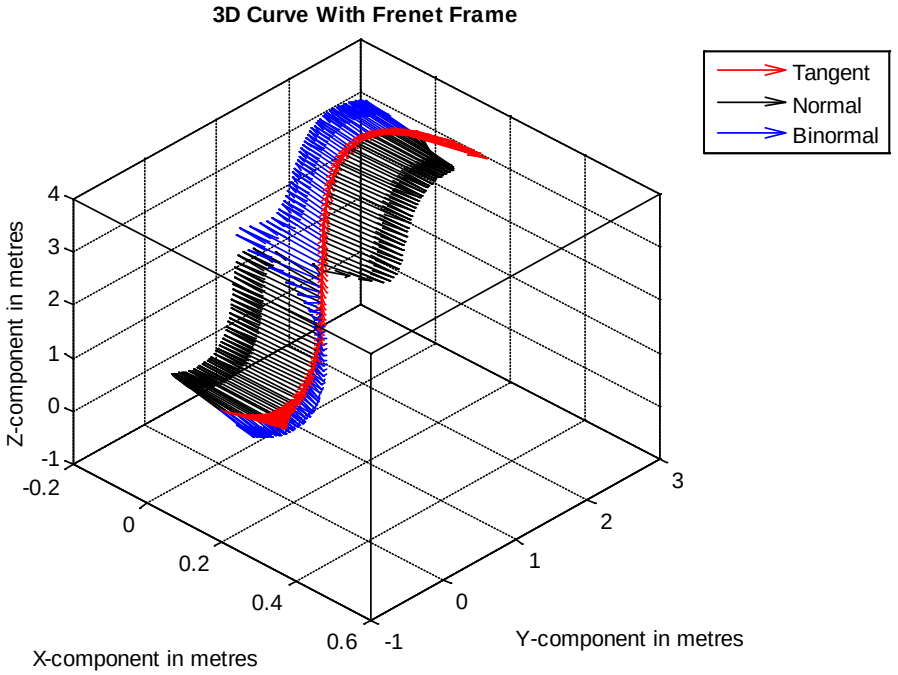

Figure 4.7: Path1 with Frenet frame

Rotation Minimizing Frames (RMF) has been used as a solution to the problem of indeterminacy of Frenet frame at the point of inflection as shown in figure 4.8. Figure 4.8 shows the calculated RMF for path. It can be seen that there is no reversal of the unit normal vector $\vec{n}$ at the point of inflection of the curve. It can also be seen that the unnecessary rotations of the Frenet frame are minimized. For comparison figures 4.7 and 4.8 can be seen where the rotation of the Frenet frame has been eliminated.

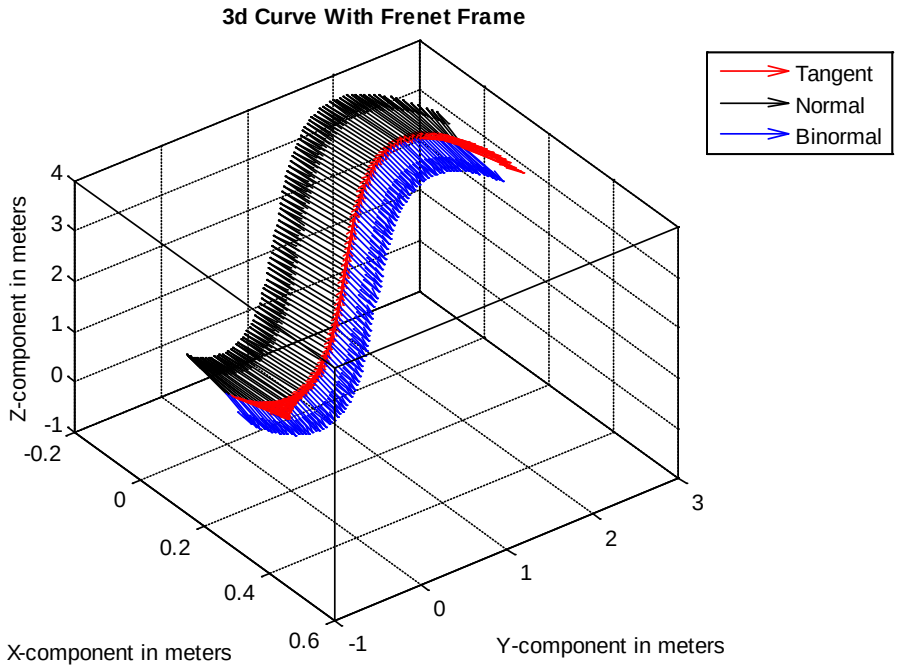

Figure 4.8: Path with Rotation Minimizing frame

4.5 Conclusion

Rotation Minimizing Frames (RMF) are proposed as a solution to the problem of indeterminacy of Frenet Frames at the point of inflection such that the intrinsic properties (curvature and torsion) of the curve are retained in the proposed frame. The Rotation Minimizing Frames are defined at every point of the curve and can be used as a reference frame for any sensor at any point of the curve without any risk of being abruptly rotated with undesirable consequences. The Rotation minimizing frame is calculated from the Frenet frames by minimizing the rotation around unit tangent vector. It eliminates the unnecessary rotation. The smooth and gradual transition of rotation of the frame between the initial and final orientations makes the calculation easier, results understandable and the process logical.

In chapter 5, the camera based perception system uses RMF as camera reference for 3D cooperative position estimation of an object in the scene. In chapter 6 and 7, the Rotation Minimising Frame is used as a camera reference frame for position estimation of pop-up obstacle and motion estimation of moving ground target.

In the rest of the thesis, the Frenet Frame will be replaced by the Rotation Minimizing Frame as camera reference frame for onboard UAV camera whenever there is a point of inflection in the path of UAV.

Chapter 5

Cooperative Perception

The future holds an ambitious vision of the autonomous mission execution by fully autonomous UAV in dynamic environments. In order to realize such a vision, multiple sensors need to be integrated in the UAV platform to enable it to perform in dynamic environments autonomously. In the dynamic environment there may be objects of interest which may be stationary, or moving. These objects may be known, partially known or unknown. Since the complete knowledge about the objects can not be guaranteed in advance in case of dynamic environment, therefore, there is a need for some sensory mechanism to obtain information about the objects dynamically throughout the mission. The integration of sensors to UAV platform makes this possible and increases the autonomy and safety of the vehicle during mission.

In this research the operating environment is assumed to be dynamic. The absence of complete prior knowledge about the environment necessitates the perception system to gather information about the operating environment. Therefore each vehicle is provided with a perception system to get the 3D information and motion estimations of the objects of interest in cooperative way. The perception system consists of a camera (vision sensor), an onboard processing capability of the information gathered by the camera and a fusion algorithm to fuse data from cameras on different vehicles. The motivation of using video cameras on board UAV is because of the richness of information that can be extracted from each frame of the video stream. Also the camera can give very accurate relative position of the UAV with respect to an object; therefore camera vision can be used as an effective sensor for close proximity flights. In this research, since the position of the target relative to UAV is needed to be known with certain degree of precision, therefore computer vision is used as reliable mean.

Using multiple cooperative UAVs in vision based estimation and tracking problem is advantageous for a number of reasons. First multiple UAVs provide redundancy, allowing for continued tracking even when the individual vehicle experience failures. Second, the presence of obstructions in the environment may temporarily block the field of view of the UAV as it attempts to observe the target. Using multiple UAVs with different lines of sight increases the probability that the target will remain observable to the group of the UAVs even when individual vehicle’s line of sight is blocked. Third, because more observations are available at a given time, multiple UAVs working together can estimate the target state more accurately than a single UAV could.

In addition, multiple cooperative UAVs for vision based estimation and tracking are needed to get the depth perception of the target position. A single UAV can only determine a line along which the target lies; therefore the UAV must be manoeuvred around the target to multiple points for observations to get the accurate 3D position estimates of the target. However if the target is moving then it may not be possible for the UAV to complete the manoeuvres to get the accurate estimates.

In this research cooperative perception is used to cooperatively locate the object of interest in 3D space using onboard camera on each vehicle. Frenet Frame and Rotation Minimising Frame of chapter 4 are used as camera reference frame in this research.

As a starting point, the problem of locating the object was simplified by ignoring the sensor noises and assuming that the target remains visible to at least two cooperating UAVs at all the times. The solution to the simplified problem was approached through multi-view method [70]. The individual views from multiple cameras constitute a multi-view setup. The 3D target trajectory was calculated from 2D image trajectory of the object in image frames of the multi-view setup. The cooperating UAVs exchange their estimates and fuse them to get the position estimates of the target [92, 93].

The problem of cooperative perception is then complicated by taking the sensors noise in consideration to accommodate the associated challenges in real missions. The cooperative UAVs vision based tracking problem is reformulated as linear estimation problem [55]. A target was tracked using two cameras on two different UAVs. Using the observed bearings of the target from each UAV, an estimate of the absolute target position was obtained by minimizing the errors in distance from the estimate to each measurement. This estimate is then used as a measurement input to a simple linear Kalman filter that uses the target location as the state vector.

Before going into the details of the methods of locating the target in 3D stated above, a thorough literature review of the problem is carried out. The model of the camera and the basics of Kalman filter are then discussed briefly to put the reader in context. After the discussion of the ingredients of the perception system (camera and Kalman filter) the problem statement is given and the two solutions to the problem is discussed in detail along with their simulation results.

In literature computer vision is used for various applications like forest fire detection, target tracking, navigation, border patrol, traffic monitoring, simultaneous localization and map building, etc. In the following a detail literature review of vision based techniques that have been developed for various applications has been discussed.

Differential Geometry techniques are used in Vision [44-50] to understand the motion of the object by taking camera images of the object at different instants. The trajectory of the object was traced by studying the orientations of the object in a sequence of frames. The natural axis of the object tends to remain aligned with the local tetrahedron defined by the object's trajectory. Based on this observation a model called Frenet Serret Frame is used which corresponds to the motion of moving tetrahedron

along the space curve. Knowing how Frenet Serret Frame is changing relative to the observer gives essential information for understanding object motion.

Vision is used in path planning of the UAV and camera is used to navigate the UAV through the desired trajectory [51-54]. In this approach a camera is used to guide the UAVs through the path in the environment which is cluttered with obstacles. The UAVs maintain up to date perception of the environment and its own states through a camera, IMU (Inertial Measurement Unit) and GPS. Each obstacle requires the UAV to pass through with correct 3D position as well as aligned pitch and yaw angles.

Estimation Algorithms are used in cooperative vision for tracking the obstacles [55- 64]. A vision based estimation and tracking algorithm for cooperative multiple UAVs has been used to give an accurate target state estimation. The method uses an optimization technique to combine the instantaneous observation of multiple UAVs, allowing very rapid estimation. The algorithm is distributed among all the participating UAVs with a very modest communication band width requirement. The image processing is done in a way which is robust to the noise. The approach is applied to the automatic forest fire detection [61, 62].

The filtering approach is used for pose estimation of object in 3D space [65-69]. A combination of two filters is used. The first filter is manoeuvre detection filter with no need of prior knowledge about the object trajectory. The second filter is numerical based analysis filter. The outcome of using this filter approach is to increase the robustness of overall 3D pose estimation of moving target from monocular image sequence. This filtering technique can deal with the dynamic environment in which there are sudden and unexpected movements of targets. This approach is applied in a project deploying a cooperative UAV and a UGV simultaneously in a mission of simulated mines detection and disposal [67]. The UAV explore the mines site with its camera. It locates the mines and prepares a map. This map is transmitted to a ground computer. The computer processes the information in the map for mines locations and sends it to UGV. The UGV uses this information for guidance to reach the correct mine's locations. Once reached the correct location, the UGV disposes the mines.

Cooperative tracking approach is a common place in vision literature [58, 61, 62, 67, 68, 69]. In this approach several UAVs operate in the same area. The cooperating UAVs have some area of investigation which is common in their field of views. Each UAV is inspecting the area from its own perspectives and share the acquired information among the group. The shared information is fused with the local information on each platform. This approach gives maximum robustness against noise and individual sensor failure. Cooperative tracking approach for UAVs with camera based sensor is developed and verified with flight data [67]. The approach utilizes a square root sigma point information filter. This approach details the theoretical development of a distributed cooperative estimation methodology for multiple UAVs, which tracks stationary and moving ground targets with onboard cameras.

Feature based position estimation in case of multiple UAVs is another development [64, 68]. In this approach, different UAVs identify a common object in the scene using their cameras, and then the relative pose displacement between the UAVs is computed from these correspondences. The common object is identified in robust and repeatable way in images from different cameras at different locations and different orientations. This method uses a robust feature extraction technique capable of identifying natural landmarks of the scene. The use of these features called blobs obtains greater robustness than the traditional point like image matching.

3D reconstruction and matching of curve from multiple views using differential geometry intrinsics is another approach used in the literature [70-83]. In this approach the 3D space curve is reconstructed from the 2D images of curve taken from many different views using the differential geometry intrinsics. The curvature and torsion of the curve are intrinsics to the curve because they remain the same in the images as that of the original curve. Therefore they can be used for matching the 2D curve in different view. The curvatures of all corresponding points in the 2D curves from multiple views are checked. If the curvatures of the corresponding points in image curves are not consistent then those points do not belong to the curve and removed. This technique provides a very powerful mean of noise reduction for geometric curves.

Stereo vision is a very powerful technique for object localization in 3D. In literature there is a lot of material on the use of stereo vision for various purposes [85-91]. Stereo vision is also used for automatic camera calibration [66]. The 2D image information from uncalibrated stereo camera is used first to control the robot hand to approach the target object. Subsequently the cameras are calibrated automatically using the information from robot hand movement, making the robot hand approaching destination rapidly and smoothly. This method overcomes the conventional camera calibration on one hand, and achieves higher efficiency and accuracy than uncalibrated camera on the other hand.

5.1 Cooperative Perception System

In this research the environment is assumed to be partially known dynamic environment, therefore each UAV is assumed to be equipped with a perception system to perceive the unknown obstacles/target within the certain range, during its travel.

In the context of this research the cooperative perception is defined as the process of acquiring information about an object in the scene from different viewpoints through vision sensor on board each vehicle, fuse and exchanging this visual information with each other, to obtain the optimal 3D location and motion estimates of the object.

The purpose of the cooperative perception is to acquire the 3D information of the objects cooperatively and fed it back to the path planner for effective decision making in re-planning the reference path to avoid the obstacles or track the target. The perception system comprises of onboard vision sensors (camera), an estimation algorithm (Kalman

filter) and a distributed processing capability. A brief discussion of these important components of the perception system is given in the following sections.

5.1.1 Camera

In this research, light intensity video camera is used as a vision sensor to perceive changes in the dynamic environment. The motivation of using camera as a vision sensor in cooperative perception is because of its model simplicity and convenient use. The vision system based on camera has a very rich literature in terms of efficient techniques and powerful algorithms. These two aspects of camera based perception system make it an attractive option to analyse the changes being taking place in the dynamic environment.

Since camera is used as a vision sensor, therefore understanding the camera model is the starting point of our perception system.

5.1.1.1 Perspective Camera or Pinhole Camera Model

The most common geometric model of an intensity camera is the perspective or pinhole [94] model as shown in figure 5.1. The model consists of an image plane and a 3D point 'O', the centre or focus of projection. The distance between the image plane and 'O' is called focal length f. the line through 'O' and perpendicular to the image plane is the optical axis. 'p' is a scene point in the field of view of camera.

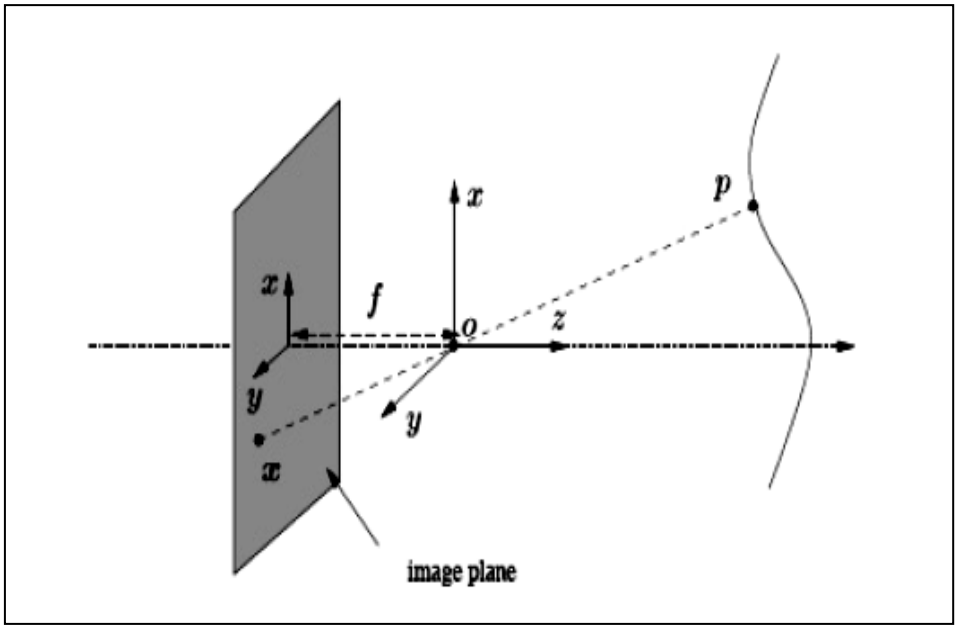

Figure 5.1: Geometric model of pinhole camera

The point at which the line through *p* and O intersect the image plane is the image of *p*. Consider the 3-D reference in which O is the origin and the image plane is orthogonal to z-axis. This reference frame is called camera reference frame. Let the coordinates of the scene point *p* be $({}^{c}X, {}^{c}Y, {}^{c}Z)$ and coordinates of the image of the scene point be

$({}^{c}x, {}^{c}y, {}^{c}z)$ both expressed in camera frame. The coordinates of the scene point *p* and the coordinates of corresponding image point are related by the following equations:

$$ {}^{c}x = f\frac{{}^{c}X}{Z} \tag{5.1} $$

$$ {}^{c}y = f\frac{{}^{c}Y}{Z} \tag{5.2} $$

$$ {}^{c}z = f \tag{5.3} $$

Since ${}^{c}z = f$ at all the times, therefore $({}^{c}x, {}^{c}y, {}^{c}z)$ is written as $(f\frac{{}^{c}X}{Z}, f\frac{{}^{c}Y}{Z})$ instead of $(f\frac{{}^{c}X}{Z}, f\frac{{}^{c}Y}{Z}, f)$.

5.1.1.2 Weak Perspective Camera Model

A classical approximation that turns equations (5.1) and (5.2) into linear equations is the weak perspective camera model. This model requires that the relative distance along the optical axis δz, of any two scene points (scene depth) is much smaller than the average distance $\bar{Z}$ of the point from the viewing camera. In this case for any scene point:

$$ {}^{c}x = f\frac{{}^{c}X}{\bar{Z}} \tag{5.4} $$

$$ {}^{c}y = f\frac{{}^{c}Y}{\bar{Z}} \tag{5.5} $$

To understand the camera model the intrinsic and extrinsic parameters of the camera needed to be defined.

5.1.1.3 Extrinsic Parameter

The extrinsic parameters of the camera are the parameters which define the location and orientation of the camera reference frame with respect to world reference frame. The 3D translation vector describing the relative position of the origins of the two reference frames, and 3x3 rotation matrices that bring the corresponding axes of the two frames onto each other, are the extrinsic parameters of the camera.

Consider a point P with coordinates ${}^{w}P({}^{w}X, {}^{w}Y, {}^{w}Z)$ in a known world reference frame and the same point with coordinates ${}^{c}P({}^{c}X, {}^{c}Y, {}^{c}Z)$ in a camera reference frame as shown in the following figure 5.2:

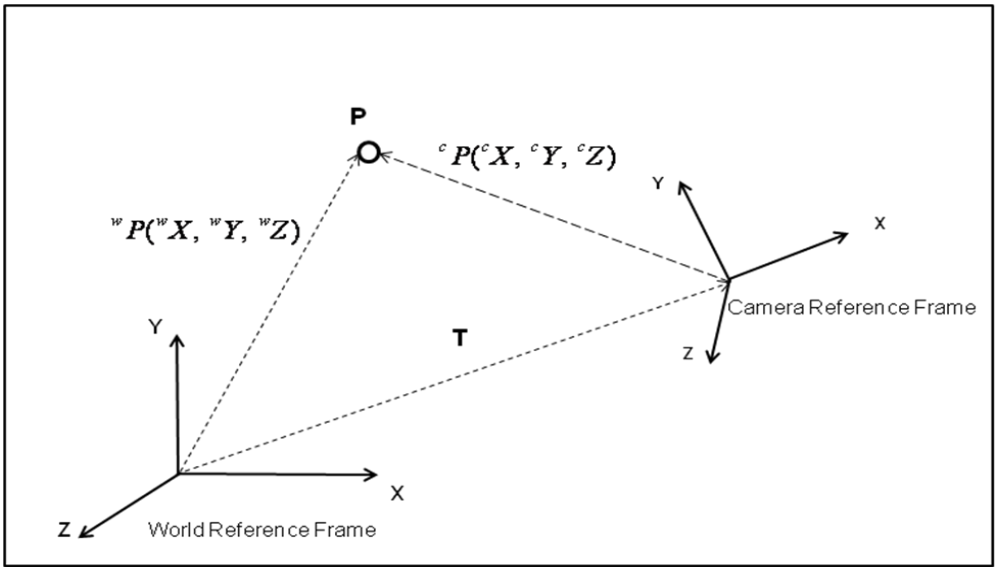

Figure 5.2: World reference frame and camera reference frame

The relation between the camera frame and world frames is given by the following equation [94]:

$$^{c}P = R^{T} \times (^{w}P - T) \tag{5.6}$$

Where R^T is the transpose of the 3x3 rotation matrix given by equation (5.7), and T is a 3D translational vector.

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{5.7}$$

Where r_{ij} for $i, j = 1,2,3$ are the unit vectors in the principal directions of camera reference frame expressed in world reference frame. The translation vector is the UAV position vector expressed in universal frame and is given by GPS positioning sensors.

5.1.1.4 Intrinsic Parameter:

The intrinsic parameters characterise the optical geometry and the digital characteristic of the viewing camera. The intrinsic parameters set comprise of the focal- length f, the coordinates in pixel of image centre O_x, O_y, and the effective size of the pixel (in millimetres) s_x, s_y, in the horizontal and vertical direction respectively.

5.1.1.5 Transformation from Camera to Pixel Coordinates

The camera reference frame has been introduced for the purpose of writing the fundamental equation of perspective projection in a simple form. However the camera reference frame is often inaccessible and the camera related coordinates needed to be eliminated and replaced with pixel coordinates and universal frame coordinates. For this

purpose referring again to figure 5.1, Let $({}^{c}X, {}^{c}Y, {}^{c}Z)$ be the co-ordinate of a scene point *p*, and $({}^{c}x, {}^{c}y, {}^{c}z)$ are the coordinates of the corresponding image point both expressed in camera reference frame. If the radial geometric distortion introduced by the optics is ignored then the coordinates of the image point in camera reference frame and the pixel coordinates of the same point in the image frame are related by the following equations [94]:

$$ {}^{c}x = -({}^{im}x - O_x)s_x \tag{5.8} $$

$$ {}^{c}y = -({}^{im}y - O_y)s_y \tag{5.9} $$

Where ${}^{im}x$ and ${}^{im}y$ are the pixel coordinates of the image point in the image frame.

Eliminating ${}^{c}x$ and ${}^{c}y$ from equations: (5.1), (5.2) and (5.8), (5.9):

$$ -({}^{im}x - O_x)s_x = f\frac{X^{c}}{Z} \tag{5.10} $$

$$ -({}^{im}y - O_y)s_y = f\frac{{}^{c}Y}{Z} \tag{5.11} $$

Putting equation (5.6) in equation (5.10) and (5.11)

$$ ({}^{im}x - O_x)s_x = f\frac{R_1^T({}^{w}P - T)}{R_3^T({}^{w}P - T)} \tag{5.12} $$

$$ ({}^{im}y - O_y)s_y = f\frac{R_2^T({}^{w}P - T)}{R_3^T({}^{w}P - T)} \tag{5.13} $$

Where R_i for $i = 1,2,3$, is a 3D vector formed by i[th] row of a rotation matrix R given by equation (5.7). Equations (5.12) and (5.13) give relation between the 3D coordinates of the point in world reference frame to corresponding image point through camera intrinsic and extrinsic parameters.

Neglecting the radial distortion equations (5.12) and (5.13) can be put as a simple matrix product. For this purpose, two matrices M_{int}, M_{ext} are defined as:

$$ M_{\text{int}} = \begin{bmatrix} \frac{-f}{s_x} & 0 & O_x \\ 0 & \frac{-f}{s_y} & O_y \\ 0 & 0 & 1 \end{bmatrix} \tag{5.14} $$

$$M_{ext} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -R_1^T T \\ r_{21} & r_{22} & r_{23} & -R_2^T T \\ r_{31} & r_{32} & r_{33} & -R_3^T T \end{bmatrix} \tag{5.15}$$

It is clear that the matrix M_{int} depends on the intrinsic parameters and the matrix M_{ext} depends on extrinsic parameters. If ${}^{w}P$ is expressed as homogenous i.e. $({}^{w}X, {}^{w}Y, {}^{w}Z, 1)$ and form the product $M_{\text{int}} M_{ext} {}^{w}P$, a linear matrix equation describing the perspective projections, is obtained.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = M_{\text{int}} M_{ext} \begin{bmatrix} {}^{w}X \\ {}^{w}Y \\ {}^{w}Z \\ 1 \end{bmatrix} \tag{5.16}$$

In the vector (x_1, x_2, x_3) the ratios $\frac{x_1}{x_3}$ and $\frac{x_2}{x_3}$ are the image coordinates:

$$\frac{x_1}{x_2} = {}^{im}x \tag{5.17}$$

$$\frac{x_2}{x_3} = {}^{im}y \tag{5.18}$$

Thus the two steps of world image projection is separated. M_{ext} performs the transformations from the camera reference frame to the image reference frame. M_{int} performs the transformations from the world reference frame to the camera reference frame. If M_{int} and M_{ext} are multiplied, another matrix known as Projection Matrix, is obtained, thus:

$$M = \begin{bmatrix} \frac{-f}{s_x} & 0 & O_x \\ 0 & \frac{-f}{s_y} & O_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & -R_1^T T \\ r_{21} & r_{22} & r_{23} & -R_2^T T \\ r_{31} & r_{32} & r_{33} & -R_3^T T \end{bmatrix} \tag{5.19}$$

The projection matrix M describes the full perspective camera model.

5.1.2 Kalman Filter

The next relevant area of the perception system is to deal with the target's motion. The main interest is the extraction of visual information from spatial and temporal changes

occurring in an image sequence. From this image information the motion of an object can be inferred. The apparent motion of the object onto the image plane is a strong visual cue for understanding 3D structure and motion. From the changes occurring in a sequence of image frames of an object in time, the motion of the object can be found out. Filtering is an essential part of motion detection system because sometimes there are failures to acquire the object's image information (frame) for various reasons, and the 3D motion needed to be estimated using some filtering scheme, as a backup measure. Kalman Filter [92, 93] is generally used for this purpose which is described in the following paragraphs.

Kalman filter addresses the general problem of trying to estimate the state $x \in R^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation:

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{5.20}$$

With a measurement $Z \in R^m$ that is:

$$Z_k = Hx_k + v_k \tag{5.21}$$

The random variables w_k, v_k represent the process and measurements noise. They are assumed to be independent (of each other), white, and with normal probability distributions

$$P(w) \sim N(0,Q)$$
$$P(v) \sim N(0,R)$$

In practice, the process noise covariance Q and measurement noise covariance R matrices might change with each time step or measurement, however here they are assumed to be constant. The $n \times n$ matrix A in the difference equation (5.20) relates the state at the previous time step $k-1$ to the state at the current step k, in the absence of either a driving function or a process noise. Note that in practice A might change with each time step, but here it is assumed to be constant. The $n \times 1$ matrix B relates the optional control input $u \in R^1$ to the state x. The $m \times n$ matrix H in the measurement equation (5.21) relates the state x to the measurement Z_k. In practice H might change with each time step or measurement, but here it is assumed to be constant.

5.1.2.1 The Computational Origins of the Filter

We define $\hat{x}_k^- \in R^n$ to be a *priori* state estimate at step k given knowledge of the process prior to step k, and $\hat{x}_k \in R^n$ to be a *posteriori* state estimate at step k given measurement Z_k. We can then define a *priori* and a *posteriori* estimate errors as

$$e_k^- \equiv x_k - \hat{x}_k^- \tag{5.22}$$

And $$e_k \equiv x_k - \hat{x}_k \tag{5.23}$$

The *priori* estimate error covariance is then:

$$P_k^- = E\left[e_k^- \; e_k^{-T}\right], \tag{5.24}$$

And the *posteriori* estimate error covariance is:

$$P_k = E\left[e_k \; e_k^T\right], \tag{5.25}$$

In deriving the equations for the Kalman filter, the goal is to find an equation that computes *a posteriori* state estimate $\hat{x}_k$ as a linear combination of *a priori* estimate $\hat{x}_k^-$ and a weighted difference between an actual measurement Z_k and a measurement prediction $H\hat{x}_k^-$:

$$\hat{x}_k = \hat{x}_k^- + K(Z_k - H\hat{x}_k^-) \tag{5.26}$$

The difference $Z_k - H\hat{x}_k^-$ in (5.26) is called the measurements *innovation*, or the *residual*. The residual reflects the discrepancy between the predicted measurement and the actual measurement. A residual zero means that the two are in complete agreement.

The $n \times n$ matrix K in (5.26) is chosen to be the gain or *blending factor* that minimizes the *posteriori* error covariance in equation (5.25). This minimization can be accomplished by first substituting equation (5.26) into the equation (5.23) i.e. definition for e_k, substitution that into (5.25), performing the indicated expectations, taking the derivative of the trace of the result with respect to the K, setting that results equal to zero and then solving for K. One form of the resulting K that minimizes (5.25) is given by:

$$K_k = -P_k^- H^T (HP_k^- H^T + R)^{-1} \tag{5.27}$$

$$= \frac{P_k^- H^T}{HP_k^- H^T + R} \tag{5.28}$$

The Kalman filter is summarised as follow:

$$P_k^- = A_{k-1} P_{k-1} A_{k-1}^T + Q_{k-1} \tag{5.29}$$

$$K_k = P_k^- H_k^{\mathrm{T}} (H_k P_k^- H_k^{\mathrm{T}} + R_k)^{-1} \tag{5.30}$$

$$\hat{x}_k = A_{k-1}\hat{x}_{k-1} + K_k(z_k - H_k A_{k-1}\hat{x}_{k-1}) \tag{5.31}$$

$$P_k = (1 - K_k)P_k^- (1 - K_k)^{\mathrm{T}} + K_k R_k K_k^{\mathrm{T}} \tag{5.32}$$

The output is the optimal estimation of the state vector at time t_k, $\hat{x}_k$ and their uncertainties, given by the diagonal elements of P_k. The whole process of filtering is shown diagrammatically as follow:

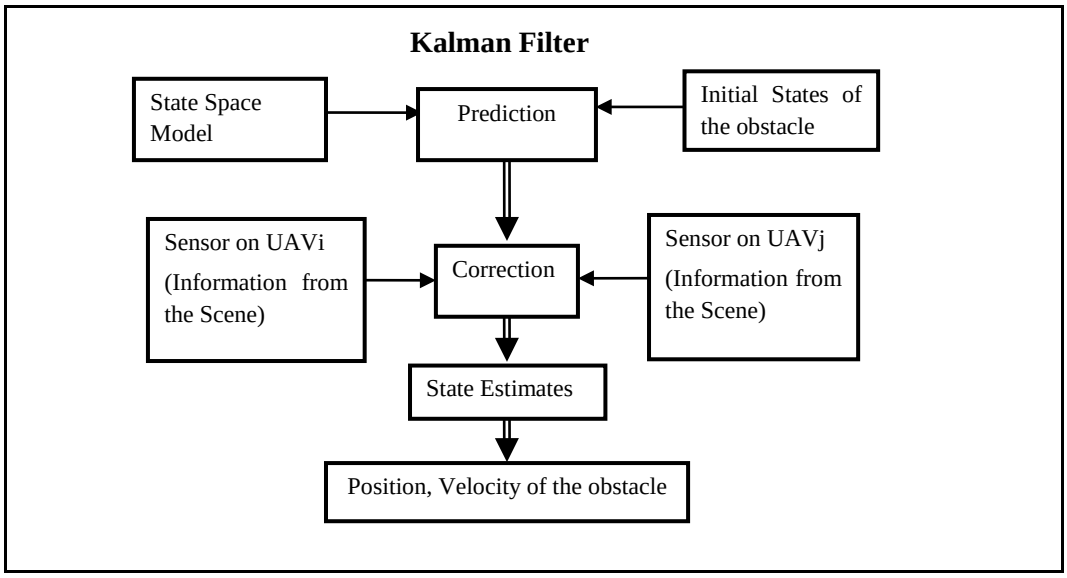

Figure 5.3: Block diagram of Kalman filter

5.2 Problem Formulation:

A target was tracked using two cameras on two different UAVs. The UAVs are assumed to be positioned such that the target is in the field of view of both the UAVs at all the time unless the line of sight is obstructed. In case of failure to acquire the target position information through UAVs cameras for a minute interval the Kalman filter is used for acquisition of target position estimates. The position of the target is denoted as $[x, y, z]$ and the state vector used is:

$$X = [x, y, z, \dot{x}, \dot{y}, \dot{z}]^T \tag{5.33}$$

Where $\dot{x}, \dot{y}, \dot{z}$ are the velocities components in x, y, z directions.

The discrete time target dynamics are assumed to be:

$$X_{k+1} = AX_k + w_k \tag{5.34}$$

$$Y_k = CX_k + v_k \tag{5.35}$$

Where w_k is the process noise and v_k is the measurement noise. The system and measurement matrices A and C used are given as follows:

$$A = \begin{bmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.36}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{5.37}$$

Where Δt is a small interval of time.

The details of the solution approaches to the problem of locating the target is discussed in the following sections.

5.3 Methods to Locate the Object in 3D (Solution Approach)

The main purpose of the cooperative perception is to locate an unknown static or dynamic object in 3D in a dynamic environment optimally and robustly. Since the 3D location of the obstacle cannot be calculated from single camera therefore cooperative perception system employs multiple cameras from multiple UAVs.

The solution to the above problem is approached in two different ways. The first solution is a simple solution as a starting point in which the location of the target has been calculated from multi-view ignoring the sensors (GPS and Camera) noises. The second solution is more complex and realistic in which the sensors (GPS and Camera) noises have been considered and optimised. These solution approaches are:

1. 3D target locating from multiple views
2. Optimization based 3D target locating.

The details of the above techniques are described in the following sections.

5.3.1 3D Target Locating from Multiple Views

In this method the position of the target at different intervals is acquired continuously with UAV cameras. The multiple UAVs constitute a multiple view scenario. The multiple view formulation consists of n pinhole camera models [70] as shown in figure 5.4. All vectors are written with respect to a common, global frame with origin O. The *ith* image, for $i = 1,2,.....n$, has a camera centre at $c_i(x_{ci}, y_{ci}, z_{ci})$, unit focal vector F_i and focal length f_i. For simplicity, let put $f_i = 1$.

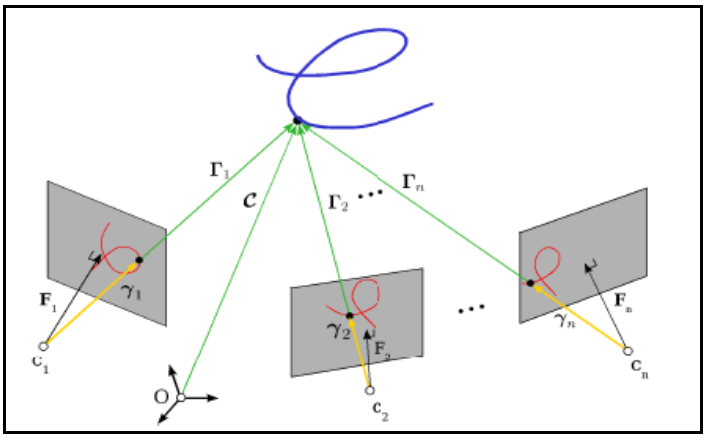

Figure 5.4: Multi-view setup.

Referring to figure 5.4:

$$\Gamma_i = C - c_i \tag{5.38}$$

relative to the camera frame.

Similarly, a 2D curve γ_i is the image of space curve C_i in image frame. Dropping the index i when not necessary, the space curve and the projected image curve are related by:

$$\Gamma(s) = \lambda(s)\gamma(s) \tag{5.39}$$

Where λ is a positive scalar.

Equation (5.38) can be written as:

$$C = c_i + \Gamma_i \tag{5.40}$$

For simplicity, it is assumed that $i = 1,2$ i.e. only two UAVs. Putting equation (5.39) in (5.40) yields:

$$C = c_1 + \lambda_1\gamma_1 \tag{5.41}$$

$$C = c_2 + \lambda_2\gamma_2 \tag{5.42}$$

The reconstruction of a point on the space curve C from two corresponding image curves points $\gamma_1 = (x_1, y_1, z_1)$ and $\gamma_2 = (x_2, y_2, z_2)$ can be obtained by equating equation (5.41) and (5.42), i.e:

$$c_1 + \lambda_1\gamma_1 = c_2 + \lambda_2\gamma_2$$

$$\lambda_1\gamma_1 - \lambda_2\gamma_2 = c_2 - c_1 \tag{5.43}$$

Or more explicitly in component form:

$$x_1\lambda_1 - x_2\lambda_2 = x_{c2} - x_{c1} \tag{5.44}$$

$$y_1\lambda_1 - y_2\lambda_2 = y_{c2} - y_{c1} \tag{5.45}$$

$$z_1\lambda_1 - z_2\lambda_2 = z_{c2} - z_{c1} \tag{5.46}$$

This system of three equations in two unknown λ_1 and λ_2 can only be solved if the lines $c_1\gamma_1$ and $c_2\gamma_2$ intersect.

Putting the calculated values of λ_1 and λ_2 in equation (5.41) or (5.42) the object can be located in 3D. The position of the object calculated in this way is used as measurement in the Kalman filter for estimation purpose.

5.3.2 Optimization Based 3D Target locating

Since camera is used as a sensor in this research to locate the target therefore like any sensor there will be accompanied noise. The camera is attached to the UAVs and the GPS sensors are used to locate the UAVs therefore there will be uncertainty in position of the UAVs and hence in the location of camera because of GPS sensor noise. To reduce these combined sensors noises optimisation is used to optimally locate the target. This method of locating the obstacle is based on optimization.

The cooperative UAVs vision based tracking problem is reformulated as linear estimation problem [55]. A target was tracked using two cameras on two different UAVs. Using the observed bearings of the target from each UAV, an estimate of the absolute target position was obtained by minimizing the errors in distance from the estimate to each measurement. This estimate is then used as a measurement input to a simple linear Kalman filter that uses the target location as the state variable.

The location of each UAV as given by the on board navigation sensor is:

$$\vec{x}_i = \hat{\vec{x}}_i + \delta\vec{x}_i \ , \quad \forall\vec{x}_i \in R^3 \tag{5.47}$$

Where $\hat{\vec{x}}_i$ is the position estimate and $\delta\vec{x}_i$ is the uncertainty in position estimate.

For each UAV there is a director vector to the target given by:

$$\vec{d}_i = \hat{\vec{d}}_i + \delta\vec{d}_i \ , \quad \forall\vec{d}_i \in R^3 \tag{5.48}$$

Where $\hat{\vec{d}}_i$ is the direction estimate and $\delta\vec{d}_i$ is the uncertainty in direction estimate. The unit direction vectors $\hat{\vec{d}}_i$ are calculated from the direction cosines of the target's images.

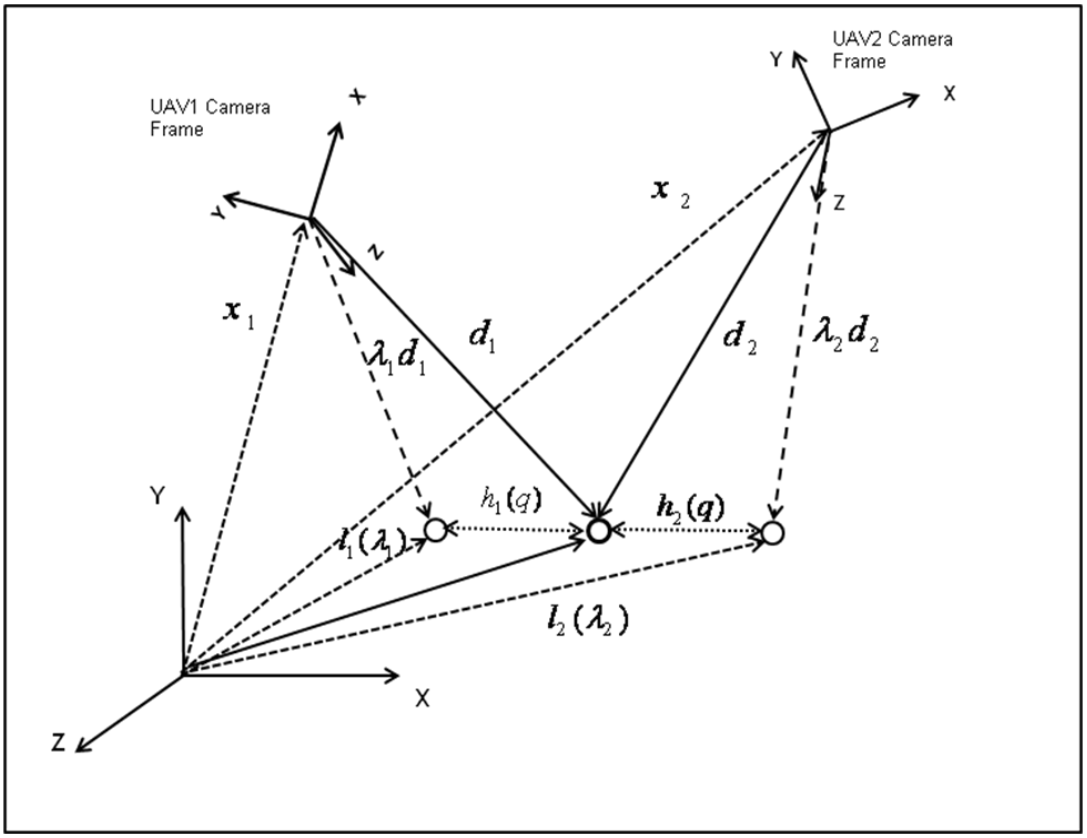

Figure 5.5: Optimization based target locating set up involving two UAVs

Note that given $\vec{x}_i$ and $\vec{d}_i$ the target must lie along the ray:

$$\vec{l}_i(\lambda_i) = \vec{x}_i + \lambda_i \vec{d}_i, \quad \lambda_i \geq 0 \tag{5.49}$$

In order to solve this problem the true position of the object $\vec{q}$, must be estimated given the set of all measurements $\{\hat{\vec{x}}_i, \hat{\vec{d}}_i : i = 1,.....,n\}$. This estimate $\vec{q}$ should minimize the error:

$$E(\vec{q}) = \sum_{i=1}^{n} w_i h_i(\vec{q}) \tag{5.50}$$

Where $h_i(q)$ is the square of the minimum distance from q to ray $\vec{l}_i(\lambda_i)$

$$h_i(\vec{q}) = \min \left\| \vec{q} - \vec{l}_i(\lambda_i) \right\|^2 = \min \left\| \vec{q} - (\vec{x}_i + \lambda_i \vec{d}_i) \right\|^2 \tag{5.51}$$

$$h_i(\vec{q}) = [\vec{q} - (\vec{x}_i + \lambda_i \vec{d}_i)]^T [\vec{q} - (\vec{x}_i + \lambda_i \vec{d}_i)] \tag{5.52}$$

$$h_i(\vec{q}) = [\vec{q}^T - (\vec{x}_i^T + \lambda_i \vec{d}_i^T)][\vec{q} - (\vec{x}_i + \lambda_i \vec{d}_i)] \tag{5.53}$$

Differentiating with respect to λ_i:

$$\frac{\partial h_i(\vec{q})}{\partial \lambda_i} = -\vec{d}_i{}^T[\vec{q} - (\vec{x}_i + \lambda_i \vec{d}_i)] + [\vec{q} - (\vec{x}_i + \lambda_i \vec{d}_i)]^T(-\vec{d}_i) \tag{5.54}$$

$$\frac{\partial h_i(\vec{q})}{\partial \lambda_i} = -\vec{d}_i{}^T\vec{q} + \vec{d}_i{}^T\vec{x}_i + \vec{d}_i{}^T\lambda_i\vec{d}_i - \vec{q}^T\vec{d}_i + \vec{x}_i{}^T\vec{d}_i + \lambda_i\vec{d}_i{}^T\vec{d}_i \tag{5.55}$$

$$= -\vec{d}_i{}^T\vec{q} + \vec{d}_i{}^T\vec{x}_i + \vec{d}_i{}^T\lambda_i\vec{d}_i - \vec{q}^T\vec{d}_i + \vec{x}_i{}^T\vec{d}_i + \lambda_i\vec{d}_i{}^T\vec{d}_i \tag{5.56}$$

As $\vec{d}_i{}^T\lambda_i\vec{d}_i = \lambda_i\vec{d}_i{}^T\vec{d}_i = \lambda_i$, for $\vec{d}_i$ is a unit vector, therefore equation (5.56) can be written as:

$$\frac{\partial h_i(\vec{q})}{\partial \lambda_i} = -\vec{d}_i{}^T\vec{q} + \vec{d}_i{}^T\vec{x}_i + \lambda_i - \vec{q}^T\vec{d}_i + \vec{x}_i{}^T\vec{d}_i + \lambda_i \tag{5.57}$$

For maximum or minimum, $\frac{\partial h_i(\vec{q})}{\partial \lambda_i} = 0$ in equation (5.57) yields:

$$-\vec{d}_i{}^T\vec{q} + \vec{d}_i{}^T\vec{x}_i + \lambda_i - \vec{q}^T\vec{d} + \vec{x}_i{}^T\vec{d}_i + \lambda_i = 0 \tag{5.58}$$

$$\Rightarrow 2(-\vec{d}_i{}^T\vec{q} + \vec{d}_i{}^T\vec{x}_i + \lambda_i) = 0 \tag{5.59}$$

$$\Rightarrow \lambda_i = \vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i \tag{5.60}$$

Now putting this value of λ_i in the equation (5.49):

$$h_i(\vec{q}) = [\vec{q} - \vec{x}_i - (\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)\vec{d}_i]^T[\vec{q} - \vec{x}_i - (\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)\vec{d}_i]$$

$$\Rightarrow h_i(\vec{q}) = [(\vec{q} - \vec{x}_i)^T - \{(\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)\vec{d}_i\}^T][\vec{q} - \vec{x}_i - (\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)\vec{d}_i]$$

$$\Rightarrow h_i(\vec{q}) = (\vec{q}^T - \vec{x}_i{}^T)(\vec{q} - \vec{x}_i) - (\vec{q} - \vec{x}_i)^T(\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)\vec{d}_i +$$
$$-\{(\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)\vec{d}_i\}^T(\vec{q} - \vec{x}_i) + \{(\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)\vec{d}_i\}^T(\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)\vec{d}_i$$

$$\Rightarrow h_i(\vec{q}) = \vec{q}^T\vec{q} - \vec{q}^T\vec{x}_i - \vec{x}_i{}^T\vec{q} + \vec{x}_i{}^T\vec{x}_i - 2(\vec{q} - \vec{x}_i)^T(\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)\vec{d}_i + (\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)^2$$

$$\Rightarrow h_i(\vec{q}) = \vec{q}^T\vec{q} - 2\vec{q}^T\vec{x}_i + \vec{x}_i{}^T\vec{x}_i - 2(\vec{q} - \vec{x}_i)^T(\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)\vec{d}_i + (\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)^2$$

$$\Rightarrow h_i(\vec{q}) = \vec{q}^T\vec{q} - 2\vec{q}^T\vec{x}_i + \vec{x}_i{}^T\vec{x}_i - 2(\vec{q} - \vec{x}_i)^T\vec{d}_i{}^T(\vec{q} - \vec{x}_i)\vec{d}_i + (\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)^2$$

$$\Rightarrow h_i(\vec{q}) = \vec{q}^T\vec{q} - 2\vec{q}^T\vec{x}_i + \vec{x}_i{}^T\vec{x}_i - 2(\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)^2 + (\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)^2$$

$$\Rightarrow h_i(\vec{q}) = \vec{q}^T\vec{q} - 2\vec{q}^T\vec{x}_i + \vec{x}_i{}^T\vec{x}_i - (\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)^2 \tag{5.61}$$

Which is the minimum value of $h_i(q)$ and can also be written as:

$$h_i(\vec{q}) = \vec{q}^T\vec{q} - 2\vec{q}^T\vec{x}_i + \vec{x}_i{}^T\vec{x}_i - (\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)^T(\vec{d}_i{}^T\vec{q} - \vec{d}_i{}^T\vec{x}_i)$$

$$\Rightarrow h_i(\vec{q}) = \vec{q}^T\vec{q} - 2\vec{q}^T\vec{x}_i + \vec{x}_i^T\vec{x}_i - (\vec{q}^T\vec{d}_i - \vec{x}_i^T\vec{d}_i)^T(\vec{d}_i^T\vec{q} - \vec{d}_i^T\vec{x}_i)$$

$$\Rightarrow h_i(\vec{q}) = \vec{q}^T\vec{q} - 2\vec{q}^T\vec{x}_i + \vec{x}_i^T\vec{x}_i - (\vec{q}^T\vec{d}_i\vec{d}_i^T\vec{q} - \vec{q}^T\vec{d}_i\vec{d}_i^T\vec{x}_i - \vec{x}_i^T\vec{d}_i\vec{d}_i^T\vec{q} + \vec{x}_i^T\vec{d}_i\vec{d}_i^T\vec{x}_i) \quad 5.62)$$

Substituting this result in equation (5.50) and minimizing with respect to q:

$$E(\vec{q}) = w_1\left[\vec{q}^T\vec{q} - 2\vec{q}^T\vec{x}_1 + \vec{x}_1^T\vec{x}_1 - (\vec{q}^T\vec{d}_1\vec{d}_1^T\vec{q} - \vec{q}^T\vec{d}_1\vec{d}_1^T\vec{x}_1 - \vec{x}_1^T\vec{d}_1\vec{d}_1^T\vec{q} + \vec{x}_1^T\vec{d}_1\vec{d}_1^T\vec{x}_1)\right]$$

$$+ w_2\left[\vec{q}^T\vec{q} - 2\vec{q}^T\vec{x}_2 + \vec{x}_2^T\vec{x}_2 - (\vec{q}^T\vec{d}_2\vec{d}_2^T\vec{q} - \vec{q}^T\vec{d}_2\vec{d}_2^T\vec{x}_2 - \vec{x}_2^T\vec{d}_2\vec{d}_2^T\vec{q} + \vec{x}_2^T\vec{d}_2\vec{d}_2^T\vec{x}_2)\right]$$

Differentiating:

$$\Rightarrow \frac{\partial E(\vec{q})}{\partial q} = w_1\left[\vec{q}^T + \vec{q}^T + 2\vec{x}_1^T + 0 - \{(\vec{d}_1\vec{d}_1^T\vec{q})^T + \vec{q}^T\vec{d}_1\vec{d}_1^T - (\vec{d}_1\vec{d}_1^T\vec{x}_1)^T - \vec{x}_1^T\vec{d}_1\vec{d}_1^T\right]$$

$$+ w_2\left[\vec{q}^T + \vec{q}^T + 2\vec{x}_2^T + 0 - \{(\vec{d}_2\vec{d}_2^T\vec{q})^T + \vec{q}^T\vec{d}_2\vec{d}_2^T - (\vec{d}_2\vec{d}_2^T\vec{x}_2)^T - \vec{x}_2^T\vec{d}_2\vec{d}_2^T\right]$$

$$\Rightarrow \frac{\partial E(\vec{q})}{\partial q} = w_1\left[\vec{q}^T + \vec{q}^T + 2\vec{x}_1^T - \{\vec{q}^T\vec{d}_1\vec{d}_1^T + \vec{q}^T\vec{d}_1\vec{d}_1^T - \vec{x}_1^T\vec{d}_1\vec{d}_1^T - \vec{x}_1^T\vec{d}_1\vec{d}_1^T\}\right]$$

$$+ w_2\left[\vec{q}^T + \vec{q}^T + 2\vec{x}_2^T - \{\vec{q}^T\vec{d}_2\vec{d}_2^T + \vec{q}^T\vec{d}_2\vec{d}_2^T - \vec{x}_2^T\vec{d}_2\vec{d}_2^T - \vec{x}_2^T\vec{d}_2\vec{d}_2^T\}\right]$$

$$\Rightarrow \frac{\partial E(\vec{q})}{\partial q} = w_1\left[2\vec{q}^T + 2\vec{x}_1^T - \{2\vec{q}^T\vec{d}_1\vec{d}_1^T - 2\vec{x}_1^T\vec{d}_1\vec{d}_1^T\}\right]$$

$$+ w_2\left[2\vec{q}^T + 2\vec{x}_2^T - \{2\vec{q}^T\vec{d}_2\vec{d}_2^T - 2\vec{x}_2^T\vec{d}_2\vec{d}_2^T\}\right]$$

$$\Rightarrow \frac{\partial E(\vec{q})}{\partial q} = 2w_1\left[\vec{q}^T + \vec{x}_1^T - (\vec{q}^T - \vec{x}_1^T)\vec{d}_1\vec{d}_1^T\right] + 2w_2\left[\vec{q}^T + \vec{x}_2^T - (\vec{q}^T - \vec{x}_2^T)\vec{d}_2\vec{d}_2^T\right] \quad (5.63)$$

For minimum value of $E(\vec{q})$, putting $\frac{\partial E(\vec{q})}{\partial q} = 0$ in equation (5.63)

$$2w_1\left[\vec{q}^T + \vec{x}_1^T - (\vec{q}^T - \vec{x}_1^T)\vec{d}_1\vec{d}_1^T\right] + 2w_2\left[\vec{q}^T + \vec{x}_2^T - (\vec{q}^T - \vec{x}_2^T)\vec{d}_2\vec{d}_2^T\right] = 0$$

$$\Rightarrow w_1\left[\vec{q}^T(I - \vec{d}_1\vec{d}_1^T) - (\vec{x}_1^T - \vec{x}_1^T\vec{d}_1\vec{d}_1^T)\right] + w_2\left[\vec{q}^T(I - \vec{d}_2\vec{d}_2^T) - (\vec{x}_2^T - \vec{x}_2^T\vec{d}_2\vec{d}_2^T)\right] = 0$$

$$\Rightarrow w_1\left[\vec{q}^T(I - \vec{d}_1\vec{d}_1^T)\right] + w_2\left[\vec{q}^T(I - \vec{d}_2\vec{d}_2^T)\right] - w_1\left[\vec{x}_2^T - \vec{x}_2^T\vec{d}_2\vec{d}_2^T\right] + w_2\left[\vec{x}_2^T - \vec{x}_2^T\vec{d}_2\vec{d}_2^T\right] = 0$$

$$\Rightarrow \sum_{i=1}^{2} w_i\left[\vec{q}^T(I - \vec{d}_i\vec{d}_i^T)\right] - \sum_{i=1}^{2} w_i\left[\vec{x}_i^T - \vec{x}_i^T\vec{d}_i\vec{d}_i^T\right] = 0$$

$$\Rightarrow \sum_{i=1}^{2} w_i\left[\vec{q}^T(I - \vec{d}_i\vec{d}_i^T)\right] = \sum_{i=1}^{2} w_i\left[\vec{x}_i^T - \vec{x}_i^T\vec{d}_i\vec{d}_i^T\right]$$

$$\Rightarrow \sum_{i=1}^{2} w_i\left[\vec{q}^T(I - \vec{d}_i\vec{d}_i^T)\right]^T = \sum_{i=1}^{2} w_i\left[\vec{x}_i^T - \vec{x}_i^T\vec{d}_i\vec{d}_i^T\right]^T$$

$$\Rightarrow \sum_{i=1}^{2} w_i \left[(I - \vec{d}_i \vec{d}_i{}^T) \right]^T \vec{q} = \sum_{i=1}^{2} w_i \left[\vec{x}_i - \vec{d}_i \vec{d}_i{}^T \vec{x}_i \right]$$

$$\Rightarrow \sum_{i=1}^{2} w_i \left[(I - \vec{d}_i \vec{d}_i{}^T) \right] \vec{q} = \sum_{i=1}^{2} w_i \left[\vec{x}_i - \vec{d}_i \vec{d}_i{}^T \vec{x}_i \right] \tag{5.64}$$

As $\vec{q}$ is optimal therefore denoting $\vec{q} = \vec{q}^*$ and let:

$$A = \sum_{i=1}^{2} w_i \left[(I - \vec{d}_i \vec{d}_i{}^T) \right]$$

$$B = \sum_{i=1}^{2} w_i \left[\vec{x}_i - \vec{d}_i \vec{d}_i{}^T \vec{x}_i \right]$$

For a group of two UAVs i.e. $i = 1,2$

$$A = w_1 \left[(I - \vec{d}_1 \vec{d}_1{}^T) \right] + w_2 \left[(I - \vec{d}_2 \vec{d}_2{}^T) \right] \tag{5.65}$$

$$B = w_1 \left[\vec{x}_1 - \vec{d}_1 \vec{d}_1{}^T \vec{x}_1 \right] + w_2 \left[\vec{x}_2 - \vec{d}_2 \vec{d}_2{}^T \vec{x}_2 \right] \tag{5.66}$$

The equation (5.64) can be written as:

$$A \vec{q}^* = B \tag{5.67}$$

Now the matrices A and B needed to be calculated. Since $\vec{d}_1, \vec{d}_2$ and $\vec{x}_1, \vec{x}_2$ are not known therefore A and B can not be calculated from above equations. Hence putting equation (5.47) and (5.48) in equations (5.65) and (5.66) yield:

$$A = w_1 \left[I - (\hat{\vec{d}}_1 + \delta \vec{d}_1)(\hat{\vec{d}}_1 + \delta \vec{d}_1)^T \right] + w_2 \left[I - (\hat{\vec{d}}_2 + \delta \vec{d}_2)(\hat{\vec{d}}_2 + \delta \vec{d}_2)^T \right] \tag{5.68}$$

$$B = w_1 \left[\hat{\vec{x}}_1 + \delta \vec{x}_1 - (\hat{\vec{x}}_1 + \delta \vec{x}_1)^T (\hat{\vec{d}}_1 + \delta \vec{d}_1)(\hat{\vec{d}}_1 + \delta \vec{d}_1) \right] +$$
$$w_2 \left[\hat{\vec{x}}_2 + \delta \vec{x}_2 - (\hat{\vec{x}}_2 + \delta \vec{x}_2)^T (\hat{\vec{d}}_2 + \delta \vec{d}_2)(\hat{\vec{d}}_2 + \delta \vec{d}_2) \right] \tag{5.69}$$

After multiplication dropping the second order terms and making A and B the subject of equations (5.68) and (5.69) we have:

$$A = \hat{A} + \delta A \tag{5.70}$$

$$B = \hat{B} + \delta B \tag{5.71}$$

Where:

$$\hat{A} = w_1 \left[(I - \hat{\vec{d}}_1 \hat{\vec{d}}_1{}^T) \right] + w_2 \left[(I - \hat{\vec{d}}_2 \hat{\vec{d}}_2{}^T) \right]$$

$$\delta A = w_1\left[\delta\vec{d}_1\hat{\vec{d}}_1^{\,T} - \hat{\vec{d}}_1^{\,T}\delta\vec{d}_1\right] + w_2\left[\delta\vec{d}_2\hat{\vec{d}}_2^{\,T} - \hat{\vec{d}}_2^{\,T}\delta\vec{d}_2\right]$$

$$\hat{B} = w_1\left[\vec{x}_1 - \vec{x}_1^{\,T}\hat{\vec{d}}_1\hat{\vec{d}}_1\right] + w_2\left[\vec{x}_2 - \vec{x}_2^{\,T}\hat{\vec{d}}_2\hat{\vec{d}}_2\right]$$

$$\delta B = w_1\left[\delta\vec{x}_1 - \hat{\vec{x}}_1^{\,T}\delta\vec{d}_1\hat{\vec{d}}_1 - \delta\vec{x}_1^{\,T}\hat{\vec{d}}_1\hat{\vec{d}}_1 - \delta\vec{x}_1^{\,T}\hat{\vec{d}}_1\delta\vec{d}_1\right] +$$
$$w_2\left[\delta\vec{x}_2 - \hat{\vec{x}}_2^{\,T}\delta\vec{d}_2\hat{\vec{d}}_2 - \delta\vec{x}_2^{\,T}\hat{\vec{d}}_2\hat{\vec{d}}_2 - \delta\vec{x}_2^{\,T}\hat{\vec{d}}_2\delta\vec{d}_2\right]$$

Now $\hat{A}$ and $\hat{B}$ involve quantities that are measured directly by the sensors therefore they are known. δA and δB are random variables because they involved uncertainties δx_i and δd_i whose distributions are known, so they are known as well, therefore the optimal estimates is given by:

$$\vec{q}^{\,*} = A^{-1}B \tag{5.72}$$

Putting equations (5.70) and (5.71) in (5.72):

$$\vec{q}^{\,*} = (\hat{A} + \delta A)^{-1}(\hat{B} + \delta B) \tag{5.73}$$

Expanding $(\hat{A} + \delta A)^{-1}$ by Taylor series and ignoring the higher power terms:

$$(\hat{A} + \delta A)^{-1} = \hat{A}^{-1} - \hat{A}^{-1}\delta A\hat{A}^{-1} \tag{5.74}$$

Putting equation (5.74) in (5.73):

$$\vec{q}^{\,*} = (\hat{A}^{-1} - \hat{A}^{-1}\delta A\hat{A}^{-1})(\hat{B} + \delta B)$$

$$\Rightarrow \vec{q}^{\,*} = \hat{A}^{-1}\hat{B} + \hat{A}^{-1}\delta B - \hat{A}^{-1}\delta A\hat{A}^{-1}\hat{B} - \hat{A}^{-1}\delta A\hat{A}^{-1}\delta B \tag{5.75}$$

Ignoring the term $-\hat{A}^{-1}\delta A\hat{A}^{-1}\delta B$ in equation (5.75):

$$\vec{q}^{\,*} \approx \hat{A}^{-1}\hat{B} + \hat{A}^{-1}\delta B - \hat{A}^{-1}\delta A\hat{A}^{-1}\hat{B} \tag{5.76}$$

Putting $\hat{q}^{\,*} = \hat{A}^{-1}\hat{B}$ and $\delta q^{\,*} = \hat{A}^{-1}\delta B - \hat{A}^{-1}\delta A\hat{A}^{-1}\hat{B}$ in the equation (5.76):

$$\vec{q}^{\,*} = \hat{q}^{\,*} + \delta q^{\,*} \tag{5.77}$$

$\hat{q}^{\,*}$ is the optimal estimate which can be calculated from the measurements. $\delta q^{\,*}$ is the error in the estimates. Once the $\hat{q}^{\,*}$ is known it is used as a measurement in the Kalman filter by putting in equation (5.35) as follows:

$$Y_k = \hat{q}^{\,*} = CX_k + \delta q^{\,*}$$

5.4 Simulation Results

The simulation results are obtained for both methods separately. In the following first the simulation results for target locating from multi-view are discussed and then the simulation results for the optimization based target locating are given.

5.4.1 Simulation Results of Target locating in 3D using Multiple View

In this simulation the target trajectory is calculated from cooperating UAVs constituting multiple views. In this simulation the sensors noises are completely ignored.

Figure 5.6 shows the original target trajectory that is to be calculated from cooperating UAVs constituting multi-view.

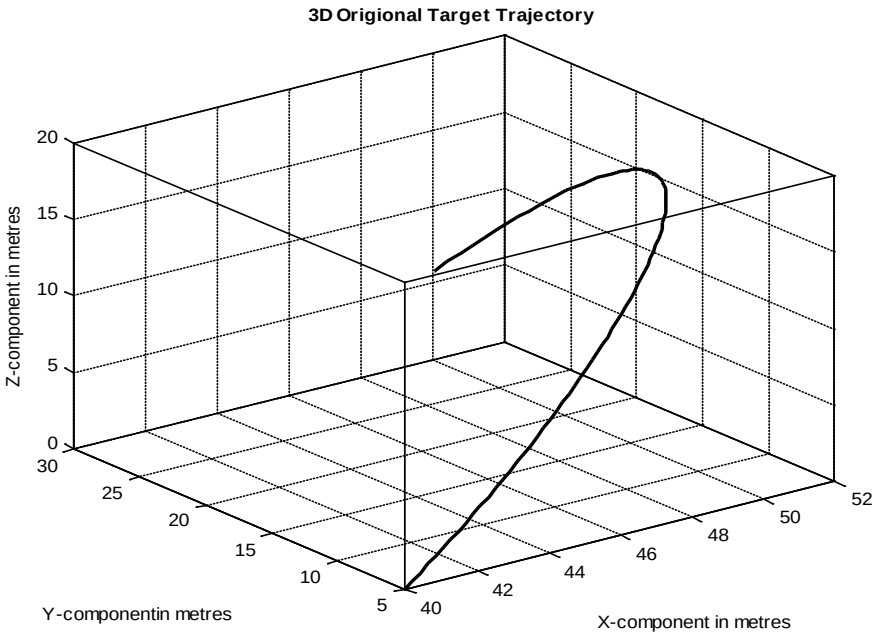

Figure 5.6: The original target trajectory

Figure 5.7 shows the original target trajectory and the target trajectory calculated from cooperating UAVs constituting multiple views. The original trajectory and the calculated trajectory from multiple views almost agree completely.

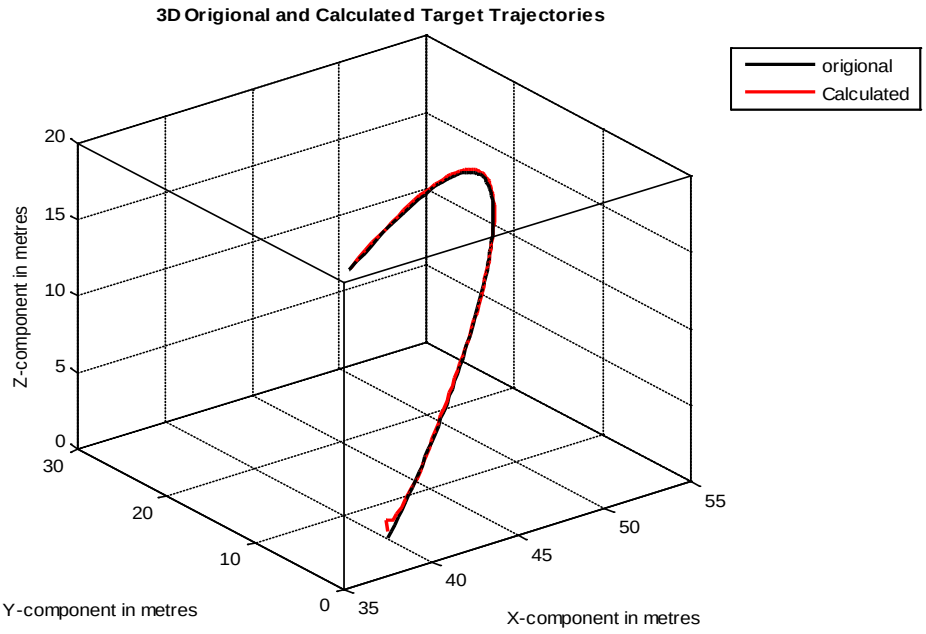

Figure 5.7: Comparison of the original and calculated target trajectory

Figure 5.8, figure 5.9 and figure 5.10 compare the x-component, the y-component, and the z-component of the original target trajectory and the calculated target trajectory respectively.

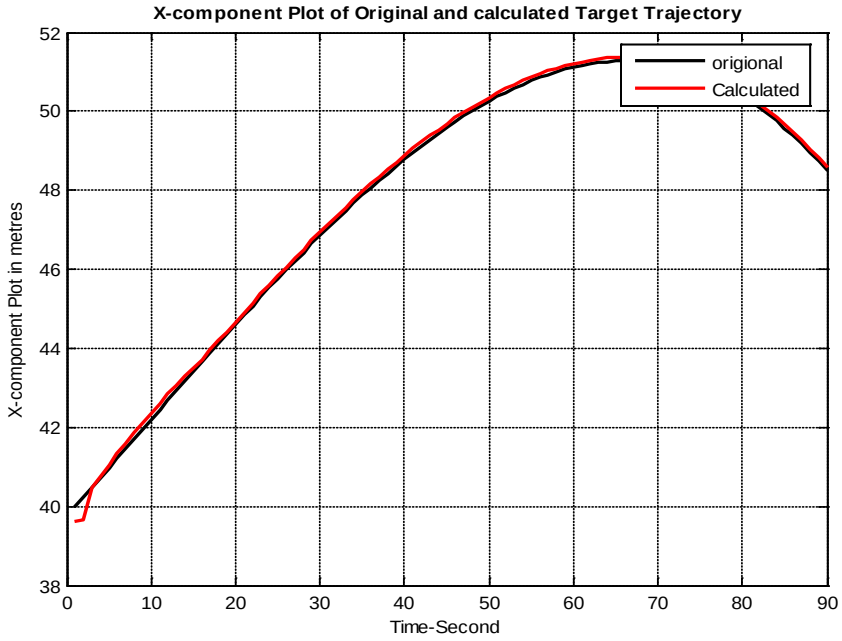

Figure 5.8: Comparison of the x-components of original and calculated trajectory with time

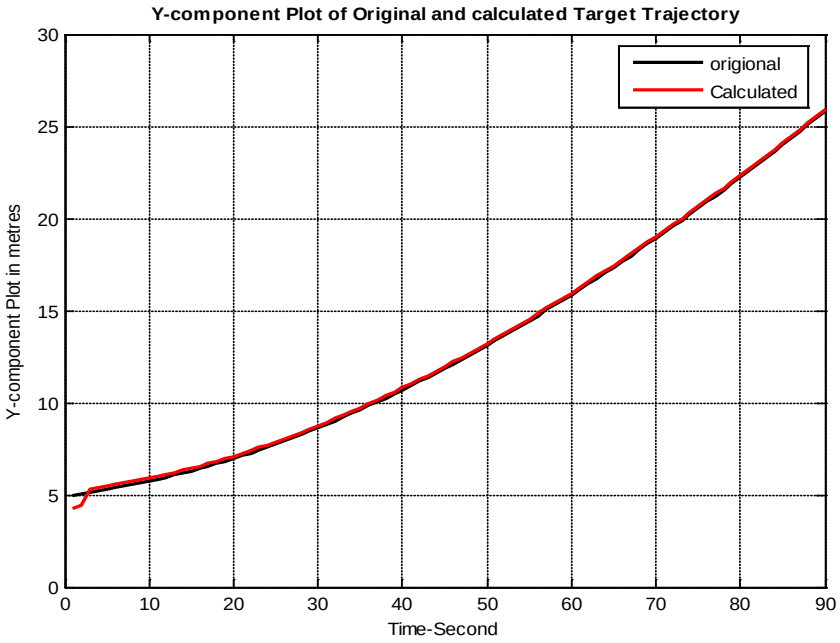

Figure 5.9: Comparison of the y-components of original and calculated trajectory with time

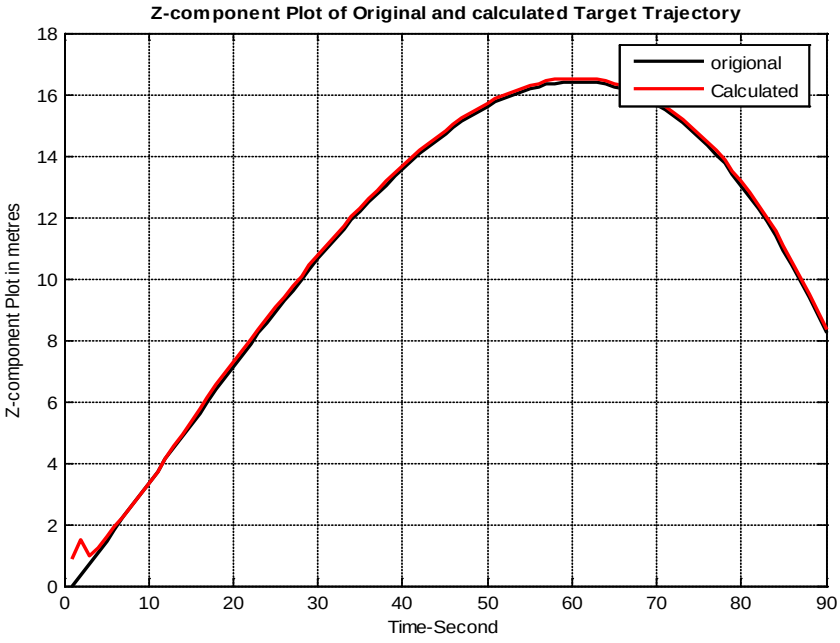

Figure 5.10: Comparison of the z-components of original and calculated trajectory with time

Figure 5.11, figure 5.12 and figure 5.13 show the error in x-component, the error in y-component and the error in z-component of the calculated target's trajectory respectively.

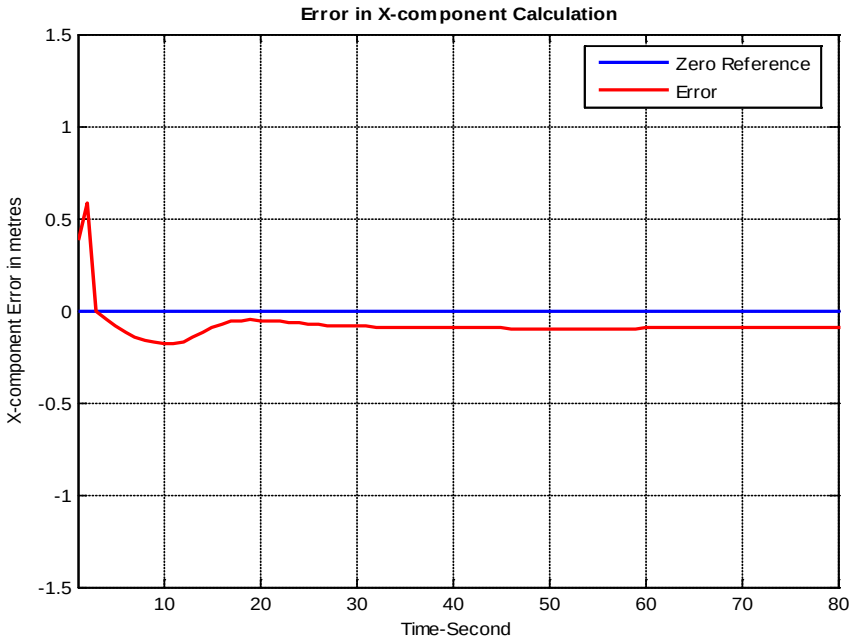

Figure 5.11: The error plot of x-component of calculated trajectory with time

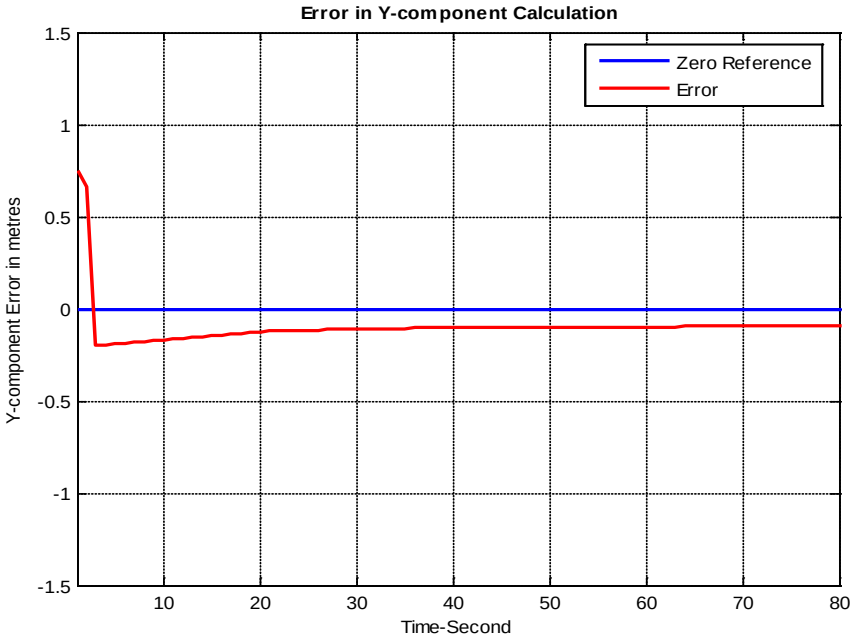

Figure 5.12: The error plot of y-component of calculated trajectory with time.

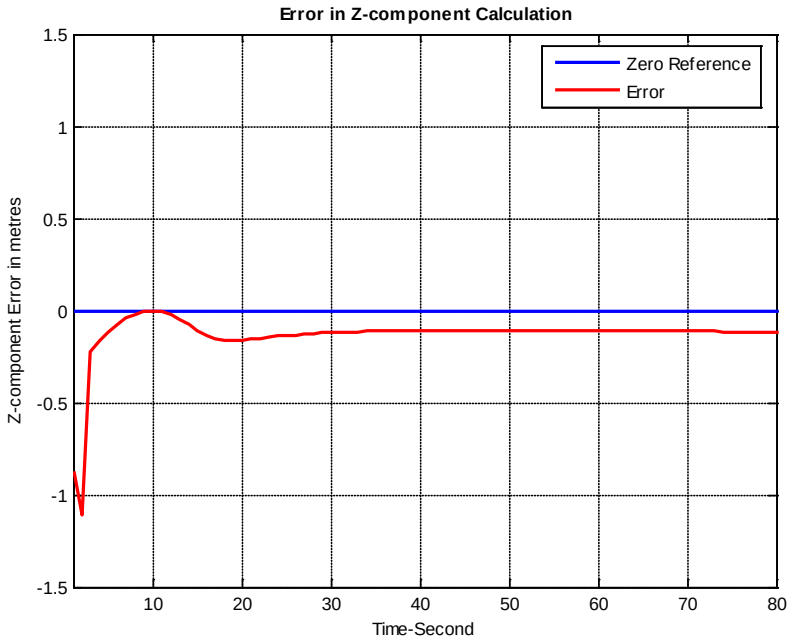

Figure 5.13: The error plot of z-component of calculated trajectory with time

5.4.2 Simulation Results of 3D Target locating using Optimization

In this simulation the sensors noises are taken into consideration to accommodate the real challenges involved in practical missions. The cooperative results obtained are optimised to get the optimal estimates of the target location.

The simulation results for 3D target tracking based on optimisation are given in the following series of figures. Figure 5.14 shows the real trajectory that is to be estimated by cooperative UAVs.

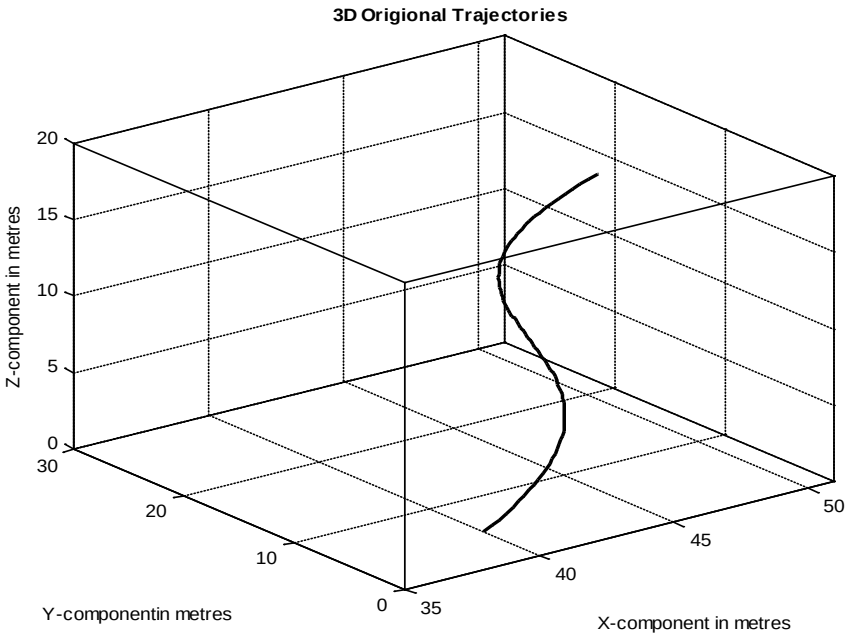

Figure 5.14: Original trajectory of the target

Figure 5.15 shows the original target trajectory, and the estimated target trajectory. The result shows that the original and estimated trajectories agree. The estimated trajectory deviates in the start but converges after some time.

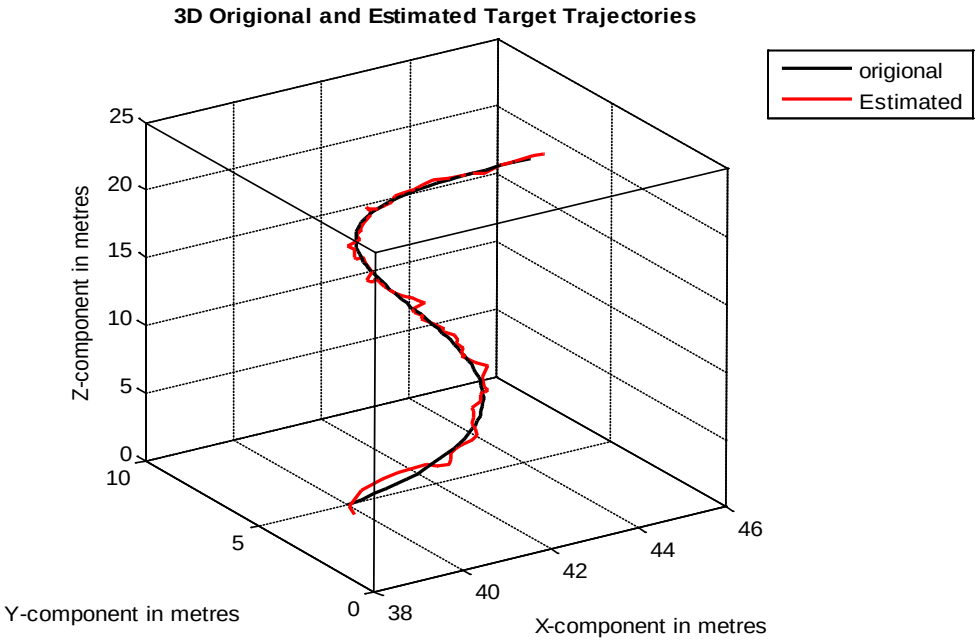

Figure 5.15: Comparison of original and estimated trajectory

Figures 5.16, figure 5.17 and figure 5.18 show the comparison of x, y and z components of original and estimated trajectories. The result shows an acceptable convergence.

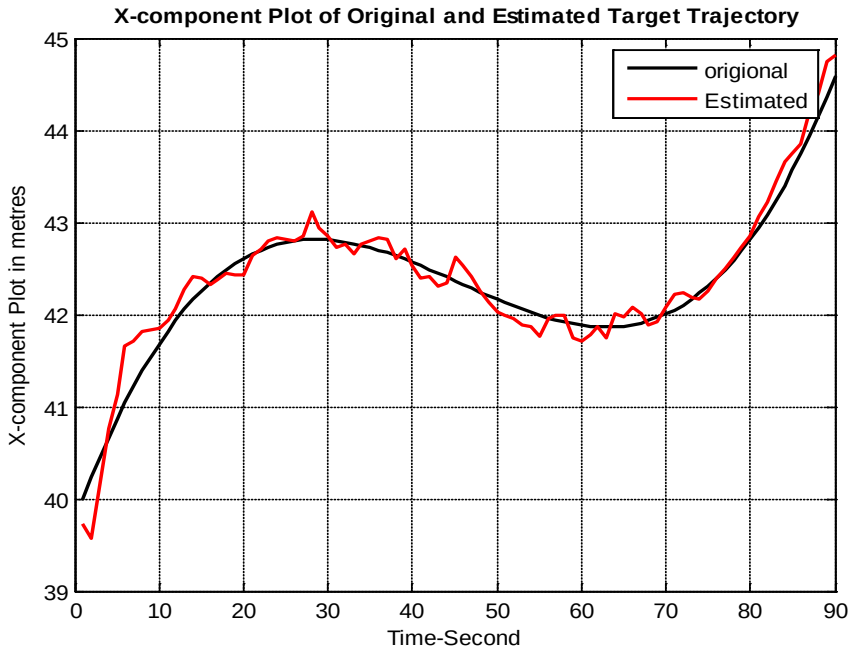

Figure 5.16: Comparison of the x-components of original and estimated trajectories

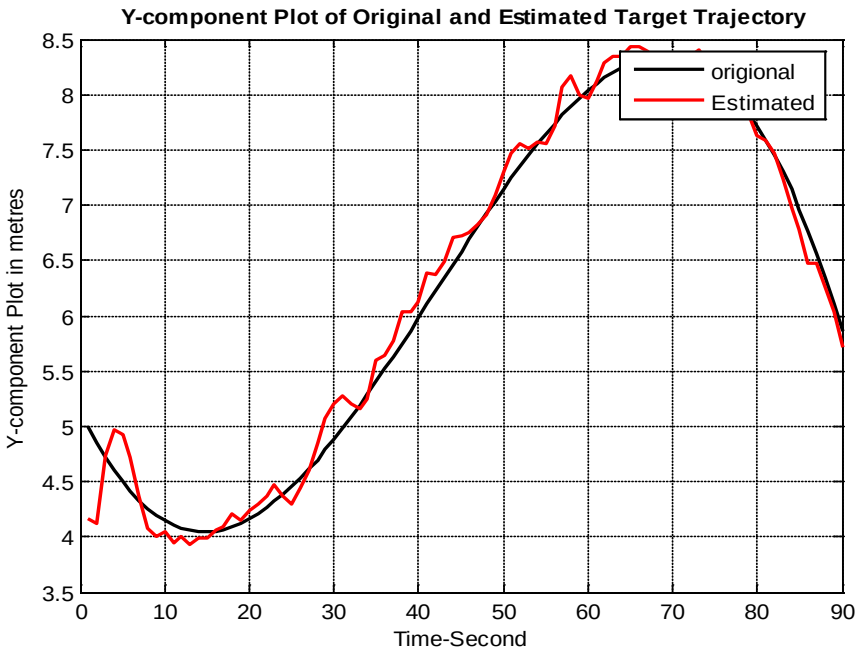

Figure 5.17: Comparison of the y-components of original and estimated trajectories

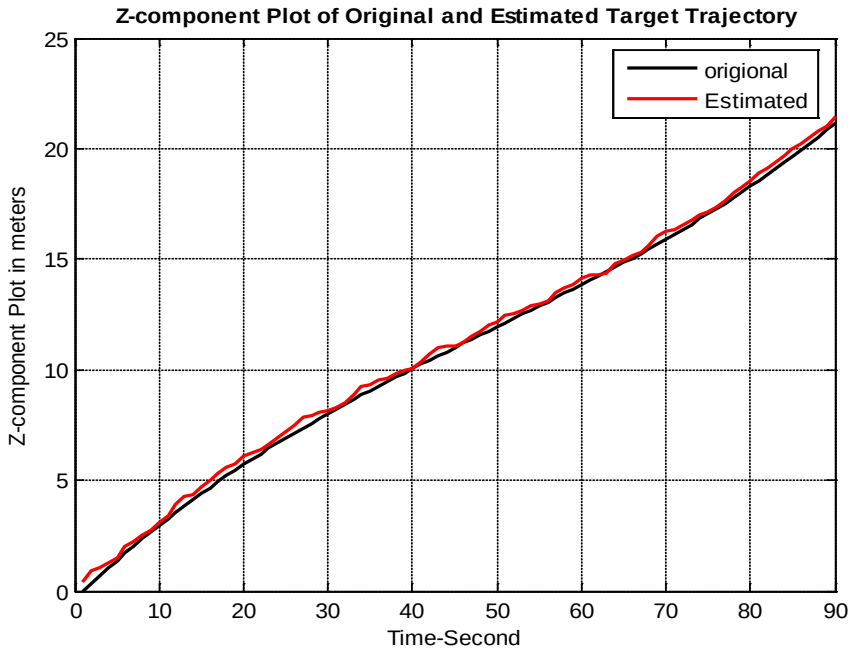

Figure 5.18: Comparison of the z-components of original and estimated trajectories

Figure 5.19, figure 5.20 and figure 5.21 show the component wise errors in the estimated trajectory. The error is large at the beginning but gradually reduces with time.

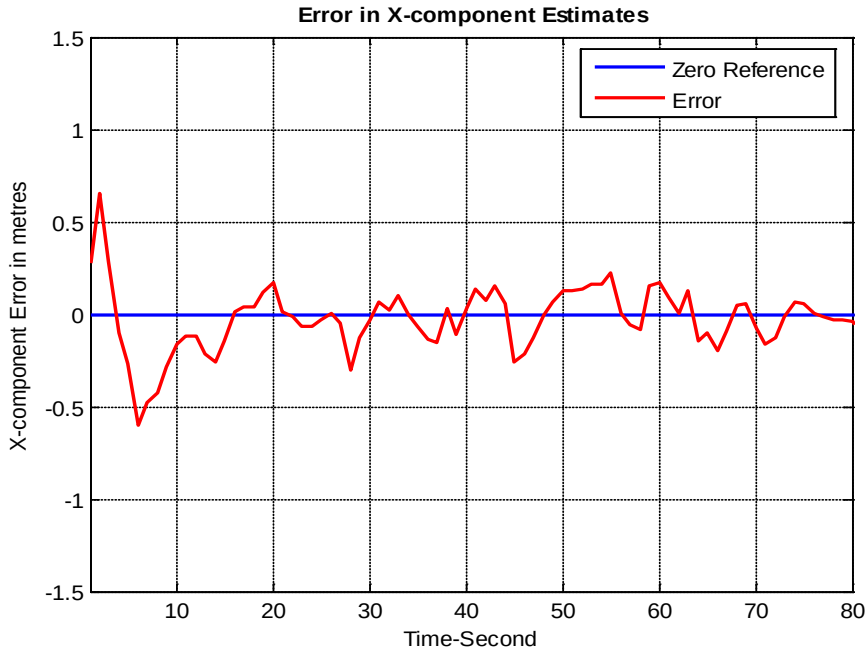

Figure 5.19: The error plot of x-component of estimated trajectory with time

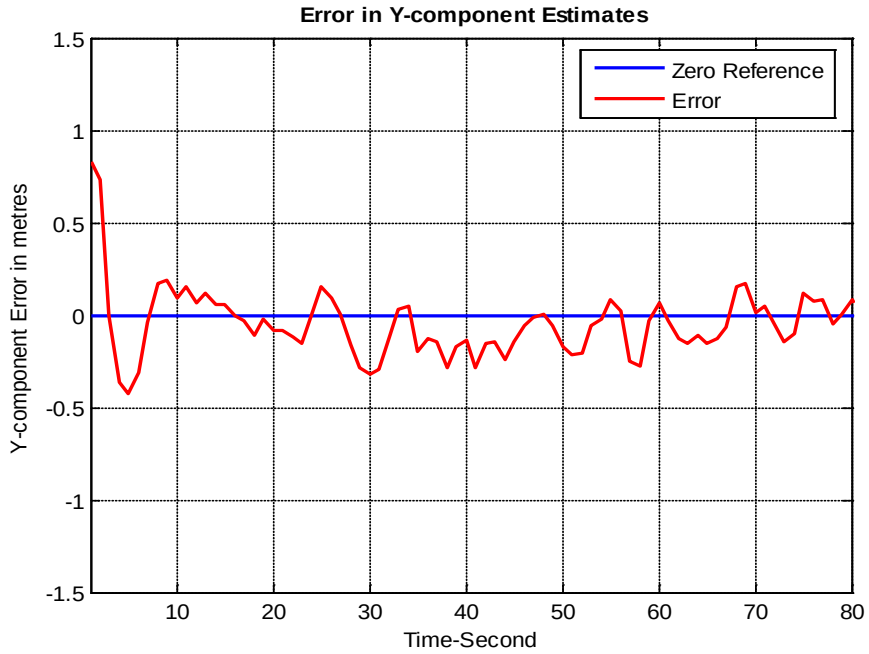

Figure 5.20: The error plot of y-component of estimated trajectory with time.

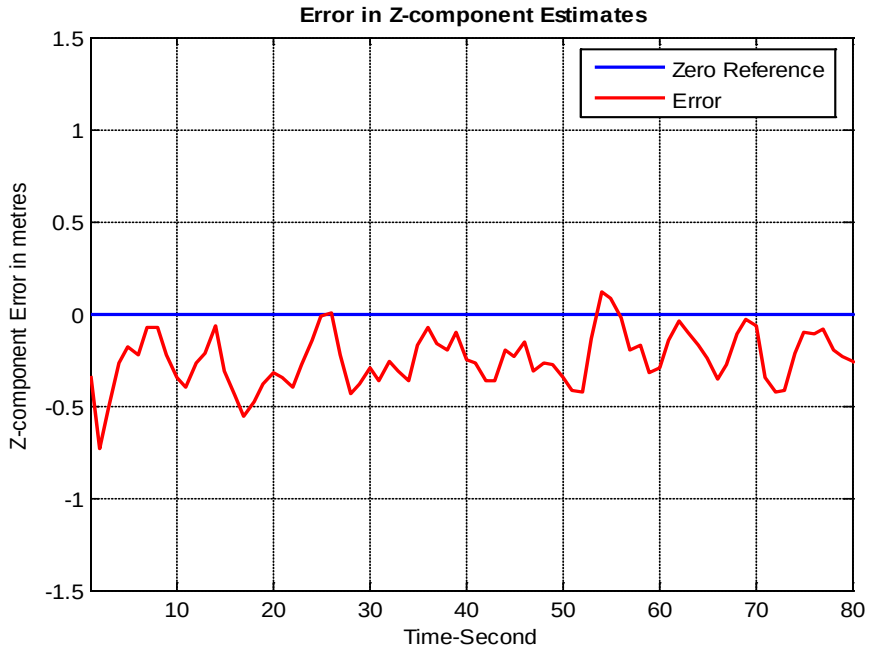

Figure 5.21: The error plot of z-component of estimated trajectory with time

5.5 Conclusion

In this chapter camera based cooperative perception is discussed. A camera based cooperative target locating algorithm is developed. This algorithm is then used by the group of cooperating UAVs to cooperatively locate the object in the scene. The simulation results show the efficiency of the proposed algorithm. From the simulation

results it can be seen that the position estimates converges in acceptably small duration. The accuracy of the position estimates is also seen to be acceptable. In the subsequent chapters this algorithm is used in combination with path planning algorithm for reactive obstacle Avoidance (reactive obstacle localisation and avoidance) and ground target tracking and surveillance.

Chapter 6

Reactive (Online) Obstacles Avoidance in the Frame Work of Pythagorean Hodograph Based Path Planning

Much of the benefits of deploying unmanned aerial vehicles (UAVs) in missions can be derived from autonomous operations. Path planning with collision avoidance is an important problem which needed to be addressed to ensure safety of the UAVs in autonomous operations. UAVs are expected to be ubiquitous in near future, autonomously performing complex military and civil operations such as border patrol, search and rescue operation, disaster relief, traffic monitoring, convey escorting, target tracking and surveillance etc. Many of these applications require the UAV to fly at low altitudes in proximity with manmade and natural structures. A collision with these structures could end in disaster. Therefore it is required that the UAVs must be able to sense and avoid obstacles and at the same time pursue its goal. This requires robust and computationally feasible collision avoidance algorithm to be implemented onboard the UAV. UAVs have limited resources therefore several challenges are needed to overcome for any such algorithm to be implemented:

- The processing speed and memory available on board is limited. Therefore the algorithm must be computationally efficient.
- The algorithm must respond quickly, so that it can be implemented in real time.
- The payload is extremely limited. Therefore the sensors used to gain information from surrounding must be light weight.
- In addition UAV may need to avoid detection in certain secret mission like enemy reconnaissance. Active sensors which send out energy into the environment are not suitable since they can be detected.

Camera based vision sensing as discussed in detail in Chapter 5 is therefore suitable option for sensing because they are light weight, compact and they do not send energy in the environment that could be detected.

UAV collision avoidance approaches can be classified in to global path planning and reactive collision avoidance algorithms (local path planning). Global path planning algorithms assumes a complete prior knowledge of the environment and find an obstacle free trajectory from start point to the goal point. The global path planning methods take into account the global knowledge of the environment and kinematic constraints of the UAV and avoid known obstacles only. In Chapter 2 and 3 this approach has been discussed in detail.

Local or reactive collision avoidance methods do not require the complete knowledge of the entire environment. The information about the local or immediate environment and

nearby popup obstacles is provided by the on board sensors. This information is sufficient to compute an avoidance manoeuvre for UAV. These reactive collision avoidance methods must consist of efficient algorithms that react quickly to changes in the environment; however they may not guarantee optimality. These algorithms can be embedded into any global path planning algorithm under the assumption that after avoiding the obstacle the UAV comes back to the global path as soon as possible. It is essential that the reactive avoidance manoeuvre planning methods are fast since the speed of UAV is limited by them.

Small UAVs are used for low altitude surveillance where unknown obstacles can be encountered. This chapter addresses the problem of avoiding the unexpected popping up obstacle(s) appearing on the path planed for small fixed wings UAV. When the information about the location of an obstacle is known *priori*, the obstacle can be easily avoided while planning the path for the vehicle offline. However if the obstacle is not known in advance and appear unexpectedly, the pre-planned path may become unsafe if no online amendment is made to it. Avoiding the unexpected obstacle, while maintaining close path tracking of UAV, is a challenging problem. This chapter presents an online obstacle avoidance method for small UAVs based on the Pythagorean hodograph curves. The case that the obstacles appear along the pre-planned path unexpectedly is considered. By using camera sensors the position of the obstacle is estimated and timely online adjustments are made to the pre-planned path to avoid the pop up obstacle.

This chapter discuss the problem of reactive obstacle avoidance based on Pythagorean Hodograph path planning. Obstacle avoidance can be achieved by changing the curvature when the path planning is described as a Pythagorean Hodograph curve. However in some cases, obstacle avoidance by changing the curvature of the path results in a tremendous increase in bending energy. An increase in bending energy of a path as a result of curvature change above a certain limit makes the path very difficult or impossible to fly. Therefore some alternative method for obstacle avoidance is needed such that the resultant path is safe and flyable. A solution based on PH curve is proposed for obstacle avoidance which results in safe and flyable paths of minimum bending energy.

In path planning for UAVs the obstacle avoidance is a common problem. In literature different researchers have adopted different ways to solve the problem of obstacle avoidance depending on the context of the problem. A brief literature review of the problem is conducted in the following paragraphs.

Cubic splines and Dubin sets are used for collision avoidance [95-97]. A method is presented for solving path planning problem by bending smooth cubic splines to avoid obstacle collision [95]. The use of cubic splines guarantees continuous vehicle heading and turning angle which are compulsory for the UAVs trajectories. Monte Carlo simulation is used to validate the performance of the algorithm. A smooth path

planning algorithm for UAVs in cluttered natural environment is presented [96]. A Rapid Exploring Random Tree (RRT) is used to produce piece wise collision free linear path amongst the obstacles, and cubic Bezier spiral curves are used to generate a continuous curvature path which satisfies the minimum curvature constraint. The inverse kinematics and Dubins set are used for path planning and obstacle avoidance for the case of steerable needle used in medical applications [97]. In 3D the needle can be interpreted as airplane with constant speed and pitch, zero yaw and controllable roll angle.

Polynomials are used for path planning to avoid the obstacle [98-100]. A planning methodology based on polynomials for non holonomic mobile manipulator in the presence of obstacle is developed [98]. To find collision free path with smaller length, two techniques are used. The first uses the intermediate path points and the second exploits the periodicity of the trigonometric function involved. Polynomials are used for path planning and obstacle on the path is avoided by increasing the order of the polynomial that is used in planning the initial trajectories [99]. An efficient Bezier curve based approach for the path planning in a multi agent robot soccer system is presented [100]. An obstacle avoidance scheme is incorporated for dealing with the stationary and moving obstacle. When the robot is approaching a moving obstacle in the field, it is decelerated and deviated to another Bezier path leading to the estimated target position. The radius of the curvature of the path at its end point is determined from the known terminal velocity constraint of the robot.

Dynamic programming is used to pick the edges of the consecutive hostile Voronoi cells with lowest cost [101]. This results in a coarse path with discontinuous curvature profile. The clothoid or Cornu Spiral curves are used to make the coarse path smooth. This results in a path of continuous curvature profile but not yet flyable. The continuous curvature path is further processed by imposing the minimum curvature constraint to make it flyable.

Fuzzy logic based approach is used for path tracking and obstacle avoidance [102]. The unexpected pop up obstacles either still or moving appearing along the pre-determined flight path is addressed. Suitable sensors are used to calculate the relative distance between the vehicle and obstacle and timely adjustments are made to the pre-planned flight path. Fuzzy logic control algorithms are developed to achieve close path tracking while avoiding obstacle.

A vision based obstacle avoidance technique is presented in [103]. This method uses 2D passive sensors and extended Kalman filter for position estimation of the obstacle. A collision cone approach is used as collision criteria in order to examine if there is any obstacle that is critical to the UAV. A guidance strategy for collision avoidance is designed based on minimum-effort guidance (MEG) method.

The optimization of a chosen trajectory for the UAV within the dynamic constraints of the UAV is used for collision avoidance [104]. Particle swarm

optimization is used for mid flight obstacle avoidance in [105]. The UAVs have limited sensor and communication ranges. When they detect a pop-up or moving threat that is in collision course with the UAV flight path, then it has to re-plan a new optimal path from its current location to the goal. Determining optimal path with short time interval is not feasible, hence anytime algorithm using particle swarm optimization that yield paths whose quality increases with increase in available computational time is developed. To track a given path by anytime algorithm in 3D, a new UAV guidance law that is based on a combination of pursuit guidance law and line of sight guidance is developed.

Non-linear lateral guidance control law and Proportional navigation guidance law is used for collision avoidance in [106, 107]. A non-linear lateral guidance control law with a simple adaptive path planning algorithm is used to model no fly zones to generate a path in real time in order to avoid pop-up obstacle is presented [106]. Proportional navigation guidance law is used for collision avoidance by guiding the relative velocity vector to collision avoidance vector [107]. The optimal navigation co-efficient is obtained if the obstacle moves with constant velocity vector. The convergence of algorithm is obtained by choosing proper navigation coefficients.

Velocity field path planning is used for collision avoidance [108]. This algorithm is the extension of potential field method often used for path planning.

A topological spaces approach is used to describe the obstacle space and free space for collision avoidance purposes [109].

A solution is proposed to avoid the obstacle in the case of Pythagorean Hodograph based path planning for UAVs by manipulating the curvature of the PH path [23, 31]. By increasing the curvature of the path for fixed initial and final poses, an obstacle can be avoided but the path length may increase tremendously as a result. For example Figures 6.1, 6.2 and 6.3 illustrate this fact:

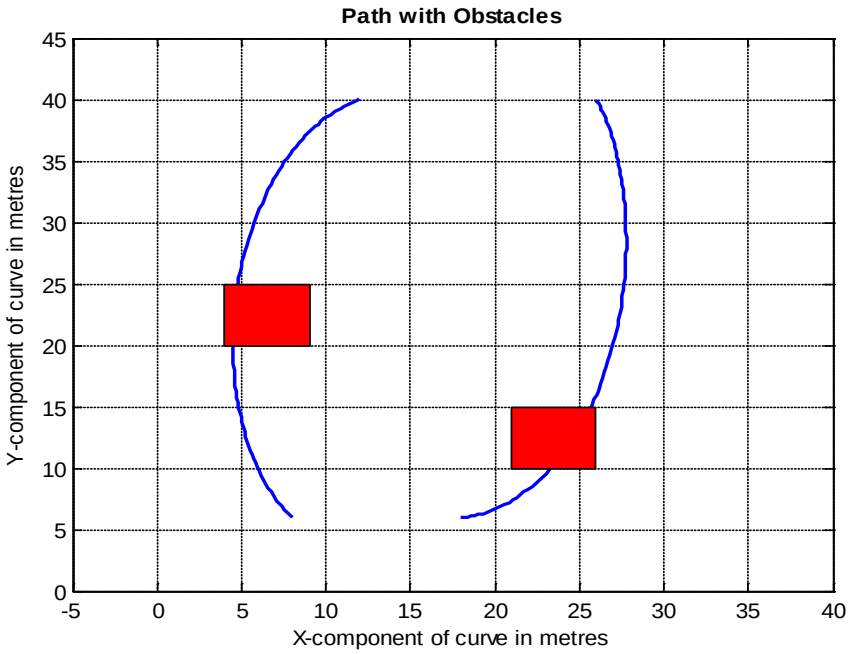

Figure 6.1: Obstacle avoidance by curvature change.

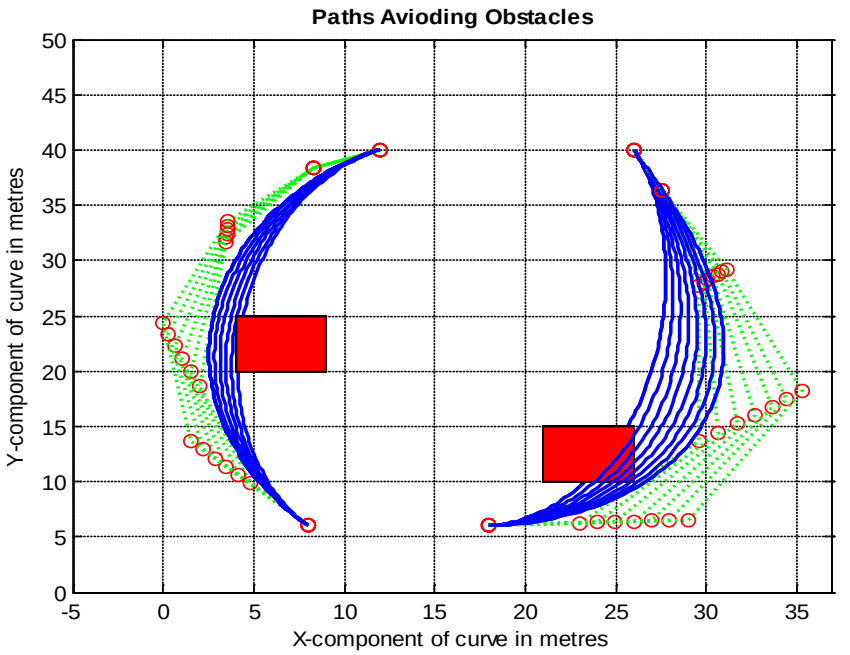

Figure 6.2: Obstacle avoidance by curvature change.

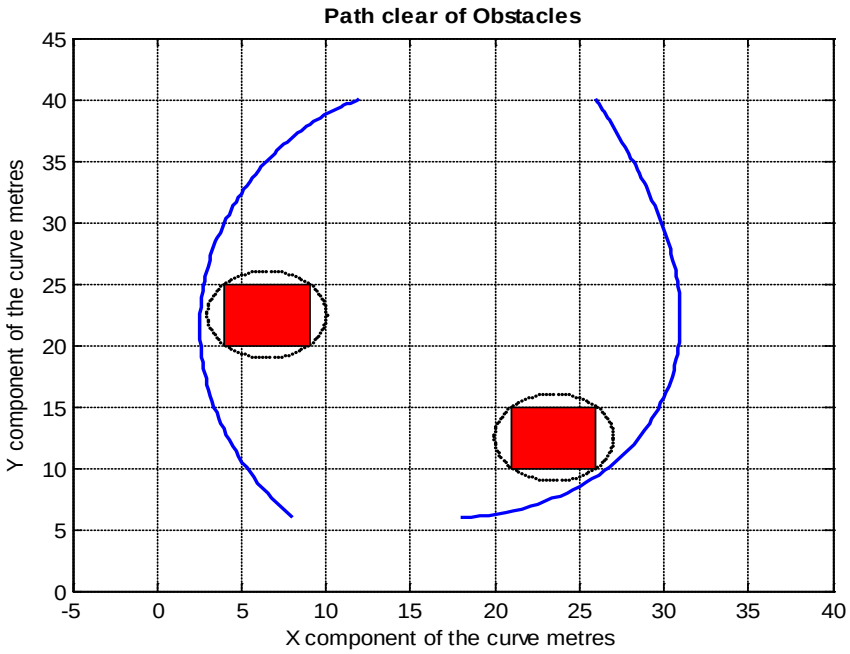

Figure 6.3: Resultant paths obtained by curvature change avoiding obstacle

However sometimes situations may arise, where obstacle avoidance by increasing the curvature of the path for fixed initial and final poses results in an increase in path length as well as bending energy. A path with a very high bending energy is very difficult or impossible for the UAVs to follow and hence not a feasible path. Figure 6.4, 6.5 and 6.6 show this fact.

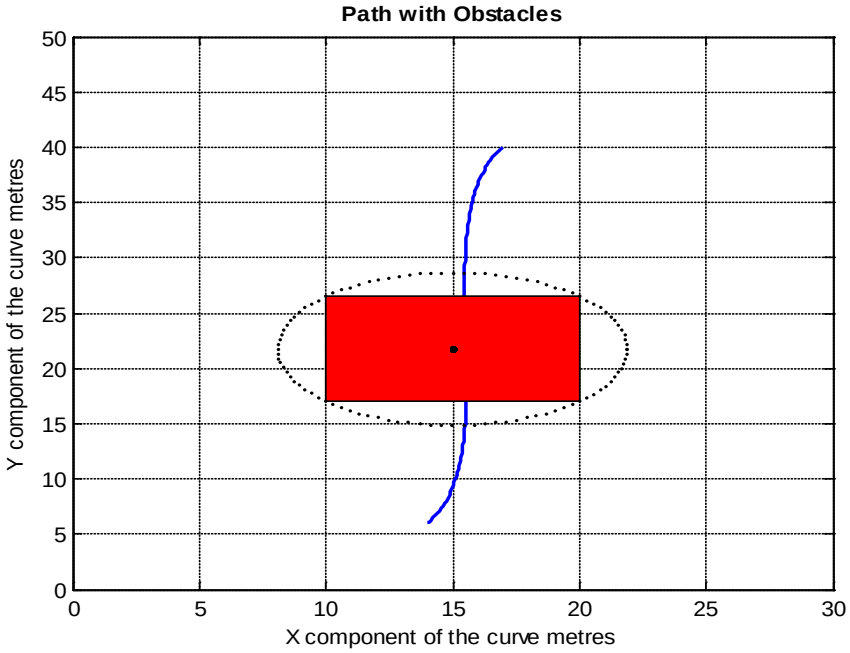

Figure 6.4: Obstacle on a path

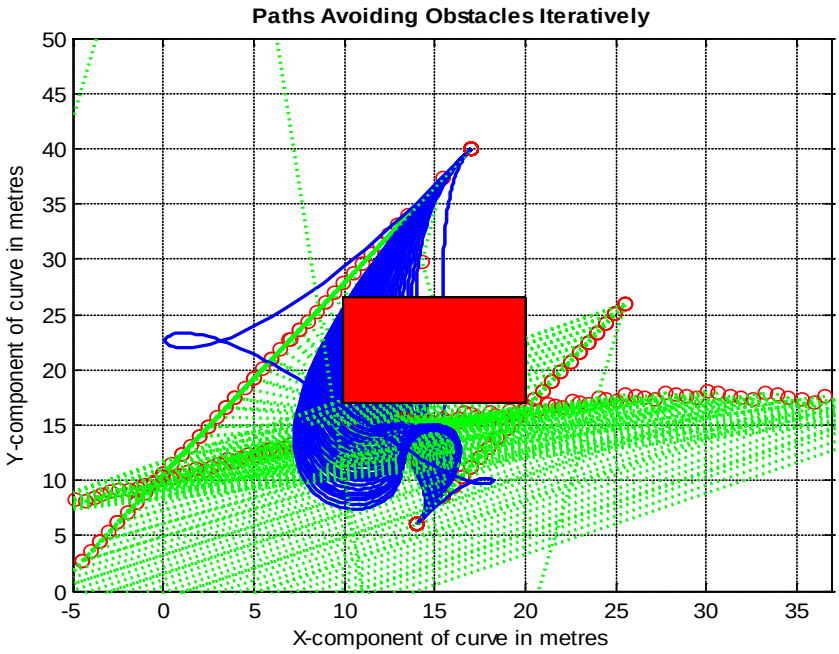

Figure 6.5: Obstacle avoidance by curvature change

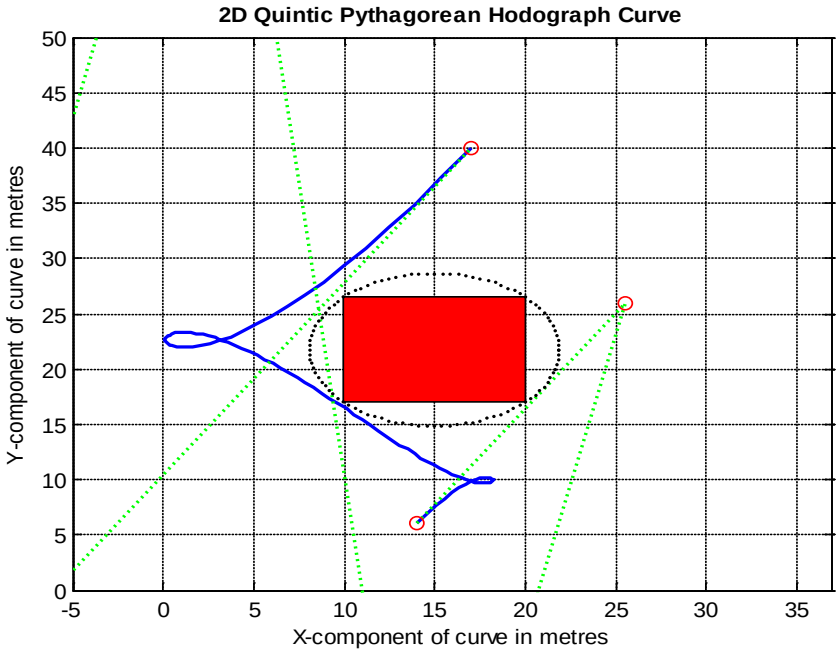

Figure 6.6: Resultant path obtained by curvature change

Since curvature manipulation methods cannot be solely used for obstacle avoidance in PH based path planning because it is bound to fail for some cases. Therefore an alternative method is needed for obstacle avoidance which can guarantee feasible (safe and flyable) paths. To elaborate such a method is the subject of this chapter.

In the proposed method the obstacle is avoided by introducing an intermediate waypoint between the initial and final configurations (poses), and Pythagorean hodographs are

used to connect all the three poses such that the path is clear of the obstacle. The motivation of using the Pythagorean hodograph polynomials for UAV paths, in the proposed method of obstacle avoidance, is because of their computational suitability which makes them a suitable option for path planning in the dynamic environment. Their ability to yield guaranteed curvature and tangent continuous paths make them a favourite option for UAVs path planning. Moreover their ability to yield path of exact length is seen to be very useful aspect to ensure the safety and coordination of the UAVs. Also for PH curves there exists a systematic procedure which guarantees to yields a path with minimum bending energy. The advantage of the minimum bending energy path is that it can be followed by the UAVs very conveniently. Their analytic and predictive nature makes them very alternative among all the candidate methods of UAVs path planning for dynamic environments.

The first contribution of the proposed technique is the rapid generation of flyable paths with optimal lengths (tangent and curvature continuous paths obeying the kinematic constraints of the vehicle). The second contribution of the technique is their ability to avoid an obstacle of any shape and any size. The method has the potential to avoid closely located small obstacles with no safe corridor between them by treating them as a single big obstacle. The third contribution of the proposed technique is to produce paths with minimum bending energies. The minimum bending path within the vehicle kinematic constraints is the easiest traceable path because no hard acceleration (turning rate) demands are put on the vehicle. The forth contribution is time efficiency compared to the curvature manipulation methods. Because of the time efficiency the algorithm is real time implementable. The fifth contribution of the proposed algorithm is that it can be repeated successively until the obstacle is avoided.

6.1 Scenario

A mission is planned to fly a group of unmanned aerial vehicles safely from base B to target T as shown in Figure 6.7. All the vehicles start from the base at the same time. The environment contains stationary obstacles which may be known or unknown. The known obstacles are those which are given in the map (terrain database). The unknown obstacles are those which will be detected and located by the sensor system during the execution of the mission.

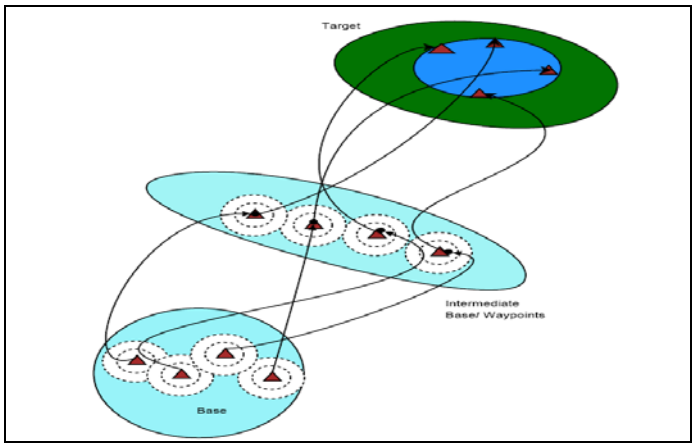

Figure 6.7: Swarm of UAVs with obstacles

Initially, UAVs paths are planned offline by path planning algorithm (PH Path Planner developed in chapter 2 and 3) on the basis of the available knowledge of the environment avoiding known obstacles.

The UAVs start following these paths during mission execution. During their flight, the perception system of the UAVs detects and locates a popup obstacle interrupting the path of any UAV. The proposed algorithm makes online changes to the initial path of the corresponding UAV to avoid the obstacle.

For simplicity the path planning and the obstacles, in the proposed method of obstacle avoidance, are restricted to 2D. Therefore it is assumed in the development of this chapter that the UAVs are flying at constant altitudes.

Before going into the details of the proposed method, a thorough analysis of the PH paths is conducted.

6.2 Analysis of the PH Paths on the Basis of the Orientation of Initial and Final Poses

Since the expression for Pythagorean hodograph curves is derived from the Hermite interpolation of known initial and final poses. Therefore the poses play a pivotal role in generating and modifying these trajectories. Pose is the combination of position and orientation. For given initial and final position in poses, the initial or final orientation in the poses can take any value from 0^0 to 360^0 i.e. any angle from the four quadrants of Cartesian plane. On the basis of the orientations of initial and final poses there are the following four possible groups. Each group contains four combinations of initial and final poses:

1. Paths with the orientation of initial pose fixed in first quadrant and orientation of final pose rotated in quadrants I, II, III and IV.
2. Paths with the orientation of initial pose fixed in second quadrant and orientation of final pose rotated in quadrants I, II, III and IV.

3. Paths with the orientation of initial pose fixed in third quadrant and orientation of final pose rotated in quadrants I, II, III and IV.
4. Paths with the orientation of initial pose fixed in fourth quadrant and orientation of final pose rotated in quadrants I, II, III and IV.

6.2.1 Paths with the Orientation of Initial Pose Fixed in First Quadrant and Orientation of the Final Pose Rotated in Quadrant I, II, III and IV.

In this case the initial pose with arbitrary position has assigned an orientation from first quadrant ($0^0 \prec \theta_{si} \prec 90^0$) and final pose has been assigned orientation from one of the four quadrants (first, second, third, fourth) turn by turn. i.e.

$(0^0 \prec \theta_{fi} \prec 90^0)$ or $(90^0 \prec \theta_{fi} \prec 180^0)$ or $(180^0 \prec \theta_{fi} \prec 270^0)$ or $(270^0 \prec \theta_{fi} \prec 360^0)$ (6.1)

Thus there are the following four possible combinations of initial and final poses giving four paths.

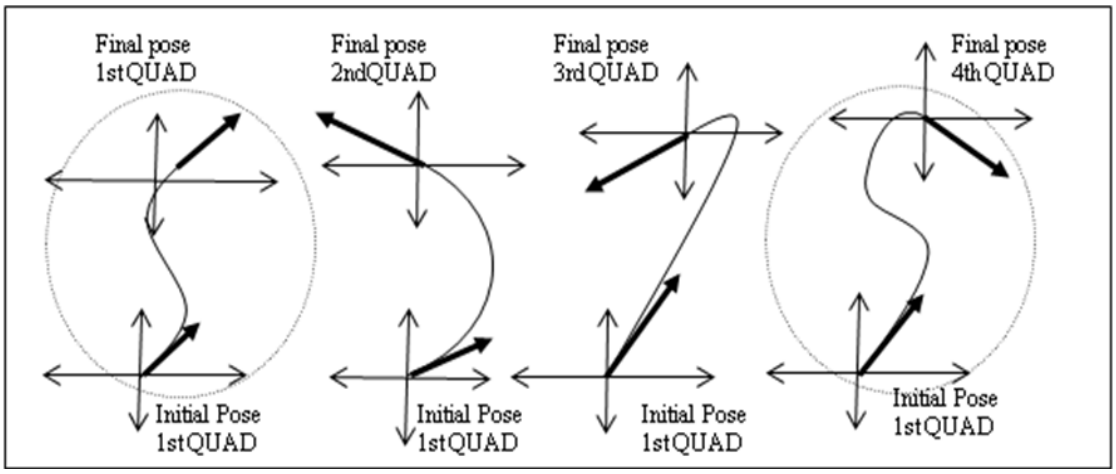

Figure 6.8: Paths resulting from four combinations of the initial and final poses where the initial pose is fixed in first quadrant and final pose is rotated in quadrant I, II, III, IV.

6.2.2 Paths with the Orientation of Initial Pose Fixed in Second Quadrant and Orientation of the Final Pose Rotated in Quadrant I, II, III and IV.

In this case the orientation of the initial pose is taken from second quadrant ($90^0 \prec \theta_{si} \prec 180^0$) and final pose can have orientation taken from one of the four quadrants (first, second, third, fourth) turn by turn, as expressed by expression (6.1).

Thus there are the following four possible combinations of initial and final poses giving four paths.

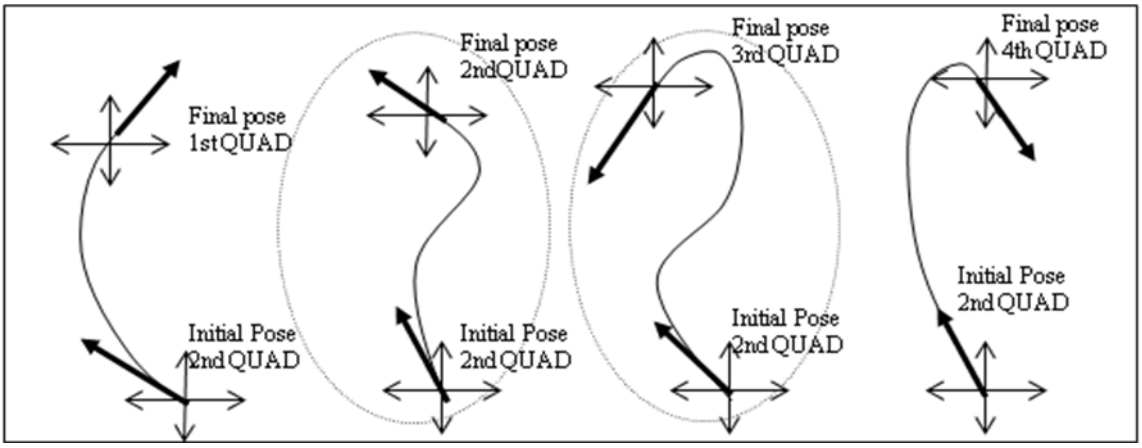

Figure 6.9: Paths resulting from four combinations of the initial and final poses where the initial pose is fixed in second quadrant and final pose is rotated in quadrant I, II, III, IV.

6.2.3 Paths with the Orientation of Initial Pose Fixed in Third Quadrant and Orientation of the Final Pose Rotated in Quadrant I, II, III and IV.

In this case the orientation of the initial pose is taken from third quadrant ($180^0 \prec \theta_{si} \prec 270^0$) and final pose can have orientation assigned from one of the four quadrants (first, second, third, fourth) One by one, as expressed by expression (6.1).

Thus there are the following four possible combinations of initial and final poses giving four paths.

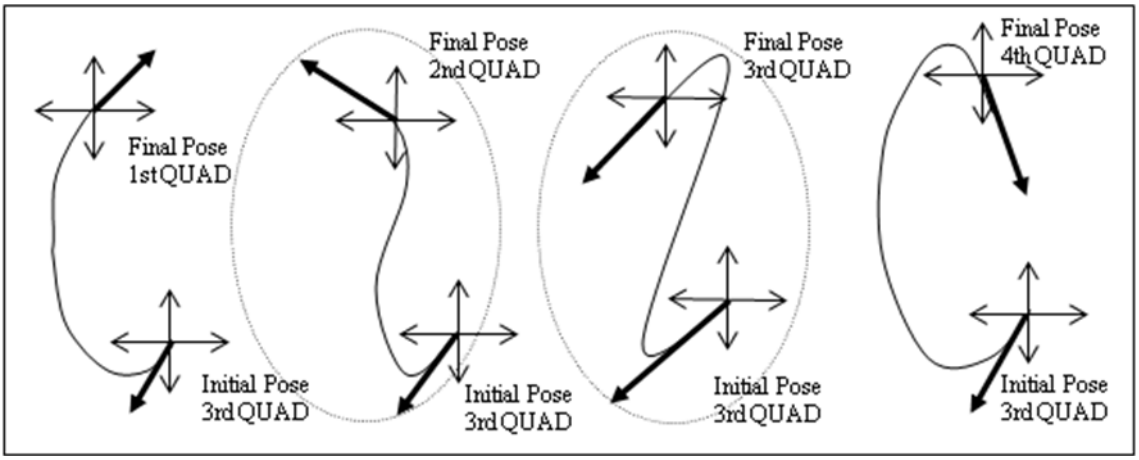

Figure 6.10: Paths resulting from four combinations of the initial and final poses where the initial pose is fixed in Third quadrant and final pose is rotated in quadrant I, II, III, IV.

6.2.4 Paths with the Orientation of Initial Pose Fixed in Fourth Quadrant and Orientation of the Final Pose Rotated in Quadrant I, II, III and IV.

In this case the orientation of the initial pose is taken from fourth quadrant ($270^0 \prec \theta_{si} \prec 360^0$) and final pose have orientation taken from one of the four quadrants (first, second, third, fourth) turn by turn, as expressed by expression (6.1).

Thus there are the following four possible combinations of initial and final poses giving four paths.

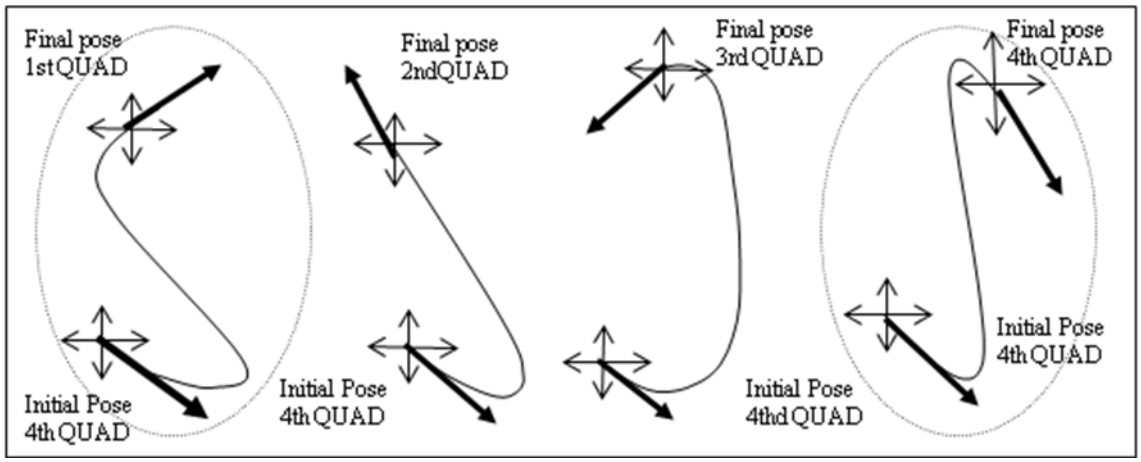

Figure 6.11: Paths resulting from four combinations of the initial and final poses where the initial pose is fixed in fourth quadrant and final pose is rotated in quadrant I, II, III, IV.

Thus, there are sixteen possible combinations of orientations of initial and final poses taken from four quadrants of Cartesian plane. These sixteen combinations of initial and final poses yield sixteen PH paths shown in figures 6.8, 6.9, 6.10, and 6.11. For the purpose of midflight obstacle avoidance, these sixteen paths can be classified into two classes as discussed in the following section.

6.3 Classification of the Pythagorean Hodograph Paths

The paths yielded by all possible combinations of orientations of initial and final poses are shown in figures 6.8, 6.9, 6.10, and 6.11. From these figures it can be seen that there are two types of paths:

- ‘C’ Shaped Paths
- ‘S’ Shaped Paths.

6.3.1 ‘C’ Shape Paths

In ‘C’ shape paths the normal vector (the vector in the direction of centripetal acceleration and perpendicular to the direction of motion) of the path does not undergo a

reversal of direction. These paths changes in lengths only when their curvature is manipulated. The paths which are not encircled in the above figures 6.8, 6.9, 6.10, and 6.11 belong to the class of ‘C’ Shape Paths. ‘C’ Shaped Paths are considered as safe paths, because an obstacle can be avoided just by curvature manipulation no matter which ever point the obstacle interrupts on these paths. This is shown in figure 6.1 and figure 6.2.

6.3.2 ‘S’ Shape Paths

In ‘S’ shape paths the normal vector of the path undergoes a reversal of the direction. These paths changes in lengths as well as in bending energy when their curvature is manipulated. The paths which are encircled in the above figures 6.8, 6.9, 6.10, and 6.11 belong to the class of ‘S’ Shape Paths.

To understand the reversal of the direction of normal vector, refer to the figure 6.12, the normal vector of the path reverses its direction at point ‘P’. Before point ‘P’, the direction of the normal vector is the opposite of its direction after the point ‘P’. Thus the normal vector of the path undergoes a reversal of the direction.

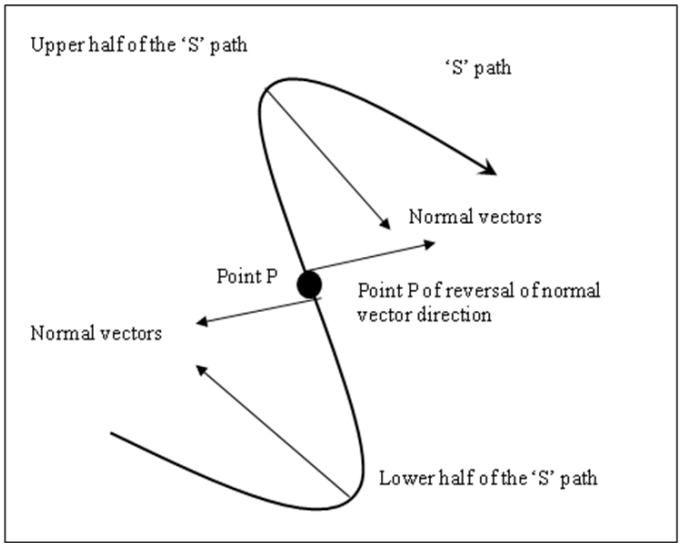

Figure 6.12: Path where the normal vector reverse its direction at point ‘P’

‘S’ Shaped Paths may be safe or unsafe depending on the location of the interrupting obstacle on the path. If the obstacle is interrupting before point ‘P’ i.e. in the lower half of the path, then for given initial and final poses of the path, it can be avoided by increasing the absolute curvature of the lower segment of the path as shown in the figure 6.13.

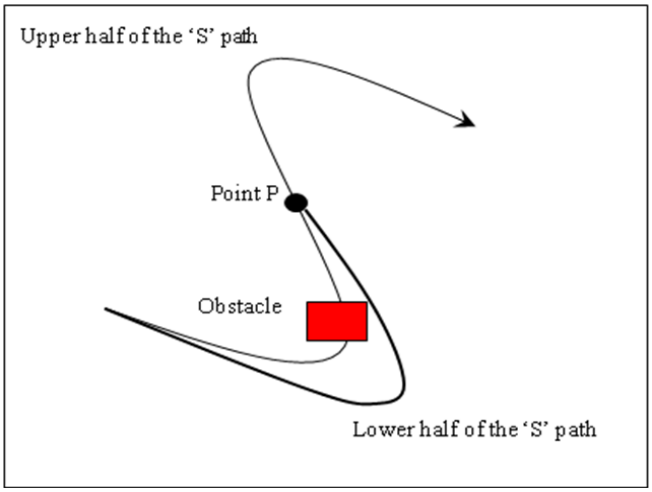

Figure 6.13: Obstacle interrupting before point 'P'

If the obstacle is interrupting after the point 'P', i.e. the upper half of the path, then for given initial and final poses, it can be avoided by increasing the absolute curvature of the upper segment of the path as shown in the figure 6.14.

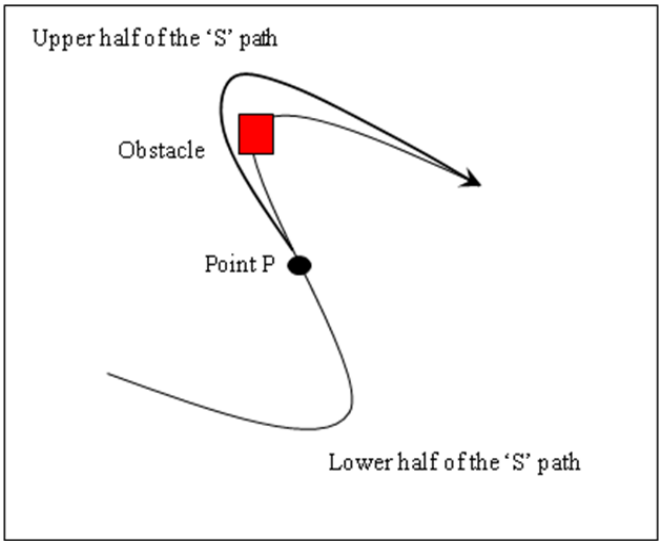

Figure 6.14: Obstacle interrupting after point 'P'

However, if the obstacle is located in the closed neighbourhood of the point 'P', where curvature is zero, then it cannot be avoided by curvature manipulation of the path for the given initial and final poses as shown in figure 6.15.

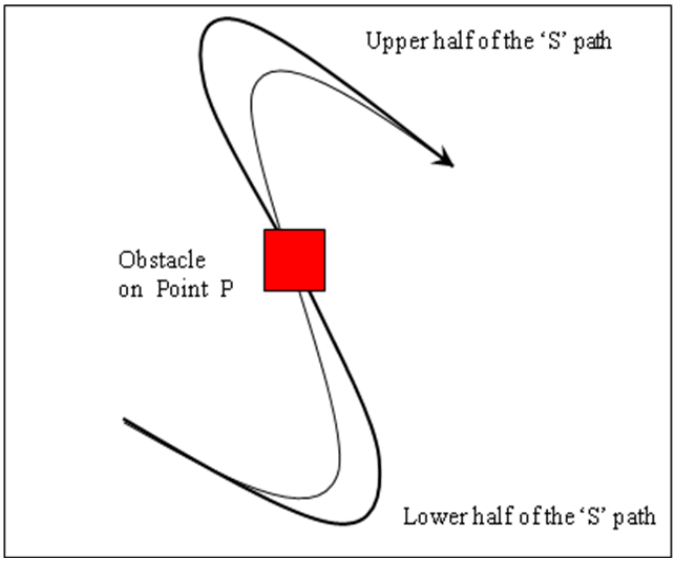

Figure 6.15: Obstacle interrupting the path at or near point 'P'

The proposed method addresses this situation where manipulating the curvature fails to avoid the obstacle or results in a path of unacceptably high bending energy as shown in figure 6.5.

6.4 Proposed Obstacle Avoidance Technique

The proposed technique yields feasible path (safe and flyable path) of minimum bending energy to avoid the obstacle. It consists of the following three steps:

1. Description of the detected obstacle.
2. Insertion of a new waypoint (intermediate pose) between the initial and final pose to avoid the obstacle.
3. Smooth connection of the three poses by two 'C' shape Pythagorean hodographs curves of minimum bending energy.

The whole process is shown in figure 6.16. It should be noted that the pose of the UAV from which the obstacle is located during its flight is taken as the initial pose for the evasive path. The final pose is the same final pose which was used to calculate the initial trajectory of the UAV offline. It cannot be changed because it is dictated by the mission.

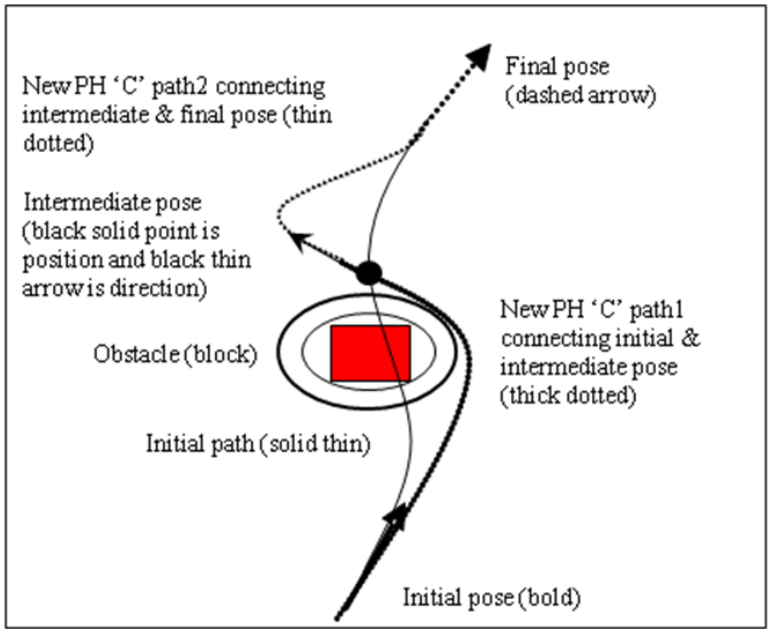

Figure 6.16: An unsafe initial path and a new amended path resulted from the proposed technique

6.4.1 Description of Detected Obstacle

This research assumes that the obstacle is detected and located midflight by the onboard camera sensors. The detected obstacle may be of any regular or irregular shape; therefore an enclosing circle/ellipse is the best way to describe the obstacle mathematically. The depth of the obstacle cannot be determined from a single UAV. Therefore the depth of the obstacle is either determined cooperatively by two neighbouring UAVs from different viewpoints [55], or it is revealed progressively as the UAV manoeuvres to different viewpoints during its flight on the amended path. If the cooperation is not possible for some reasons then the depth is calculated as the UAV proceed along its amended path.

Let the extreme visible points of the obstacle are (x_1, y_1) and (x_2, y_2) as viewed by the on board cameras in the first instant. For the time being the obstacle is assumed to be of square shape with vertices $(x_1, y_1), (x_2, y_2)$, (x_3, y_3) and (x_4, y_4) as shown in the figure 6.17. (x_1, y_1) and (x_2, y_2) are known from the image while (x_3, y_3) and (x_4, y_4) can be calculated as follows:

$$x_3 = x_2 \tag{6.2}$$

For square obstacle:

$$y_3 = y_2 + (x_2 - x_1) \tag{6.3}$$

And $$x_4 = x_1 \tag{6.4}$$

Similarly $$y_4 = y_1 + (x_2 - x_1) \tag{6.5}$$

To find out the centre of the assumed square obstacle (x_{cen}, y_{cen}) refer to the figure 6.17.

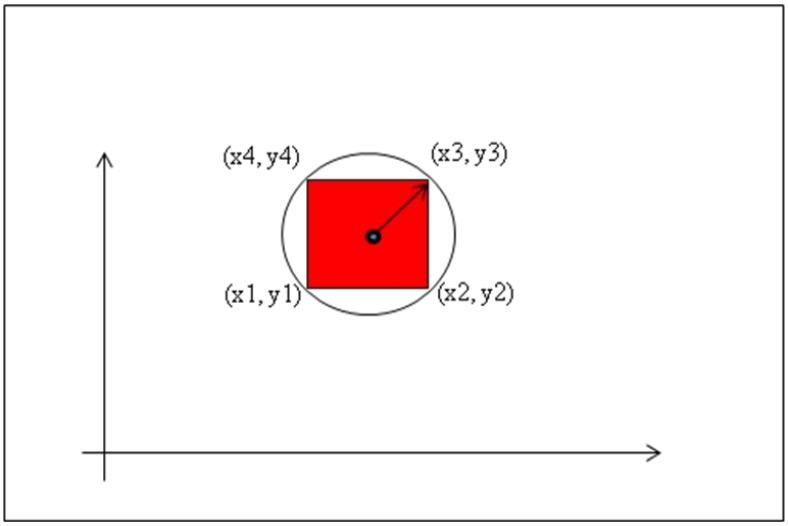

Figure 6.17: Obstacle description by an enclosing circle

$$x_{cen} = x_1 + 0.5(x_2 - x_1) \tag{6.6}$$

$$y_{cen} = y_1 + 0.5(y_4 - y_1) \tag{6.7}$$

The obstacle is enclosed in a circle to describe no fly area. The centre of the obstacle (x_{cen}, y_{cen}) is taken as the centre of the enclosing circle and the radius of the enclosing circle is:

$$r_d = \frac{\sqrt{(x_2 - x_1)^2 + (y_4 - y_1)^2}}{2} \tag{6.8}$$

Once (x_{cen}, y_{cen}) and r_d are known, then the coordinates of the points of the enclosing circle (x, y) are given by the following parametric equations:

$$x = x_{cen} + r_d \cos\theta \tag{6.9}$$

$$y = y_{cen} + r_d \sin\theta \tag{6.10}$$

Where $\theta \in [0 \;\; 2\pi]$.

In figure 6.16 the outer circle is to keep the UAV at minimum safety distance from the obstacle. If the radius of the inner circle is r_d and the minimum safety distance is R then the radius of the outer circle R_d is

$$R_d = r_d + R \tag{6.11}$$

And the centre of the outer circle is the same as the inner circle because both the circles are concentric as shown in figure 6.16. The coordinates (X,Y) of the points of the outer circle enclosing the obstacle are given by the following parametric equations:

$$X = x_{cen} + R_d \cos\theta \tag{6.12}$$

$$Y = y_{cen} + R_d \sin\theta \tag{6.13}$$

Where $\theta \in [0 \;\; 2\pi]$. The obstacle is avoided if the equation for the PH path (2.58) and equations (6.12), (6.13) do not intersect.

The reality that whether the assumed square has covered the entire depth of the obstacle will be revealed when the UAV will start journey on the avoidance path of figure 6.13 calculated by the proposed method. During this process, the updated views of the obstacle are taken from different perspectives. From these updated views, if the assumed square based on initial perspective is proved to covers the entire obstacle as in figure 6.16, then the avoidance path of figure 6.16 is the final safe path. If on the other hand, the assumed square based on initial perspective, misses the correct depth, and the new depth based on current perspective is not covered by the initially assumed square, then the whole process is repeated until the obstacle is avoided as shown in figure 6.18.

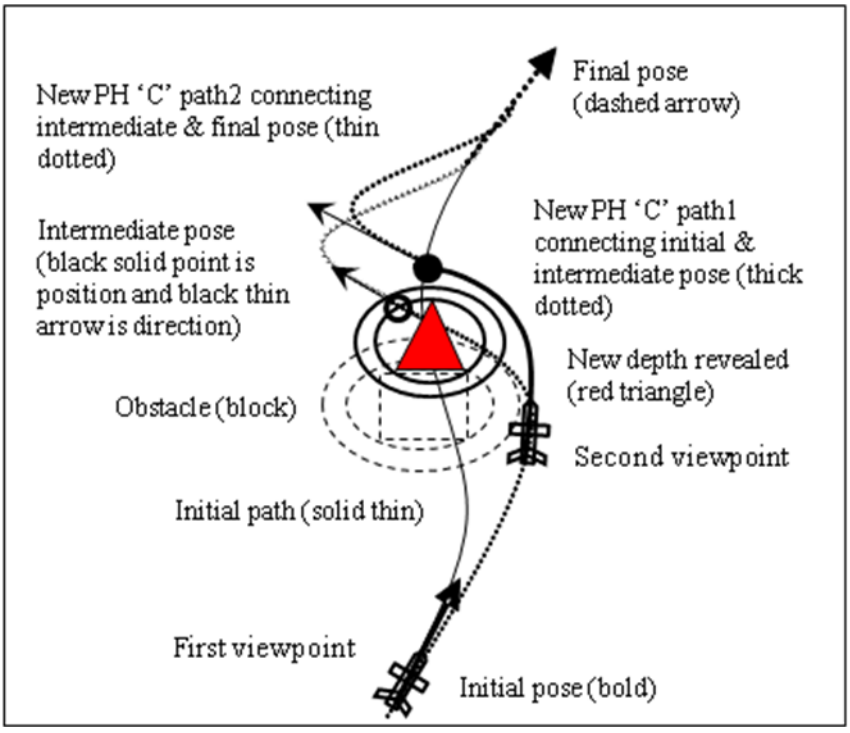

Figure 6.18: New depth of the obstacle is revealed as the UAV camera takes an updated view from new position on the avoidance path and the avoidance path is amended accordingly

6.4.2 New Waypoint (Intermediate Pose) Insertion between the Initial and Final Pose to Avoid the Obstacle

After describing the obstacle by an enclosing circle the next stage is the insertion of intermediate pose (way point) such that the amended path passing through this waypoint is clear of the obstacle. Since pose is a combination of position and direction, therefore the position and direction of the inserted intermediate pose must be determined. The following paragraphs describe the determination of the position and direction of the intermediate waypoint.

6.4.2.1 Determination of the Location of the Intermediate Pose

In this section a suitable location would be determined for the insertion of the intermediate pose in order to avoid the obstacle. The location determination of the intermediate pose is based on the following theorem.

Theorem:

The point dividing the line –segment with end-points $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ is:

$$\left(\frac{k_1 x_2 + k_2 x_1}{k_1 + k_2}, \frac{k_1 y_2 + k_2 y_1}{k_1 + k_2} \right)$$

(k_1, k_2 being positive integers)

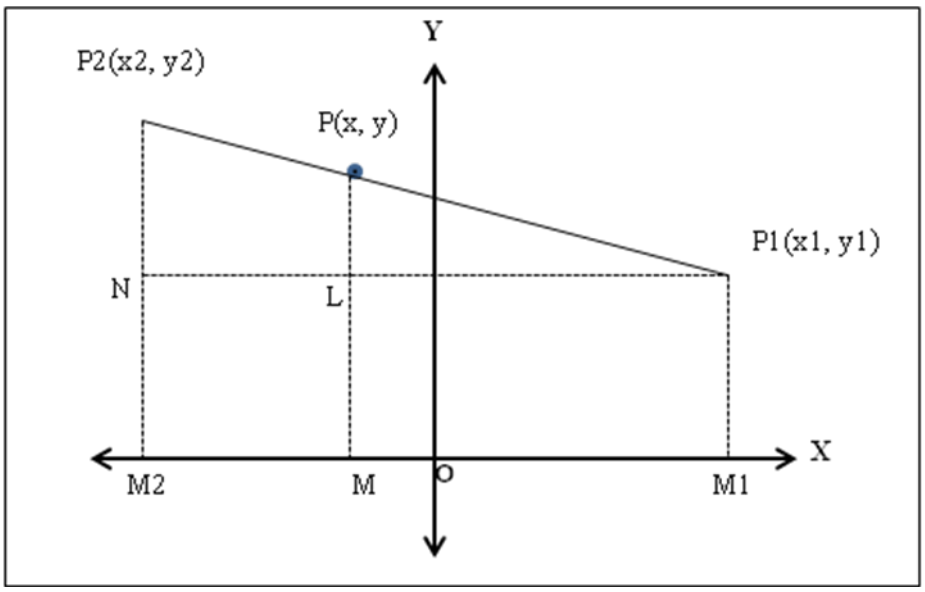

Figure 6.19: Point dividing the line segment in ratio k_1:k_2

Proof: Referring to the figure 6.19, let $P(x, y)$ be the point which divides the line segment $\overline{P_1P_2}$. $\overline{P_1M_1}$ and $\overline{P_2M_2}$ be $\perp^s$ from P_1 and P_2 respectively on to the x-axis. Let $\overline{P_1N}$ be $\perp$ to $\overline{M_2P_2}$, and L be the Point of intersection of $\overline{P_1N}$ and $\overline{PM}$. Then:

$$\frac{k_1}{k_2} = \frac{\overline{P_1P}}{\overline{PP_2}} \tag{6.14}$$

Because of similarity of ΔP_1PL and ΔP_1P_2N , we can write:

$$= \frac{\overline{P_1L}}{\overline{LN}}$$

$$\frac{k_1}{k_2} = \frac{\overline{M_1M}}{\overline{MM_2}}$$

$$\frac{k_1}{k_2} = \frac{x - x_1}{x_2 - x} \tag{6.15}$$

$$\Rightarrow k_1x_2 - k_1x = k_2x - k_2x_1$$

$$\Rightarrow (k_1 + k_2)x = k_1x_2 + k_2x_1$$

$$\Rightarrow x = \frac{k_1x_2 + k_2x_1}{k_1 + k_2} \tag{6.16}$$

Similarly, drawing perpendiculars from P_1, P_2 on to the y-axis we can prove that:

$$y = \frac{k_1y_2 + k_2y_1}{k_1 + k_2} \tag{6.17}$$

The above theorem is used for the determination of location (x_{iwp}, y_{iwp}) of the intermediate waypoint. Refer to figure 6.20, let the current position of the UAV be (x_{uav}, y_{uav}) given by onboard position sensors (GPS & INS), the centre of the obstacle is (x_{cen}, y_{cen}) determined by equations (6.6) and (6.7), then the distance d of the UAV from the obstacle is:

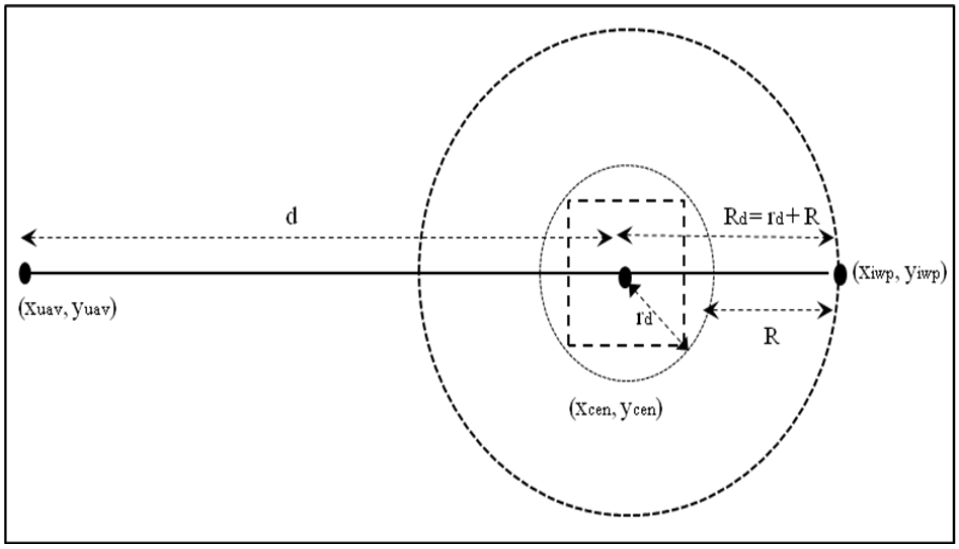

Figure 6.20: Position of the UAV (x_{uav}, y_{uav}), position of centre of the obstacle (x_{cen}, y_{cen}) and position of the newly inserted way point (x_{iwp}, y_{iwp})

$$d = \sqrt{(x_{cen} - x_{uav})^2 + (y_{cen} - y_{uav})^2} \tag{6.18}$$

And the radius of the outer circle is known which is:

$$R_d = r_d + R \tag{6.19}$$

Now (x_{cen}, y_{cen}) is the point which divides the line segment with starting point (x_{uav}, y_{uav}) and finish point (x_{iwp}, y_{iwp}) internally in the ratio $k_1 : k_2 = d : R_d$ then by equations (6.16) and (6.17) of above theorem it follows:

$$x_{cen} = \frac{k_1 x_{iwp} + k_2 x_{uav}}{k_1 + k_2} \tag{6.20}$$

$$y_{cen} = \frac{k_1 y_{iwp} + k_2 y_{uav}}{k_1 + k_2} \tag{6.21}$$

$$\Rightarrow x_{iwp} = \frac{(k_1 + k_2)x_{cen} - k_2 x_{uav}}{k_1}$$

$$\Rightarrow x_{iwp} = \frac{(d + R_d)x_{cen} - R_d x_{uav}}{d} \tag{6.22}$$

$$\Rightarrow y_{iwp} = \frac{(k_1 + k_2)y_{cen} - k_2 y_{uav}}{k_1}$$

$$\Rightarrow y_{iwp} = \frac{(d + R_d)y_{cen} - R_d y_{uav}}{d} \tag{6.23}$$

Equation (6.22) and (6.23) give the coordinates of the location of the new waypoint (pose).

6.4.2.2 Determination of the Direction of the Intermediate Pose

After finding the location for insertion of the intermediate pose, the next issue is the orientation that the UAV must have at this location in order to avoid the obstacle. The orientation of the UAV θ_{iwp} at the intermediate waypoint should be such that the two consecutive poses could be connected smoothly by PH quintic curves of minimum bending energy, each belonging to the class of 'C' shape path. In the following subsections the orientation or direction determination for the intermediate pose is considered in cases of all 'S' shape paths shown in figures 6.8, 6.9, 6.10 and 6.11.

6.4.2.2.1 Initial Poses in First Quadrant and Final Poses from Any Quadrant

In this case there are four possible paths as mentioned before. Two of the four possibilities are 'S' shape paths as shown in figure 6.8. The proposed method is applied to 'S' shape paths only.

In figure 6.8 the directions of initial poses of both 'S' shape paths belong to quadrant 1, and the direction of final poses of one of the paths belongs to quadrant 1 and the direction of final pose of the second path belongs to quadrant 4. The direction of the newly inserted pose is chosen in such a way that the three poses can be smoothly connected by two PH quintic paths of 'C' shape. This can be done if we choose the direction to be an angle belonging to the range:

$$\theta_{iwp} \in (\frac{\pi}{2}, \frac{3\pi}{2})$$

From figure 6.21, it is apparent that in the case of 'S' shape paths, the orientations of UAVs initial and final poses belong to the right half of the Euclidean plane. The orientation of the intermediate pose is selected from the left half of the Euclidean plane. Thus the left half of the Euclidean plane is the range of angle from which the orientation of intermediate pose is chosen.

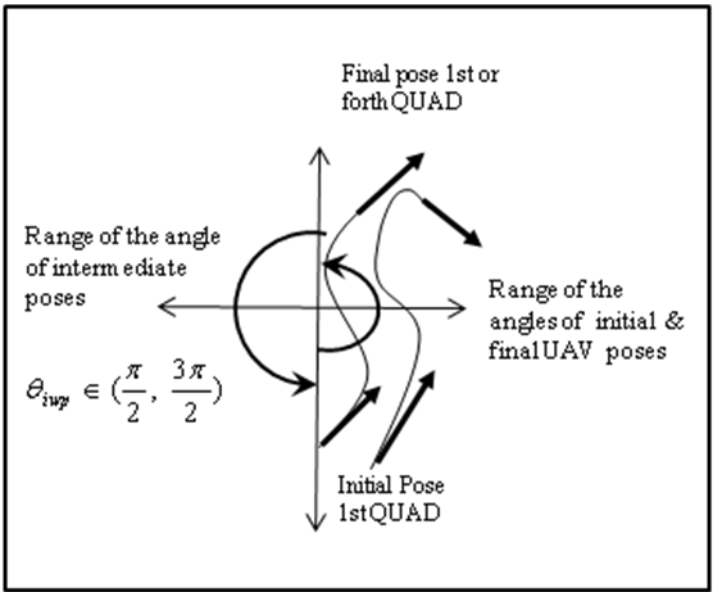

Figure 6.21: The left half of the Euclidian plane is the range from where the orientation of the intermediate pose is selected

From the range $(\frac{\pi}{2}, \frac{3\pi}{2})$ a suitable angle is selected iteratively such that the initial & intermediate, and the intermediate & final poses give two minimum energy PH quintic curves of 'C' shape. Each curve is clear of the obstacle and mutually joins smoothly.

6.4.2.2.2 Initial Poses in Second Quadrant and Final Poses from Any Quadrant

In this case there are four possible paths as mentioned before. Two of the four possibilities are unsafe as shown in figure 6.9.

In figure 6.9 the directions of initial poses of both 'S' shape paths belong to quadrant 2, and the direction of final poses of one path belongs to quadrant 2 and the direction of final pose of the second path belongs to quadrant 3. The direction of the newly inserted pose is chosen in such a way that the three pose can be smoothly connected by two PH quintic path of 'C' shape. This can be done if the direction is chosen to be an angle belonging to the range:

$$\theta_{iwp} \in (-\frac{3\pi}{2}, \frac{\pi}{2})$$

From figure 6.22, it is apparent that in the case of 'S' shape paths, the orientations of UAVs initial and final poses belong to the left half of the Euclidean plane. The orientation of the intermediate pose is selected from the right half of the Euclidean plane. Thus the right half of the Euclidean plane is the range of the orientation of intermediate pose.

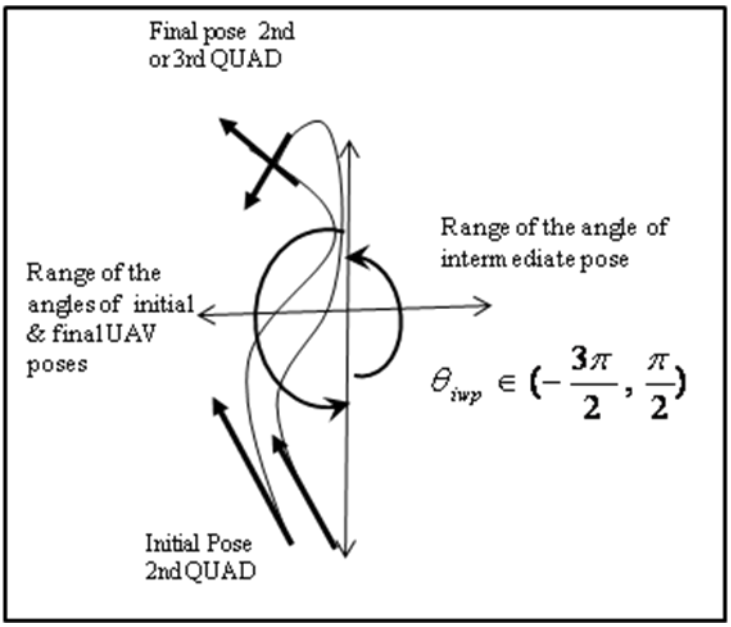

Figure 6.22: The right half of the Euclidian plane is the range from where the orientation of the intermediate pose is selected

From the range $(-\frac{3\pi}{2}, \frac{\pi}{2})$ a suitable angle is selected iteratively such that the initial & intermediate, and the intermediate & final poses give two minimum energy PH quintic curves of 'C' shape. Each curve is clear of the obstacle and mutually joins smoothly.

6.4.2.2.3 Initial Poses in Third Quadrant and Final Poses from Any Quadrant

In this case there are four possible paths as mentioned before. Two of the four possibilities are unsafe as shown in figure 6.10.

In figure 6.10 the directions of initial poses of both 'S' shape paths belong to quadrant 3, and the direction of final poses of one path belongs to quadrant 2 and the direction of final pose of the second path belongs to quadrant 3. The direction of the newly inserted pose is chosen in such a way that the three pose can be smoothly connected by two PH quintic path of 'C' shape. This can be done if the orientation of intermediate pose is chosen to be an angle belonging to the range:

$$\theta_{iwp} \in (-\frac{3\pi}{2}, \frac{\pi}{2})$$

From figure 6.23, it is apparent that in the case of 'S' shape paths, the orientations of UAVs initial and final poses belong to the left half of the Euclidean plane. The orientation of the intermediate pose is selected from the right half of the Euclidean plane. Thus the right half of the Euclidean plane is the range of the orientation of intermediate pose.

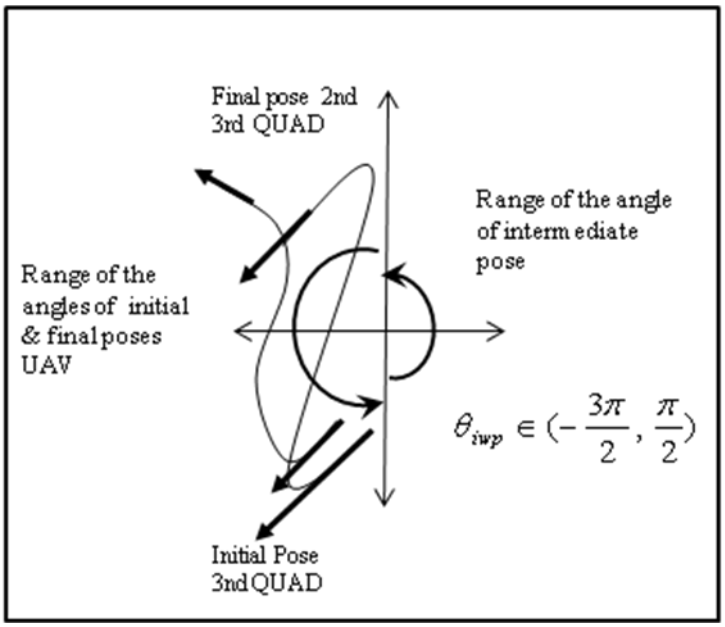

Figure 6.23: The right half of the Euclidian plane is the range from where the orientation of the intermediate pose is selected

From the range $(-\frac{3\pi}{2}, \frac{\pi}{2})$ a suitable angle is selected iteratively such that the initial & intermediate, and the intermediate & final poses give two minimum energy PH quintic curves of 'C' shape. Each curve is clear of the obstacle and mutually joins smoothly.

6.4.2.2.4 Initial Poses in Fourth Quadrant and Final Poses from Any Quadrant

In this case there are four possible paths as mentioned before. Two of the four possibilities are unsafe as shown in figure 6.11.

In figure 6.11 the directions of initial poses of both 'S' shape paths belong to quadrant 4, and the direction of final poses of one path belongs to quadrant 1 and the direction of final pose of the second path belongs to quadrant 4. The direction of the newly inserted pose is chosen in such a way that the three pose can be smoothly connected by two PH quintic path of 'C' shape. This can be done if the orientation of intermediate pose is chosen to be an angle belonging to the range:

$$\theta_{iwp} \in (\frac{\pi}{2}, \frac{3\pi}{2})$$

From figure 6.24, it is apparent that in the case of 'S' shape paths, the orientations of UAVs initial and final poses belong to the right half of the Euclidean plane. The orientation of the intermediate pose is selected from the left half of the Euclidean plane. Thus the left half of the Euclidean plane is the range of the orientation of intermediate pose.

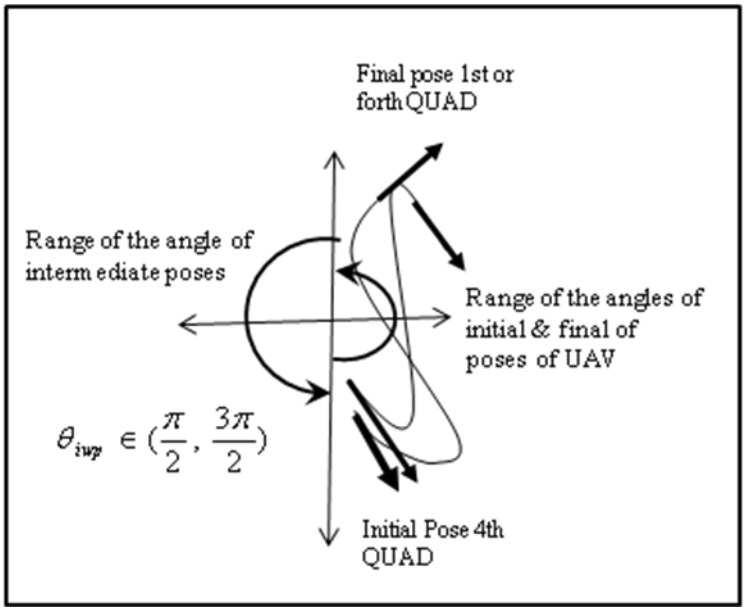

Figure 6.24: The left half of the Euclidian plane is the range from where the orientation of the intermediate pose is selected

From the range $(\frac{\pi}{2}, \frac{3\pi}{2})$ a suitable angle is selected iteratively such that the initial & intermediate, and the intermediate & final poses give two minimum energy PH quintic curves of 'C' shape. Each curve is clear of the obstacle and mutually joins smoothly.

6.4.3 Smooth Connection of the Three Poses by Two 'C' Shape Pythagorean Hodographs Curves of Minimum Bending Energy

After the determination of the position and direction of the intermediate pose, the three poses are finally connected by two Pythagorean 'C' shape curve of minimum bending energy. For this purpose the path planner of chapter 2 is used to produce two Pythagorean curves of minimum bending energy and of 'C' shape. One curve connects the initial and intermediate pose. The other curve connects the intermediate and final pose. The two curves join smoothly. For the general case of initial and final poses of the UAV, this whole process is shown in figure 6.25. The PH curve which safely connects the initial and intermediate pose is shown as dashed curve. The PH curve which connects the intermediate and final pose is shown as dashed-dotted curve. The original unsafe 'S' shape path is shown as solid curve.

For obstacle free path the set of points of intersection of UAV path and the circle enclosing the obstacle must be empty. However if the set of points of intersection is not empty then the absolute curvature of the new path is increased iteratively till the intersection set is empty. i.e. the path is clear of the obstacle. This whole process is shown in figure 6.25. The dotted paths show the iterative process of obstacle clearance by the curvature increase for the conflicting 'C' shape path.

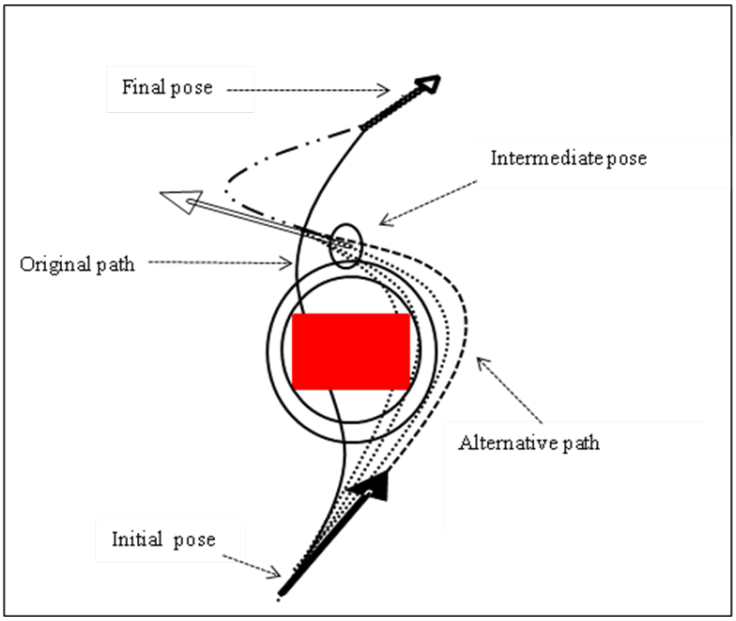

Figure 6.25: Obstacle clearance by gradual increase of curvature.

Mathematically the curvature manipulation is achieved by manipulating the control points P_1, P_2, P_3 and P_4 of the dashed curve iteratively using the technique discussed in chapter 2, such that the path is clear of the obstacle. The first and the last control points P_o, P_5 are kept unchanged. The control points are changed by increasing the values of constants k_1, k_2 iteratively in the equations (2.90) and (2.92) as mentioned in chapter 2.

Thus the final feasible paths resulted from the smooth connection of the three poses by two 'C' shape PH quintic paths for each case of the section 6.6.2.2 using the above connection scheme are shown in figures 6.26, 6.27, 6.28, and 6.29.

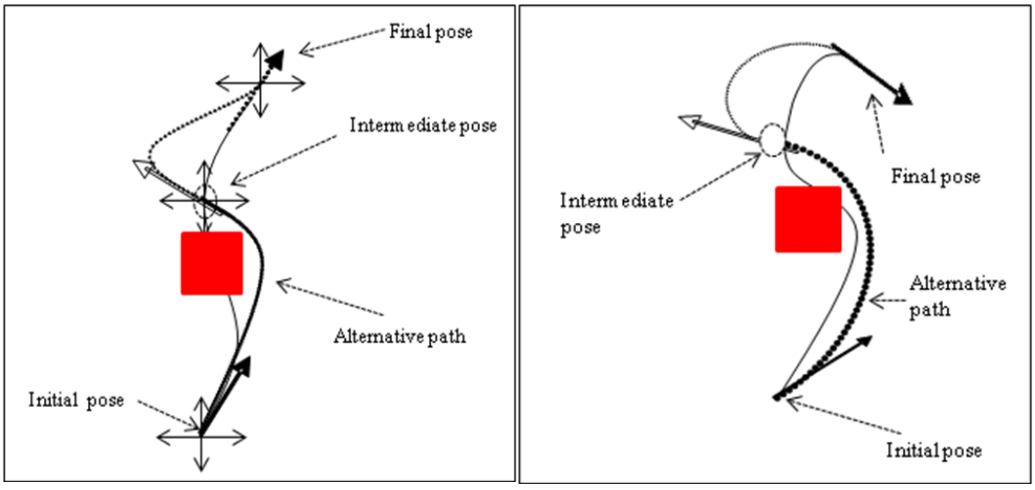

Figure 6.26: The original unsafe 'S' shape paths (initial pose first quadrant, final pose any quadrant) and the amended safe paths

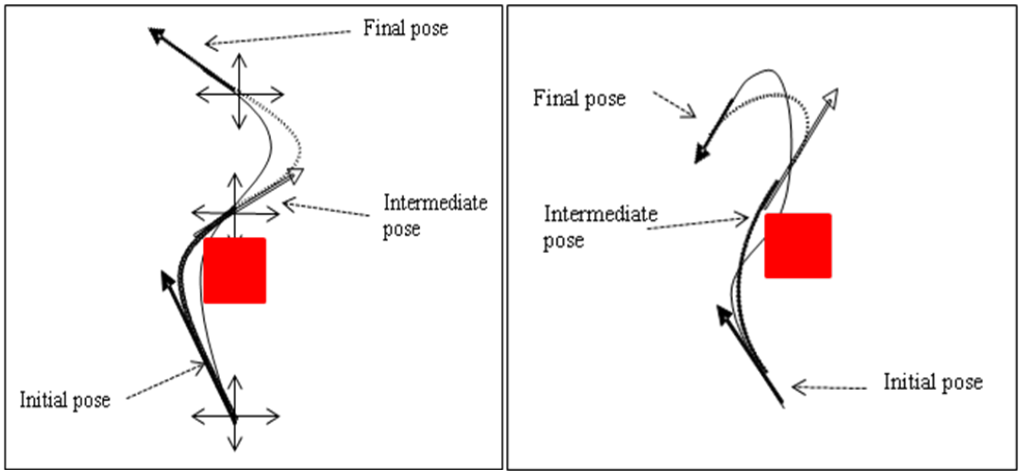

Figure 6.27: The unsafe ‘S’ shape paths (initial pose second quadrant, final pose any quadrant) and the amended safe paths

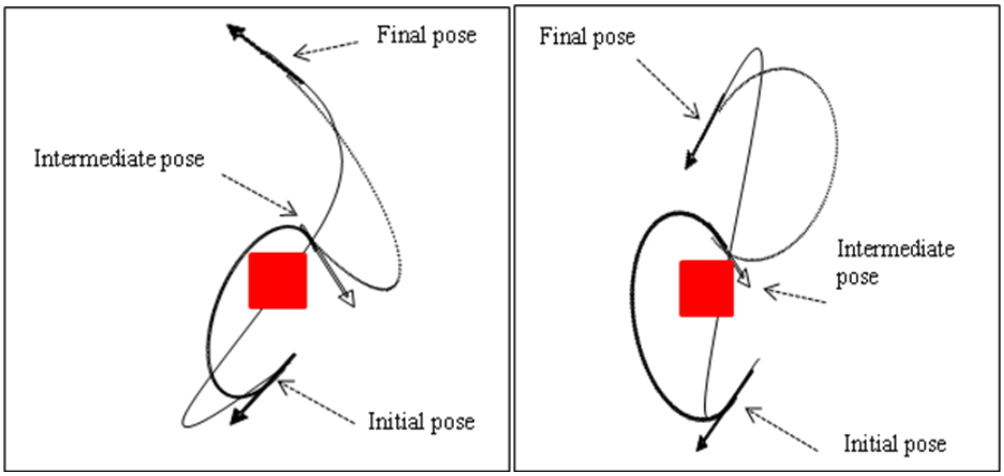

Figure 6.28: The unsafe ‘S’ shape paths (initial pose third quadrant, final pose any quadrant) and the amended safe paths

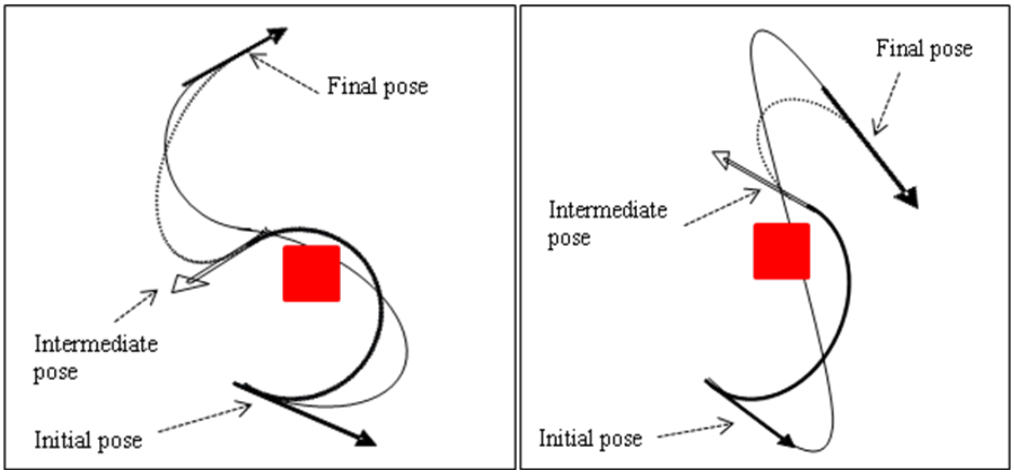

Figure 6.29: The unsafe ‘S’ shape paths (initial pose fourth quadrant, final pose any quadrant) and the amended safe paths

6.5 Simulation Results

The simulation results are obtained by taking only one case of 'S' shape path in which the direction of initial and final poses belong to quadrant 1. The method can be tested for the rest of cases. For simplicity the obstacle is assumed to be of square shape or rectangle shape with wider side facing the UAV.

The two methods (the curvature manipulation method and the proposed method) were simulated by taking the initial pose (14, 6, 60^o) and final pose (17, 40, 60^o) for the initial UAV path generation by PH path planner of chapter 2. The square obstacle was described by vertices (10, 17), (20, 17), (20, 27) and (10,27). The following results were obtained.

Figure 6.30 shows the initial path generated by PH path planner and the obstacle.

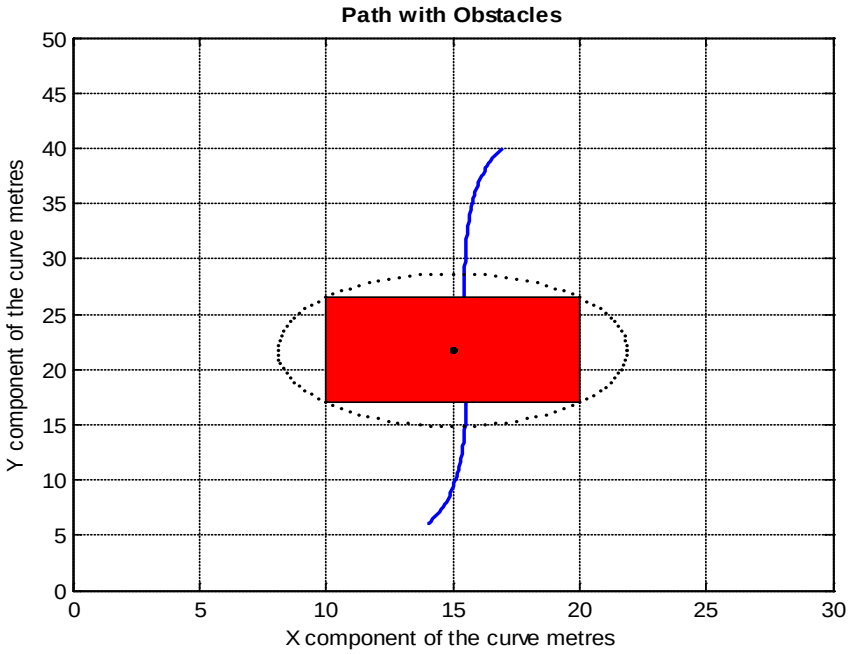

Figure 6.30: Path with the obstacle.

When the curvature manipulation method was applied to avoid the obstacle, the result was a path with an unacceptably high bending energy as shown in figure 6.31. The resultant path shown in figure 6.31 is not a flyable path because its high bending energy makes it difficult to obey the kinematic constraints of the UAV. If the kinematic constraints are tried to impose then the length of the path will increase tremendously. The proposed method can offer a remedy to this problem.

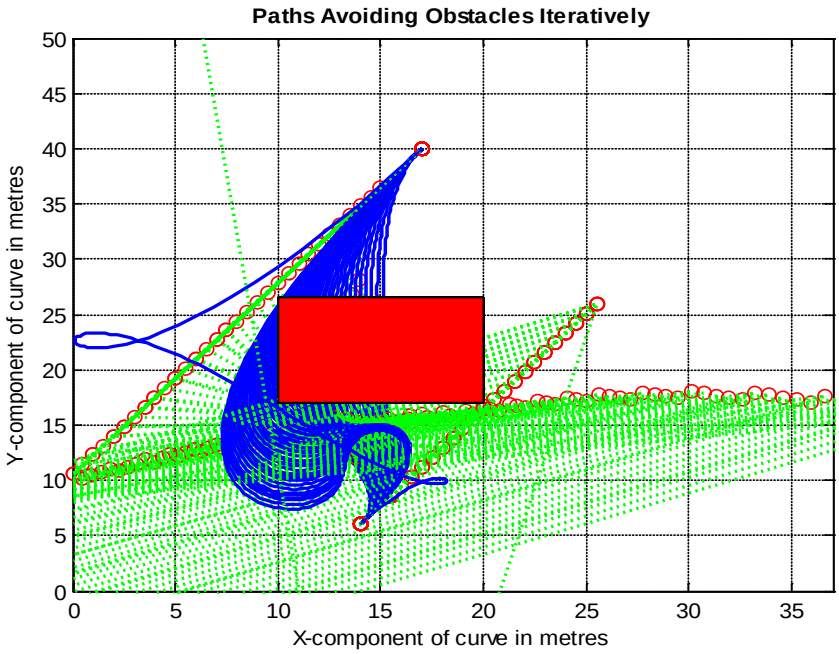

Figure 6.31: Path with different stages of curvature manipulation to avoid the obstacle.

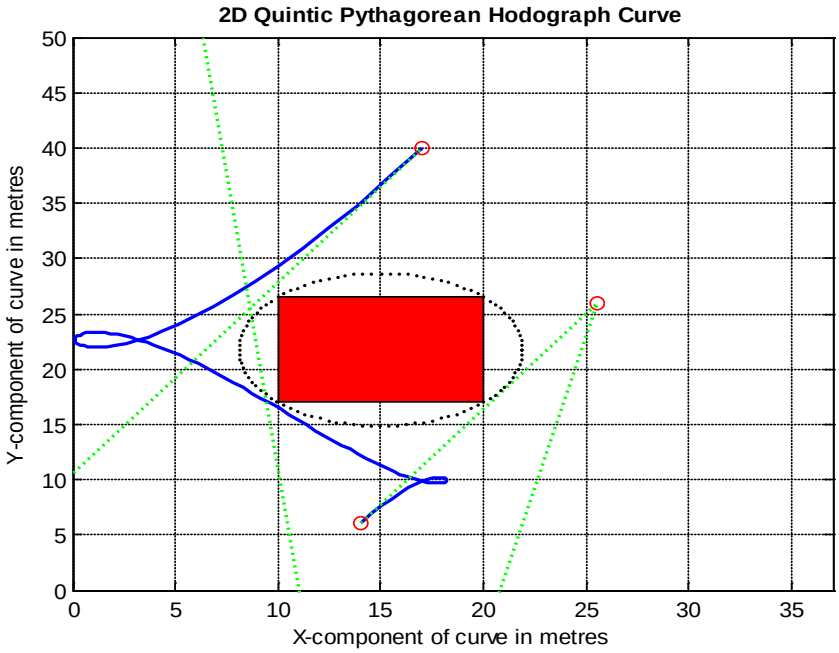

Figure 6.32: Final obstacle free path with very high bending energy.

The proposed method is applied by inserting an intermediate pose between the initial and final poses. Then the three poses were connected by Pythagorean Hodograph curves such that the individual curves are 'C' shape paths of figures 6.8, 6.9, 6.10 and 6.11. The result is shown in figure 6.33.

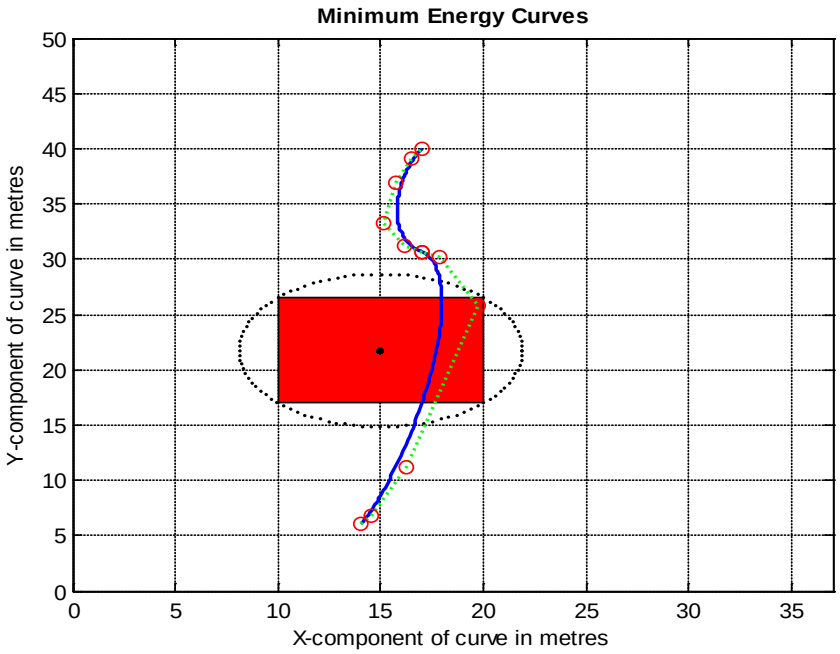

Figure 6.33: Path with same initial and final poses resulted from the proposed method.

Figure 6.34 shows the same path as that of figure 6.33 after imposing the kinematic constraints of the UAV. Although the resulting path shown in figure 6.34 comprises of two PH curves of least bending energy each complying with the safe cases of figures 6.8, 6.9, 6.10 and 6.11 but it is still not clear of the obstacle.

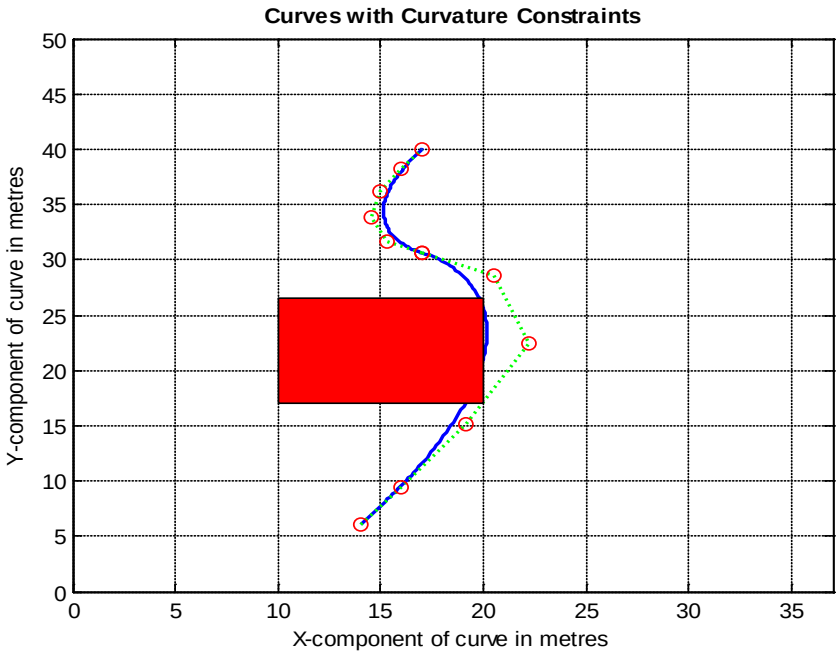

Figure 6.34: Path with curvature constraint been imposed

Now the absolute curvature of the first curve is increased iteratively to avoid the obstacle. Figure 6.35 shows the iterative process to avoid the obstacle.

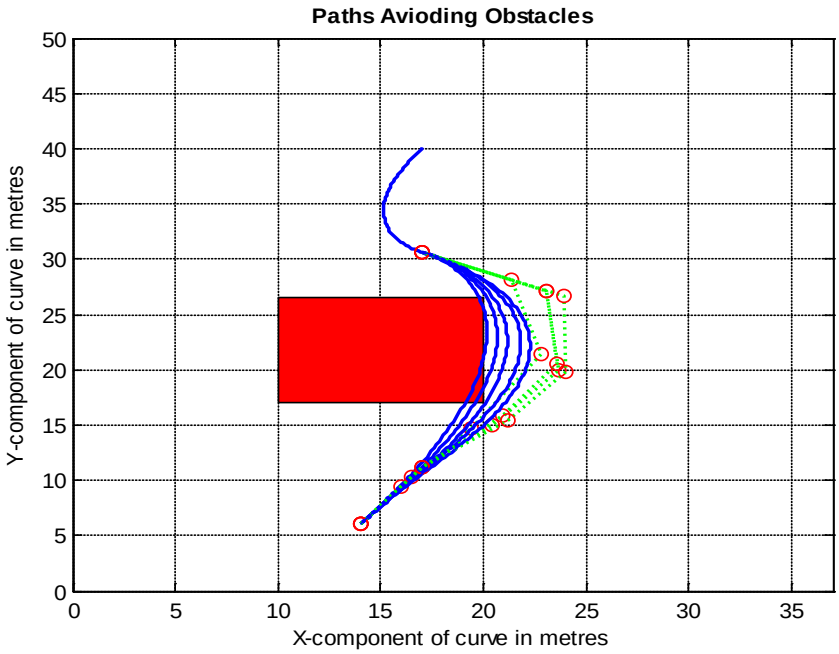

Figure 6.35: Curvature manipulation of the path to avoid the obstacle.

Finally figure 6.36 shows the final obstacle free feasible path. The path shown in figure 6.36 obeys the kinematic constraints of the vehicle. The bending energy is much lower compared to the path achieved with the curvature manipulation method. The length of the path is optimal as well.

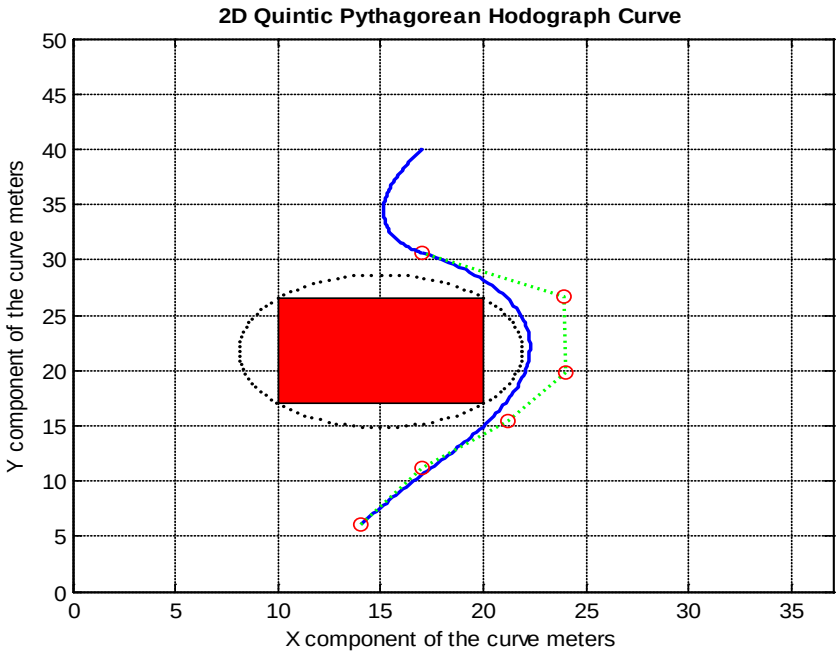

Figure 6.36: Obstacle free path with optimal bending energy and optimal path length.

Figures 6.37, 6.38 and 6.39 show curvature profiles of the two paths. From these figures it is clear that, the path generated by conventional method (path of figure 6.32) clearly

violates the kinematic constraint while the path generated by proposed method (path of figure 6.36) obeys the constraint.

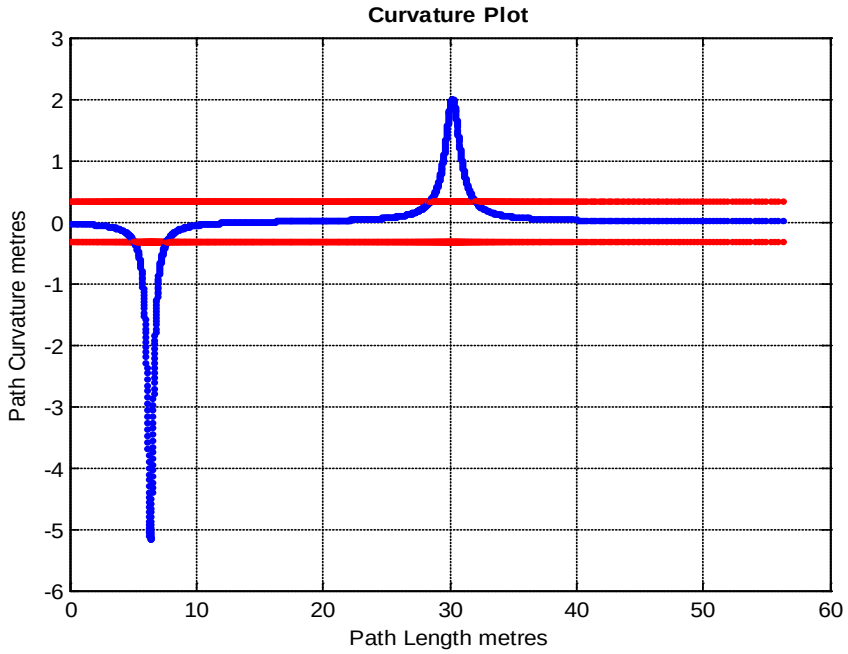

Figure 6.37: Curvature profile of avoidance path generated by curvature manipulation method

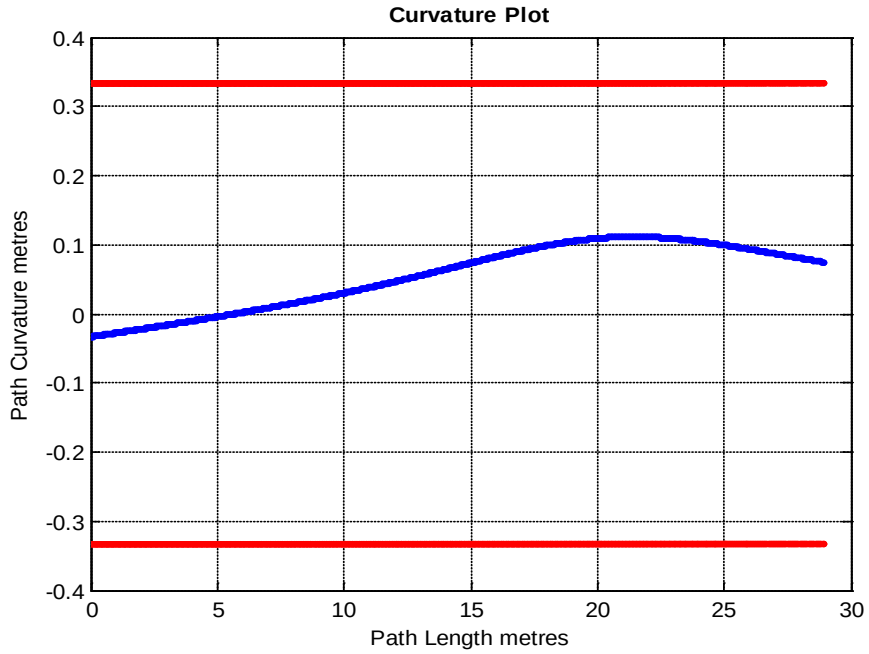

Figure 6.38: Curvature profile of avoidance path C1 generated by proposed method

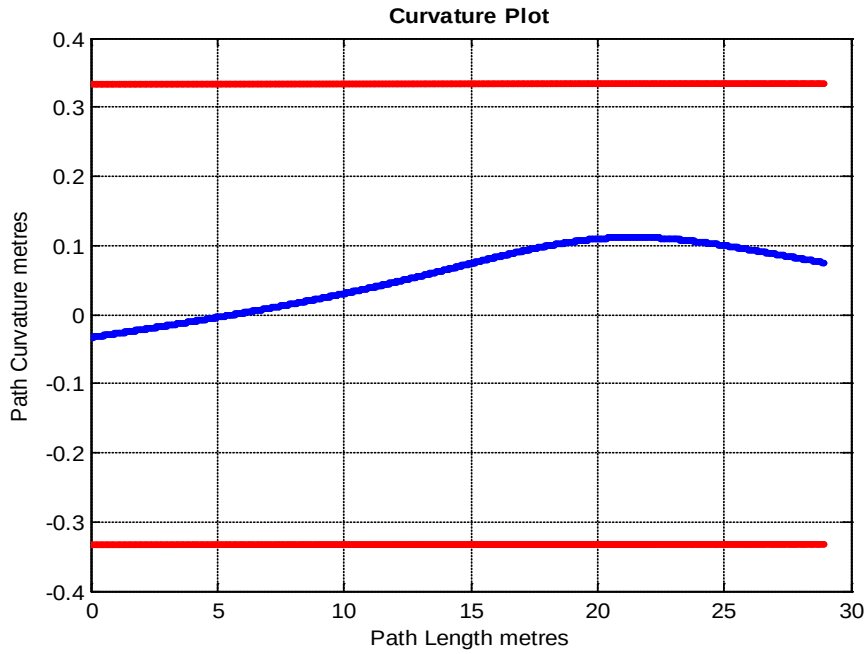

Figure 6.39: Curvature profile of avoidance path C2 generated by proposed method.

6.6 Conclusions

Looking at the path of figure 6.32 which is calculated by the curvature manipulation method and the path of the figure 6.36 calculated by the proposed method, some conclusions can be drawn which are summarised in table 6.1. It should be noted that the path generated by the proposed method has two parts C1 and C2. C1 is the path segment which connects the initial and intermediate configuration. C2 is the path segment which connects the intermediate and final configuration. The data in the table 6.1 in case of the path generated by the proposed method is given separately for C1and C2.

No	Paths	Length	Bending Energy	Iterations	Curvature Limits
1	Path of figure 6.32	56.087	9.5938	Many	Violates
2	Path of figure 6.36 (part C1)	28.9012	0.3602	Few	Obeys
	Path of figure 6.36 (part C2)	10.3651	0.1619	None	Obeys

Table 6.1: Summary of the results.

From table 6.1, the length of the path generated by conventional method (path of figure 6.32) is much greater than that of the path generated by the proposed method (path of figure 6.36). This will affect the time of flight.

The bending energy of the path in figure 6.32 is much more than that of the path in figure 6.36. This will affect the convenience with which the paths could be traced by the UAV.

The number of iterations after which the path of figure 6.32 is obtained is more than the iterations after which the path of figure 6.36 is obtained. Consequently the computational time is more for the path of figure 6.32 than the later. In other words the curvature manipulation method is much slower.

Chapter 7

Pythagorean Hodograph Paths for Cooperative Vision Based Target Surveillance and Tracking

Target surveillance and tracking is one of the most important operations in the UAVs context. The superiority of the UAVs over its manned counterpart to do the dull, dirty, dangerous and tedious repetitive tasks has made UAV a favourite option to do such operations. Target surveillance and tracking falls in this category of operations, therefore it is natural to consider UAV as a suitable option for surveillance and tracking. UAVs deployment for autonomous target surveillance and tracking is the subject of this chapter.

Vision based target tracking is one of the most important operations undertaken by the UAVs. UAVs are very effective in tracking operation due to their vantage position and freedom of movement. UAVs undertake such an operation during surveillance, search and rescue, forest fire monitoring, traffic monitoring, border patrol, convoy escorting and in combat military missions.

Surveillance and tracking of a moving target on the ground by multiple cooperative UAVs using onboard vision sensor (camera) is a challenging operation in the UAVs missions. The challenges associated with this operation are:

- Keeping updated knowledge of the target position.
- Maintaining close proximity of the target.

In this chapter a method is proposed for surveillance and tracking of a moving ground target autonomously by multiple cooperative UAVs. The proposed method uses camera based cooperative perception algorithm (developed in chapter 4 and 5) for locating the target and Pythagorean Hodograph (PH) bases path planning algorithm (developed in chapter 2 and 3) for keeping close proximity of the target.

The UAVs considered here for tracking are fixed wings UAVs having certain dynamic and kinematic constraints. The target however, is free from these constraints. The speed of the UAVs is assumed to be greater than that of the target, otherwise the target cannot be tracked. The UAVs are kept at different constant heights to avoid inter collisions. The simulation results at the end of the chapter show the efficiency of the proposed method.

In literature there are various methods proposed for the problem of target tracking. Each method addresses the problem from its own perspective, arriving at the solutions which are complimenting each other, rather than offering a single best solution. Therefore

there is a room for more research in order to arrive at more compact solution. To achieve a single compact solution for target tracking is the motivation of this research.

In [110] a sinusoidal path pattern to follow a moving target has been proposed. This sinusoidal pattern becomes a circular or rose pattern when the moving target stops. The amplitude of the wave is varied to keep the proximity of the target. This method addresses the tracking problem from the path planning perspective and there is less attention given to the details of the simultaneous position acquisition of the target.

In [111] dynamic path planning for routing UAV in order to track ground target is formulated as an optimal control problem consisting of system dynamics, a set of boundary condition, control constraint and a cost criterion. This method seems to ignore some of the kinematic constraints such as curvature continuity, because the path used are Dubins type paths which have a discontinuous curvature profile.

In [55] the tracking problem is formulated as an estimation problem using the target position as a state vector. The optimal estimate of the target position is obtained by optimizing the sensors noise entailed with the target position measurement. This method assumes the proximity of the target, therefore it does not consider the path planning to manoeuvre the UAVs in order to maintain the proximity of the target.

In [112] an algorithm to identify target size, location and orientation, based on machine vision is derived. The algorithm uses target measurement taken from video camera to update estimates of the target. The EKF and UKF were used for estimation. Again this method mentions no strategy for chasing the target.

In [113] vision based tracking is considered using circular pattern for path planning and onboard gimbal camera for target position acquisition. The gimbal camera is controlled by an algorithm according to the position and orientation of the aircraft and the position of the target. The proposed technique presented in this chapter is similar to that mention in [113], but instead of circular patterns Pythagorean hodographs curves are used to route the UAVs to follow the moving target. PH curves are very flexible and can adopt itself to any shapes dynamically as demanded by the manoeuvring target. In [113] the gimbal camera is required because of the use of less flexible circular pattern. Since more flexible PH curves are used in the proposed tracking technique, therefore gimbal camera is not needed. In this research camera is rigidly attached, to the aircraft. More over the target is tracked cooperatively by at least two UAVs.

7.1 Problem Formulation

A moving ground target needed to be tracked autonomously by two UAVs in a cooperative way. The target is moving and the UAVs need to chase the target in order to track it. As the speed of the UAVs is considerably greater than that of the ground target, and have certain kinematic constraints, therefore they (UAVs) must be manoeuvred such that the manoeuvring paths are feasible and can provide the UAVs a close proximity of the ground target. The close proximity must ensure that the target is in the

field of view of at least one of the tracking UAVs. The manoeuvring paths of the UAV are subjected to the following constraints:

1. $$|\kappa_i(t)| \le \kappa_{\max} \text{ and } |\tau_i(t)| \le \tau_{\max}$$

   Where κ_i and τ_i is the curvature and torsion of the path of *ith* UAV respectively, $\kappa_{\max}$ and $\tau_{\max}$ are the maximum curvature and torsion attainable by the UAVs.

2. The vehicles must keep a safe distance to avoid inter collision while tracing their manoeuvring paths.

   $$R_{si} \cap R_{sj} = \varphi$$

   That is, the intersection of the safety spheres of radius R_{si} and R_{sj} corresponding to *ith* and *jth* UAVs must be empty. Or equivalently the minimum distance d between the two vehicles must be greater than $2R_s$. i.e. $d \ge 2R_s$.

3. The vehicles must stay within communication range while travelling on these paths. i.e.

   $$d \le R_c$$

   d is the distance between the cooperating UAVs and R_c is the radius of communication range sphere. The wireless communication between the UAVs ensures the exchange of position information of the target for fusion purposes.

4. The union of the fields of view of cameras of cooperating UAVs must contain the object of interest at all the time while travelling on these paths. If R_{fov_i} and R_{fovj} are the radii of the fields of view of the *ith* and *jth* UAVs cameras respectively and x_{obj} is the object of interest then:

   $$x_{obj} \in R_{fovi} \cup R_{fovj} \qquad \text{for } \forall\ i \ne j$$

   Where the suffix *i* represents the *ith* UAV and *j* represents the *jth* UAV.

Successive feasible manoeuvring paths are planned using the path planning algorithm of chapter 2 or 3. While moving on these manoeuvring paths, target position estimates are acquired by the cooperative perception algorithm developed in chapter 4 and 5.

The UAVs follow the target autonomously as long as the tracking mission continues.

7.2 Proposed Method

In this method of ground target tracking, first of all, the manoeuvring paths are planned using the path planning algorithm developed in chapter 2 and 3, for the cooperating UAVs each time the target moves. These paths must satisfy all the conditions stated in

the section 7.1. The camera reference frame is specified as stated in chapter 4. The target is located in camera frame by the perception system using the technique developed in chapter 5. The acquired target position information is then transformed from respective camera frames to a universal frame. This transformed target position information is used to generate a waypoint for extending the chasing manoeuvres. The initial and final poses of the UAVs are updated each time the paths are extended. The chasing paths must enable the cooperative UAVs to maintain close proximity to the moving target.

The steps of the proposed method are:

1. Calculating the reference frame for camera.
2. Locating target in camera frame.
3. Locating the target position in universal frame.
4. Planning the chasing paths.
5. Updating the poses of UAV to extend the chasing paths.

The step by step details of the method are given in the following sections.

7.2.1 Calculating the Reference Frames for Camera

The starting point of the proposed method is to specify a reference frame for the camera used in perception system. In this method the Frenet frames of the UAVs flight are used as the camera reference frame.

At each point of a regular curve the Frenet frame is defined by the orthonormal basis $(\vec{t},\vec{n},\vec{b})$ in R^3 aligned with local intrinsic curve geometry. The elements of this basis are the unit tangent vector $\vec{t}$ (tangent to the curve), unit normal vector $\vec{n}$ and a unit binormal vector $\vec{b}$. The Frenet frame of the UAV's flight path is considered to be UAV's local frame with its origin at the centre of mass of the UAV. The following system of equations relates the Frenet Frame with its derivatives:

$$\begin{bmatrix} \vec{t}\,' \\ \vec{n}\,' \\ \vec{b}\,' \end{bmatrix} = \begin{bmatrix} 0 & \kappa & 0 \\ -\kappa & 0 & \tau \\ 0 & \tau & 0 \end{bmatrix} \begin{bmatrix} \vec{t} \\ \vec{n} \\ \vec{b} \end{bmatrix} \tag{7.1}$$

Where $\vec{t}\,'$, $\vec{n}\,'$ and $\vec{b}\,'$ are the derivatives of the unit tangent vector $\vec{t}$, unit normal vector $\vec{n}$ and a unit binormal vector $\vec{b}$ respectively. $\kappa(t)$ is the trajectory's curvature at time t, given by:

$$\kappa(t) = \frac{\left|\dot{r}(t) \times \ddot{r}(t)\right|}{\left|\dot{r}(t)\right|^3} \tag{7.2}$$

$\tau(t)$ is the torsion of the trajectory at time t and is given by:

$$\tau(t) = \frac{\left[\dot{r}(t) \times \ddot{r}(t)\right] \bullet \dddot{r}(t)}{\left|\dot{r}(t) \times \ddot{r}(t)\right|^2} \tag{7.3}$$

$(\vec{t}, \vec{n}, \vec{b})$ is calculated for all points on each UAV planned trajectory using equation (7.1), and then used as UAV local reference frame for the corresponding UAV. Figure 7.1 shows the UAV planned trajectory and the calculated Frenet Frames throughout the trajectory.

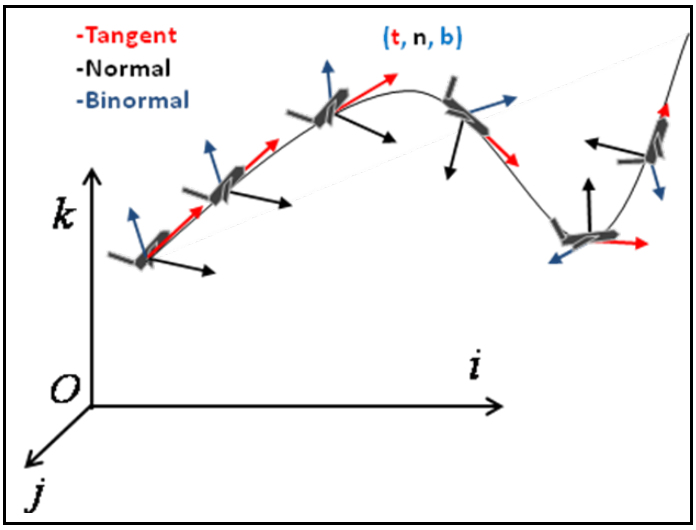

Figure 7.1: UAV path with Frenet frame $(\vec{t}, \vec{n}, \vec{b})$

As the camera is attached to the UAV at the point coinciding with the centre of the mass of UAV, Therefore the Frenet frame can be used as camera reference frame. If the camera is attached at the centre of mass of the UAV with a little tilt angle ζ then the camera reference frame can be derived from the UAV frame by rotating the UAV frame by the tilt angle about the normal unit vector.

If the UAV body frame is $(\vec{t}, \vec{n}, \vec{b})$, then the camera reference frame $(\vec{t}_{cam}, \vec{n}_{cam}, \vec{b}_{cam})$ with the origin C can be obtained from the following system of equations:

$$\begin{bmatrix} \vec{t}_{cam} \\ \vec{n}_{cam} \\ \vec{b}_{cam} \end{bmatrix} = \begin{bmatrix} \cos\zeta & 0 & -\sin\zeta \\ 0 & 1 & 0 \\ -\sin\zeta & 0 & \cos\zeta \end{bmatrix} \begin{bmatrix} \vec{t} \\ \vec{n} \\ \vec{b} \end{bmatrix} \tag{7.4}$$

Where ζ is the tilt angle which the optical axis of the camera is making with the direction of motion of the UAV as shown in figure 7.2.

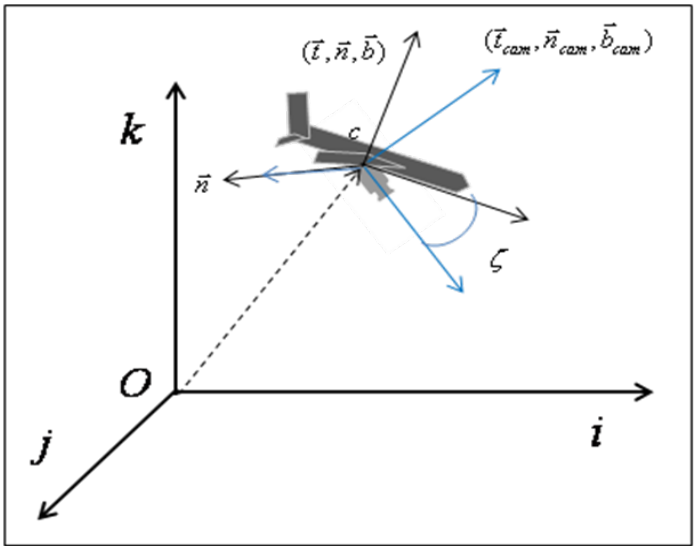

Figure 7.2: UAV with Frenet Frame $(\vec{t},\vec{n},\vec{b})$ as body frame and the derived camera frame $(\vec{t}_{cam},\vec{n}_{cam},\vec{b}_{cam})$

The Rotation Minimising Frame derived in chapter 4 is the most suitable reference frame for camera, but the simulation results obtained in this chapter is based on Frenet Frames.

7.2.2 Target Locating in Camera frame

The most important step in tracking operation is to acquire the target position information through the perception system in camera reference frame. In order to locate the target in camera frame, the techniques developed in chapter 5 are used.

Referring to figure 7.3, let the coordinates of the target expressed in camera frame is $({}^{c}X,{}^{c}Y,{}^{c}Z)$ and coordinates of the image of the target expressed in the camera frame is $({}^{c}x,{}^{c}y,{}^{c}z)$ then the relation between these two points in camera frame is given by the following equations:

$$ {}^{c}x = f\frac{{}^{c}X}{{}^{c}Z} \tag{7.5} $$

$$ {}^{c}y = f\frac{{}^{c}Y}{{}^{c}Z} \tag{7.6} $$

Where is f is the focal length of the camera.

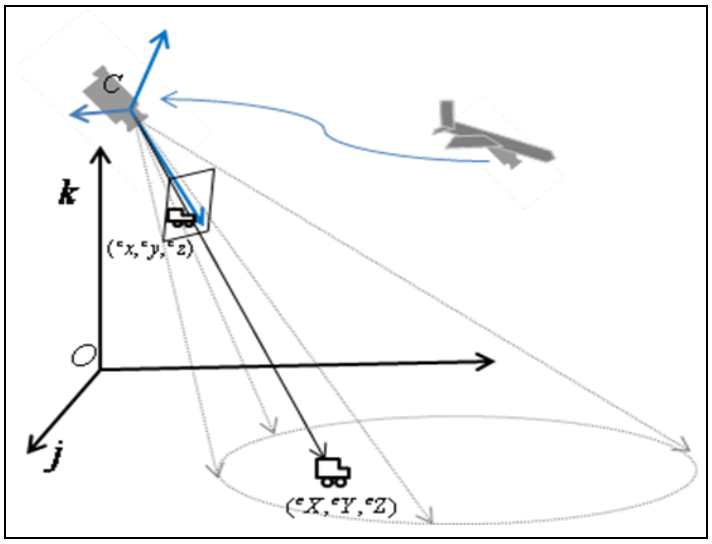

Figure 7.3: Camera with universal frame, image frame and field of view containing the target.

Referring to figure 7.4, the relation between the coordinates of the image point expressed in image reference frame and the coordinates of the image point expressed in camera reference frame are given by the following equations:

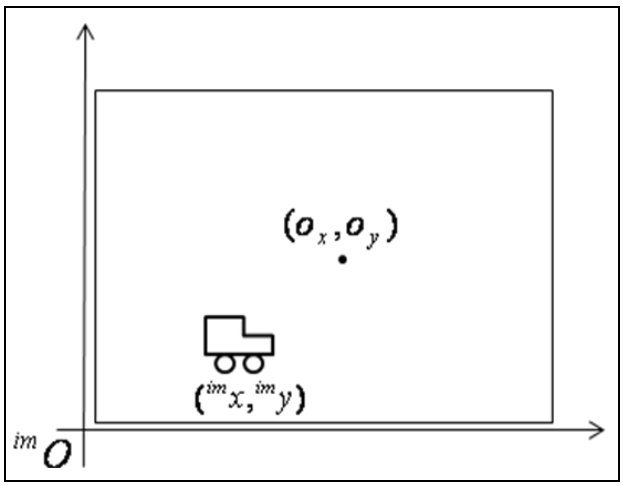

Figure 7.4: Image frame containing the image of the target.

$$^{c}x = -(^{im}x - o_x)s_x \tag{7.7}$$

$$^{c}y = -(^{im}y - o_y)s_y \tag{7.8}$$

$$^{c}z = f \tag{7.9}$$

Where o_x, o_y are the coordinates in pixel of the principal point. s_x, s_y are the effective sizes of the pixel in millimetres in the horizontal and vertical direction respectively. $^{im}x, ^{im}y$ are the pixel coordinates of the image point in the image reference frame and f is the focal length of the camera. o_x, o_y, s_x, s_y and f are the intrinsic

parameter and are readily known for the camera in use. ${}^{im}x, {}^{im}y$ are known from the image frame. From these known quantities the position vector of the target in camera $({}^{c}X, {}^{c}Y, {}^{c}Z)$ is calculated by equating equations (7.5), (7.7) and (7.6), (7.8):

$$ {}^{c}X = -\frac{{}^{c}Z}{f}({}^{im}x - o_x)s_x \tag{7.10}$$

$$ {}^{c}Y = -\frac{{}^{c}Z}{f}({}^{im}y - o_y)s_y \tag{7.11}$$

${}^{c}Z$ is the ray from the camera centre through the target image to the terrain. This ray can be calculated from commercially available Digital Elevation Map (DEM) of the terrain.

At this point the technique developed in chapter 5 is integrated in this method to ensure cooperation in perception. This is shown in figure 7.5.

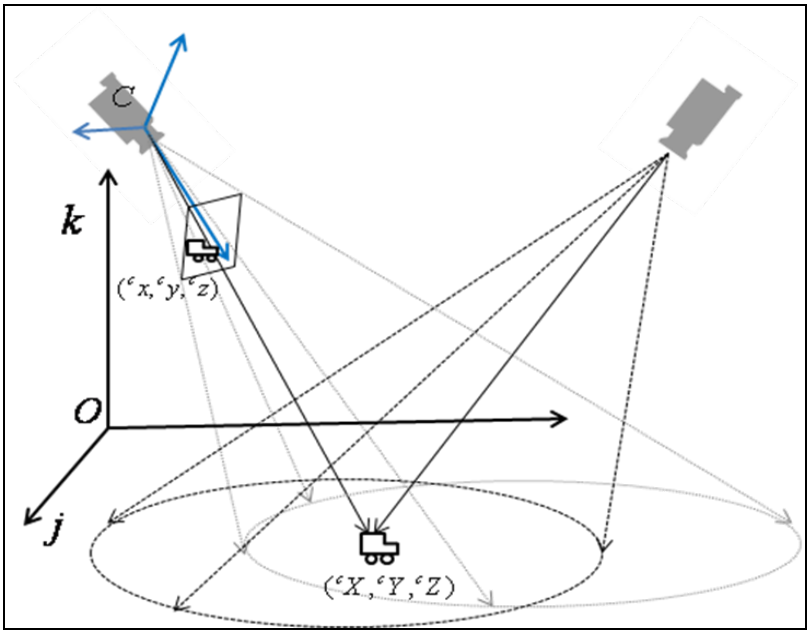

Figure 7.5: Cooperative cameras with universal frame, image frame and field of view containing the target.

The target location information is obtained cooperatively in camera frame, which is then transformed to a universal frame. In the next subsection the detail of the transformation from camera frame to the universal frame is discussed.

7.2.3 Calculating the Target Position in Universal Frame

Once the target is located in camera reference frame, the transformation between the camera reference frame and the Universal frame is performed through rotation matrix and translation vector. The detail of the transformation process is as follow.

Referring to figure 7.6, ${}^{O}\vec{P}_1 = [\vec{x}_{uav}, \vec{y}_{uav}, \vec{z}_{uav}]^T$ is the position vector of the centre of mass of the UAV (or equivalently the position vector of the camera reference frame origin C) expressed in Universal frame with the origin O. This is known from the GPS sensor onboard UAV.

${}^{C}\vec{P}_2 = [{}^{c}X, {}^{c}Y, {}^{c}Z]^T$ is the position vector of the target expressed in camera frame with the origin C. This is known from the cameras onboard the UAVs as given by equation (7.10) and (7.11).

${}^{O}\vec{P}_3 = [\vec{x}_{tar}, \vec{y}_{tar}, \vec{z}_{tar}]^T$ is the position vector of the target expressed in Universal frame with the origin O, and is needed to be calculated.

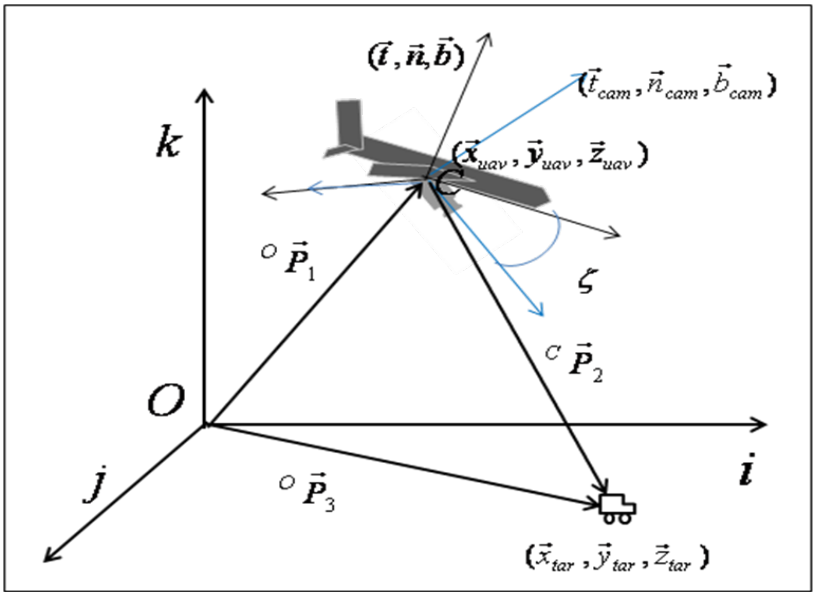

Figure 7.6: UAV with camera, Universal frame, UAV frame, camera frame and target

In order to calculate ${}^{O}\vec{P}_3$ in Universal frame, all the vectors must be expressed in Universal frame. ${}^{O}\vec{P}_1$ is already expressed in Universal frame, ${}^{C}\vec{P}_2$ is expressed in camera frame therefore it needs to be expressed in Universal frame. After this conversion the matter reduces to a simple vector addition.

${}^{C}\vec{P}_2$ is transformed to Universal frame by multiplying with rotation matrix. The required rotation matrix can be found by describing the principal unit vectors of the camera frame i.e. $(\vec{t}_{cam}, \vec{n}_{cam}, \vec{b}_{cam})$ in terms of coordinate system of the Universal frame (i, j, k) and then stake these vectors as columns of a 3×3 matrix. Since this particular rotation describes frame C relative to frame O it will be denoted by ${}^{O}_{C}R$ (denotes rotation of frame C with respect to frame O) which is given by:

$$ {}^{O}_{C}R = \begin{bmatrix} \vec{t}_{cam}.i & \vec{n}_{cam}.i & \vec{b}_{cam}.i \\ \vec{t}_{cam}.j & \vec{n}_{cam}.j & \vec{b}_{cam}.j \\ \vec{t}_{cam}.k & \vec{n}_{cam}.k & \vec{b}_{cam}.k \end{bmatrix} \tag{7.12} $$

${}^{C}\vec{P}_2$ (denotes vector $\vec{P}_2$ expressed in frame C) when expressed in Universal frame will be denoted by ${}^{O}_{C}\vec{P}_2$ (denotes vector $\vec{P}_2$ of frame C but expressed in frame O) and is given by:

$$ {}^{O}_{C}\vec{P}_2 = {}^{O}_{C}R\,{}^{C}\vec{P}_2 $$

$$ {}^{O}_{C}\vec{P}_2 = \begin{bmatrix} \vec{t}_{cam}.i & \vec{n}_{cam}.i & \vec{b}_{cam}.i \\ \vec{t}_{cam}.j & \vec{n}_{cam}.j & \vec{b}_{cam}.j \\ \vec{t}_{cam}.k & \vec{n}_{cam}.k & \vec{b}_{cam}.k \end{bmatrix} \begin{bmatrix} {}^{c}X \\ {}^{c}Y \\ {}^{c}Z \end{bmatrix} \tag{7.13} $$

Now the absolute target position is given by vector addition:

$$ {}^{O}\vec{P}_3 = {}^{O}\vec{P}_1 + {}^{O}_{C}\vec{P}_2 $$

$$ \begin{bmatrix} \vec{x}_{tar} \\ \vec{y}_{tar} \\ \vec{z}_{tar} \end{bmatrix} = \begin{bmatrix} \vec{x}_{uav} \\ \vec{y}_{uav} \\ \vec{z}_{uav} \end{bmatrix} + \begin{bmatrix} \vec{t}_{cam}.i & \vec{n}_{cam}.i & \vec{b}_{cam}.i \\ \vec{t}_{cam}.j & \vec{n}_{cam}.j & \vec{b}_{cam}.j \\ \vec{t}_{cam}.k & \vec{n}_{cam}.k & \vec{b}_{cam}.k \end{bmatrix} \begin{bmatrix} {}^{c}X \\ {}^{c}Y \\ {}^{c}Z \end{bmatrix} \tag{7.14} $$

Thus the absolute target position is calculated, which will be used in planning the chasing paths for pose specification in the next subsection. This position will be converted to final pose of the UAV for path planning by appending some convenient orientation. The third dimension in pose specification will be replaced by the altitude of the corresponding UAV.

7.2.4 Planning the Chasing Paths

Pythagorean Hodograph curves are used to plan the chasing path for cooperating UAVs. In path planning for each UAV, the current pose of the each UAV is used as the initial pose and the final pose for each UAV is derived from the acquired target position given by equation (7.14).

Consider *n* UAVs deployed in a tracking mission. For path planning purpose let the starting pose of the *ith* UAV is $Pose_{si}(x_{si}, y_{si}, z_{si}, \theta_{si}, \phi_{si})$ and the final pose of the same UAV is $Pose_{fi}(x_{fi}, y_{fi}, z_{fi}, \theta_{fi}, \phi_{fi})$, then the path of the UAV is defined as the parametric curve $r_i(t) = [x(t), y(t), z(t)]$ with curvature κ_i and torsion, τ_i connecting the initial and final pose:

$$Pose_{si}(x_{si}, y_{si}, z_{si}, \theta_{si}, \phi_{si}) \xrightarrow{r_i(t)} Pose_{fi}(x_{fi}, y_{fi}, z_{fi}, \theta_{fi}, \phi_{fi})$$

Subjected to the constraints mentioned in section 7.1.

These constrained paths are Pythagorean hodograph of chapter 2 and 3. Pythagorean hodograph path for each tracking UAV can be successfully produced by Hermite interpolation of initial and final configuration. Thus to produce the Pythagorean Hodograph path for each UAV, the starting or initial pose and the final pose of UAVs must be known:

For UAV1 the poses are specified as follows:

Initial pose of the UAV1=

$$Pose_{Suav1}(x_{Cuav1}, y_{Cuav1}, z_{Cuav1}, \theta_{Cuav1}, \phi_{Cuav1})$$

Where $(x_{Cuav1}, y_{Cuav1}, z_{Cuav1})$ is the current UAV1 position and $(\theta_{Cuav1}, \phi_{Cuav1})$ is the current UAV1 orientation. The finish or final pose of UAV1 is:

Final pose of UAV1=

$$Pose_{Fuav1}(x_{Ctar}, y_{Ctar}, z_{Cuav1}, \theta_{Cuav1}, \phi_{Ctuav1})$$

Where $(x_{Ctar}, y_{Ctar}, z_{Cuav1})$ is the current target position determined by the perception system with the z-dimension been replaced by the UAV1 altitude z_{Cuav1}. The orientation of the final pose is taken the same as that of the starting pose of UAV1 i.e. $(\theta_{Cuav1}, \phi_{Cuav1})$.

To produce the PH path for UAV2 the poses are specified as follows:

Initial pose of the UAV2=

$$Pose_{Suav2}(x_{Cuav2}, y_{Cuav2}, z_{Cuav2}, \theta_{Cuav2}, \phi_{Cuav2})$$

Where $(x_{Cuav2}, y_{Cuav2}, z_{Cuav2})$ is the current UAV2 position and $(\theta_{Cuav2}, \phi_{Cuav2})$ is the current UAV2 orientation. The finish or final pose of UAV2 is taken:

Final pose of UAV2=

$$Pose_{Fuav2}(x_{Ctar}, y_{Ctar}, z_{Cuav2}, \theta_{Cuav2}, \phi_{Ctuav2})$$

Where $(x_{Ctar}, y_{Ctar}, z_{Cuav2})$ is the current target position acquired by the perception system with z-dimension been replaced by the UAV2 altitude z_{Cuav2}. The orientation of the final pose is taken the same as that of the starting pose of UAV2 i.e. $(\theta_{Cuav2}, \phi_{Cuav2})$.

Once the initial and final poses for both the UAVs are specified, the PH technique of chapter 3 can be applied to produce the chasing path for the UAVs.

7.2.5 Updating the Poses of the UAVs to Extend the Chasing Paths

This is the last and very important stage of the proposed method. Each time when the new target position is acquired by the perception system, the UAVs are required to follow the target new position in order to maintain its close proximity. This is achieved by updating the initial and final poses of the tracking UAVs every time the new target position information is received. This way the paths of the UAVs extend along the moving target. The poses of the UAVs are updated as follow:

New Initial pose of the *ith* UAV= Old final pose of the *ith* UAV

New final pose of the *ith* UAV= Latest calculated position of the target.

This is repeated until the tracking operation is over. This whole process is shown in figure 7.7.

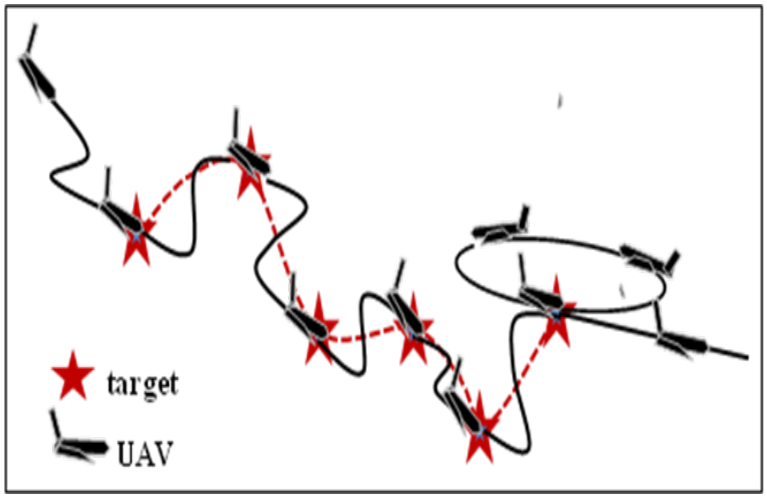

Figure 7.7: UAV tracking the target

7.3 Simulation Results

The simulation has the following two parts:

1. Tracking a target in a known domain.
2. Tracking a target in an unknown domain.

Each of the above case is discussed in the following subsections.

7.3.1 Tracking a Target in a Known Domain

The algorithm was tested by assuming that the domain of the target is known. The target tracking operation was simulated in known rectangular area as shown in figure 7.8. The paths for the two UAVs to chase the moving target were planned using Pythagorean

hodograph curves. The dotted path of the target is not known at this stage but is drawn for illustration purpose. It will be estimated by the UAVs in the tracking operation.

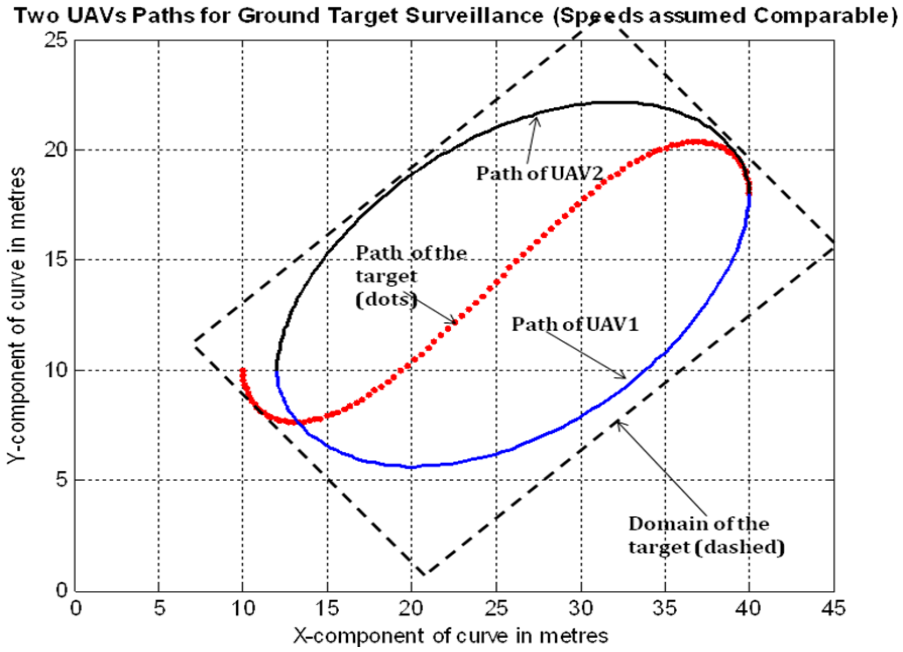

Figure 7.8: The domain of the target motion (dashed rectangle) with the initial planned paths of the UAVs and the target trajectory (dotted)

In figure 7.9 the UAVs are shown to manoeuvre in order to keep the target in the field of view. After each manoeuver the UAVs return to its original planned path. The congestion of the manoeuvering paths depend on the difference of speed of the UAVs and the target.

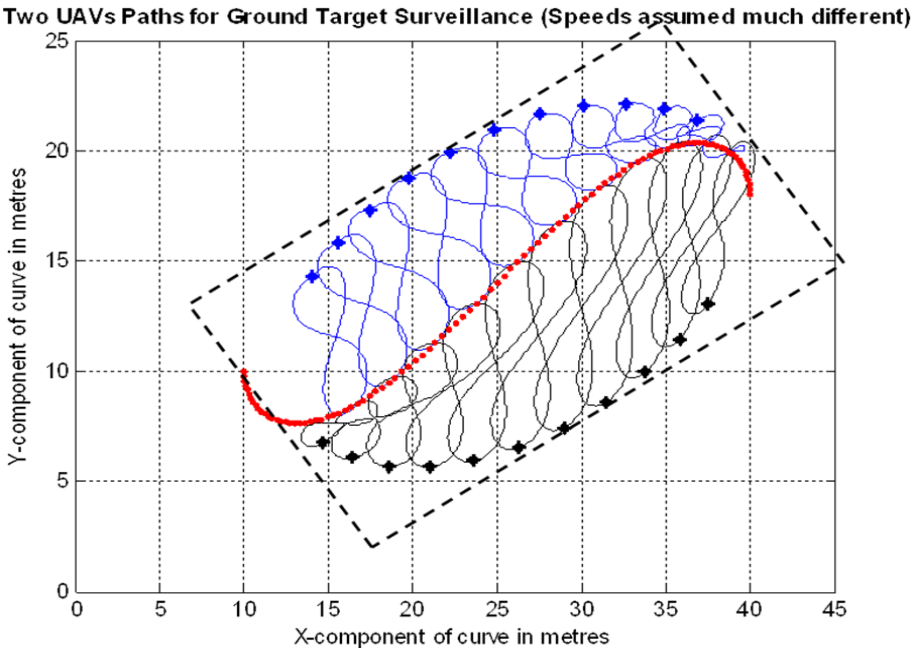

Figure7.9: The manoeuvres of the UAV1 and UAV2 in domain for target tracking.

While manoeuvring, the UAVs track the target. Figures 7.10, 7.11 and 7.12 show the component wise errors in estimates of the target trajectory plotted against time. The error in the start is very large, but converges to acceptable limits (within 0.5metre) with the passage of time. The large error is due to the initialization of the state vector arbitrarily.

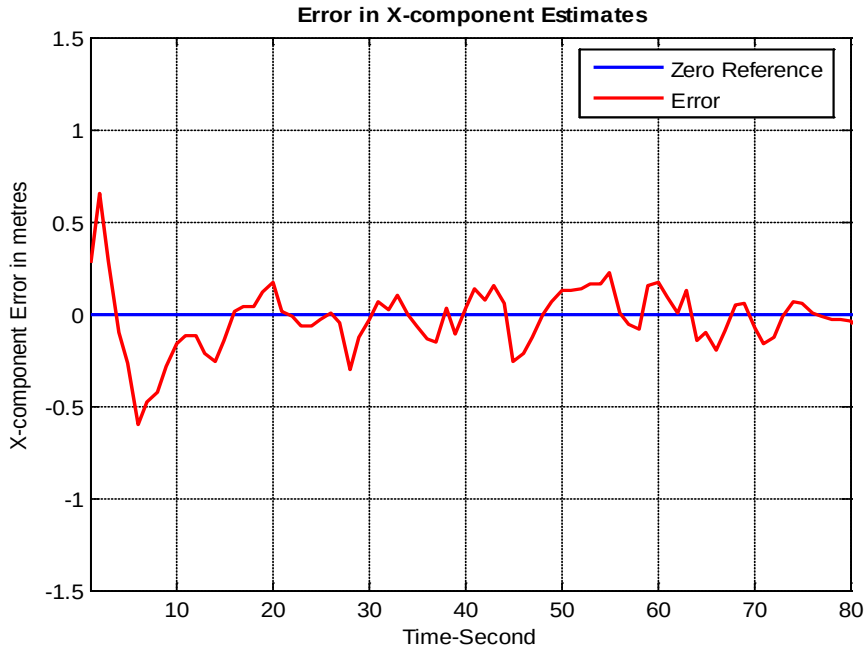

Figure 7.10: The X-estimates error of the target trajectory plotted against time

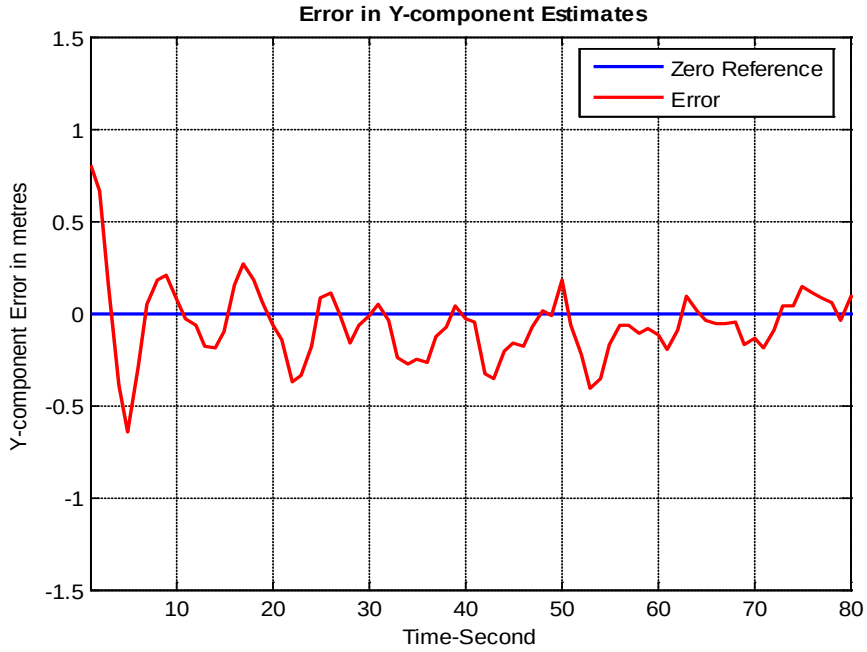

Figure 7.11: The Y-estimates error of the target trajectory plotted against time

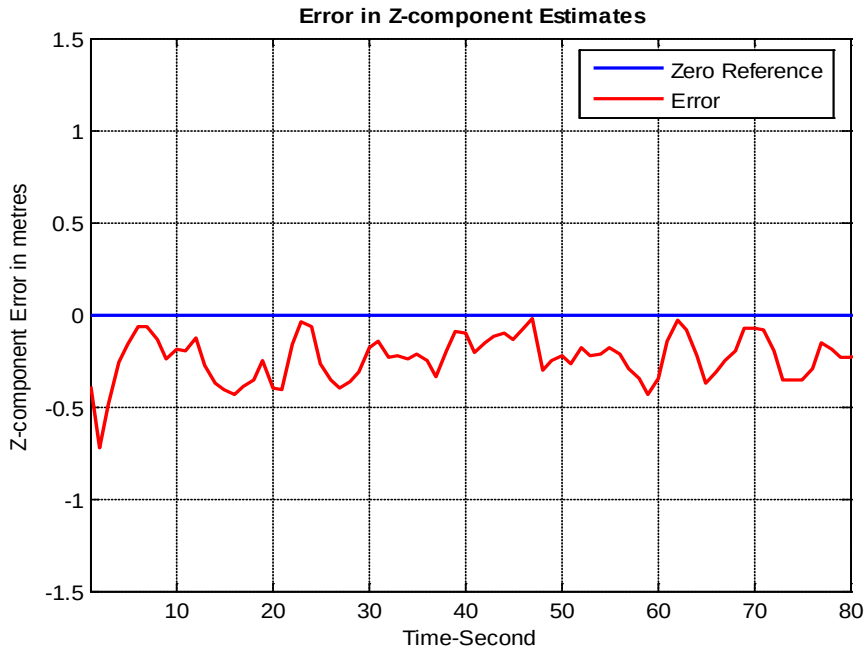

Figure 7.12: The Z-estimates error of the target trajectory plotted against time

7.3.2 Tracking a Target in an Unknown Domain

In this simulation, the domain of the target movement is assumed to be unknown. The vision sensors are used to obtain the position information of the ground target as it moves. The updated positions of the ground target are then passed on to the PH path planning algorithm in sequence. Based on the ground target position the proposed algorithm finds the extended path for UAV1 to chase the target. Figure 7.13 shows path of UAV1 extending along with the motion of ground target.

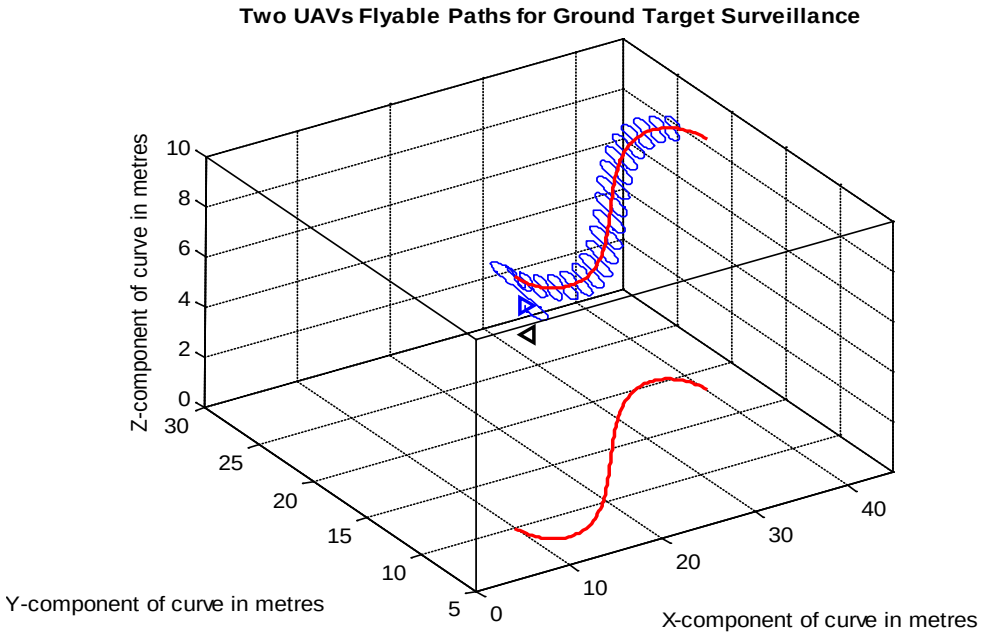

Figure 7.13: Path of the UAV1 (blue) to track the ground target (red)

In a similar way the extended path of UAV2 is calculated by the proposed algorithm to track the target. Figure 7.14 shows the path of UAV2 along with the ground target path.

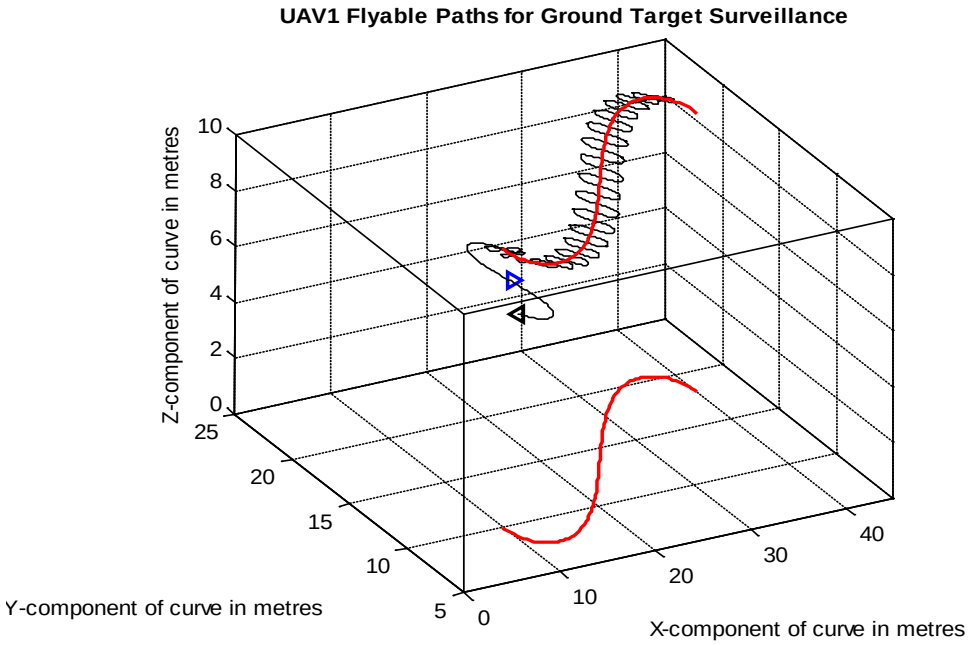

Figure 7.14: Path of the UAV2 (black) to track the ground target (red)

Figure 7.15 shows the paths of UAV1 and UAV2. The UAVs are kept close enough to impose the cooperation between them. This cooperation must enable the UAVs to have the ground target either in the field of view of both UAVs simultaneously or in the field

of view of at least one UAV at any time. Since the UAVs are communicating all the time therefore the ground target is position information is available to both UAVs all the times even if the target is visible to only one UAV. The cooperative paths of UAV1 and UAV2 are shown in figure 7.15.

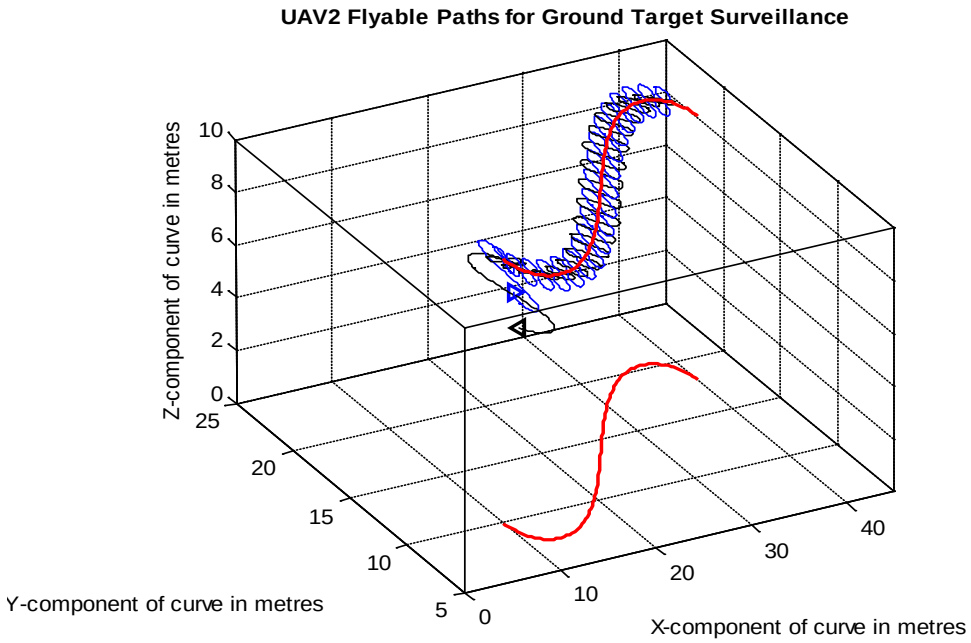

Figure 7.15: Cooperative paths of the UAV1 (blue) and UAV2 (black) to track the ground target (red)

The path of the ground target was then changed and the algorithm was applied. The results along with the changed paths of ground target are shown in figure 7.16, 7.17 and 7.19.

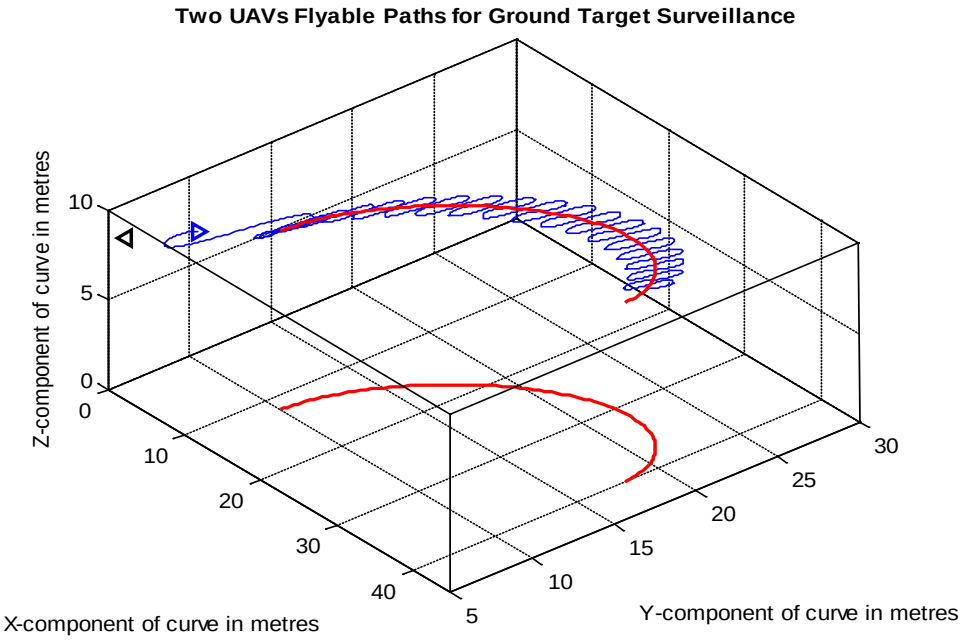

Figure 7.16: Path of the UAV1 (blue) to track the ground target (red)

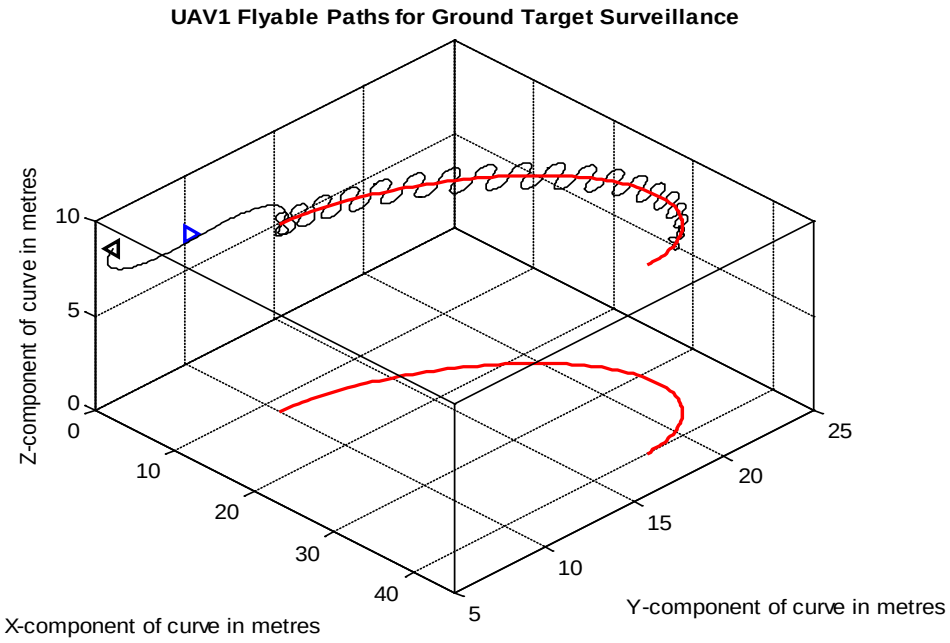

Figure 7.17: Path of the UAV2 (black) to track the ground target (red)

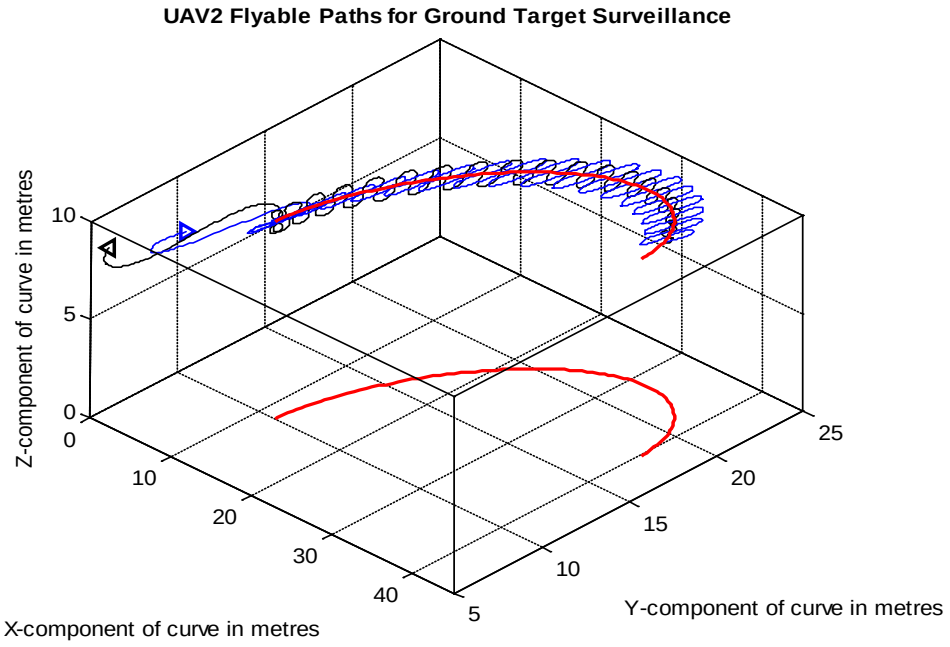

Figure 7.18: Cooperative paths of the UAV1 (blue) and UAV2 (black) to track the ground target (red)

7.4 Conclusions

The proposed algorithm was tested to track target traversing paths of different shape and found to yield satisfactory results. The proposed algorithm has the ability to adapt the length of the paths of the UAVs automatically to the speed of the target because every time the final pose of each tracking UAV is derived from the updated position of the target (calculated by perception system) by adding to it the height and orientation of the UAV. The algorithm also provides a full control over the overall path lengths of UAVs (for entire tracking) by controlling the rate at which the perception system passes the target position information to the path planning algorithm of the proposed method. Figure 7.19, 7.20, 7.21 shows the tracking path of UAV1 with increased rate of target positions acquisition by the perception system and passing it over to the path planning algorithm. This must be done sensibly within the speed limits of the UAVs.

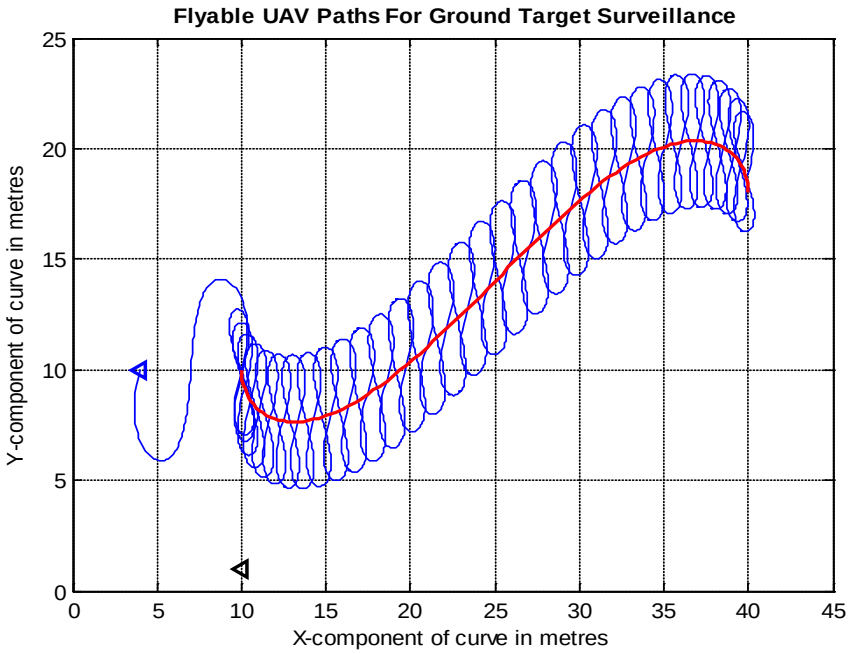

Figure 7.19: Path of the UAV1 (blue) to track the ground target (red)

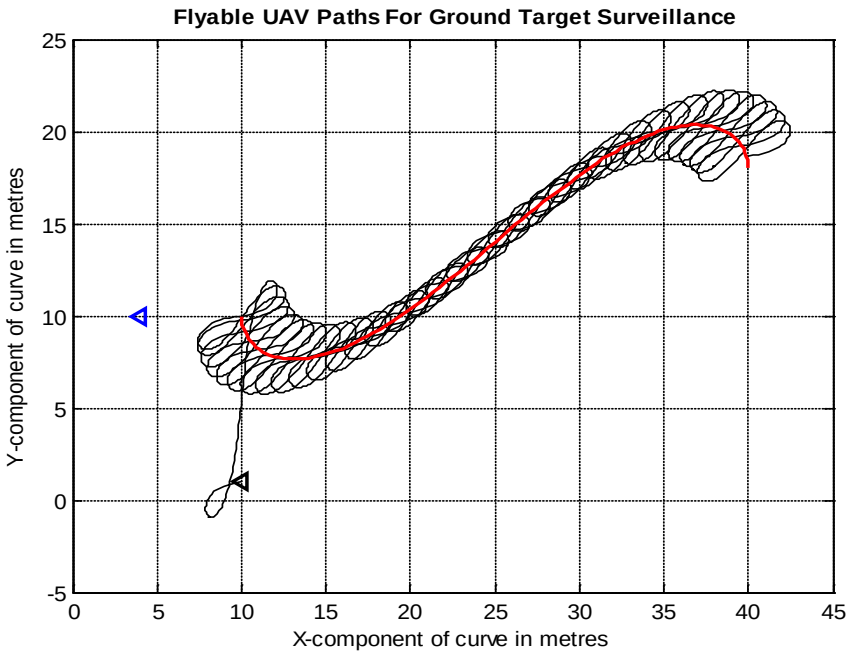

Figure 7.20: Path of the UAV2 (black) to track the ground target (red)

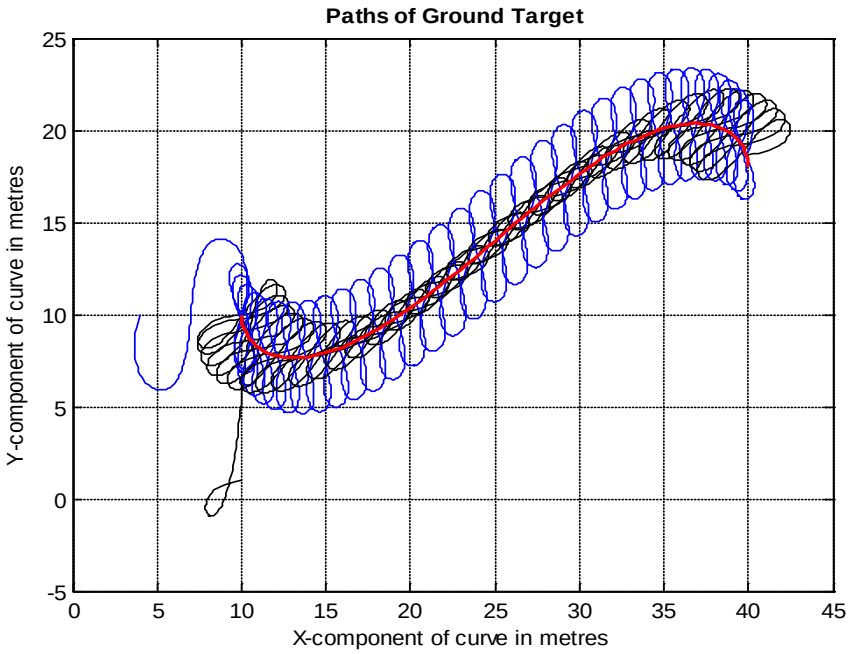

Figure 7.21: Cooperative paths of the UAV1 (blue) and UAV2 (black) to track the ground target (red)

Chapter 8

Conclusion and Further Work

In this research the path planning and cooperative perception for UAVs swarm was considered. The path planning in this research is based on Pythagorean Hodographs, and the cooperative perception is based on camera sensor. The operating environment of the swarm was assumed to be dynamic in which some popup obstacles could appear which may interrupt the UAV paths. More over it was assumed that some object of interest may appear in the environment, which may need to be tracked or invigilated. To perceive the popup obstacle or the object of interest dynamically each UAV platform is equipped with a perception system which consists of an onboard camera sensor and filtering scheme for estimation purposes. Each UAV uses its camera to watch the environment. The estimation algorithm uses the optimized camera output as a measurement in Kalman filter to estimate the obstacle/object position. The path planning and cooperative perception were then integrated for use in reactive obstacle avoidance and dynamic target tracking and surveillance.

In chapter 2, efficient feasible (flyable and safe) paths were produced using 2D Pythagorean hodograph based path planner as a starting point of this reasearch. The known obstacles in terrain database were successfully avoided during the offline path planning stage.

In chapter 3, the 2D PH paths were extended to 3D PH paths to accommodate the real challenges of path planning.

In chapter 4, an algorithm for calculating the Frenet frame of the PH paths derived in chapter 3 was developed. The algorithm calculates the Frenet frame at each point of the PH path. The Frenet frame was then converted to Rotation Minimizing frame. The Rotation Minimizing frame was used as camera reference frame in the perception system.

In chapter 5 camera based cooperative perception algorithm was developed to locate an object in 3D in the dynamic environment.

In chapter 6, the path planning of chapter 3 and 4, and the cooperative perception of chapter 4 and 5 were integrated to sense and avoid the popup obstacles along the trajectory. The cooperative perception was used to locate the obstacle and path planning was used to plan the evasive manoeuvre.

In chapter 7, the integration of path planning and cooperative perception was used for target tracking and surveillance. The cooperative perception was used to get the information about the position of the obstacle dynamically and the path planning was

used to maintain the close proximity of the target in order to keep it in the field of view of the camera.

Each algorithm developed in the course of this research is simulated. The simulation results are presented in a separate section at the end of the corresponding the chapter to prove the validity and efficiency of the technique.

In this thesis Chapters 6 and 7 are dedicated to the integration of path planning and cooperative perception for two different applications (obstacle avoidance and target tracking). The integration between these two totally diverse areas to achieve maximum UAV autonomy is a novel approach and a major contribution of this research. During the execution of the mission in dynamic environments the UAVs are faced with multiple challenges which develop in mid course, like, sense and avoid, continuous surveillance of an object or area, target tracking etc. These tasks are not predetermined or pre-assigned but they are situational and may develop mid course because of the dynamic nature of the environment. The employed UAVs must be able to deal with these challenges autonomously without the human intervention for the successful completion of the mission. To enable the UAVs to deal with such kind of situational developments autonomously is the primary purpose of this integration.

The mutual dependence of the path planning and cooperative perception is outlined as follows:

In case of obstacle avoidance, the perception system is needed to locate the popup obstacle cooperatively, and the path planner is needed to plan an evasive path in order to avoid it.

In case of surveillance and tracking of moving target the perception system is needed to give the waypoints based on the position of moving target at every time instant. The path planner is needed to plan chasing paths to maintain close target proximity by connecting the current pose of the UAV to the new waypoints based on the target location information given by the perception system.

This mutual need of the path planning and cooperative perception is satisfied by the integration of the path planning and cooperative perception.

The integration of the path planning and cooperative perception is a novel approach and major contribution of this research. The approach was successfully applied to avoid the popup threat and to track a moving target autonomously. However in the world of scientific research nothing is ultimate and final, there is always a room for further improvements. During this research, some areas of future research and improvement were identified. The areas where further investigation and research could be fruitful are discussed in the following section.

8.1 Future work

1. Transition from Differential Geometry to Linear Algebra for path planning:

The derivation of the PH paths is based on differential geometry. One area of improvement could be a possible shift from differential geometry to linear algebra and vector space in the context of path planning. A constraint vector space with vectors in the principal directions as basis vectors can be defined such that they satisfy the Pythagorean condition. The vector space is consistent with the implementation of state space based controller for guidance to trace the planned trajectory. If the state vector of the process is defined to be x, y, z, pitch, roll and yaw then the planned path of the UAV completely defines the state trajectory of the process.

2. The distance relationship between the control points location and the actual curve

The current technique of obstacle avoidance is to find the set of points of intersection of the curve enclosing the obstacle and the calculated UAV path. If the intersection is not empty, then the curvature of the path is manipulated till the intersection set is empty. This technique is iterative in nature. If the known control points give the exact whereabout of the curve it creates, then this can be used to eliminate the iterative process of finding the set of intersection of the path with the obstacle by choosing the control point such that the UAV path is clear of the known obstacle. By eliminating the iterative intersection process the computational load will decrease considerably and the algorithm will be quicker to respond. This will make the algorithm more suitable for dynamic environments.

3. Obstacle avoidance in 3D

In chapter 3 the dynamic obstacle avoidance is considered by restricting the evasive manoeuvre to 2D only. If this restriction of planar motion is removed then we could have more freedom in planning the evasive manoeuvre. This freedom can give us the ability to obtain length optimal paths. This will entail some further research to choose the most suitable initial orientation for the UAV that can yield length optimal path for evasive manoeuvre.

4. Obstacle avoidance in 3D using PH-Circle-PH combination

The current reactive obstacle avoidance algorithm can be improved further by PH-Circle-PH composite curve. In the current algorithm an obstacle free path is obtained by finding the iterative intersection set of the path and the curve enclosing the obstacle until an empty intersection set is achieved. The use of PH-Circle-PH combination in evasive manoeuvre will exclude the gradual iterative process of clearing of the path from the obstacle. This can enhance the performance of the algorithm.

5. Avoidance of an obstacle moving with speed comparable to that of UAV

In our research we have considered the popup obstacles which are stationary or moving very slow. It will be interesting if the fast moving obstacles are considered for reactive avoidance. Such a research will of course take into consideration the aviation regulation while undertaking the evasive manoeuvre. The future position estimation of the manoeuvring obstacle for the purpose of avoidance will be another interesting aspect of the research.

6. Using the cooperative perception for autonomous navigation

Object localisation in 3D by mean of vision based cooperative perception is the inverse operation of vision based navigation. In the cooperative perception the target was located cooperatively in 3D, while the positions of the UAVs were given by the GPS sensors. In navigation the target is replaced by known landmark and the cooperative perception is used to locate the UAV with respect to the landmark. There is need for further work to use the cooperative perception in automatic navigation to back up the GPS sensor in case of absence of GPS signals.

7. Maximising the combined field of view of the onboard cooperating cameras

In the operation of target surveillance and tracking, the target was located cooperatively. The cooperation in locating a moving target is possible if the target stays in the field of view of both UAVs. Since the target is kept by the UAVs in their field of view by planning the chasing manoeuvres, therefore there is a need for further research which maximise the common area in the fields of views of the UAVs by planning suitable chasing paths. This will increase the possibility of target being in field of view of both UAVs simultaneously.

8. Classification of PH paths, and replacing PH quintic with Cubic where possible

During this research the PH quintic paths are divided in S shape and C shape paths. S shape path can be produced by PH quintic while C shape path can be produced by PH quintic as well as cubics. The quintic paths are more flexible to manipulate but less easy to calculate. The C shape paths are less flexible to manipulate but more easy to calculate. Therefore the tasks which do not require lot of flexibility the quintic C shape path could be replaced by cubic C shape path. This will reduce the computational load considerably. There is a need for further research to classify the tasks into two groups: the group of tasks for which C shape paths are suitable and the group of tasks for which S shape path are suitable.

9. Search of free corridor.

The obstacle avoidance algorithm is able to avoid a cluster of small obstacles, if there is no free corridor among them, by assuming the aggregation to be one big obstacle, by the virtue of its ability to deal with the obstacle of any size. The search for free corridors among the obstacles however needs some further work. This could help in obtaining length optimal paths.

10. Covariance Intersection

The position estimates of the target obtained by the application of cooperative perception can be made more accurate if the covariance intersection approach is used. This needs some further work.

11. Optimal target acquisition rate in dynamic target tracking

In the ground target tracking, the chasing path of the UAV is extended when the new position information of the target is received. High rate of position information would mean more manoeuvres, which could lead to unattainable speed demand on the UAVs because of its speed limitation. Therefore some further investigation is needed to find a sensible way of deciding on the rate of position information that is passed to the path planner for planning chasing manoeuvres.

Appendix A

A.1 Quaternions

Quaternion is the elegant way to deal with PH curve of the higher orders. Introduced by an Irish mathematician William Hamilton in ninth century, Quaternion are four dimensional numbers having a scalar as well as a vector part [30]. It can be thought of as an extension of three dimensional vectors i.e. a three dimensional vector having a scalar component as well. Denoting by A and B the quaternion's can be represented as:

$$A = a + a_x i + a_y i + a_2 k \tag{A1}$$

And

$$B = b + b_x i + b_y i + b_2 k \tag{A2}$$

The quaternion's ''basis element'' *1, i, j, k* are governed by the following rules

$$i^2 = j^2 = k^2 = ijk = -1 \tag{A3}$$

$$ij = -ji = k_i, \quad jk = -kj = i, \quad ki = -ik = j$$

A.2 Quaternion's Algebra

A.2.1 Sum

The sum of the quaternion is performed component wise:

$$AB = (a + b) + (a_x + b_x)i + (a_y + b_y)j + (a_2 + b_2)k \tag{A4}$$

A.2.2 Product

The product of the Quaternion is performed as follow:

$$AB = (a + a_x i + a_y j + a_2 k)(b + b_x i + b_y + b_2 k)$$

$$\begin{aligned} &= ab + ab_x i + aab_y j + ab_2 k \\ &\quad - a_x b_x + a_x bi - a_x b_z j + a_x b_y k \\ &\quad - a_y b_y + a_y b_2 i + a_y b_j - a_y b_x k \\ &\quad - a_z b_z - a_2 b_y i + a_2 b_2 j + a_z bk \end{aligned}$$

$$\begin{aligned} AB = (ab - a_x b_x - a_y b_y - a_z b_z) &+ (ab_x + a_x b + a_y b_z - a_z b_y)i \\ &+ (ab_y + ab_y + a_z b_x - a_x b_z)j \\ &+ (ab_z + a_x b_y + ba_z - a_y b_x)k \end{aligned} \tag{A5}$$

If we denote $A = (a, \bar{a})$ where a is the real part and $\bar{a}$ is vector part and also $B = (b, \bar{b})$ then a convenient short hand for the above product AB is:

$$AB = (a, \bar{a})(b, \bar{b})$$

$$= (ab - \bar{a}.\bar{b}, \quad a\bar{b} + \bar{a}b + \bar{a} \times \bar{b}) \tag{A6}$$

Hamilton also consider the quaternion operator.

$$\nabla = \frac{\partial}{\partial x} i + \frac{\partial}{\partial y} j + \frac{\partial}{\partial z} k \tag{A7}$$

Taking the quaternion product of ∇ with a vector function $v = v_x i + v_y j + v_z k$ yields a quaternion $(-\nabla .v, \nabla \times v)$ in which the divergence $\nabla .v$ and curl $\nabla \times v$ of the vector field v are apparent in the scalar and vector parts.

A.2.3 Conjugate

Each quaternion $A=(a, \bar{a})$ has a conjugate $A^* = (a, -\bar{a})$ i.e. the conjugate of

$$A = a + a_x i + a_y j + a_z k \tag{A8}$$

is

$$A^* = a - a_x i - a_y j - a_z k \tag{A9}$$

A.2.4 Magnitude

The magnitude of a quaternion A is a non negative number $|A|$ given by

$$|A^2| = A^* A$$

$$= A A^*$$

$$= a^2 + |\bar{a}^2| \tag{A10}$$

One can easily verify the conjugate of quaternion product satisfy

$$(AB)^* = B^* A^* \tag{A11}$$

A.2.5 Inverse

In term of conjugate and magnitude we can specify the inverse:

$$A^{-1} = \frac{A^*}{|A|} \tag{A12}$$

A.2.6 Unit Quaternion

When $|A| = 1,$ A is a unit quaternion, in fact we can identify the unit quaternion with a point of unit sphere defined by

$$p^2 + {p_x}^2 + {p_y}^2 + {p_z}^2 = 1 \tag{A13}$$

A.2.7 Matrix and Quaternion

Quaternion can be represented by a real skew symmetric 4x4 matrix the form:

$$\begin{bmatrix} q & -q_x & -q_y & -q_z \\ q_x & q & -q_z & q_y \\ q_y & q_z & q & -q_x \\ q_z & -q_y & q_x & q \end{bmatrix}$$

If A and B are the matrices of the above form representing the quaternion, we can check that their matrix product yields a 4x4 skew symmetric matrix of the same form with the elements defined by the components of AB. Note that the transpose A^t represent the conjugate and $\det|A| = |A|.$ thus the unit quaternion are defined by the matrix of the above form with det(A) =1. If:

$$A = \begin{bmatrix} a & -a_x & -a_y & -a_z \\ a_x & a & -a_z & a_y \\ a_y & a_z & a & -a_x \\ a_z & -a_y & a_x & a \end{bmatrix}$$

And

$$B = \begin{bmatrix} b & -b_x & -b_y & -b_z \\ b_x & b & -b_z & b_y \\ b_y & b_z & b & -b_x \\ b_z & -b_y & b_x & b \end{bmatrix}$$

Then

$$AB = \begin{bmatrix} a & -a_x & -a_y & -a_z \\ a_x & a & -a_z & a_y \\ a_y & a_z & a & -a_x \\ a_z & -a_y & a_x & a \end{bmatrix} \begin{bmatrix} b & -b_x & -b_y & -b_z \\ b_x & b & -b_z & b_y \\ b_y & b_z & b & -b_x \\ b_z & -b_y & b_x & b \end{bmatrix}$$

$$= \begin{bmatrix} ab - a_x b_x + a_y b_y - a_z b_z & -ab_x - ba_x - a_y b_z + a_z b_y & -ab_y + a_x b_z - ba_y - a_z b_x & ab_z - a_x b_y + a_y b_x - a_z b \\ a_x b + ab_x - a_z b_y + b_y b_z & -a_x b_x + ab - a_z b_z - a_y b_y & a_x b_y - ab_z - a_z b + a_y b_x & -a_x b_z + ab_y + a_z b_x + a_y b \\ a_y b + a_z b_x + ab_y - a_x b_z & ab_z - a_x b_y + a_y b_x - a_z b & \ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots \\ ab_z + a_x b_y - a_y b_x + a_z b & -ab_y + a_x b_z - ba_y - a_z b_x & \ldots\ldots\ldots\ldots & \ldots\ldots\ldots\ldots \end{bmatrix}$$

$$\begin{aligned} &= (ab - a_x b_x - a_y b_y - a_z b_z) + (ab_x + ba_x + a_y b_z - a_z\ b_y)\, i \\ &\quad + (a_y b + a_z b_x + ab_y - a_x b_z)\, j \\ &\quad + (ab_z - a_x b_y + a_y b_x - a_z b)\, k \end{aligned} \tag{A14}$$

A.3 Quaternion and spatial rotation

A unit quaternion u has the form

$$u = [(\cos(\frac{1}{2}\theta),\ \sin(\frac{1}{2}\theta)n] \tag{A15}$$

For some angle $\frac{\theta}{2}$ and a unit vector $n = i + j + k$

$$\begin{aligned} |u|^2 &= \left|\cos(\frac{1}{2}\theta)\right|^2 + \left|\sin\frac{1}{2}\theta\right|^2 |(i^2 + j^2 + k^2|^2 \\ &= \cos^2\frac{\theta}{2} + \sin^2\frac{\theta}{2}.1 \\ &= 1 \end{aligned} \tag{A16}$$

If $\mathrm{v} = (0, \mathrm{v})$ is any pure vector, then quaternion product $\mathrm{u\,v\,u}^*$ also defines a pure vector, corresponding to a rotation of v through angle θ about the axis defined by n.

$$\begin{aligned} uv &= (\cos\theta,\ \sin\theta n)(0,\ v) \\ &= (0 - v\sin\theta n,\ \ v\cos\theta + 0 + v\times\sin\theta n) \\ &= (-v\sin\theta n,\ \ \ v\cos\theta + v\times n\sin\theta) \end{aligned}$$

$$uvu^* = (-v\sin\theta n,\ \ v\cos\theta + v\times n\sin\theta)(\cos\theta,\ \sin\theta n) \tag{A17}$$

To prove that for any quaternion A the basis vectors are Ai, Aj and Ak and hence any quaternion can be written in term of A, Ai, Aj, Ak.

By multiplying the matrix of basis vectors:

$$[A\ \ Ai\ \ Aj\ \ Ak] = \begin{bmatrix} a & -a_x & -a_y & -a_z \\ a_x & a & -a_z & a_y \\ a_y & a_z & a & a_x \\ a_z & -a_y & a_x & a \end{bmatrix}$$

Let:

$$A = (a, \quad a_x\ i, \quad a_y j, \quad a_z k)$$

$$\begin{aligned} Ai &= (a \quad a_x i \ \ a_y j \ \ a_z \ \ k)\, i \\ &= ai \ - a_x \ - a_y k - a_z j \\ &= - a_x + \ a_i \ - a_z j - a_y \ k \end{aligned}$$

Or by quaternion multiplication:

$$\begin{aligned} Ai &= (a \quad a_x i \quad a_y j \ \ a_z k)(0,\ i) \\ &= (0 + a_x i.i, \quad ai + 0 + (a_x i + a_y j + a_z k) \times i) \\ &= (a_x \ \ ai \ + a_z j \ \ - a_y k) \end{aligned}$$

$$Ai = -a_x + a\, i + a_z j - a_y k \tag{A18}$$

Similarly:

$$\begin{aligned} Aj \ &= (a \ \ a_x i \ \ a_y j a_z k)\, j \\ &= a_j \ \ \ a_x k \ - a_y \ - a_z i \end{aligned}$$

$$Aj = -a_y - a_z i + a\ j + a_x k \tag{A19}$$

$$\begin{aligned} Ak &= (a \quad a_x i \quad a_y j \ \ a_z k) k \\ &= ak - a_x j \ \ a_y i - a_z \end{aligned}$$

$$Ak = -a_z + a_y i - a_x j + ak \tag{A20}$$

Since the matrix of the basis vectors has its columns composed of basis vectors therefore it can be written as:

$$M = \begin{bmatrix} A & Ai & Aj & Ak \end{bmatrix}$$

$$A = \begin{bmatrix} a & -a_x & -a_y & -a_z \\ a_x & a & -a_z & a_y \\ a_y & a_z & a & -a_x \\ a_z & -a_y & a_x & a \end{bmatrix} \tag{A21}$$

Solution of $u\ i\ u^*$

$$\sqrt{\frac{1}{2}(1+\lambda)}\ \ [i + \frac{u}{1+\lambda} j + \frac{\lambda}{1+\lambda} k]$$

Proof:

Suppose for simplicity we choose co-ordinates so that first unit vector coincides with i the second unit vector v has a general orientation. We are interested in a quaternion solution u to the equation

$$u\ i\ u^{*} = v\ \mathrm{I} \tag{A22}$$

That specifies a spatial rotation of i i into v.

For any vector say v and quaternion A the form AvA^{*} always yield pure vector writing $u = u_o + u_x i + u_y j + u_z k$ and $v = \lambda i + \mu j + \nu k$ is equivalent to the system of three quadratic scalar equations

$$u_o{}^2 + u_x{}^2 - u_y - u_z{}^2 = \lambda \tag{A23}$$

$$2(u_o u_z + u_x u_y) = \mu \tag{A24}$$

$$2(u_x u_z - u_o u_y) = \nu \tag{A25}$$

in four unknowns (u_o, u_x, u_y, u_z).

Squaring the above three equations and adding gives after simplification:

$$(u_o{}^2 + u_x{}^2 + u_y{}^2 + u_z{}^2)^2 = \lambda^2 + \mu^2 + \gamma^2 = 1 \tag{A26}$$

Such that the unit quaternion is automatically satisfied. Putting $u_o = 0$ a particular solution to the above system of equation can be obtained as:

$$u_x{}^2 - u_y{}^2 - u_z{}^2 = \lambda \tag{A27}$$

$$2u_x u_y = \mu \tag{A28}$$

$$2u_x u_z = \gamma \tag{A29}$$

Implies:

$$u_y = \frac{\mu}{z u_x} \tag{A30}$$

$$u_z = \frac{\nu}{z u_x} \tag{A31}$$

Putting in equation (A27) and solving for u_x we get:

$$u_x = \sqrt{\frac{1}{2}(\lambda + 1)} \tag{A32}$$

$$u_y = \frac{\mu}{\sqrt{\frac{1}{2}(\lambda+1)}} \tag{A33}$$

$$u_z = \frac{\nu}{\sqrt{\frac{1}{2}(\lambda+1)}} \tag{A34}$$

Now putting these results of u_x, u_y and u_z in equation $u = u_x i + u_y j + u_z k$:

$$u = \sqrt{\frac{1}{2}(\lambda+1)}\,(i + \frac{\mu}{1+\lambda} j + \frac{\nu}{1+\lambda} k) \tag{A35}$$

more over if q is any quaternion satisfying the equation

$$q_i q^* = i \tag{A36}$$

Then u_q must also be a solution to equation (A36) above, since:

$$(u_q) i (u_q)^* = (uq) i (q^* u^*) \tag{A37}$$

$$= u q_i q^* u^* \tag{A38}$$

$$(u_q) i (u_q)^* = u_i u^* \tag{A39}$$

The quaternion q satisfying equation (A36) is necessary of the form:

$$q = \cos\phi + \sin\phi i \tag{A40}$$

As can be deduced from component equations of (A22):

$$q_o^{\,2} + q_x^{\,2} - q_y^{\,2} - q_z^{\,2} = 1 \tag{A41}$$

$$2(q_o q_z + q_x q_y) = 0 \tag{A42}$$

$$2(q_x q_z - q_o q_y) = 0 \tag{A43}$$

Putting $q_o = 0$

Squaring equations (A41), (A42), (A43) and summing:

$$(q_o^{\,2} + q_x^{\,2} - q_y^{\,2} - q_z^{\,2})^2 = 1 \tag{A44}$$

When we take

$$q = q_o + q_x i - q_y j - q_z k \tag{A45}$$

the most general solution to the equation

$$u\, i\, u^* = v \tag{A46}$$

Can thus be parameterized in term of the variable ϕ as

$$u(\phi) = \sqrt{\frac{1}{2}(1+\lambda)}\left(-\sin\phi + \cos\phi i + \frac{\mu\cos\phi + \nu\sin\phi j}{1+\lambda} + \frac{\nu\cos\phi - \mu\sin\phi}{i+\lambda}k\right)$$

We can gain better geometric by writing

$$u = (\cos\frac{1}{2}\theta,\ \sin\frac{1}{2}\theta n). \tag{A47}$$

The rotation axis $n(n_x, n_y, n_z)$ and the angle θ must then satisfy:

$$n_x{}^2(1-\cos\theta) + \cos\theta = \lambda \tag{A48}$$

$$n_x n_y(1-\cos\theta) + n_z\sin\theta = \mu \tag{A49}$$

$$n_z n_x(1-\cos\theta) - n_y\sin\theta = \nu \tag{A50}$$

Writing $\alpha = \cos\lambda,$ for $(\alpha \le \theta \le 2\pi - \alpha)$ then the general solution

$$n_x = \frac{\sqrt{\cos^2\frac{\alpha}{2} - \cos^2\frac{\theta}{2}}}{\sin\frac{\theta}{2}} \tag{A51}$$

$$n_y = \frac{\sqrt{\cos^2\frac{\alpha}{2} - \cos^2\frac{\theta}{2}}}{\sin\frac{\theta}{2}} \tag{A52}$$

$$\mathrm{n}_z = \frac{\sqrt{\cos^2\frac{\alpha}{2} - \cos^2\frac{\theta}{2}}}{\sin\frac{\theta}{2}} \tag{A53}$$

References:

[1] A. Kolushev, Alexander A. Bodanov. Multiple-Agent Optimal Path Planning for Mobile Robot in Environment with Obstacles. Proceedings of the third international Andrei Ershov memorial conference on Perspective System Informatics, PSI 99.
http://citeseer.ist.psu.edu/666991.html

[2] Zbigniew Tarapata. Military Route Planning in Battle Field Simulation, Effectiveness Problem and Potential Solution. Proceeding of 4th WSEAS international conference on Computer Engineering and Application CEA 10. Pp. 47-56 2003.
http://www.tarapata.strefa.pl/publikacje/jtit_2003.pdf

[3] P. E. Hart, N. J. Nilsson. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. Syst Sci Cybern Vol. SSC-4 No. 2, PP100-107, 1968.

[4] D. H. Yang, S. K. Hong. Road Map Construction Algorithm for Mobile Robot Path Planning using Skeleton Map. Advance Robotic Vol.21 No 1-2 PP. 51-63 (2007)

[5] Ganish Swaminathan. Robot Motion Planning. Technical Report, School of Engineering ScienceSimon Fraser University Canada. Feb 27, 2007
http://ergodicity.iamganesh.com/uploads/ensc400-gsw-mpreport.pdf

[6] Alade Tokuta. Extending the Visibility Graph Algorithm for Robot Path Planning. Technical Report Department of Mathematics and Computer Science, North Carolina Central University.
http://wscg.zcu.cz/wscg98/papers98/Tokuta_98.pdf

[7] R. R. D. Santos, Valder Steffen Jr. Robot Path Planning in Constrained Work Space using Optimal Control Techniques. Multibody System Dynamics Volume 19 Page 159-177.

[8] D. Constantinescu, E.A. Croft. Smooth and Time Optimal Trajectory Planning for Industrial Manipulators along Specified Paths. J. Robotics syst. Volume 17, Issue 5, Pages 223-249, May 2000.

[9] W. Gerke. Collision Free and Shortest Paths Found by Dynamic Programming. Robotersysteme 1, 43-52 (1985).

[10] M. Vukdbralovic, M. A. kircanski. Method for Optimal Synthesis of Manipulations Robot Trajectories. Journal of Dynamic Systems and Control. Volume 104 188-193 (1982)

[11] F. Pfeifer, R. A. Johanni. Concept for Manipulator Trajectory Planning. IEEE Trans. Robotic Automat 3(3), 115-123 (1987)

[12] S. Singh, M. C. Leu. Optimal Trajectory Generation for Robot Manipulators Using Dynamic Programming. J. Dyn. Sys. Meas Control 109 (2), 88-96 (1987)

[13] M. A. Lee, H. Takagi. Integrating Design Stages of Fuzzy Systems Using Genetic Algortithms. Proc. 2nd IEEE Int. Conf. Fuzzy Systems, San Francisco, 1993; 612-617

[14] K. Belarbi, F. Titel. Genetic algorithm for the design of a class of fuzzy controllers: An alternative approach. IEEE Transactions on Fuzzy Systems 2000, Vol 8, Iss 4, pp 398-405.

[15] D. E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, 1989.

[16] D. Whitley, "A Genetic Algorithm Tutorial" Technical Report CS-93-103, Dept. of Computer Science, Colorado State University, Nov. 1993.

[17] L. E. Dubins. On curves of minimal length with a constraint on average Curvature and with prescribed initial and terminal positions and tangents. American Journal of Mathematics, 79:497– 516, 1957.

[18] M. Hwangbo, J. Kuffner, T. Kanade, Efficient 3D Motion Planning for Small Fixed Wing UAVs. IEEE International Conference on Robotics and Automation Roma, Italy, 10-14 April 2007.

[19] X. N. Bui, P. Soueres, J. D. Boissonat and J. P. Laumond. Shortest Path Synthesis for Dubins Non Holomonic Robot. In Proc. IEEE Int'l Conf. on Robotic and Automation1994.

[20] S, Madhavan, A. Tsourdos, B. White. Differential Geometric Path Planning of Multiple UAVs. Journal of Dynamic System, Measurement and Control, vol. 129 September 2007.

[21] E. W. Weisstein. Cornu Spiral. From Maths World Web Resource http://mathworld.wolfram.com/CornuSpiral.html

[22] R. Dai, John E. Cochran Jr. Path Planning for Multiple Unmanned Aerial Vehicles by Parameterized Corno-Spiral. American Control Conference Hyatt Regency Riverfront St. Louis, MO, USA June 10-12, 2009.

[23] S. Madhavan, A. Tsourdos, B. White. A Solution to Simultaneous Arrival of Multiple UAVs using Pythagorean Hodograph. Proceeding of American Control Conference Minneapolis, Minnesota USA, June 14-16 2006.

[24] S. Subchan, B. white, A Tsourdos. Pythagorean Hodograph Path Planning for Tracking Airborne Contaminant using Sensor Swarm. IEEE International Instrumentation Technology Conference, Victoria Vancouver Canada, May 12-15, 2008.

[25] Armando A. Neto, Mario F. M. Campos. A Path Planning Algorithm for UAVs with Limited Climb Angle. IEEE International Conference on Intelligent Robots and Systems October 11-15 2009 St. Louis, USA.

[26] Armando Alves Neto, Douglas G. Macharet, Mario F. M. Campos. On Generation of Trajectories for Multiple UAVs in Environment with Obstacles. Journal of Intelligent & Robotic Systems, Volume 57, no. 1-4, pp. 123-141.

[27] T. R. Farouki, T. Skallis. Pythagorean Hodograph. IBM J. RES. Develop. Vol. 34 No. 5 September, 1990.

[28] R. T. Farouki, C. A. Neff. Hermite interpolation by Pythagorean hodograph quintics. Mathematics of Computation, v.64 n.212, p.1589-1609, 1995

[29] H.P. Moon, R.T. Farouki, and H.I. Choi. Construction and shape analysis of PH quintic Hermite interpolants. Comput. Aided Geom. Design. 18 (2001) 93-115.

[30] R.T. Farouki. The elastic bending energy of Pythagorean hodograph curves. Comput. Aided Geom. Design 13 (1996) 227-241.

[31] H. Bruyninckx, 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.

[32] R. T. Farouki. Pythagorean-Hodograph Curves: Algebra and Geometry Inseparable. Springer-Verlag Berlin Heidelberg 2008.

[33] R. T. Farouki, Mohammad al-Kandari, T. Sakkalis. Hermit interpolation by rotation-invariant spatial Pythagorean hodograph. Advances in Computational Mathematics, 17:369-383, 2002.

[34] Rida T. Farouki, Chang Yong Han. Algorithm for Spatial Pythagorean Hodograph Curves.
http://mae.ucdavis.edu/~farouki/phspace.pdf

[35] R.T. Farouki, T. Sakkalis, “Pythagorean-hodograph space curves”, Adv. Comput.Math.2 (1994)41-66.

[36] K. Shoemake. Animating Rotation with Quaternion Curves. ACM SIGGRAPH, 19(3): 245-254, 1985.

[37] John C. Hart. George K. Francis, Louis H. Kauffman. Visualising Quaternion Rotation. ACM Transactions on Graphics, Vol. 13, No. 3, July 1994, Pages 256-276.

[38] Berthold K. P. Horn. Closed form Solution to Absolute Orientation using Unit Quaternions. Journal of Optical Society of America, Volume 4, 629-642(1987).

[39] H. I. Choi, D. S. Lee and H. P. Moon. Clifford algebra, spin representation, and rational parameterization of curves and surfaces. Adv. Comput. Math. 17, 5-48

[40] Z. Duric, A. Rosenfield, L. S. Davis. Egomotion analysis based on Frenet Serret motion model. International Journal of Computer Vision Archive. Volume 15, Issue 1-2 (June 1995), pp 105-122.

[41] John J. Craig. *Introduction to Robotics Mechanics & Control*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA – 1989

[42] R. T. Farouki. Exact Rotation Minimizing Frames for Spatial Pythagorean-Hodoraph Curves. Graphical Model 64(2003) 382-395.

[43] H. Guggenheimer. Computing Frames along a Trajectory. Computer Aided Geometric Design 6 (1989) 77-78 North-Holland.

[44] Zoran Duric, Ehud Rivilin, Azriel Rosenfield. Understanding Object Motion. Image and Vision Computing, Volume 16, Number 11, 1 August 1998, pp. 785-797(13)

[45] Liangyin Yu. Active 3D Surface Modelling Using Perception-Based Differential- Geometry Primitive. PhD thesis, University of Wisconsin – Madison 1999.
http://ftp.cs.wisc.edu/computer-vision/repository/PDF/yu.1999.thesis.pdf

[46] K. Gould, M. Shah. The trajectory primal sketch: A multi-scale scheme for representing motion characteristic. In Proc. Conference on Computer Vision and Pattern Recognition, 1989, pp 79-85.

[47] T. J. Broida, R. Chellapa. Estimation of object motion parameter from noisy images sequences. IEEE Transaction on Pattern Analysis and Machine Intelligence 9 (1987) 370-389.

[48] J. Weng, T. S. Huang, N. Ahuja. 3-D motion estimation, understanding and prediction from noisy image sequences. IEEE Transaction on Pattern Analysis and Machine Intelligence 9 (1987)370-389

[49] C. Cedras, M. shah. Motion based recognition: A survey. Image and Vision Computing. Volume 13, no 2 (1995)129-155.

[50] S. Engel, J. Rubin. Detecting visual motion boundaries. In proc. workshop on motion (1986), pp. 107-111.

[51] M. Hwangbo, J. Kuffner, Takeo Kanade. Efficient Two Phase 3D Motion Planning for Small fixed Wing UAVs. Robotics and Automation, IEEE International Conference April 2007, pp. 1035-1041.

[52] T. Kanade, O. Amidi and Q. Ke. Real-time and 3D vision for autonomous small and micro vehicles. IEEE Conference on Decision and Control (2004), pp. 1655-1662.

[53] B. Sinopoli, M. Michelli, G. Donata and T. Koo. Vision based navigation for an unmanned Aerial vehicle. In Proceeding IEEE International Conference on Robotic and Automation 2001, Vol. 2, pp.1757-1764.

[54] R. Zapata, P. Lepinay. Flying among obstacles. In European Work Shop on Advance Mobile Robots, Sept. 1999, pp. 81-88.

[55] Brett. Bethke, Mario. Valenti, Jonathan How. Cooperative Vision Based Estimation and Tracking Using Multiple UAVs. Proceedings of the Conference on Cooperative Control and Optimization Vol: 369/2007 page: 179-189.

[56]- N. Gordon, B. Ristic, S. Arulampalam. Beyond kalman filter: particle filters for tracking apptications. Artech House Boston, 2004.

[57] C. sharp, O. Shakernia, S. Sastry. A vision system for landing an unmanned aerial vehicle. In Proceeding of the IEEE International Conference on Robotic and Automation vol. 2, pages 1720-1727(2001).

[58] D. Casbeer, S. Li, R. Beard, R. Mehra, T. Mclain. Forest fire monitoring with multiple small UAVs. Proceeding of American Control Conference 2005, pp. 3530-3535, Porland, Oregon.

[59] E. Wan, R. Van, Der Merwe. The unscented Kalman filter for non linear estimation. IEEE Adaptive System for Signal Processing, Communication and Control Symposium, Alta Canada, Oct 2000, AS-SPPC. Pp.153-158.

[60] G. Goebel. In public domain unmanned aerial vehicles. http://www.vectorsite.net/twuav.html

[61] L. Merino, F. Caballero, J. R. Martinez de Dios, A. Allero. Cooperative fire detection using unmanned aerial vehicles. In Proceedings of the IEEE International Conference on Robotic and Automation, Barcelona Spain April 2005, pp. 1884-1889.

[62] L. Merino, F. Caballero, J .R. Martinez De Dios, J. Ferruz, A. Allero. A cooperative perception system for multiple UAVs: Application to automatic detection of forest fire. Journal of Field Robotics 23: 165-184, 2006.

[63] S. Jullier, J. Uhlmann. A new extension of Kalman filter to non linear system. In Proceedings of 11th International Symposium on Aerospace/Defence Sensing, Simulation and Control, 1997.

[64] T. McGee, R. Sengupta, K. Hedrick. Obstacles detection for small autonomous aircraft using sky segmentation, In Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona Spain April 2005.

[65] Feili Hou, Feng Zhu**.** A Filter Method for Pose Estimation of Maneuvering Target. IEEE International Conference on Intelligent Robots and Systems (2004), Vol. 3, pp. 2396-2403.

[66] Wei Xiong, Yi Jin, Ming Xie. Effective and Efficient 3D Object Location using Automatic Camera Calibration. http://citeseer.ist.psu.edu/464831.html

[67] Mark E. Cambell, William W. Whitacre. ‘’Cooperative Tracking Using Vision Measurement on Sea scan UAVs’”, IEEE Transaction on Control Systems Technology Vol. 15, No. 4 (July 2007), pp. 613-626.

[68] Luis Merino, J. Wiklund, C. Fernando, Anders Moe. Vision Based Multi-UAV Position Estimation, Localization Based on Blob Feature for Exploration Mission. Robotics & Automation Magazine, IEEE, Vol. 12, Page 46-57(2005).

[69] Erica Z. MacArthur, Donald MacArthur, Carl Crane. Use of Cooperative UAV and UGV for Detection and Disposal of Simulated Mines. Proceeding of the SPIE, Vol. 5999, pp. 94-101(2005).

[70] Ricardo Fabbri, Benjamin B. Kimia. High-Order Differential Geometry of Curves for Multiview Reconstruction and Matching. Lecture Notes in Computer Science 2005, No. 3757, pages 645-660.

[71] Olivier Faugeras, Three-Dimensional Computer Vision - A Geometric Viewpoint. MIT Press, Cambridge MA, USA (1993)

[72] G. Li, S. W. Zucker. A differential geometrical model for contour-based stereo correspondence. In Proc. IEEE Workshop on Variational, Geometric, and Level Set Methods in Computer Vision, Nice, Franc (2003).

[73] L. Quan. Conic reconstruction and correspondence from two views. IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (1996) 151-160.

[74] T. Papadopoulo, O. D. Faugeras. Computing structure and motion of general 3D curves from monocular sequences of perspective images. In Proceeding of the 4th European Conference on Computer Vision, London, UK, Springer-Verlag (1996) 696-708.

[75] S. D. Ma, L. Li. Ellipsoid reconstruction from three perspective views. In Proceedings of the 13th International Conference on pattern Recognition, Vienna, Austria. Vol. 1, (1996), pp. 344-348.

[76] R. Berthilsson, K. Astrom, A. Heyden. Reconstruction of general curves, using factorization and bundle adjustment. International Journal of Computer Vision, Vol. 41, No. 3, (2001), pp. 171-182.

[77] J. Y. Kaminski, A. Shashua, Multiple view geometry of general algebraic curves. International Journal of Computer Vision, Vol. 56, No. 3, (Feb 2004), pp. 195-219.

[78] M. Spetsakis, J. Y. Aloimonos. A multiframe approach to visual motion perception. Int. J. Comput. Vision, Vol. 6, Issue 3, (1991), pp. 245-255.

[79] R. I. Hartley. A linear method for reconstruction from lines and points. In Proceedings of the Fifth International Conference on Computer Vision, Boston, Mas-sachusetts, IEEE Computer Society Press (1995) 882-887.

[80] L. Robert, O. D. Faugeras. Curve-based stereo: figural continuity and curvature. In Proceedings of IEEE Conference on Computer Vision and pattern Recognition. (1991) 57-62.

[81] C. Schmid, A. Zisserman. The geometry and matching of lines and curves over multiple views. International Journal of Computer Vision, Vol. 40 (2000). Pp. 199-233.

[82] L. Robert, R. Deriche. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. In Fourth European Con-ference on Computer Vision, Cambridge, England, Springer Verlag (1996) 439-451.

[83] F. Kahl, J. August. Multiview reconstruction of space curves. In Proceedings of the 9th IEEE International Conference on Computer Vision, (2003) Vol. 2, pp. 1017-1024.

[84] D. Mumford. Elastica and computer vision. Algebraic Geometry and Its Applications, Springer-Verlag (1994) 491-506.

[85] S. W. Zucker. Differential Geometry from the Frenet Point of View: Boundary Detection, Stereo, Texture and Color. Mathematical Models of Computer Vision, Springer (2005) 361-376.

[86] S. W. Alibhai. Contour-based correspondence for stereo. In Proc. Sixth European Conf. on Computer Vision, Dublin, Ireland (2000)

[87] R. Arnold, T. Binford. Geometric constraints in stereo vision. In Proc. SPIE. Volume 338-Image Processing for Missile Guidance, San Diego, CA (1980) 281-292

[88] D. Sherman, S. Peleg. Stereo by incremental matching of contours. IEEE Trans. On pattern Analysis and machine Intelligence, Vol. 12, Issue 11, (1990), pp. 1102-1106

[89] D. Marr, T. Poggio. A theory of human stereo vision. Proceedings of the Royal Society of London, Series B, Biological Sciences, Vol. 204, No. 1156, (May 23 1979), pp. 301-328.

[90] W. E. L. Grimson. A Computer Implementation of a Theory of Human Stereo Vision. Royal Society of London Philosophical Transactions Series B, Vol. 292 (1981), pp. 217-253.

[91] S. B. Pollard, J. E. W. Mayhew, J. P. Frisby. A stereo correspondence algorithm using a disparity gradient limit. Perception 14 (1985) 449-470.

[92] G. Welch, G. Bishop. An Introduction to the Kalman Filters. TR 95-041 Dept. Of Computer Science University of North Carolina Chapel Hill NC 27599-3175.

[93] Dan Simon. *Optimal State Estimation: Kalman, H-infinity and Non Linear Approaches.* John Wiley & Sons Inc, 2006.

[94] Emanuele Trucco, Alessandro Verri. *introductory techniques for 3-D Computer Vision* Prentice Hall, 1998.

[95] V. Duindam, J. Xu, R. Alterovitz, S. Sastry, K. Goldberg. 3D Motion Planning Algorithms for Steerable Needle Using Inverse Kinematics. International Journal of Robotics Research, 29(7), pp. 789-800, June 2010.

[96] J. Connors, G. Elkaim. Analysis of a spline based obstacle avoiding path planning algorithm. IEEE 65th Vehicular Technology Conference, 2007. VTC2007-Spring.

[97] Kwangjin Yang, Salah Sukkarieh. An Analytical continuous curvature path Smoothing Algorithm. IEEE Transaction on Robotics, Vol. 26, Issue: 3, pp 561-568.

[98] E. Papadopoulos, I. Papadimitriou, I. Poulakakis. Polynomial based obstacle avoidance techniques for nonholonomic mobile manipulator system. Journal of Robotic and Autonomous System Vol. 51, No. 4, pp. 229-247, 2005.

[99] E. Papadopoulos, I. Poulakakis. Planning and obstacle avoidance for mobile robots. Proceeding IEEE International Conference on Robotics and Automation, ICRA 01, May 21-26 2001, Seoul Korea.

[100] K. G. Jolly, R. S. Kumar, R. Vijaykumar. A Bezier curve based path planning in multi Agent robot soccer system with out violating the acceleration limits. Robotics and Autonomous System 2009, Volume 57,Issue 1, Page: 23-33.

[101] Ran Dai, John E. Cochran Jr. Path planning and state estimation for unmanned aerial vehicles in hostile environment, Journal of Guidance Control and Dynamics" Vol. 33, No. 2, March –April 2010.

[102] Tao Dong, X. H. Liao, R. Zang, Zhao Sun, Y. D. Song. Path tracking and obstacle avoidance of UAVs fuzzy logic approach. The 14th IEEE International Conference on Fuzzy Systems 2005, FUZZ '05. May 25, 2005.

[103] Yoko Watanabe, Anthony J. Calise, Eric N. Johnson. Vision based obstacle Avoidance for UAVs. AIAA Guidance, Navigation and Control Conference and Exhibit 20-23 August 2007, Hilton Head, South Carolina.

[104] R. W. Penny. Collision avoidance within flight dynamics constraints for UAV application. Aeronautical Journal, 109(1094):193-199, April 2005

[105] P. B. Sujit, R. Beard. Multiple UAV Path Planning Using Anytime Algorithm. Proceeding of 2009 Conference on American Control Conference St. Louis, Missouri, USA Pages: 2978-2983.

[106] G. Ducard, K. C. Kulling, H. P. Geering. A simple adaptive on-line path planning system for UAVs. Mediterranean Conference on Control & Automation, 2007. MED '07 Athens Greece, 27-29 June 2007, Pages 1-6

[107] Su-Cheol Han. Hyochoong Bang. Proportional navigation based optimal collision avoidance for UAVs. 2nd International Conference on Autonomous Robot and Agents December 13-15, 2004 PalmerstonNorth, New Zealand.

[108] C. R. McInnes. Velocity field path planning for single and multiple unmanned aerial vehicles. Aeronautical Journal, 107(1073). Pp. 419-426 ISSN 0368-3931, July, 2003.

[109] S. Lavelle. *Path planning algorithm*, Cambridge University Press, 2006.

[110] J. Lee, R. Haung, A. Vaughan, X. Xiao, J. K. Hedrick, M. Zennaro, R. Sengupta. Strategies of Path Planning for a UAV to Track a Ground Vehicle. In AINS Conference 2003.

[111] H. Chen, K. Chang, C. S. Agate. Tracking With UAVs Using Tangent Plus Lyapunov Vector Field Guidance. 12th International Conference on Information Fusion Seattle, WA, USA, July 6-9, 2009.

[112] G. F. Ivey, E. N. Johnson. Investigation of Methods for Target State Estimation using Vision Sensors. AIAA Guidance, Navigation, Control Conference and Exhibit, 15-18 Auqust 2005, San Francisco California.

[113] F. Rafi, S. Khan, K. Shafiq, M. Shah. Atonomous Target Following by Unmanned Aerial Vehicles. SPIE Defence and Security Symposium 2006, Orlando FL.