3D Path Planning of UAV using Model Predictive Control

Student Name: Rahul Singhal

IIIT-D-MTech-ECE-13-1321

Jul 30, 2015

Indraprastha Institute of Information Technology

New Delhi

Thesis Committee

Dr. P. B. Sujit (Advisor)

Dr. A. V. Subramanyam (Internal)

Dr. Vikram Gupta (External)

Submitted in partial fulfillment of the requirements

for the Degree of M.Tech. in Electronics & Communication Engineering,

in General

Keywords: UAV, Path Planning, Model Predictive Control, 3D terrain , 3D Vertical obstacle, Threat avoidance

Certificate

This is to certify that the thesis titled **"3D Path Planning of UAV using Model Predictive Control"** submitted by **Rahul Singhal** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Electronics & Communication Engineering* is a record of the bonafide work carried out by him under my guidance and supervision in the Security and Privacy group at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. P. B. Sujit

Indraprastha Institute of Information Technology, New Delhi

Abstract

Path planning is essential for UAVs that travel in different terrains. Designing feasible path planners for UAVs taking the dynamic constraints is difficult. In this thesis, we develop a 3D path planner using model predictive control (MPC) methods for a quadrotor. The MPC based controller optimizes the control effort required to travel a given reference path. The obtained path avoids infeasible regions of the terrain and also threats. The MPC takes constraints on jerks, position and angular acceleration on the quadcopter. Simulation results are presented to validate our approach.

Acknowledgments

I would like to first thank my adviser Prof. P.B Sujit, I am truly grateful for providing me excellent guidance, insight, attention and encouragement. I am also thankful to IIIT-Delhi for providing excellent infrastructure and support. Last but never the least, I am immensely grateful to my parents, family members and close friends, for their invaluable support and unconditional love.

Rahul Singhal

Contents

1 Introduction **2**

2 UAV Model **4**

2.1 Discrete time Model of UAV 5

3 Model Predictive Control for UAV **7**

3.1 Objective function 7

3.2 Feasible constraints condition 8

3.3 Model Parameters 8

4 2D Path Planning and Control of UAV **10**

4.1 Trajectory Evaluation 10

4.2 Results 11

4.2.1 VTP for straight line & Circular Path planning 11

4.2.2 NLGL for straight line, Circular Path & Sine Wave 12

5 3D Path Planning and Control of UAV for vertical obstacles **15**

5.1 Trajectory Evaluation 15

5.1.1 Node Selection 15

5.1.2 Visibility Graph 16

5.1.3 Cost Function 17

5.2 Result 18

6 3D Path Planning and Control of UAV for 3D Terrain 19

6.1 Trajectory Evaluation 19

6.1.1 Visibility node chart 19

6.1.2 Cost Map 20

6.1.3 Smallest weighted Path 21

6.2 Results 22

6.3 3D Path Planning and Control of UAV for 3D Terrain with threat avoidance 23

6.4 Results 23

7 Conclusion & Future Directions 26

List of Figures

2.1 Inertial frame I and Body frame B of quadcopter 4

4.1 Determining the virtual target point (VTP) for NLGL. The VTP for (a) the straight-line path and (b) the loiter path [1] 11

4.2 VTP algorithm for Straight Line following 12

4.3 VTP algorithm for circular orbit 13

4.4 NLGL algorithm for straight line as desired path in green & path in red color traced by UAV 13

4.5 NLGL algorithm for circular orbit as desired path in green & path in red color traced by UAV 13

4.6 NLGL algorithm for sine wave as desired path in green & path in red color traced by UAV 14

5.1 * as desired nodes and dotted red line a selected visible path 17

5.2 Desired 3D trajectory in black dots, UAV with MPC in red line 18

6.1 Neighbor visibility graph, o represents the neighbor to x node. 20

6.2 shows different smallest weighted path between start node (*) and goal node (o) for $\alpha = 2$ in blue line, $\alpha = 0.2$ in magenta line & $\alpha = 0$ in red line 21

6.3 shows smallest weighted path in blue line between start node (*) and goal node (o), red line shows trajectory followed by MPC for $v = 2\ ms^{-1}$ 22

6.4 shows the value of distance error and jerk of quadcopter for $v = 2\ ms^{-1}$ . . 23

6.5 shows the value of distance error and acceleration of quadcopter for $v = 2\ ms^{-1}$ 24

6.6 Top view of threat avoidance by UAV 24

6.7 Other view of threat avoidance by UAV 25

List of Tables

3.1 Real Time constraints value 9

4.1 Parameter value for VTP & NLGL 12

Chapter 1

Introduction

Quadcopter is an active area research as they have exceptional agility, mechanical simplicity and robustness. With extraordinary rotational dynamics they achieve very high thrust to weight ratio resulting in large translation acceleration.

This platform has already attracted great interest from academia as well as industries, as they are very effective in real world problem solving such as surveillance [2], agriculture for remote sensing & crop monitoring [3], [4],[5] or image based surveillance and decision [6], culvert inspection [7] and mapping [8].

Some ways for path planning implementation with RHC are: trajectory generation for urban terrain[9], Artificial Potential Field(APF) with RHC [10] ,[11], large range constrained trajectory [12], decentralized RHC [13] & multi-task allocation & path planning for cooperative UAV [14].

Different vehicle guidance algorithms have been suggested for air vehicle control [15],[16], ground vehicle and UAVs [1] for a predefined path in three or two dimensions at constant height.

In general they have to follow predefined path such as straight line or circular orbits.

Path evaluation in 3D with vertical objects is suggested [17]. Selection process for visible nodes is based on fact that shortest desired node would rarely choose nodes at the corners [18]. Desired visible nodes are midpoint of edges having a relative height greater than ground surface. Different other nodes could be chosen on edge for improved accuracy of smallest weighted path but design expands very rapidly with small improvements in accuracy and leads to very large computation [18]. Visible graph is evaluated using MILP for verification of the path not passing through vertical obstacle. The cost function penalizes based on the distance between the nodes and their relative heights. Then the smallest weighted path is obtained using Dijkstra algorithm [12]. Path is followed by

solving convex optimization problem based on receding horizon control with constraints.

The model predictive control solves quadratic cost function with linear equality constraints and affine inequality constraints falls under subset of convex optimization problem. Different tools are available to solve convex optimization problems are CVXOPT [19], FORCES [20], CVX [21],[22] and CVXgen [23] which also generates C code.

3D terrain is generation as suggested [24]. Followed by visibility nodes chart preparation along with cost map are required as input to the Dijkstra algorithm to evaluate the smallest weighted path between the start node (*) and goal node (o) based on the weights assigned in cost map for a node pair. As quadcopter should avoid threads and radar, their path should be closer to surface so cost function penalizes euclidean distance and also relative heights of visible nodes and current node. Model predictive control follows a path with constraints on acceleration, positions and jerks. MPC is implemented using CVX package for specifying and solving convex programs [21], [22] on MATLAB.

Chapter 2

UAV Model

The Quadcopter model has six degrees of freedom (DoF) : three DoF are linear translations along inertial x, y and z axis and rest three DoF are rotational translation along the body frames with respect to the inertial axis. U is a rotational matrix such that when multiplied with a vector ${}^{b}\mathrm{r}$ in body frames would yield resulted vector ${}^{i}\mathrm{r}$ in internal frame (2.1).

$$ {}^{i}\mathrm{r} = {}^{i}_{b}\mathrm{U}.{}^{b}\mathrm{r} \tag{2.1} $$

Quadcopter have rotational rate $(\omega_x, \omega_y, \omega_z)$ about the body frames. These are illustrated in figure (2.1), [25] suggested mixing of $(\omega_x, \omega_y, \omega_z)$ to motor commands.

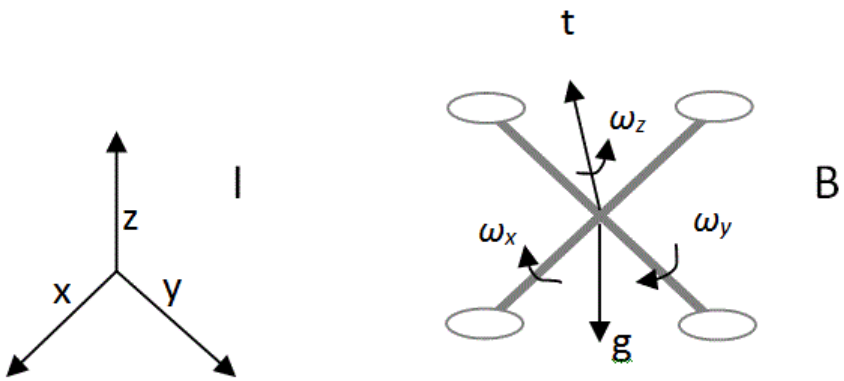

Figure 2.1: Inertial frame I and Body frame B of quadcopter

The change in quadcopter altitude in inertial frame is related to the rotational control

inputs described by [26] as

$$ {}^{i}_{b}\dot{\mathrm{U}} = {}^{i}_{b}\,\mathrm{U} \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.2} $$

The total thrust t produced which is further controlled by the differential equation governing the translation acceleration with respect to an inertial frame [27] described as

$$ \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = {}^{i}_{b}\,\mathrm{U} \begin{bmatrix} 0 \\ 0 \\ \mathrm{t} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -\mathrm{g} \end{bmatrix} \tag{2.3} $$

Here g is acceleration due to gravity.

Note : The advantage of using jerk as an input to above model would be clearly visible in MPC implementation. As it allow decoupling and hence convex optimization problem can be solved separately for each axis [28].

Smallest weighted path assumed to be thrice differentiable and jerk is expressed as below

$$ j = (j_x, j_y, j_z) = (x''', y''', z''') \tag{2.4} $$

The input thrust t obtained by applying euclidean norm to (2.3). The f is thrust transformed to inertial axis.

$$ f = {}^{i}_{b}\,\mathrm{U}.t = \left\| \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathrm{g} \end{bmatrix} \right\| \tag{2.5} $$

2.1 Discrete time Model of UAV

Three dimensional jerk is input to the quadcopter system. The states of the system are position (x, y, z), velocity $(\dot{x}, \dot{y}, \dot{z})$ and acceleration $(\ddot{x}, \ddot{y}, \ddot{z})$. The discrete linear time invariant model of quadcopter decoupled for x inertial axis would be same for y and z axis. The jerk (2.6), states (2.7) & dynamics model (2.8) for decoupled x inertial axis

represented as

$$j_x[n] = x'''[n\Delta t] \tag{2.6}$$

$$x_x[n] = \begin{bmatrix} x[n\Delta t] & \dot{x}[n\Delta t] & \ddot{x}[n\Delta t] \end{bmatrix}^T \tag{2.7}$$

$$x_x[n+1] = Ax_x[n] + Bj_x[n] \tag{2.8}$$

$$A = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

$$B = \begin{bmatrix} \frac{1}{6}\Delta t^3 & \frac{1}{2}\Delta t^2 & \Delta t \end{bmatrix} \tag{2.10}$$

Chapter 3

Model Predictive Control for UAV

Suggested [28] MPC with constraints would be solved individually to decoupled inertial axis with a uniform step size of Δt. Trajectory is followed by implementing MPC on individual decoupled inertial axis. Advantage of using MPC is, it can implement soft as well as hard constraints. Soft constraints are generally added to the cost function and hard constraints are explicitly define as linear equality constraints or affine inequality constraints.

Form the various available solver we used CVX to solve MPC with constraints, a package for specifying and solving convex programs [21] , [22] on MATLAB. The solver either returns a solution within a by default residual of $\epsilon = 1.4832 \times 10^{-8}$. That is quite sufficient for most of the case. The solver also returns the status such as solved, unbounded, failed and infeasible from which quality of optimal solution obtained can be inferred.

3.1 Objective function

In MPC cost function would try to emphasis more on error of distance which is controlled by the value of γ. Then tracking of trajectory is done by solving convex optimization problem with cost function (3.1), with equality linear constraints (2.6) and affine inequality constraints (3.4), (3.6) & (3.7) w.r.t to individual decoupled inertial axis.

$$minC_x = \sum_{i=1}^{N_p} \left(x_{path}\left[i\right] - x\left[i\right]\right)^2 + \gamma \sum_{i=1}^{N_p} \left(j_x\left[i\right]\right)^2 \tag{3.1}$$

Here N_p is prediction horizon of the MPC controller.

MPC objective is to minimize its cost function (C_x) influenced by error in path following

and magnitude of jerks input in respective inertial frames. The objective of MPC is to find the possible value of jerks which satisfies explicit constraints with least possible cost function C_x.

3.2 Feasible constraints condition

The Quadcopter trajectory as described in (2.2) and (2.3) would only be feasible if they lie in a feasible set defined as

$$0 < f_{min} \leq f \leq f_{max} \tag{3.2}$$

$$\|\omega\| \leq \omega_{max} \tag{3.3}$$

$$\begin{aligned} j_{xmin} = -\, j_{xmax} \leq j_x \leq j_{xmax} \\ j_{ymin} = -\, j_{ymax} \leq j_y \leq j_{ymax} \\ j_{zmin} = -\, j_{zmax} \leq j_z \leq j_{zmax} \end{aligned} \tag{3.4}$$

As quadcopter have fixed pitch propellers with a fixed direction of rotation

$$f_{min} > 0 \tag{3.5}$$

Squaring the equation (2.5) and decomposing to its components yield following constraints

$$\begin{aligned} \ddot{x}_{min} = -\, \ddot{x}_{max} \leq \ddot{x} \leq \ddot{x}_{max} \\ \ddot{y}_{min} = -\, \ddot{y}_{max} \leq \ddot{y} \leq \ddot{y}_{max} \\ \ddot{z}_{min} = f_{min} - g \leq \ddot{z} \leq \ddot{z}_{max} \end{aligned} \tag{3.6}$$

For z inertial axis we would add one more following constrain over predicted z position as follows only for the 3D Terrain

$$z\left[n\Delta t.. \left(N_p - 1 + n\right)\Delta t\right] \geq z_{path}\left[n\Delta t.. \left(N_p - 1 + n\right)\Delta t\right] \tag{3.7}$$

3.3 Model Parameters

The following are the values of parameters used in the simulation, [29] these values have been observed in real time to match Flying Machine Arena (FMA) at the ETH Zurich.

where:-

Table 3.1: Real Time constraints value

Parameter	Value	Units	Description
f_{min}	5	ms^{-2}	minimum thrust force
$\ddot{x}_{max}$	20	ms^{-2}	max. acceleration in x direction
$\ddot{y}_{max}$	20	ms^{-2}	max. acceleration in y direction
$\ddot{z}_{max}$	20	ms^{-2}	max. acceleration in z direction
$\ddot{x}_{min}$	-20	ms^{-2}	min. acceleration in x direction
$\ddot{y}_{min}$	-20	ms^{-2}	min. acceleration in y direction
j_{xmax}	70	ms^{-3}	max. Jerk in x direction
j_{ymax}	70	ms^{-3}	max. Jerk in y direction
j_{zmax}	70	ms^{-3}	max. Jerk in z direction
j_{xmin}	-70	ms^{-3}	min. Jerk in x direction
j_{ymin}	-70	ms^{-3}	min. Jerk in y direction
j_{zmin}	-70	ms^{-3}	min. Jerk in z direction
v	2	ms^{-1}	UAV velocity
N_p	5	steps	Prediction Horizon of MPC

- x, y, z - global position of UAV in x,y & z direction in inertial frame
- $\dot{x}, \dot{y}, \dot{z}$ - global speed in x,y & z direction

Chapter 4

2D Path Planning and Control of UAV

UAV is subjected to follow the the desired path. The most commonly and the used path are straight line and circular orbits [1]. As they are required to follow the predefined path at a prescribed height.

We used virtual target point (VTP) & Nonlinear Guidance Law (NLGL) algorithms are used for the path generation which UAV tries to follow based on real model constraints.

Model predictive control solve for x & y decoupled axis with constraints and tries to minimize the cost function individually.

4.1 Trajectory Evaluation

A Virtual target point (VTP) is point on the path which UAV tries to follow. VTP may be the point having shortest distance on path form UAV or may be obtain from other algorithm like NLGL. Here γ is an angle which helps in evaluating the VTP , it is the angle line made with UAV and shortest distance point in direction of path transverse by UAV.

A Nonlinear Guidance Law based method described in [1] is used to determine the destination intercept along the given path. For simplicity, assume the UAV is following a path, as shown in Figure 4.1 . At the current UAV position p, draw a circle of radius L. The circle will intercept the path at two points q and ql . Depending on the direction in which the UAV has to move, either q or q1 . will be selected. The advantage of this algorithm is that the same algorithm can be applied to any type of trajectory. The stability of the guidance law is shown using Lyapunov stability arguments .

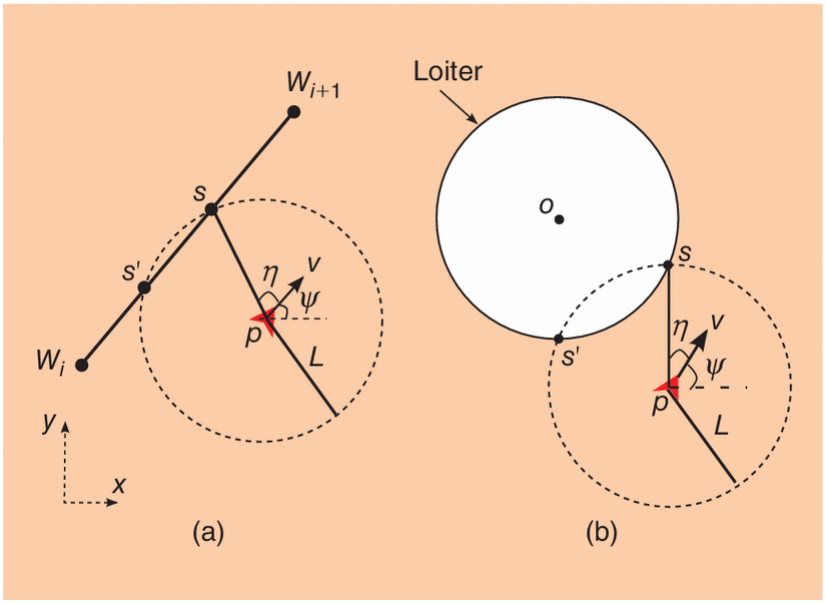

Figure 4.1: Determining the virtual target point (VTP) for NLGL. The VTP for (a) the straight-line path and (b) the loiter path [1]

Once the target intercepts are calculated, future position, velocity and acceleration of the quadcopter are predicted using convex optimization and having constraints based on the quadcopter model. The discrete time formulation (described in the next sub-section) is used to predict up to 5 values if position, velocity and acceleration. The first value of the jerk calculated from the constraints is given as input

4.2 Results

Following are the parameters value used:-

Model Predictive Control tries to minimize the cost function (3.1) & constraints (3.4 & 3.6) for only x and y decoupled axis.

4.2.1 VTP for straight line & Circular Path planning

For the straight line as a reference trajectory 4.2 shows how it follow trajectory for different gamma angles $(\pi/6, \pi/4, \pi/3)$ for VTP algorithm.
For the Circular Path Planning 4.3 shows how it follow trajectory for different gamma angles $(\pi/6, \pi/4, \pi/3)$ for VTP algorithm.

Table 4.1: Parameter value for VTP & NLGL

Parameter	Value	Units	Description
f_{min}	5	ms^{-2}	minimum thrust force
$\ddot{x}_{max}$	20	ms^{-2}	max. acceleration in x direction
$\ddot{y}_{max}$	20	ms^{-2}	max. acceleration in y direction
$\ddot{x}_{min}$	-20	ms^{-2}	min. acceleration in x direction
$\ddot{y}_{min}$	-20	ms^{-2}	min. acceleration in y direction
j_{xmax}	70	ms^{-3}	max. Jerk in x direction
j_{ymax}	70	ms^{-3}	max. Jerk in y direction
j_{xmin}	-70	ms^{-3}	min. Jerk in x direction
j_{ymin}	-70	ms^{-3}	min. Jerk in y direction
v	2	ms^{-1}	UAV velocity
N_p	5	steps	Prediction Horizon of MPC

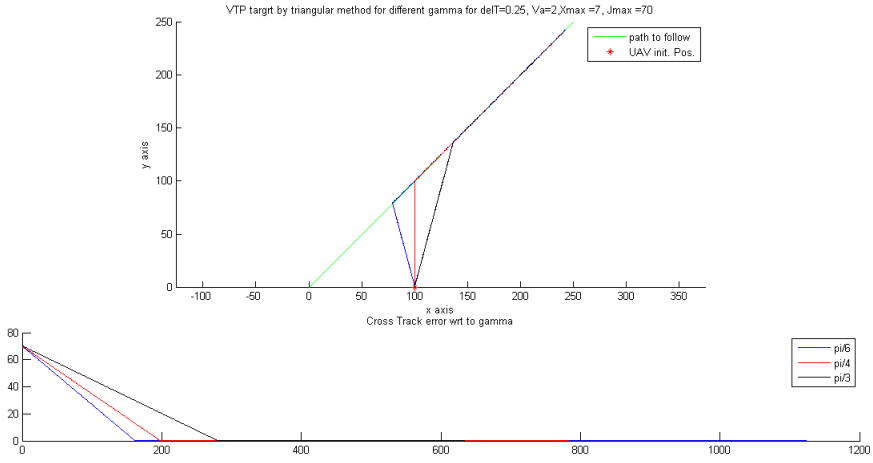

Figure 4.2: VTP algorithm for Straight Line following

4.2.2 NLGL for straight line, Circular Path & Sine Wave

For Straight Line path 4.4 show how accurately it follow the straight line as a reference path based on NLGL algorithm.

For the Circular Orbit 4.5 show how accurately it follow the circular orbit as a reference path based on NLGL algorithm. Here L is the radius of circle intercepts reference path.

For the Sine Wave path 4.6 show how accurately it follow the sine wave as a reference path based on NLGL algorithm.

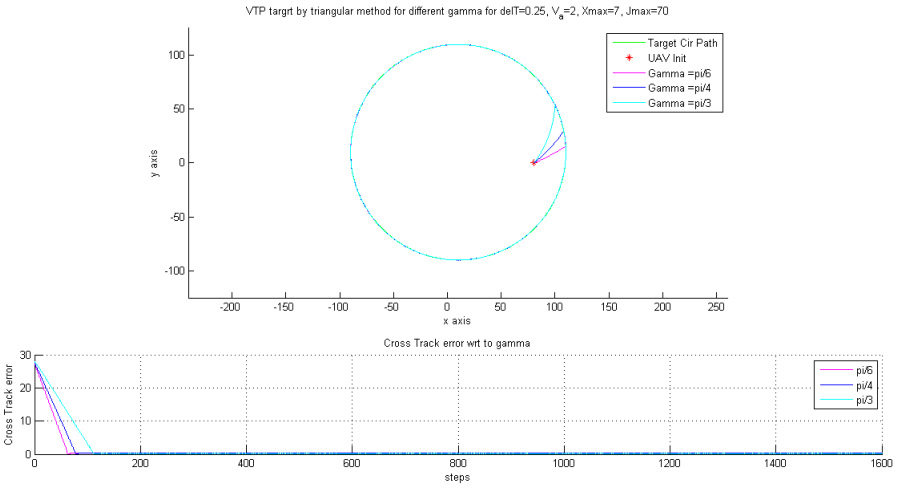

Figure 4.3: VTP algorithm for circular orbit

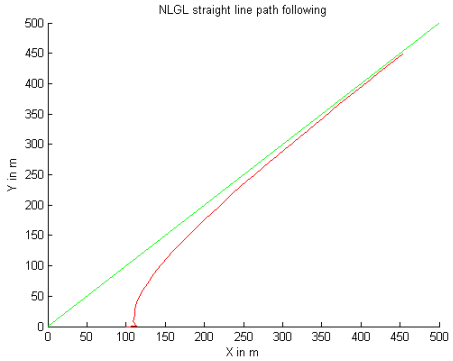

Figure 4.4: NLGL algorithm for straight line as desired path in green & path in red color traced by UAV

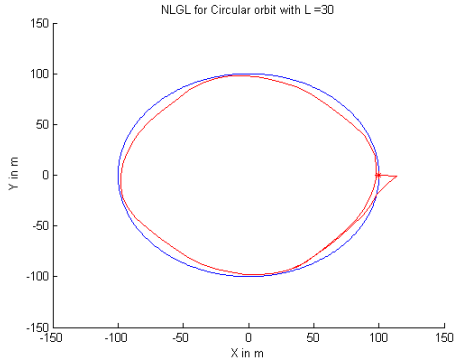

Figure 4.5: NLGL algorithm for circular orbit as desired path in green & path in red color traced by UAV

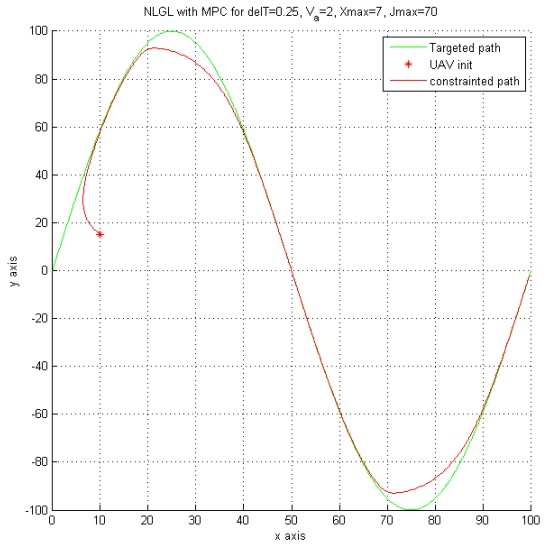

Figure 4.6: NLGL algorithm for sine wave as desired path in green & path in red color traced by UAV

Chapter 5

3D Path Planning and Control of UAV for vertical obstacles

Artificial 3D arena is created to depict Urban buildings. Cuboid building represents the obstacle. Complex structures are made with use of multi cuboid touching there faces.

5.1 Trajectory Evaluation

Path planning here requires following step.

1. Nodes for path, including start, goal points , point on cuboid as suggested by [18] shortest path rarely visit obstacle corners.
2. Nodes selected must not lies inside any of the cuboid obstacle.
3. Selection of node pair as an element of path must not pass through obstacle.
4. Weighage or cost is assigned to selected node pair based on distance between and their relative height. As we wish to closer to ground to avoid radars. So more penalty for higher relative heights.
5. desired path is obtained using dijkstra's algorithm.

5.1.1 Node Selection

As suggested by [18] for a cuboid , shortest path rarely visit obstacles corner. This paper approximated the candidate nodes of shortest path with obstacle corner on the ground (z

= 0) and a middle point of each edge above ground surface.More vertices can be introduced on each obstacle edge, but the computation load both in the cost map construction phase and in the detailed trajectory design phase grows rapidly with small improvements in the accuracy.

5.1.2 Visibility Graph

Formulation is based on some advancement above the approach suggested by [17]. This formulation is allow fast computation and can also handle convex obstacles.

In 3D each obstacle is a polygon. Let π_k denotes the k^{th} polygon. Then for a point r = $[x, y, z]^T$ position w.r.t. π_k obstacle can be understand from below equation (5.3)

$$r = x_i + l * (x_j - x_i) \tag{5.1}$$

$$0 \leq l \leq 1 \tag{5.2}$$

$$\pi_k : A_k * r + b_k \begin{cases} \text{Inside : if all elements of } \pi_k < 0 \\ \text{Boundary : if all elements of } \pi_k \leq 0 : \text{ with at least one element} = 0 \\ \text{Outside : if any elements of } \pi_k > 0 \end{cases} \tag{5.3}$$

For a cuboid π_k has dimension of 6 x 1 .

Selection of node pair $(x_i, x_j)$is based on the fact , it must not pass through any of the obstacle but its path may lie on the boundary of obstacle. We simply discard the node pair which pass inside any polygon π_k which can be determined by solving below mixed integer linear programming (MILP), where c is an integer

$$\min_{r, c_k} \left(\sum c_k \right) \tag{5.4}$$

subjected to

$$A_k * r + b_k \leq c_k * \mathbf{1} \tag{5.5}$$

$$0 \leq c_k \tag{5.6}$$

$$r = x_i + l * (x_j - x_i) \tag{5.7}$$

$$0 \leq l \leq 1 \tag{5.8}$$

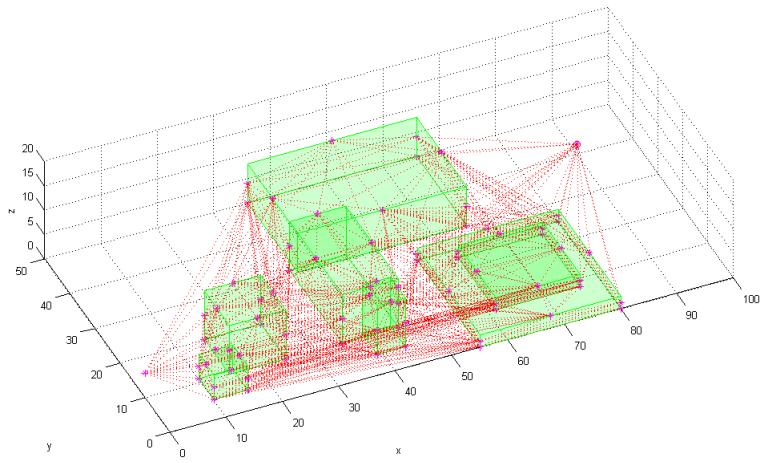

Figure 5.1: * as desired nodes and dotted red line a selected visible path

The above formula compute till it found a point r between the node pair which lies inside the $\pi_k - c * \mathbf{1}$.

So line segment formed by node pair (x_i, x_j) can't considered as a path, if any of the condition is satisfied

$$c_k > 0, \text{ for any k : it lies Inside that obstacle} \tag{5.9}$$

$$c_k = 0, \forall k \begin{cases} \text{test } r, & \text{if it lies Inside } \pi_k \\ \text{test } (r + \epsilon_{ij}, r - \epsilon_{ij}), & \text{if it lies Inside } \pi_k \\ \text{if (z=0) for } (x_i, x_j), \text{test for} \frac{x_i + x_j + \epsilon_z}{2} & \text{if it lies Inside } \pi_k \end{cases} \tag{5.10}$$

$$\epsilon_{ij} = \frac{x_j - x_i}{\|x_j - x_i\|} * \epsilon \tag{5.11}$$

$$\epsilon_z = [0, 0, 1]^T * \epsilon \tag{5.12}$$

Here $(r + \epsilon_{ij}, r - \epsilon_{ij}$) are the near by point of r lying on line segment between (x_i, x_j). The above expression also take care of the path formed below the obstacle surface.

Figure 5.1 shows the (*) in magenta as candid nodes based on above selection critera, red dotted line are the visible path , green cuboid as vertical obstacles and (*) & (o) red as start & goal position.

5.1.3 Cost Function

The Cost function C_{ij} (5.13) assigns the weight age between current node n_i with its one of visible neighbor nodes. Here node n_j represents one of the member in visible neighbors.

Relative height of nodes n_i and n_j are represented as n_i^z and n_j^z.

$$C_{ij} = \|n_i - n_j\| \, (1 + \alpha(mean(n_i^z, n_j^z)) \tag{5.13}$$

The cost function C_{ij} depends on parameter α for a given terrain as relative heights and euclidean distance would be same between the visible neighbor nodes and current node pair. α decides weight age between mean relative heights and euclidean distance.

Then based on desired path MPC control is implemented with constraints.

5.2 Result

Following 5.2 show the path planned in black dot line and UAV followed the path using MPC with constraints in red line. With start node as (*) and goal node as (o).

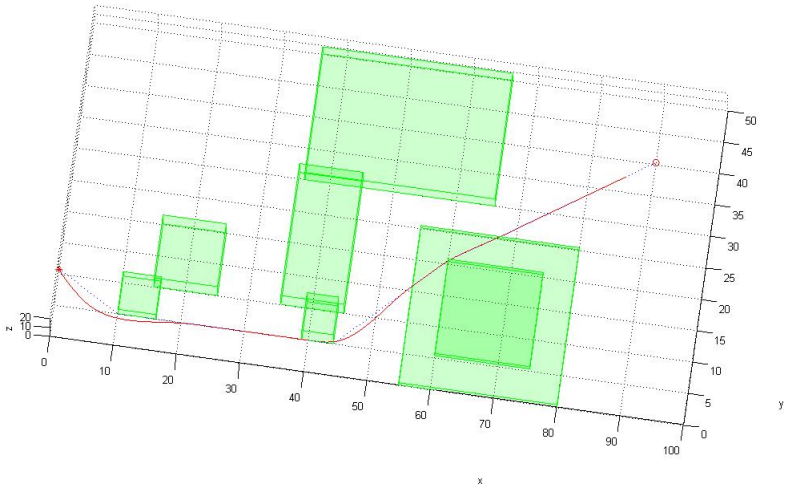

Figure 5.2: Desired 3D trajectory in black dots, UAV with MPC in red line

Chapter 6

3D Path Planning and Control of UAV for 3D Terrain

6.1 Trajectory Evaluation

For a given square terrain of size $n \times n$ m^2. The total no of nodes would be n^2. The node number n_i can be evaluated based on position(x, y) at terrain as

$$n_i = x + (n-1)\, y \tag{6.1}$$

Smallest weighted path is evaluated by Dijkstra's algorithm which is a graph search algorithm that solves single source and goal path on graph based on non negative cost function. From all the possible path between start node and goal node, smallest weighted path have smallest sum of weights assigned to the node pair in sequence forming the path from start node to goal node. For graph search it required information about neighborhood and cost map. The neighborhood scheme is demonstrated for a terrain by visibility node chart and cost map is determined by the cost function for visible neighbor nodes and current node.

6.1.1 Visibility node chart

For terrain surface visible nodes are adjacent neighbor nodes (o) to which quadcopter can move from current node (x) as its current position.

The 8 point neighbor scheme is followed. The figure (6.1) shows how the 8 point neighbor scheme varies at the central and the boundary regions. For scheme having more than

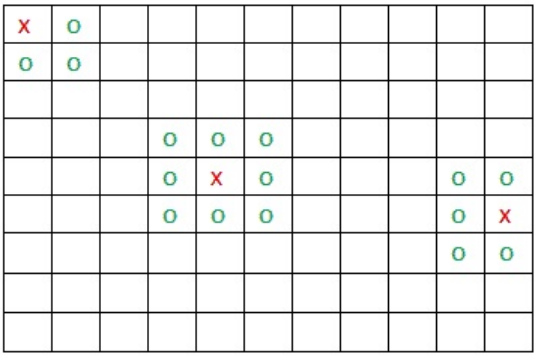

Figure 6.1: Neighbor visibility graph, o represents the neighbor to x node.

8 neighbors may not include in between nodes of path as well as there penalty and hence should not be incorporated as it may result in missing nodes from the actual path trajectory between start node to goal node. The current node of quadcopter is represented by (x) and their neighbor nodes are represented by (o). This neighbor relationship help in evaluating the visibility node chart. For current node n_i positioned at central region following would be visible neighbors based on figure (6.1)

$$\begin{bmatrix} n_i\text{-}1\text{+n} & n_i\text{+n} & n_i\text{+}1\text{+n} \\ n_i\text{-}1 & \text{-} & n_i\text{+}1 \\ n_i\text{-}1\text{-n} & n_i\text{-n} & n_i\text{+}1\text{-n} \end{bmatrix} \tag{6.2}$$

For the boundary node there visible node can be obtain with help of (2) and figure(6.1).

6.1.2 Cost Map

The Cost function C_{ij} (6.3) assigns the weight age between current node n_i with its one of visible neighbor nodes. Here node n_j represents one of the member in visible neighbors. Relative height of nodes n_i and n_j are represented as n_i^z and n_j^z.

$$C_{ij} = \|n_i - n_j\| \left(1 + \alpha(mean(n_i^z, n_j^z)\right) \tag{6.3}$$

The cost function C_{ij} depends on parameter α for a given terrain as relative heights and euclidean distance would be same between the visible neighbor nodes and current node pair. α decides weight age between mean relative heights and euclidean distance. So it controls cost function C_{ij} which indirectly influence smallest weighted path based on penalty. To avoid threads and radar detection it is preferred to move closer to terrain surface and not to climbing through ridges. In cost map (6.4) for position (i, j) represents

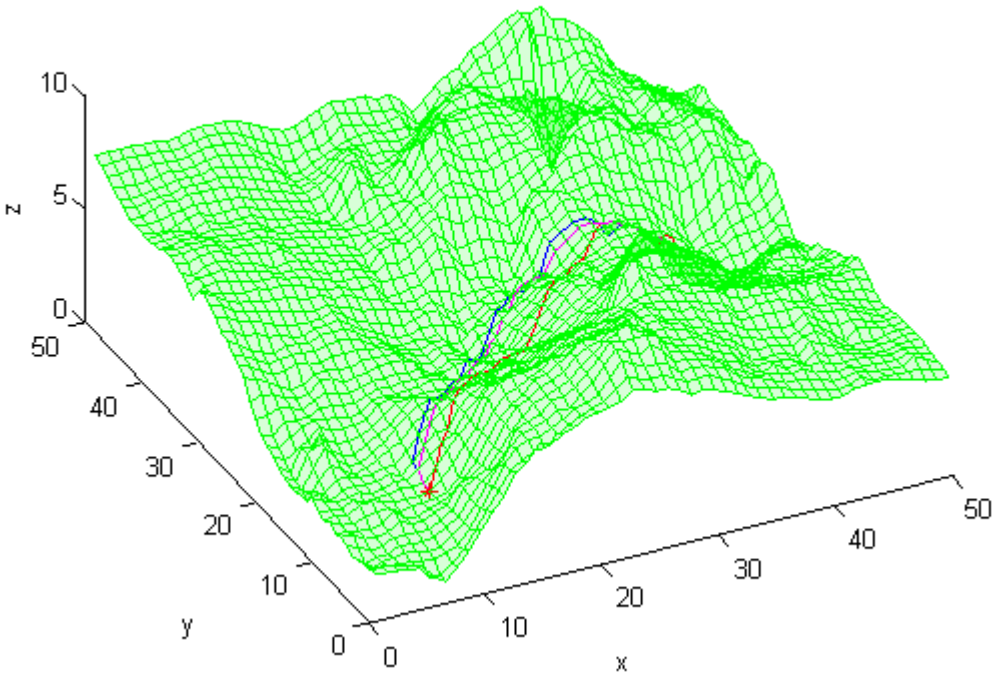

Figure 6.2: shows different smallest weighted path between start node (*) and goal node (o) for $\alpha = 2$ in blue line, $\alpha = 0.2$ in magenta line & $\alpha = 0$ in red line

the cost between current node n_i and neighbor node n_j as

$$C(i,j) = \begin{cases} C_{ij} & \text{node } n_j \text{ is visible neighbor} \\ \infty & \text{node } n_j \text{ is not visible neighbor} \end{cases} \tag{6.4}$$

6.1.3 Smallest weighted Path

Smallest weighted path is evaluated using the Dijkstra's algorithm based on the visible neighbor chart and cost map. Path evaluated not solely depends on euclidean distance as α in cost function C_{ij} decides weight age to relative height of nodes also. Hence smallest weighted trajectory may be different for different value of α.

From the figure (6.2) demonstrate impact of α on smallest weighted path. Smallest weighted paths for $\alpha = 2$ would be used as reference for MPC control for quadcopter.

6.2 Results

The 3D terrain was generated [24] of size 50×50 m^2. For the value of $\alpha = 2$ in a cost function C_{ij} smallest weighted path is evaluated by using dijkstra's algorithm based on cost map.

The following are the parameters used in the simulation, [29] these values have been observed in real time to match Flying Machine Arena (FMA) at the ETH Zurich.

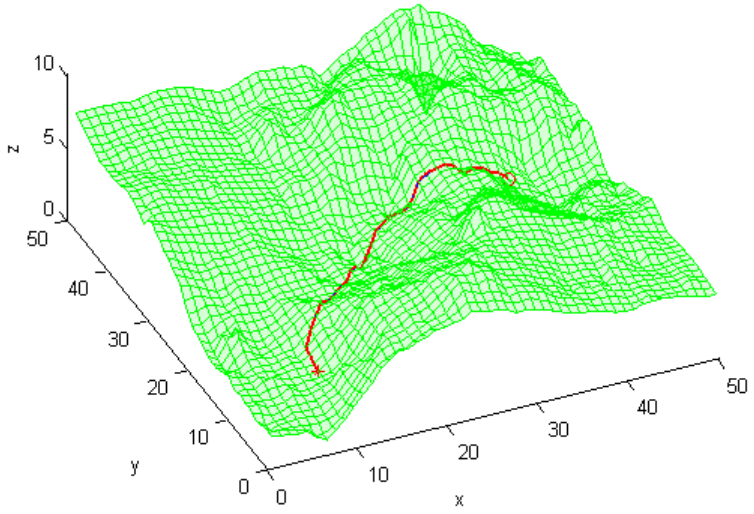

Figure 6.3: shows smallest weighted path in blue line between start node (*) and goal node (o), red line shows trajectory followed by MPC for $v = 2$ ms^{-1}

Here v is velocity of quadcopter and N_p is the prediction horizon of the MPC. The next as well as important step was selection of γ value such that constraint (3.7) remains valid as it depends on the distance error between the actual path and smallest weighted trajectory.

As the cost function C_x, C_y and C_z of MPC (3.1) comprises of two parts and we had given more weight age to the distance error part. As we knew that jerk can't be greater the 70 ms^{-3} as problem become infeasible. The desired distance error must be lower than 0.25 for constraint (3.7) to be valid. The value of $\gamma = 0.001$ would satisfies the requirement, which is equivalent of giving 3 times more weight age to max acceptable distance error w.r.t max jerk error. The figure (6.3) show good results for path followed for $v = 2$ ms^{-1}. The figure (6.4) & (6.5) shows the value of distance error less than the desired 0.25, constraints on jerks & acceleration are also satisfied.

These are calculated on a laptop running on Windows 7 with Intel Core i3-2350M dual core processor (2.5Ghz) with 4GB RAM. CVX is a Matlab-based modeling system for

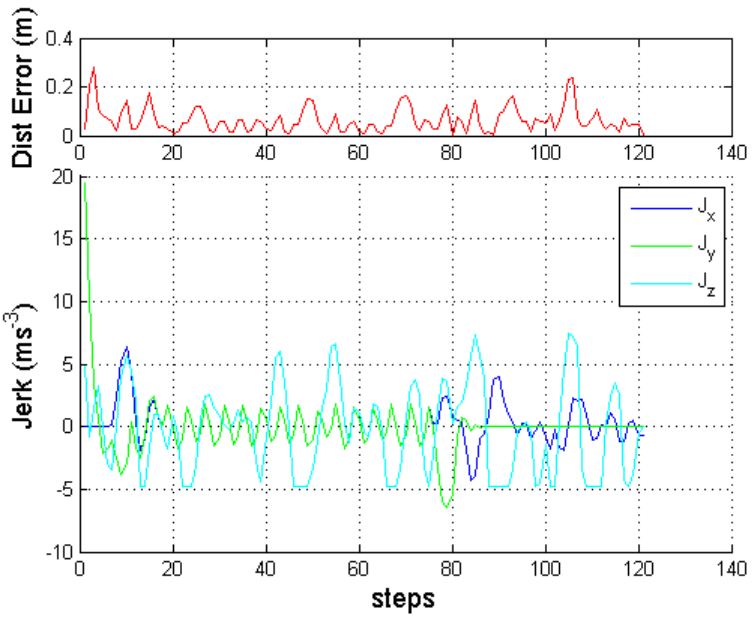

Figure 6.4: shows the value of distance error and jerk of quadcopter for $v = 2\ ms^{-1}$

convex optimization. CVX turns Matlab into a modeling language, allowing constraints and objectives to be specified using standard Matlab expression syntax.

6.3 3D Path Planning and Control of UAV for 3D Terrain with threat avoidance

In this threat are also sensing if detected try to maintain minimum distance from it as non flying zone. So when threat is encounter or identify at a particular point on the planned path while traversing. We discard the planned path and identify the non flying zone and setting its cost to inf. From the current point after updating the cost map we obtain the shortest weighted path by Dijkastra Algorithm. And then follows that path as a reference trajectory using Model Predictive Control with constraints.

6.4 Results

In this UAV tries to maintain minimum distance of 10 m from threat detected as a no fly zone to avoid it. Figure 6.6 & 6.7 shows how the UAV avoid no flying zone. Here the old planned path passing through no fly zone is shown in blue. The new planned path in blue overlaps with UAV transverse path in red. Nodes as (* , o) in red represents start and goal nodes. No fly area for UAV is presented as (*) in magenta having threat at centre.

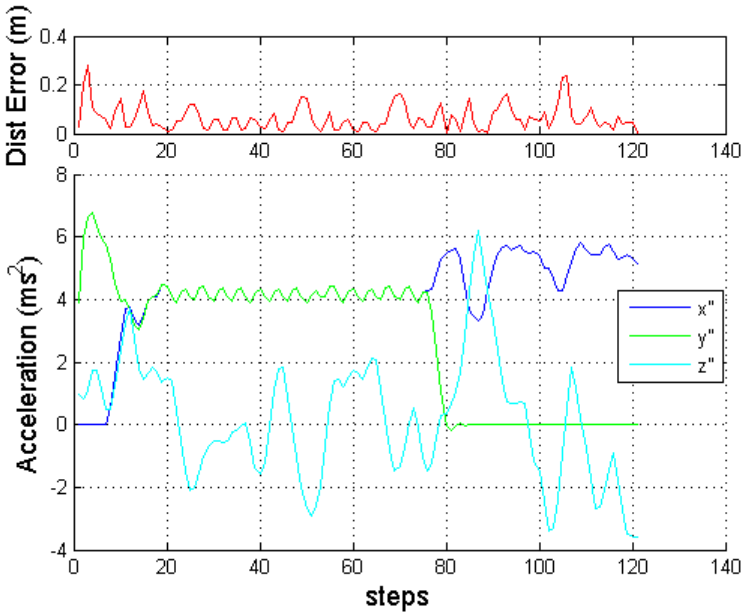

Figure 6.5: shows the value of distance error and acceleration of quadcopter for $v = 2\ ms^{-1}$

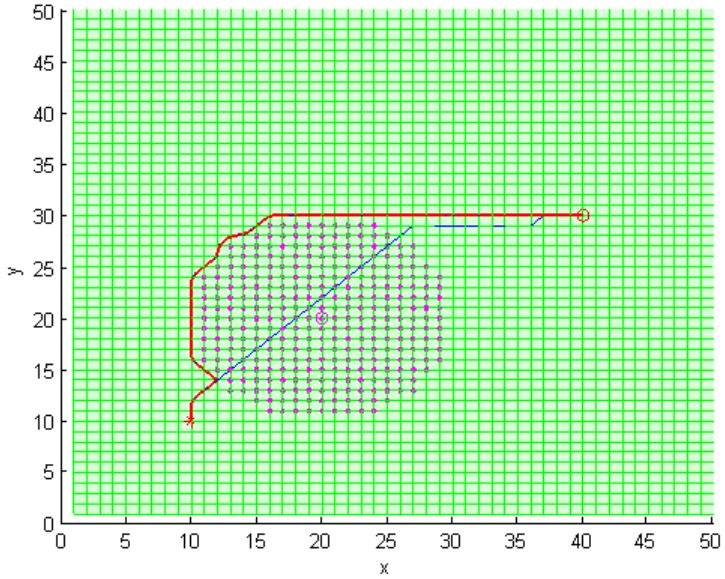

Figure 6.6: Top view of threat avoidance by UAV

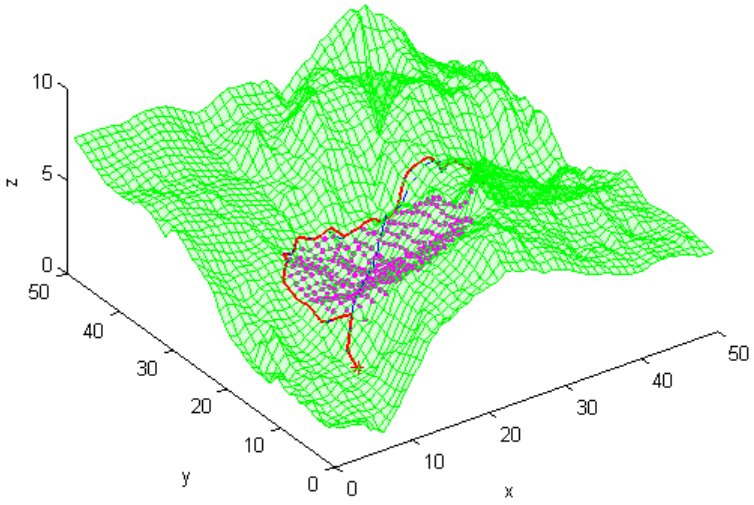

Figure 6.7: Other view of threat avoidance by UAV

Chapter 7

Conclusion & Future Directions

This thesis presented a method for 3D terrain path generation and path following with MPC with constraints by quadcopter works satisfactory well. Quadcopter is supposed to move closer to terrain and avoid going through ridges in order to evade from threats and radar detection. The proposed method had two parts: smallest weighted path evaluation and trajectory following using MPC with constraints. For evaluation of smallest weighted path for a given terrain required cost map based on knowledge of visibility node chart and cost function. Then Dijkstras algorithm evaluates smallest weighted path based on cost map. Trajectory is followed by Quadcopter using Model Predictive Control with constraints on jerks, position & acceleration. The simulation showed good results for trajectory followed by quadcopter using MPC with constraints.

Further work can be extended using Model Predicative Control with the nonlinear Quadcopter model having precise state output. The MPC may be used with external disturbances such as wind as their direction is influenced near to the terrain shape. Other Dynamics model of UAV may be used to compare the results obtained. In Path planning weightage may be added if quadcopter constraints are violated based on which dijkastra may choose a path that would be constrained friendly shortest weighted path. Threat avoidance may be extended to the moving threat approaching UAV, based on the threat information path planning and path following may be implemented.

Bibliography

[1] PB Sujit, Srikanth Saripalli, and Joao Borges Sousa. Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless. *Control Systems, IEEE*, 34(1):42–59, 2014.

[2] K Alexis, G Nikolakopoulos, A Tzes, and L Dritsas. Coordination of helicopter uavs for aerial forest-fire surveillance. In *Applications of intelligent control to engineering systems*, pages 169–193. Springer Netherlands, 2009.

[3] Jose AJ Berni, Pablo J Zarco-Tejada, Lola Suárez, and Elias Fereres. Thermal and narrowband multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(3):722–738, 2009.

[4] E Raymond Hunt, W Dean Hively, Stephen J Fujikawa, David S Linden, Craig ST Daughtry, and Greg W McCarty. Acquisition of nir-green-blue digital photographs from unmanned aircraft for crop monitoring. *Remote Sensing*, 2(1):290–305, 2010.

[5] ER Hunt, W Dean Hively, Craig ST Daughtry, Greg W McCarty, Stephen J Fujikawa, TL Ng, Michael Tranchitella, David S Linden, and David W Yoel. Remote sensing of crop leaf area index using unmanned airborne vehicles. In *Proceedings of the Pecora 17 Symposium, Denver, CO*, 2008.

[6] SR Herwitz, LF Johnson, SE Dunagan, RG Higgins, DV Sullivan, J Zheng, BM Lobitz, JG Leung, BA Gallmeyer, M Aoyagi, et al. Imaging from an unmanned aerial vehicle: agricultural surveillance and decision support. *Computers and electronics in agriculture*, 44(1):49–61, 2004.

[7] Nathan E Serrano. *Autonomous quadrotor unmanned aerial vehicle for culvert inspection*. PhD thesis, Massachusetts Institute of Technology, 2011.

[8] Masahiko Nagai, Tianen Chen, Ryosuke Shibasaki, Hideo Kumagai, and Afzal Ahmed. Uav-borne 3-d mapping system by multisensor integration. *Geoscience and Remote Sensing, IEEE Transactions on*, 47(3):701–708, 2009.

[9] Leena Singh and James Fuller. Trajectory generation for a uav in urban terrain, using nonlinear mpc. In *American Control Conference, 2001. Proceedings of the 2001*, volume 3, pages 2301–2308. IEEE, 2001.

[10] Guan-chen Luo, Jian-qiao Yu, Si-yu Zhang, and Wei Zhang. Artificial potential field based receding horizon control for path planning. In *Control and Decision Conference (CCDC), 2012 24th Chinese*, pages 3665–3669. IEEE, 2012.

[11] Yong-bo Chen, Guan-chen Luo, Yue-song Mei, Jian-qiao Yu, and Xiao-long Su. Uav path planning using artificial potential field method updated by optimal control theory. *International Journal of Systems Science*, (ahead-of-print):1–14, 2014.

[12] John Bellingham, Arthur Richards, and Jonathan P How. Receding horizon control of autonomous aerial vehicles. In *American Control Conference, 2002. Proceedings of the 2002*, volume 5, pages 3741–3746. IEEE, 2002.

[13] Arthur Richards and Jonathan How. Decentralized model predictive control of cooperating uavs. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 4, pages 4286–4291. IEEE, 2004.

[14] John Bellingham, Michael Tillerson, Arthur Richards, and Jonathan P How. Multi-task allocation and path planning for cooperating uavs. In *Cooperative Control: Models, Applications and Algorithms*, pages 23–41. Springer, 2003.

[15] Antonio Bicchi and Lucia Pallottino. On optimal cooperative conflict resolution for air traffic management systems. *Intelligent Transportation Systems, IEEE Transactions on*, 1(4):221–231, 2000.

[16] Zhi-Hong Mao, Eric Feron, and Karl Bilimoria. Stability and performance of intersecting aircraft flows under decentralized conflict avoidance rules. *Intelligent Transportation Systems, IEEE Transactions on*, 2(2):101–109, 2001.

[17] Yoshiaki Kuwata and Jonathan How. Three dimensional receding horizon control for uavs. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, volume 3, pages 2100–2113, 2004.

[18] Lakmi P Gewali, Simeon Ntafos, and Ioannis G Tollis. Path planning in the presence of vertical obstacles. *Robotics and Automation, IEEE Transactions on*, 6(3):331–341, 1990.

[19] Joachin Dahl and Lieven Vandenberghe. Cvxopt: A python package for convex optimization. In *Proc. eur. conf. op. res*, 2006.

[20] Alexander Domahidi, Aldo U Zgraggen, Melanie Nicole Zeilinger, Manfred Morari, and Colin N Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 668–674. IEEE, 2012.

[21] Michael Grant, Stephen Boyd, and Yinyu Ye. Cvx: Matlab software for disciplined convex programming, 2008.

[22] Michael C Grant and Stephen P Boyd. Graph implementations for nonsmooth convex programs. In *Recent advances in learning and control*, pages 95–110. Springer, 2008.

[23] Jacob Mattingley and Stephen Boyd. Cvxgen: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, 2012.

[24] Tucker McClure. Automatic terrain generration, matlab centralfile exchange, retrieved jan 08, 2013.

[25] Samir Bouabdallah, Pierpaolo Murrieri, and Roland Siegwart. Design and control of an indoor micro quadrotor. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4393–4398. IEEE, 2004.

[26] Peter C Hughes. *Spacecraft attitude dynamics*. Courier Corporation, 2012.

[27] Peter H Zipfel. *Modeling and simulation of aerospace vehicle dynamics*. Amer Inst of Aeronautics &, 2007.

[28] Mark W Mueller, Markus Hehn, and Raffaello D'Andrea. A computationally efficient algorithm for state-to-state quadrocopter trajectory generation and feasibility verification. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 3480–3486. IEEE, 2013.

[29] Mark W Mueller and Raffaello D'Andrea. A model predictive controller for quadrocopter state interception. In *Control Conference (ECC), 2013 European*, pages 1383–1389. IEEE, 2013.