Czech Technical University in Prague

Faculty of Electrical Engineering

Bachelor thesis

Hierarchical Approach to Motion Planning of Formations of Unmanned Helicopters in Complex Environment

Filip Rak

Supervisor: Dr. Martin Saska

2014

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne ……………………… ……………………………………….

Podpis autora práce

Czech Technical University in Prague
Faculty of Electrical Engineering

Department of Cybernetics

BACHELOR PROJECT ASSIGNMENT

Student: Filip R a k

Study programme: Cybernetics and Robotics

Specialisation: Robotics

Title of Bachelor Project: Hierarchical Approach to Motion Planning of Formations of Unmanned Helicopters in Complex Environment

Guidelines:

The aim of the thesis is to design and implement an approach enabling to integrate RRT (Rapidly exploring Random Trees [2]) techniques into the system of predictive control of formations of unmanned helicopters. A hierarchical method of reduction of RRT trajectories will be design, which is crucial for utilization of RRT in initialization of the predictive control.
Work plan:

- To study the implementation of method in [1] provided by advisor of this thesis.
- To study and implement different RRT variants [3, 4] and to integrate two of them into the system [1].
- To design and implement a method for reduction of complexity of RRT trajectories.
- To verify the algorithm in simulations and in an experiment with real robots, if the formation control system (developed in parallel) will be prepared.

Bibliography/Sources:

[1] M. Saska, Z. Kasl, L. Preucil: Motion planning and control of formations of micro aerial vehicles. Accepted for IFAC World Congress 2014.
[2] S. M. LaValle: Rapidly-exploring random trees: A new tool for path planning. Technical report, TR 98-11, Computer Science Dept., Iowa State University, 1998.
[3] S. M. LaValle: Planning algorithms. Cambridge University Press, 2006.
[4] L. Jaillet, J. Cortés, T. Siméon: Sampling-based path planning on configuration-space costmaps. IEEE Transactions on Robotics, 26(4):635-646, 2010.

Bachelor Project Supervisor: Ing. Martin Saska, Dr. rer. nat.

Valid until: the end of the summer semester of academic year 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
Head of Department

prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 10, 2014

České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra kybernetiky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Filip R a k

Studijní program: Kybernetika a robotika (bakalářský)

Obor: Robotika

Název tématu: Hierarchický přístup k plánování pohybu formací bezpilotních helikoptér v komplexním prostředí

Pokyny pro vypracování:

Cílem práce je navrhnout a implementovat metodu umožňující začlenit techniku RRT (Rapidly exploring Random Trees [2]) do systému prediktivního řízení formací bezpilotních helikoptér [1]. Konkrétně bude vyvinut hierarchický přístup redukce RRT trajektorií a navržená metoda bude uzpůsobena potřebám inicializace prediktivního řízení.
Plán prací:

- Nastudovat metodu uvedenou v [1] a její implementaci poskytnutou vedoucím práce.
- Nastudovat a naimplementovat různé varianty RRT [3, 4] a integrovat alespoň dvě z nich do systému plánování pohybu formace.
- Navrhnout a implementovat metodu pro redukci složitosti RRT trajektorie.
- Ověřit funkci algoritmu pomocí simulací a případně experimenty s reálnými roboty. Podle dostupnosti paralelně vyvíjeného systému pro řízení formací helikoptér vedoucí práce rozhodne, zda bude požadován reálný test, nebo se student zaměří na podrobnější analýzu systému v simulaci.

Seznam odborné literatury:

[1] M. Saska, Z. Kasl, L. Preucil: Motion planning and control of formations of micro aerial vehicles. Accepted for IFAC World Congress 2014.
[2] S. M. LaValle: Rapidly-exploring random trees: A new tool for path planning. Technical report, TR 98-11, Computer Science Dept., Iowa State University, 1998.
[3] S. M. LaValle: Planning algorithms. Cambridge University Press, 2006.
[4] L. Jaillet, J. Cortés, T. Siméon: Sampling-based path planning on configuration-space costmaps. IEEE Transactions on Robotics, 26(4):635-646, 2010.

Vedoucí bakalářské práce: Ing. Martin Saska, Dr. rer. nat.

Platnost zadání: do konce letního semestru 2014/2015

L.S.

doc. Dr. Ing. Jan Kybic
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 10. 1. 2014

Acknowledgment

First of all, I would like to thank to my supervisor, Dr. Martin Saska, who dedicated a lot of time, effort and patience to help me with this thesis. I would also thank to Ing. Zdeněk Kasl, who helped me and guided me in early stage, last, but not least I would like to thank to my family and friends for their support.

Abstract

The goal of this thesis was to propose in algorithm suitable for planning of initial trajectory for formations of Micro Aerial Vehicles (MAV). To find a feasible trajectory, Rapidly exploring Random Trees (RRT) are used. The advantage of RRT is that during creating of a tree, kinematic constraints are respected. Generated trajectory is needed to be reduced. This is achieved by Sequential Quadratic Programming (SQP) optimizing method. Simplified trajectory is passed to system of formation control. Created solution enables to find trajectory in complex environments and reduce computational time.

Abstrakt

Cílem této práce je navržení algoritmu pro plánování iniciální trajektorie bezpilotních mikro kvadrokoptér. K nalezení proveditelné trajektorie se využívají „Rychle náhodně rostoucí stromy“ (Rapidly exploring Random Trees - RRT). Výhodou RRT je, že při vytváření stromu jsou zahrnuty kinematické podmínky kvadrokoptéry. Výslednou trajektorii je třeba zjednodušit. To se řeší pomocí metody optimalizace Sekvenčního kvadratického Programování (Sequential quadratic Programming – SQP). Zjednodušená trajektorie se předá systému řízení formace. Vytvořené řešení umožňuje najít trajektorii ve složitějších prostředích a zkrátit čas potřebný k výpočtu řízení.

Contents

List of figures

1 Introduction

The formation of autonomous robots is getting more interesting these days. Thanks to smaller, cheaper and more powerful components such as motors and computational units, which can be used in many ways. For example, search and rescue missions, cargo delivery, surveillance etc. The Unmanned Aerial Vehicles (UAVs) we use are called quadrotors. They can be also referred as Micro Aerial Vehicles (MAV). The size of a quadrotor can vary, depending on which task it should perform. There are both, indoor and outdoor quadrotors.

Figure 1.1: Formation of UAVs. Source: Wikimedia Commons

This thesis is focused on initial trajectory planning for formations of quadrotors. The quadrotors are popular today because of their simple construction and also because they can be very small. It has four counter-rotating blades.

The Goal of the thesis is to implement a path planning algorithm, RRT [1] and reduce a complex trajectory found by RRT to an acceptable form for a formation solver. Simplified trajectory is passed to the formation control system, introduced in paper [3].

The thesis is organized as follows: in the next chapter, used tools, a formation description and a quadrotor model are described. In chapter 3, used path planning algorithms are explained. In chapter 4, path reduction with hierarchical approach is presented. In the last chapter 5, experimental verification is presented.

2 Preliminaries

In this chapter, an explanation of used methods is presented. In section 2.1, a leader-follower formation system is introduced. In section 2.2, a model of a quadrotor is described. In section 2.3, Model Predictive control is described. In section 2.4, Sequential Quadratic Programming is introduced.

2.1 Formation keeping

There are various ways of controlling a formation. This thesis is built on a thesis [3] in which is a leader – follower approach and one of the robots is a leader. Another approach is a swarm approach based on the behavior of flocks or herds of animals.

The leader is usually best equipped robot, because the leader needs to carry a device to determine its position in global space. Followers keep position relative to the leader, so only a tracking device could be mounted, reducing cost of followers. This approach also enables use of heterogeneous formations – for example, one quadrotor leader and ground units as followers. This is further expanded in [9].

The position of a follower is represented relative to the leader. The position is denoted by free parameters:p_i, q_i and r_i. Meaning of these parameters is shown on figure 2.1.

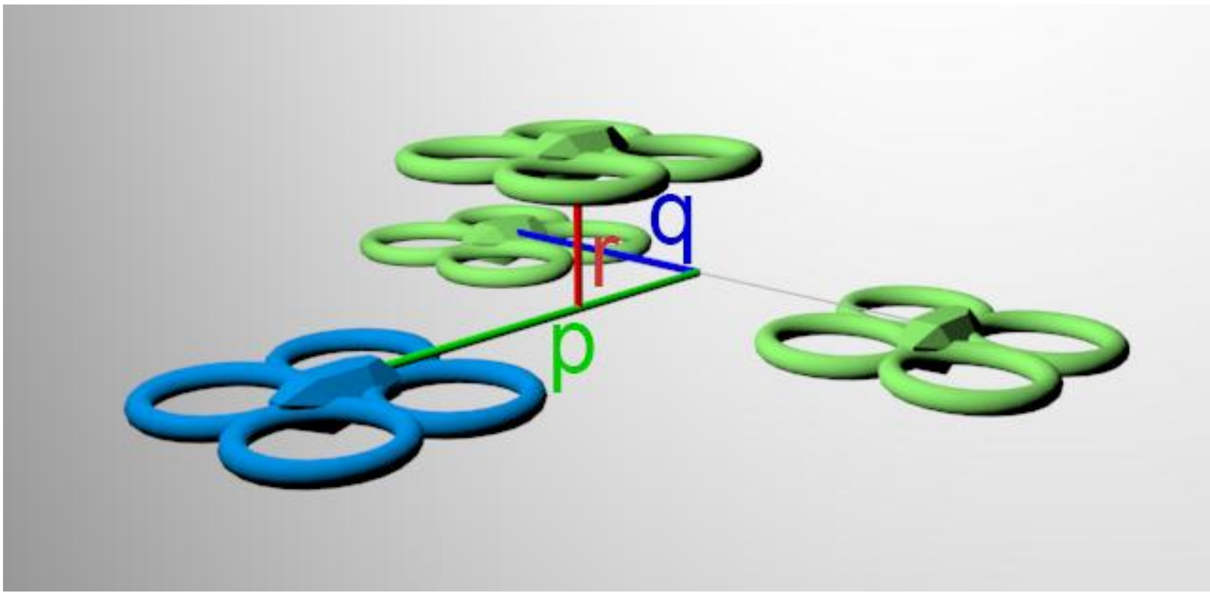

Figure 2.1: Representation of MAV formation.

2.2 Model Predictive Control

Model Predictive Control (MPC) is an approach widely used in the industry, but it also finds its usage in other fields. In this thesis, it is used to control a movement of a formation. The whole trajectory is coded into vector, with N+M control elements, where N is the length of the horizon and M is the length of the trajectory behind the horizon. The control input consists of four variables: horizontal speed v_t, vertical speed v_n, curvature c and duration Δt.

The MPC control loop goes as follows: from robot's current position, optimal trajectory is computed. Next, the first trajectory element is applied and the robot moves to a new position. The whole process is repeated until the goal region is reached.

Instead of coding only the length of the horizon, the whole trajectory is coded into an optimization vector. The advantage of this approach is ability to react to dynamical changes of environment, such as moving obstacles or other robots in the formation.

vt_1	vn_1	c_1	t_1	… … … …	vt_N	vn_N	c_N	t_N	vt_{N+1}	vn_{N+1}	c_{N+1}	t_{N+1}	… … … …	vt_{N+M}	vn_{N+M}	c_{N+M}	t_{N+M}

Figure 2.2: Trajectory representation.

2.3 Model of a Quadrotor

The motion model of a quadrotor is needed for path planning algorithms to find a feasible solution. Its kinematic movement is described by equations:

$$\dot{x}(t) = v_t(t)\cos(\theta(t)), \tag{2.1}$$

$$\dot{y}(t) = v_t(t)\sin(\theta(t)), \tag{2.2}$$

$$\dot{z}(t) = v_n(t), \tag{2.3}$$

$$\dot{\theta}(t) = v_t(t)c(t). \tag{2.4}$$

Input variables are horizontal speedv_t, vertical speedv_n, curvature c.

From these equations, position and yaw angle is as follows:

$$\theta = v_t(t)k(t)\Delta t, \tag{2.5}$$

$$x = \begin{cases} v_t(t)\Delta t\cos\big(\theta(t)\big) + x_0 & ,k = 0 \\ \frac{1}{k(t)}[\sin(v_t(t)k(t)\Delta t + \theta_0) - \sin(\theta_0)] + x_0 & ,k \neq 0 \end{cases}, \tag{2.6}$$

$$y = \begin{cases} v_t(t)\Delta t\sin\big(\theta(t)\big) + y_0 & ,k = 0 \\ \frac{1}{k(t)}[\cos(v_t(t)k(t)\Delta t + \theta_0) - \cos(\theta_0)] + y_0 & ,k \neq 0 \end{cases}, \tag{2.7}$$

$$z = v_n(t)\Delta t. \tag{2.8}$$

Duration of applied input is Δt. This model is used for initial trajectory planning, however, for MPC dynamic model is needed, to reflect real-life properties of MAV. Equations are as proposed in paper [10]:

$$\dot{x} = v \tag{2.9}$$

$$m\dot{v} = mge_3 - fRe_3 \tag{2.10}$$

$$\dot{R} = R\widehat{\Omega} \tag{2.11}$$

$$J\dot{\Omega} + \Omega \times \mathrm{J}\Omega = M. \tag{2.12}$$

In the equations:

$m \in \mathbb{R}$	is the total mass of the quadrotor,
$g = 9{,}81\ m \cdot s^{-2}$	is the gravitational acceleration,
$J \in \mathbb{R}^{3x3}$	is the inertia matrix with respect to body fixed frame,
$R \in SO(3)$	is the rotation matrix from the body fixed frame to the inertial frame,
$\Omega \in \mathbb{R}^3$	is the angular velocity with respect to the body fixed frame,
$x \in \mathbb{R}^3$	is the position of the center of the mass the quadrotor in the inertial frame,
$v \in \mathbb{R}^3$	is the velocity of the center of the mass of the qadrotor in the inertial frame,
$f \in \mathbb{R}$	*is* the total thrust of quadrotor's propellers,
$M \in \mathbb{R}^3$	is the total moment in the body fixed frame.

Vectors $e_1, e_2, e_3 \in \mathbb{R}^3$are columns of identity matrix. $\widehat{\Omega} \in SO(3)$ is a skew-symmetric matrix such that $\hat{x}y = x \times y,\ \ x, y \in \mathbb{R}^3$.

In this framework, weight of quadrotor $m = 4{,}34$ kg and the inertia matrix

$$J = \begin{bmatrix} 0.0820 & 0 & 0 \\ 0 & 0.0845 & 0 \\ 0 & 0 & 0.1377 \end{bmatrix} kg \cdot m^2.$$

For more information about how to compute thrust of each propeller and how to control quadrotor please see the paper [10].

2.4 Sequential Quadratic Programming

Sequential quadratic programming is a method for solving nonlinear constrained optimization problems. Its disadvantage lies in inability to overcome local extremes.

In this thesis, CFSQP library is used. It is used for the reduction of initial trajectory planed by Rapidly explored Random Trees (RRT) [1] and for optimizing formation control. More information about the library can be found in manual[2].

CFSQP uses two user defined functions for trajectory evaluation: Objective and Constraint.

The objective function provides an evaluation of a user-defined cost function. CFSQP solver can determine whether the trajectory is better or worse in iteration step using this function. Because of this, it is important to choose good evaluation. Not only the quality of the solution depends on it, also stability and efficiency is dependent on the function. In this thesis, it is needed to define two cost functions: one for trajectory reduction and other for formation control. The formation control cost function is defined in [3].

The purpose of constraint function is to add motion constraints. For usage in trajectory reduction and formation control, two inequality constraints are proposed. Constraint function consists of two components: obstacle distance and distance to goal. This ensures that formation during movement keeps sufficient distance from obstacles and will converge to the proposed goal. The return value for each inequality constraint has to be less than zero, otherwise trajectory is considered unfeasible. Constraint function is the same for trajectory reduction and formation control.

3 Motion planning

To drive the formation of MAVs, initial trajectory planning is needed, because formation trajectory optimization needs initial guess. This thesis aims to implement two methods – Rapidly exploring Random Trees (RRT)[1] and extension Transition based Rapidly exploring Random Trees (T-RRT)[4]. Each of those two methods will be described in sections 3.1 and 3.2 respectively. Initial path planning must find a feasible solution from the initial position of leader to the desired goal. However, these trajectories are often complex and low quality. The trajectory is proposed as low quality if it is much longer than optimal trajectory and is curvy. Reduction and optimization of the trajectory are needed. This is utilized by CFSQP library, which is described in chapter 4. RRT and T-RRT takes into account a kinematic model of a robot, more information can be found in [5,8].

Another method of finding an initial path could be geometric methods, for example visibility graph. Their disadvantage does not contain a kinematic model of robot. This means solution found can be unfeasible, when kinematic constraints are considered. The advantage of RRT is that it finds a solution, which is feasible.

3.1 RRT

RRT path planning method was found in 1998 [1]. The RRT path planning algorithm uses random approach to find a feasible solution. During path planning a kinematic model of a robot is considered. It is useful because any solution found is feasible. Another advantage is that it returns a control vector directly. However, it cannot guarantee the quality of the found solution when the found path is unnecessarily long. Too complex trajectories takes large amount time to optimize when compared to simpler ones. Because of this, trajectory reduction must be used.

We have a configuration space *C*. *C* contains obstacle space *Cobs*, *Cobs*$\subset$*C* and *Cfree*, *Cfree*=*C\Cobs*

RRT algorithm proceeds as follows:

1. Add start node *qinit* to the tree *T*

2. Then, generate random node *qrand* ∈ *Cfree*

3. Find nearest node *qnear* in tree *T*

4. Compute new state *qnew* ∈ *Cfree*, *qnew* lies in the direction of node *qrand*, and add this node to the tree *T*. Trajectory connecting *qnear* and *qnew* nodes must respect the constraints of the robot's model.

5. Repeat from step two, until stopping condition is reached. Stopping condition is when *qnew* lies within goal region.

To use RRT, we need to determine how to compute new state *qnew*. Geometric approach is used to find trajectory to get *qnew*. After that, only constant duration time interval is added to tree *T*.

Node *qnew* is determined as follows. First, we need to find the normal vector of *qnear* heading. Then, point between *qnear* and q new is found and normal vector of line intersecting *qnear* and *qnew* is created. Next, the intersection of those vectors is found. That point is centre of circle.

Horizontal speed v_t is constant during all simulations. Vertical speed v_n depends if qnew is higher or lower than *qnear*. Curvature c can be computed as $c = \frac{1}{r}$, where *r* is a distance between the intersection point and *qrand*.

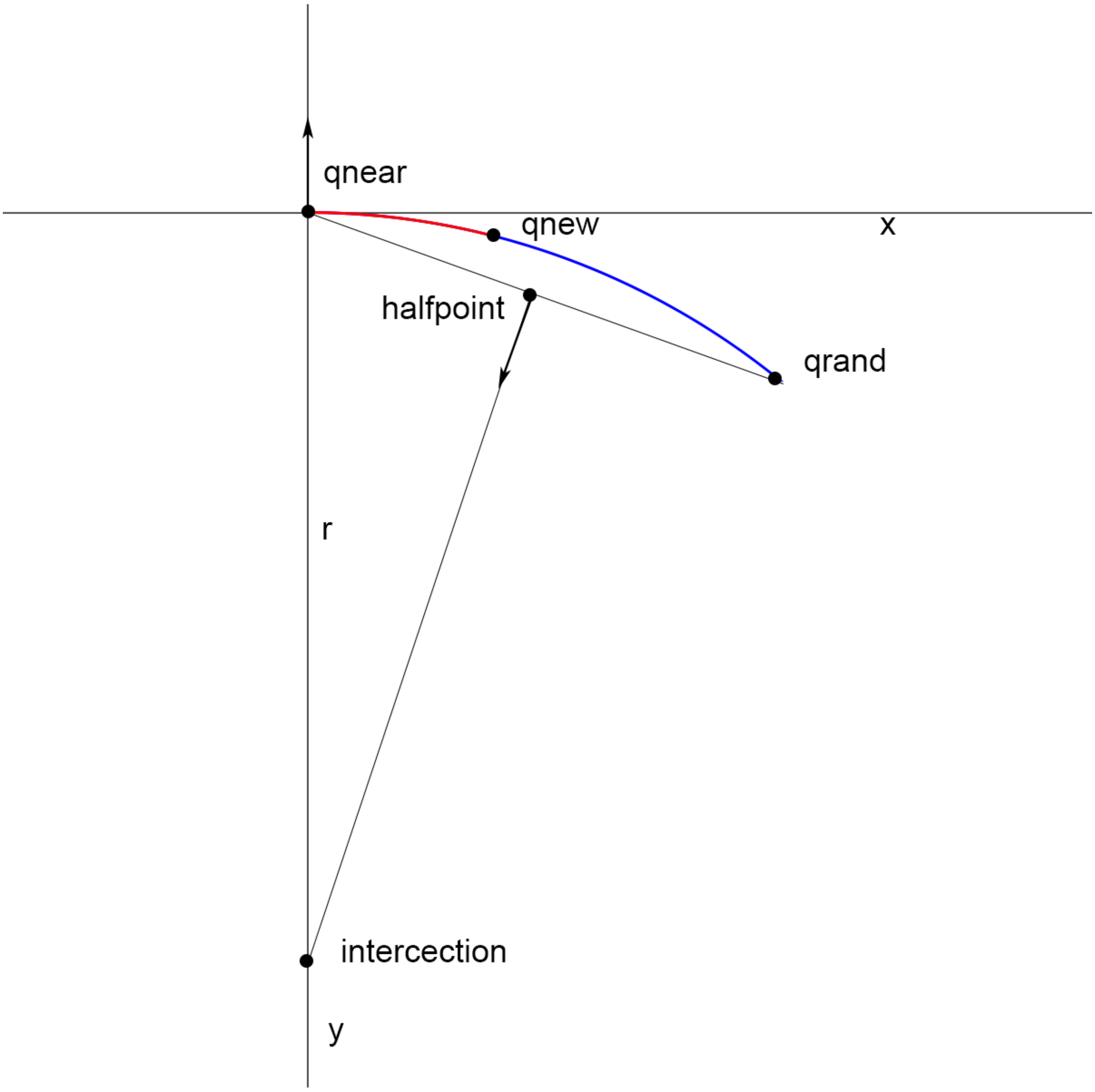

Figure 3.1: Computing control input.

During experiments with all subsystems together, it seemed that instead of constant time interval, scaling according to computed length is added. Time is added to the basic time interval, proportional to the length of the computed arc from *qnear* to *qnew* shown in figure 3.1 in red.

Also, to avoid unnecessary exploring in the wrong direction, a bias towards a goal is included. There is a probability that instead of generating random point, goal point is selected as new *qrand*. This improves performance of RRT slightly. However, in case which is shown in figure 3.2 will stuck at the border, if probability is too high.

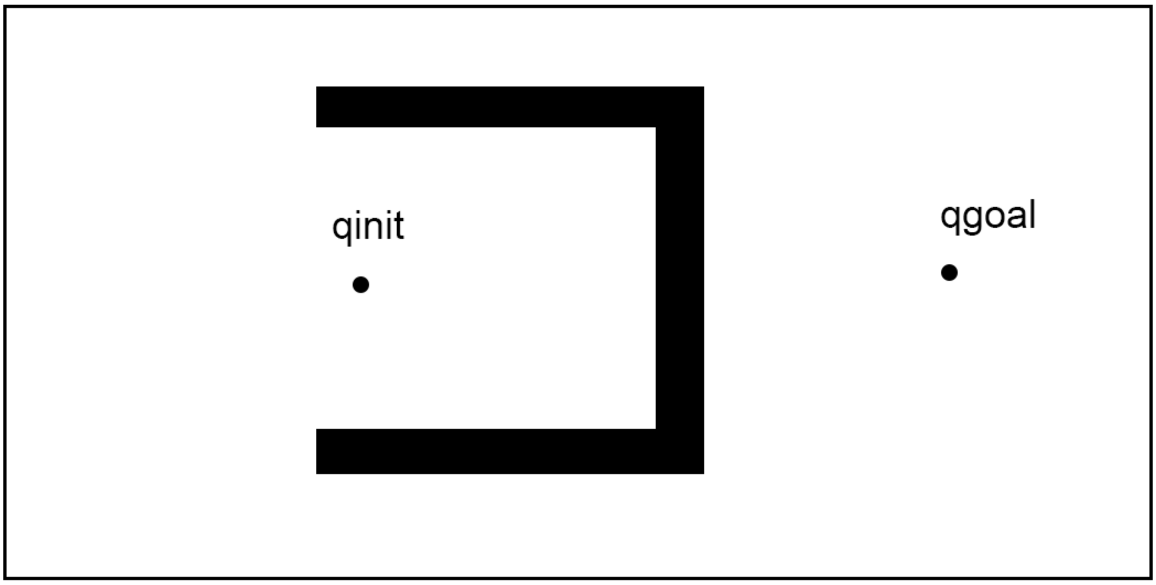

Figure 3.2: Bias to goal problem.

3.2 T-RRT

Transition based RRT is an extension of basic RRT. It was found in 2010 [4]. Addition to RRT is transition test and minimal expand control. Transition test evaluates if the newly generated edge should be accepted or not.

Advantage of Transition based RRT over RRT is that T-RRT includes cost function so trajectory will converge to goal area faster and is narrower.

T-RRT algorithm in steps:

1. Firstly, add initial point *qinit* to tree *T.*
2. Generate new point *qrand.*
3. Find nearest neighbor from tree *T.*
4. Then start transition test and Min Expand Control to test if new point *qnew* will be added to tree T or not.
5. Extend tree T to new point *qnew.*

3.2.1 Transition test

Firstly, trajectory with cost more than maximum cost *cmax* is filtered. If cost of new state is lower than its parent state, new trajectory is accepted. Otherwise, there is probability to accept new trajectory even if the cost is higher. This probability is represented:

$$p_{ij} = \exp\left(-\frac{\Delta c_{ij}}{KT}\right), if \Delta c_{ij} > 0, \qquad (3.1)$$

$$p_{ij} = 1, \qquad othewise.$$

$\Delta c_{ij} = (c_j - c_i)/d_{ij}$ is the slope of cost.

K is a constant value that is based on magnitude of cost.

T is a parameter named temperature. It controls a difficulty of transition test. Higher temperatures enables to climb steeper slopes, in contrast, lower enables to climb only slight slopes. T is tuned dynamically to reflect progress during exploring. Initial T is set to low value. If it fails to climb the slope before the maximum number of allowed fails $nFail_{max}$, is reached, temperature is multiplied by factor α, to enable climb steeper slopes. If a solution is found before $nFail_{max}$ is reached, temperature is divided by α.

3.2.2 Min Expand Control

Minimal expand function controls rate of exploring. Side effect of transition test could result in refining already explored area. To improve this behavior, Min Expand controls tries to ensure the rate of exploring. This is useful in complex environment, where we are easily trapped in one region. In simpler environment, it is not as important, because we are slowly proceeding to the goal without being confronted by obstacle.

It proceeds as follows:

1. If distance between *qnear* and *qrand* is greater than d, *qnew* is considered as exploring point and is added to tree

2. Otherwise, *qnew* is considered as tree refinement.

3. The point is not inserted into the tree if the ratio between refining points and all points is greater than threshold p.

As it will be shown in chapter 5, properties of T-RRT showed promising results in combination with reduction algorithm.

4 Trajectory reduction

In this chapter, reduction algorithm is described. Path obtained from RRT, or T-RRT is too complex to be used in the formation solver. Due to the high number of trajectory segments solver cannot find a solution. Thus, reduction algorithm must be applied, to reduce complexity of trajectory. Maximal number of trajectory segments depends on the complexity of trajectory and environment complexity. One approach is to take first S control inputs and try to reduce it to T, where T is number of control inputs in first segment. Then, move formation to point where reduced control inputs leads. Then, restart whole process of trajectory planning and reducing from new point. Another approach is to try reducing whole trajectory at once, and then moving formation. This is called hierarchical approach and is explained further.

In this hierarchical approach, we try to reduce the trajectory by removing trajectory segments. To do this, we use SQP optimizer. The goal is to minimize time of flight in this segment below threshold so we can remove it completely. This threshold is influenced by the length of trajectory and speed of MAV. Trajectory segment can be removed if time of flight of this segment compared to time of flight through the whole trajectory can be neglected.

During creation of algorithm, several approaches were tried. One of them was to divide trajectory to segments, and each of them will be reduced to fixed smaller amount. However, problem arose when we connected reduced segments together. Goal of trajectory did not lie in the goal area, so reduced trajectory was infeasible for MPC. When we tried cost function that takes in account distance to goal, optimizer failed to find next iteration step. For this reason, when reduction of first segment is done, remaining trajectory is replanned.

Reducing algorithm is as follows:

1. Take first S control segments from initial trajectory.
2. Start reducing on the first segment, until the segment is reduced to T.
3. From the point where the first segment ended, starts new planning using RRT.
4. The new trajectory consists of reduced segments and newly planned trajectory.
5. If the number of control inputs is greater than threshold, repeat from step one.

Experiments with formation control showed that around four trajectory elements will be handled. With more elements it can find solution, but it is dependent on initial trajectory and will consume much more time. During experiments good results was provided if S = 8 and T=3. The loop terminates when the trajectory is reduced to four elements.

A problem is to determine good cost function. If we consider a lot of parameters, such as curvature penalty, time of flight, etc., SQP optimizer will fail to find a solution. So, the simplified cost function is implemented, containing two basic parameters. That parameter is a minimal value and distance to current goal.

From control vector, time interval with shortest duration is taken. The optimizer tries to minimize this value. If it is lower than the threshold, control input could be completely removed, thus reducing the number of control inputs. It can be removed because of its short duration, it has very little effect on resulting trajectory.

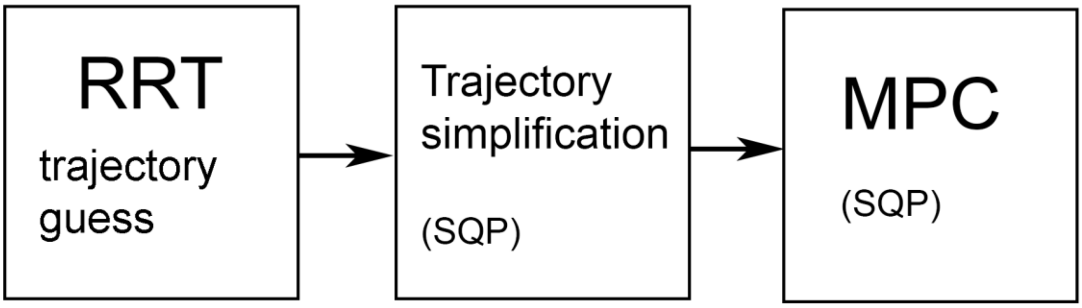

Figure 4.1: Flow of trajectory processing.

5 Experimental results

In this section, experimental result of initial trajectory planning and reduction are shown. Results of RRT and T-RRT initial trajectory planning in space without obstacles and with obstacles are presented. After planning, trajectory reduction and formation keeping is presented.

In all cases, as a goal region is chosen point (9, 6.5, 2) and initial position is (-3, -2, -1). These points are marked on figures as black dots. Red crosses are points where quadrotor moves after applying one control element. Leader's trajectory is shown in blue.

5.1 RRT with no obstacles

In figure 5.1, Tree generated by RRT is shown. Note how RRT explores and expand in whole space. This means we cannot guarantee quality of the path. This view focuses only on X-Y plane for better visibility.

In figure 5.2, 5.3, 5.4, trajectory is shown in space and how it is affected in each step. Red crosses are points, where new control input begins. Trajectory generated by RRT in figure 5.2 has a lot of control inputs, and in this way it cannot be handled by formation solver. In figure 5.3, we can see the trajectory after reduction process. Note, that trajectory is much narrower and has only few control inputs.

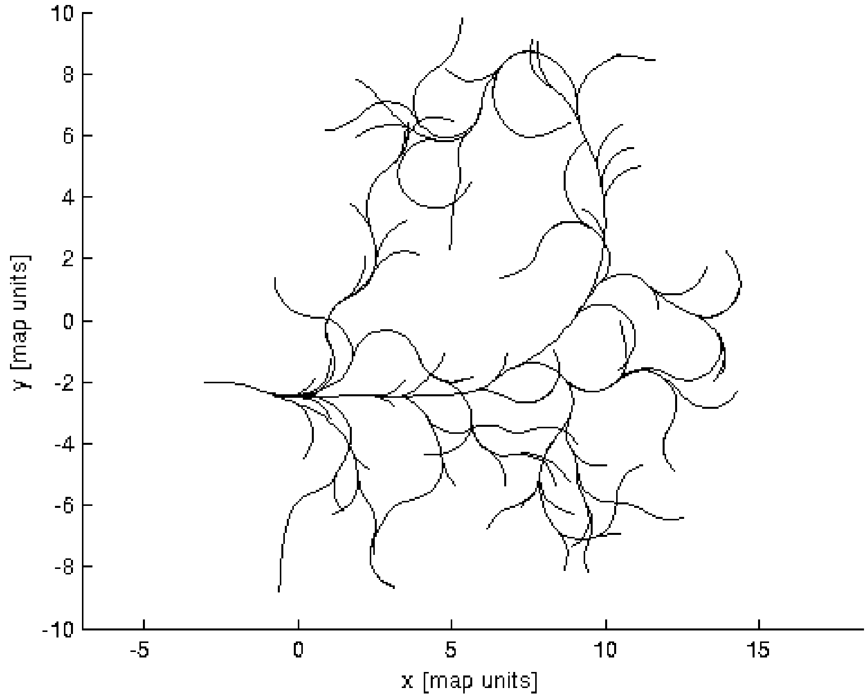

Figure 5.1: Example of generated RRT tree, after 300 iterations.

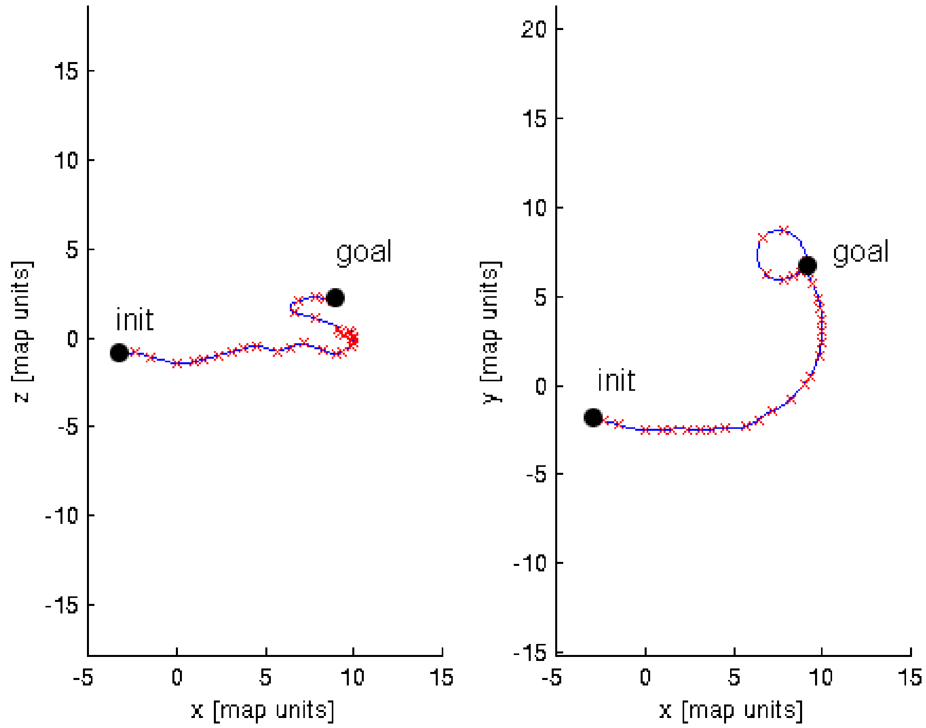

Figure 5.2: Trajectory generated by RRT. Trajectory is cluttered and crooked.

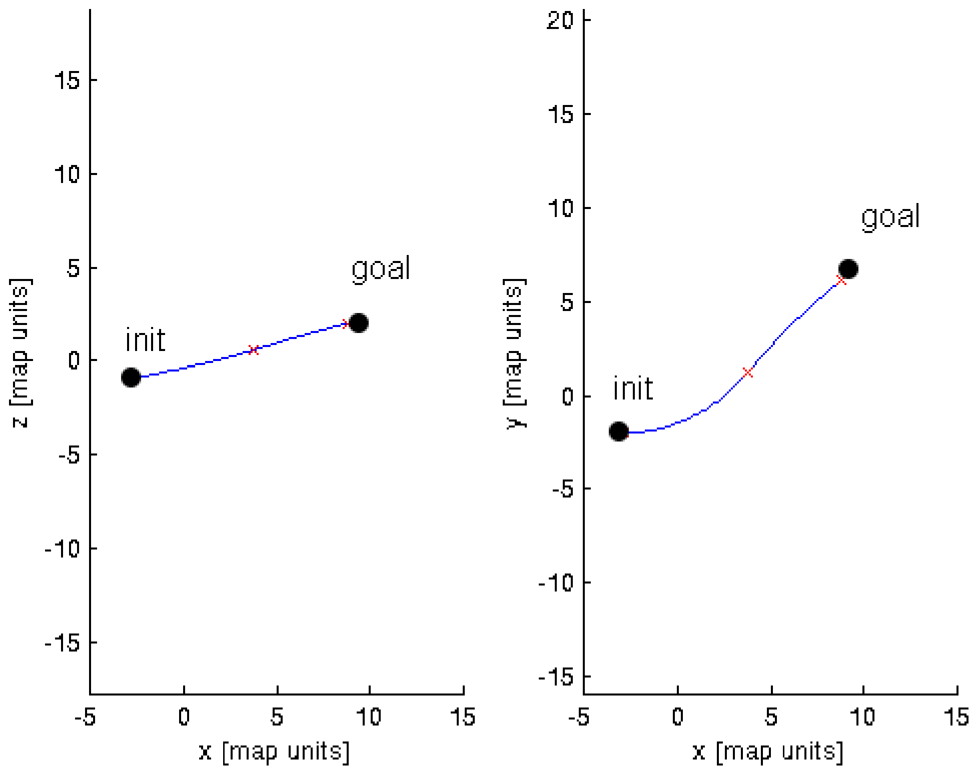

Figure 5.3: Reduced trajectory. Without obstacles, reduction result is adequate even for RRT.

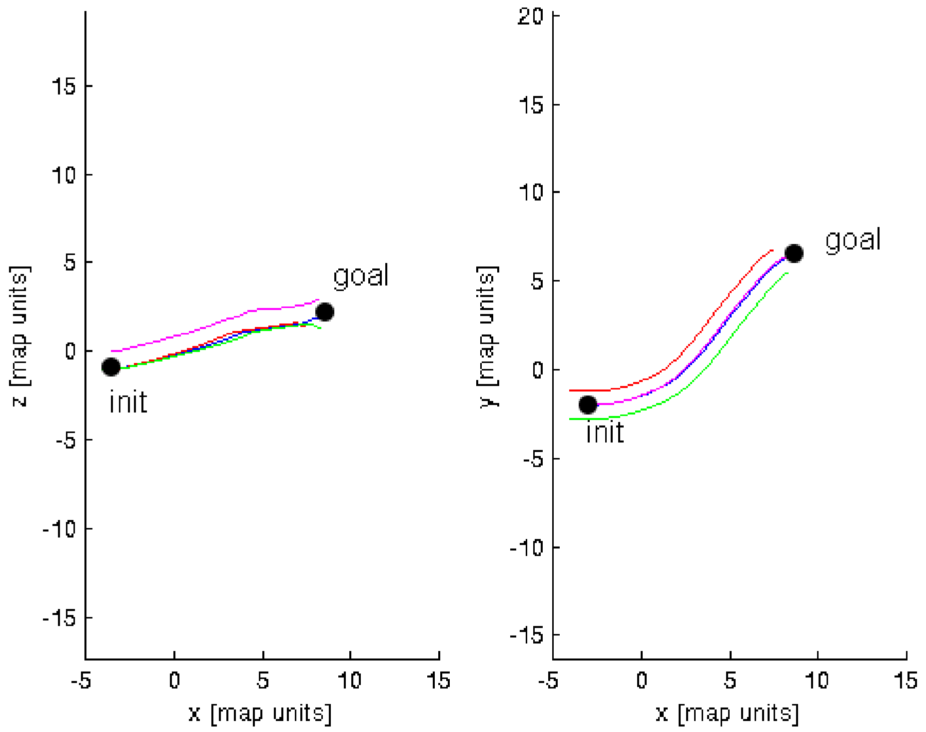

Figure 5.4: Final trajectory, blue – leader

5.2 T-RRT empty space

In this case, instead of using RRT, Transition based RRT was used. In picture 5.5 is visible that searching is strongly biased towards goal. Areas that lie away from goal lead only to more complex and long trajectories. Resulting trajectories have better quality than RRT, and leads to better results.

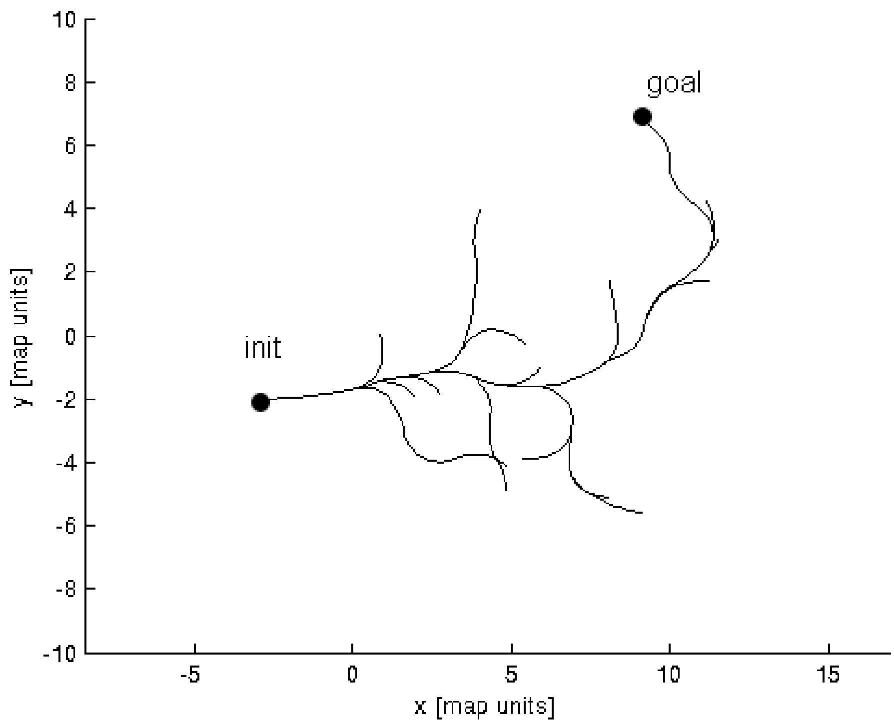

Figure 5.5: Exploration tree of T-RRT. In this case, T-RRT finds solution quickly without exploring other ideas, which is good behavior for next processing.

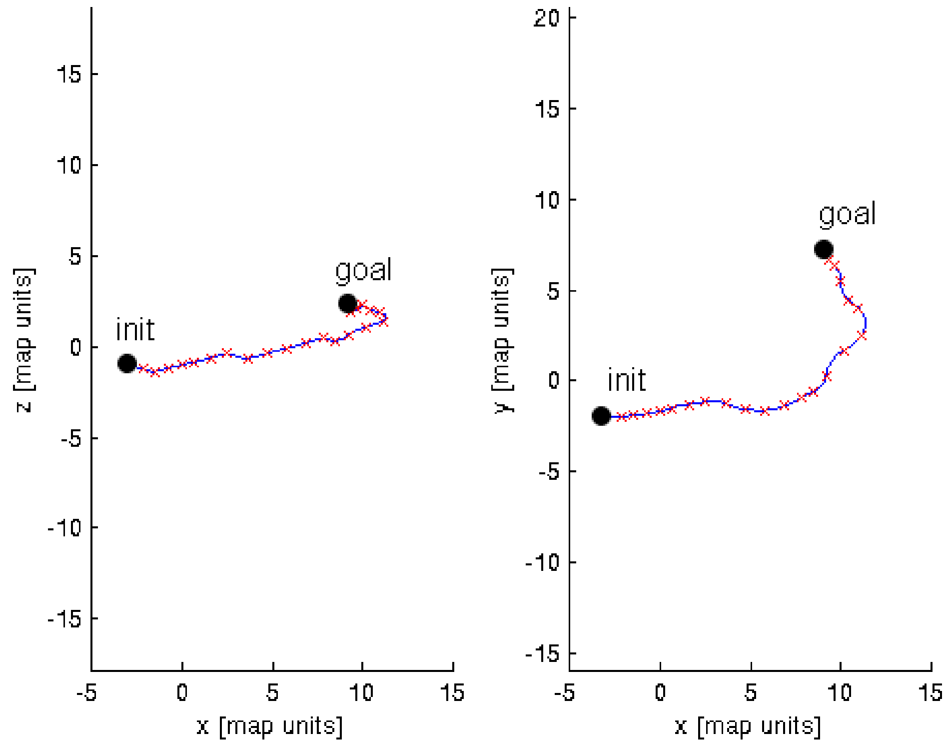

Figure 5.6: Trajectory generated by T-RRT. Too many control inputs denoted by red crosses.

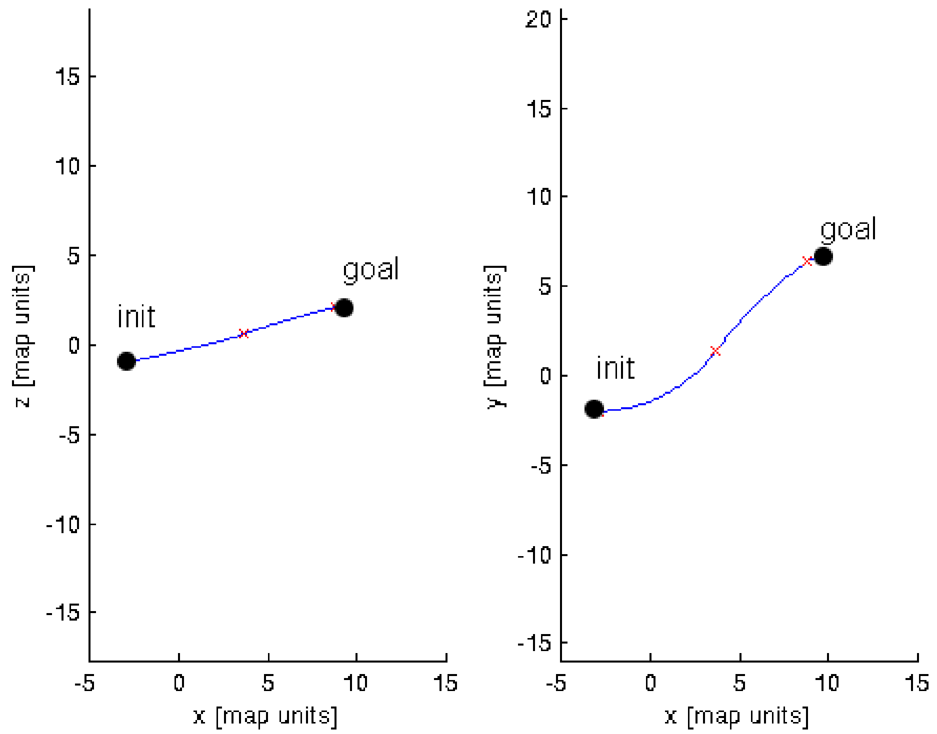

Figure 5.7: Reduced trajectory. Note in this step that trajectory has good quality.

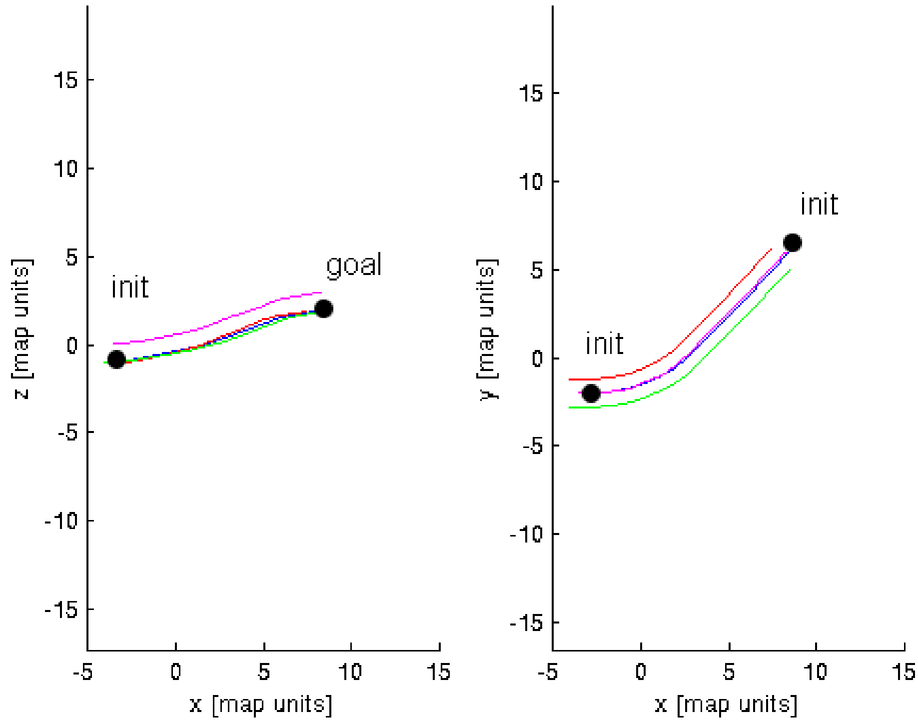

Figure 5.8: Movement of formation. Without obstacles, movement is almost straight.

5.3 RRT with obstacles

In environment with some obstacles, RRT is used. In figure 5.12 can be seen how formation turns away from obstacles. It can also be seen on 5.11, during reduction.

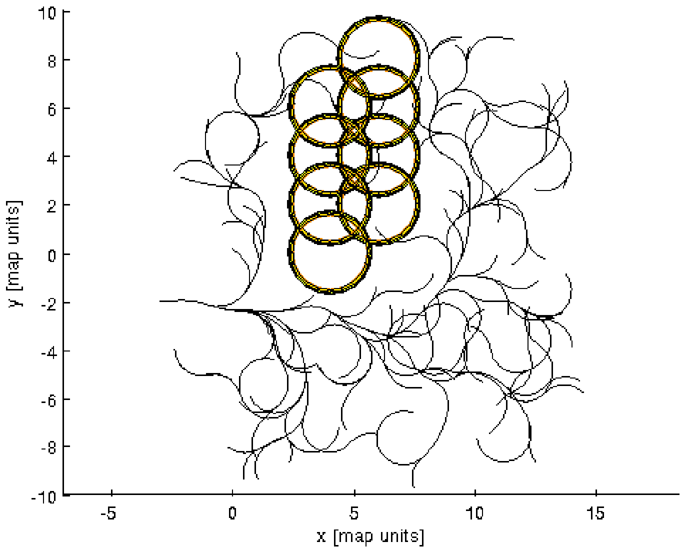

Figure 5.9: RRT tree tries to explore the whole area, which results in longer trajectories.

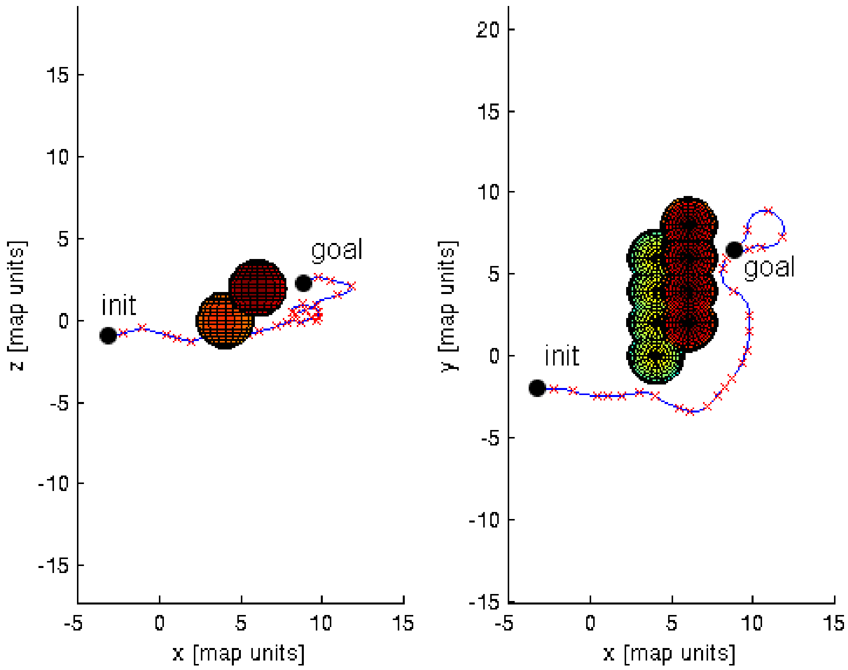

Figure 5.10: Initial trajectory

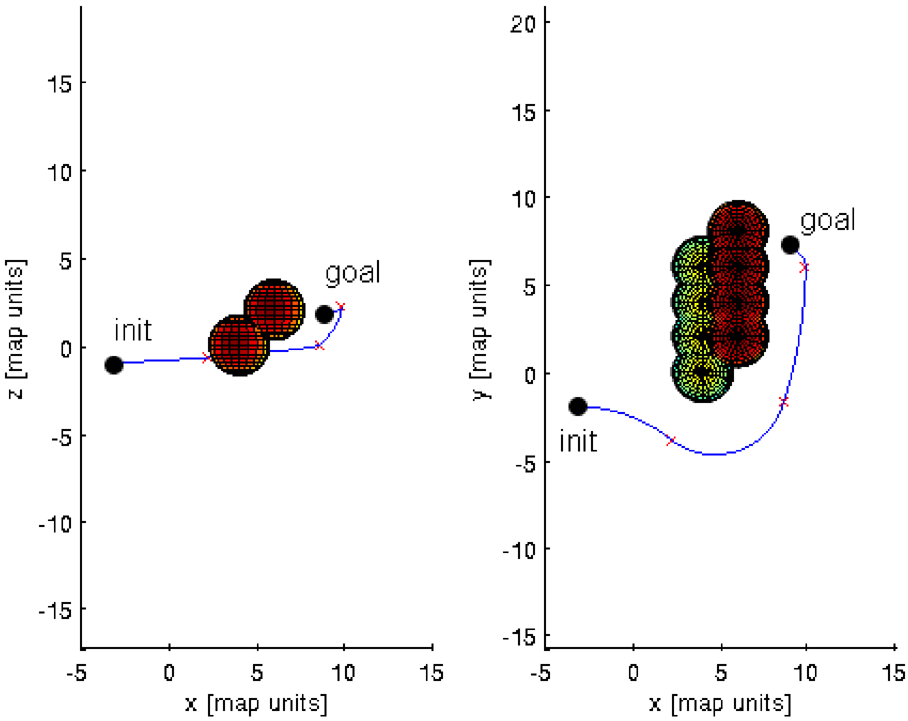

Figure 5.11: Reduced trajectory

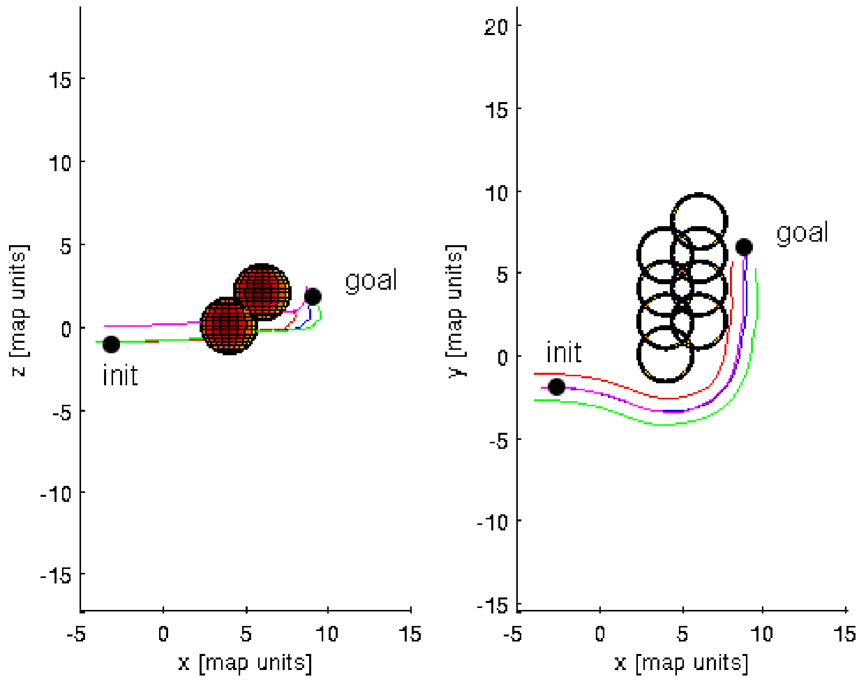

Figure 5.12: Formation keeping.

5.4 T-RRT with obstacles

In last experiment, T-RRT approach is chosen again, but with obstacles in place. In figure 5.13 tree is shown. Quality is better than if we used only RRT.

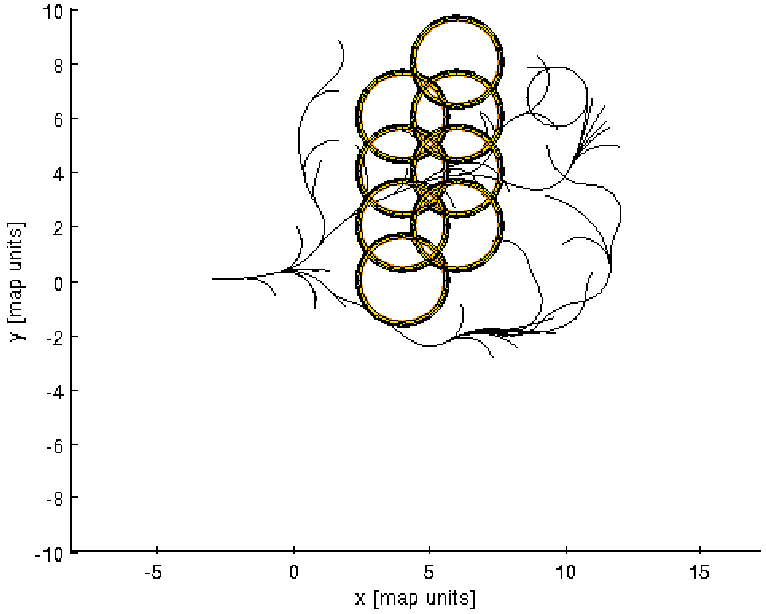

Figure 5.13: T-RRT Tree. Path that is crossing obstacles is going under them.

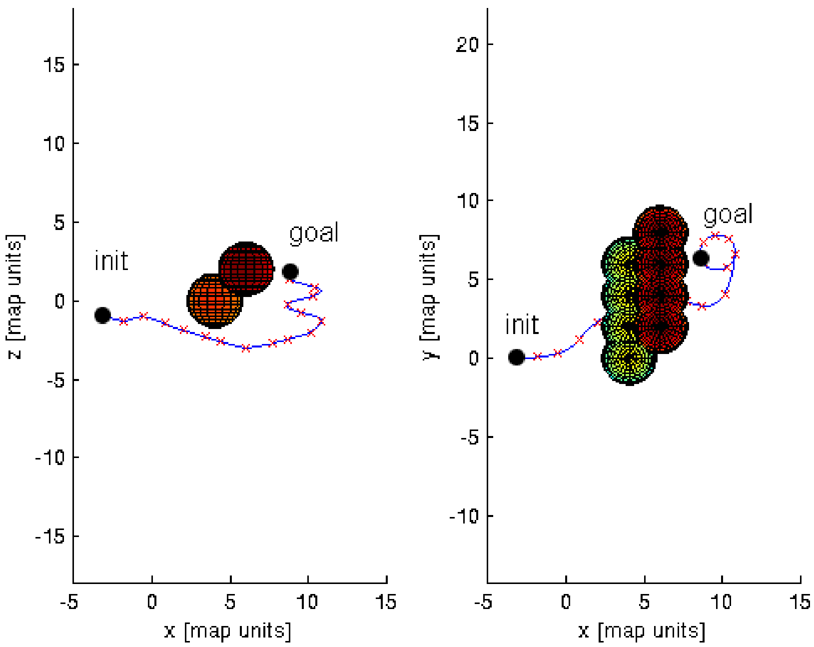

Figure 5.14: Initial trajectory. Initial trajectory leads under the obstacles.

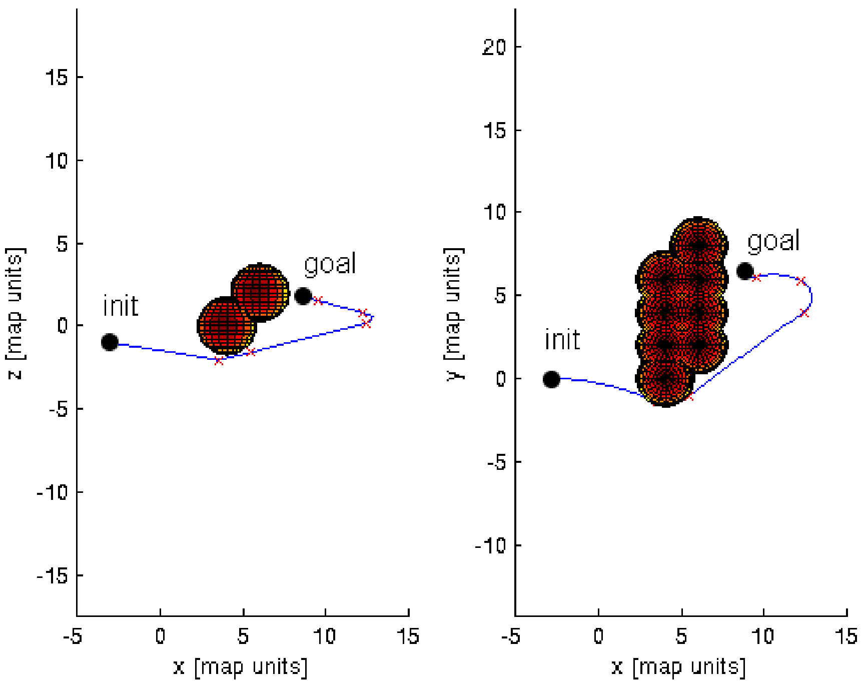

Figure 5.15 Reduced trajectory. Number of control inputs is reduced and trajectory is diverted around the obstacles.

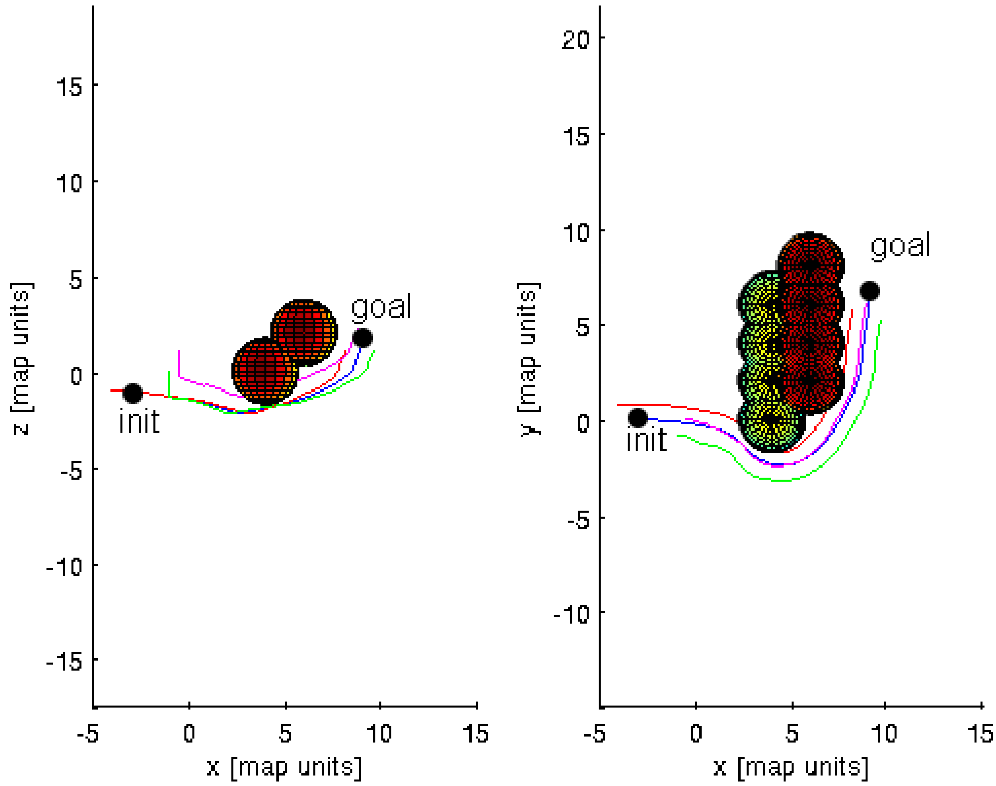

Figure 5.16: Formation keeping. From initial path plan, which has many control inputs, resulting formation movement is clean and smooth.

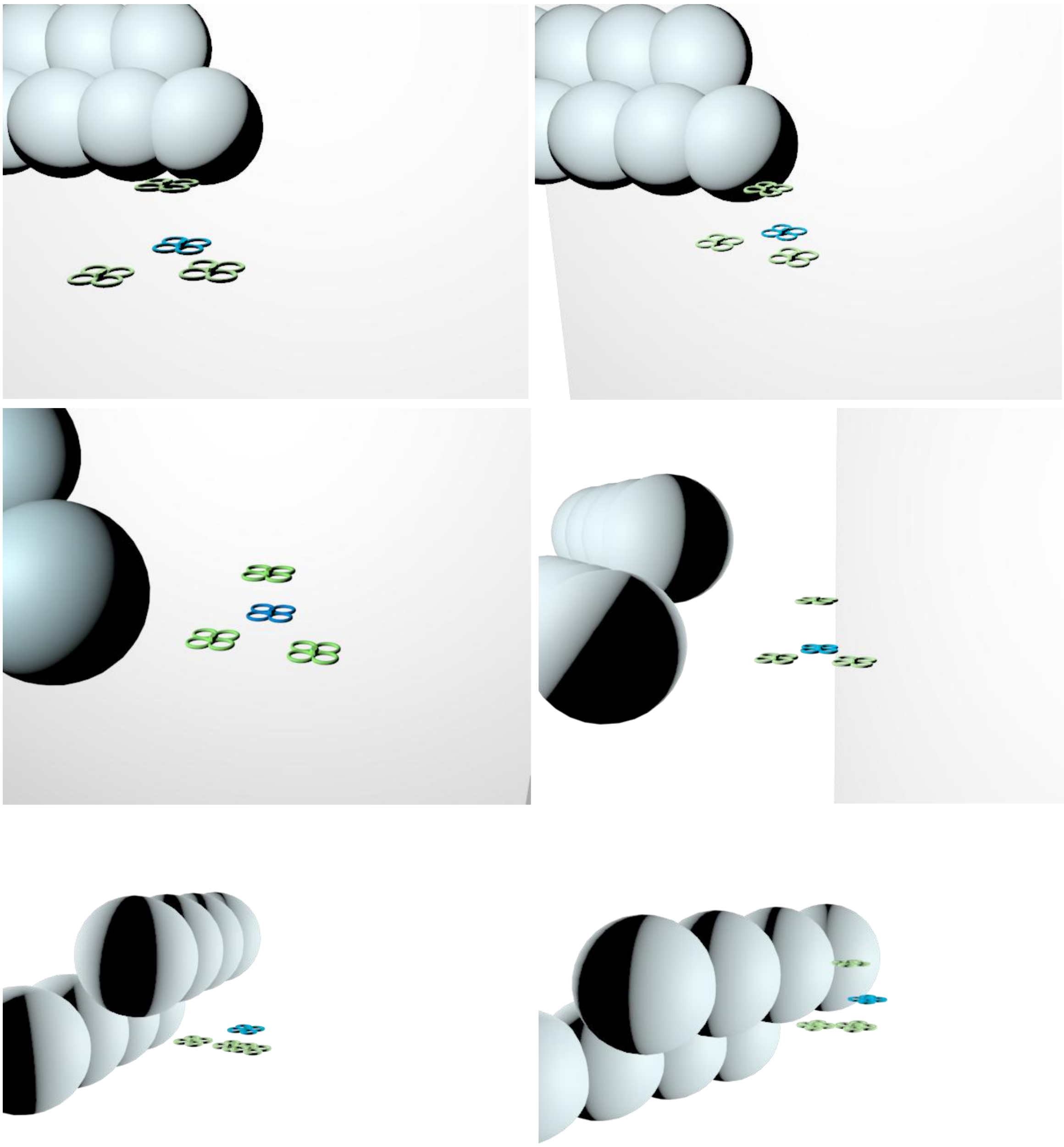

Figure 5.17: Visualization of a formation flying around obstacles. The blue quadrotor is leader and followers are green.

5.5 Summary

During experiments, some of properties and limitation occurred. T-RRT is more suitable for this problem than classic RRT thanks to its informed exploration. Classic RRT explores whole space, which is unnecessary and leads to longer trajectories.

Experiments showed that the initial trajectory plan has a lot of trajectory elements. In this form, formation control optimization takes a large amount of time. With reduced trajectory time needed is shorter. One of approaches used here tries to reduce whole trajectory at once. This approach was tested in experiments. Experiments showed that framework is able to plan the trajectory in environment with obstacles and reduce the trajectory to reduce time needed to drive formation.

6 Conclusion

The goal was to design algorithm for initial trajectory planning of MAV formations. Rapidly exploring Random Trees (RRT)[1] and Transition based RRT (T-RRT)[4] has been implemented. Description of algorithms is described in section 4. This approach was chosen because it respects kinematic constraints of MAV and generated trajectory is represented directly by control input. Model predictive control uses control input to move MAV. Because of nature of randomly generated paths, trajectory reduction is needed to be suitable for formation control that. To reduce trajectory, Sequential Quadratic Programming (SQP) [2] optimizer was used.

Results of simulations are presented in chapter 5. Created algorithm was tested in environment without obstacles and with obstacles. Each of the methods, RRT and T-RRT is shown. It can be seen here, even if initial trajectory is complex and is long, a reduction step creates a trajectory which is shorter and has less control inputs. The result is moving formation from the start to the goal. Proposed contribution of this work is reducing the time needed for optimization and trajectory planning through complex environment.

Another approach to trajectory reduction is to take interval, reduce it and move the whole formation to a new position, instead of trying to reduce the whole trajectory. One of the problems with testing is that better environment representation needed to reflect real-life conditions. For simulations, sphere obstacles were used.

Possible extension is to find a better metric for choosing nearest neighbor in RRT and T-RRT. A simple Euclidian metric works for non constrained models. Quadrocopter has kinematic model and its movement is constrained. A better metric could be implemented, as proposed in [6] and [7].

7 Bibliography

[1] S. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical report, TR 98-11, Computer Science Dept., Iowa State University, 1998.

[2] C. Lawrence, J. Zhou, and A. Tits. User's guide for cfsqp version 2.5. University of Maryland, 1997.

[3] M. Saska, Z. Kasl, L.Preucil: Motion planning and kontrol of formations of micro aerial vehicles. Accepted for IFAC World Congress 2014

[4] Léonard Jaillet, Juan Cortés, and Thierry Siméon: Sampling-Based Path Planning on Configuration-Space Costmaps, IEEE TRANSACTIONS ON ROBOTICS, VOL. 26, NO. 4, AUGUST 2010

[5] Romain Pepy, Alain Lambert and Hugues Mounier: Path Planning using a Dynamic Vehicle Model, Institut d'Electronique Fondamentale, 2006

[6] Alejandro Perez, Robert Platt Jr., George Konidaris, Leslie Kaelbling and Tomas Lozano-Perez, LQR-RRT* : Optimal Sampling-Based Motion Planning with Automatically Derived Extension Heuristics, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 2012

[7] Peng Cheng: Reducing RRT metric sensitivity for motion planning with differential constraints, Iowa State University, 2001

[8] S.M. LaValle: Planning algorithms. Cambridge University Press, 2006

[9] Petr Vaněk: Motion planning for Formations of Mobile Robots and Unmanned Aerial Vehicles, Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Cybernetics, 2012

[10] T. Lee, M. Leoky, and N. McClamroch. Geometric tracking control of a quadrotor uav on se (3). In Decision and Control (CDC), 2010 49th IEEE Conference on, pages 5420{5425. IEEE, 2010.

8 CD content

Directory	Content
application	source code of application
rakfilip_BT.pdf	electronic version of this paper