2016 16th International Conference on Control, Automation and Systems (ICCAS 2016)
Oct. 16–19, 2016 in HICO, Gyeongju, Korea

Model Predictive Control-based Trajectory Planning for Quadrotors with State and Input Constraints

Penghong Lin[1], Songlin Chen[2*] and Chang Liu

[1]Control and Simulation Center, Harbin Institute of Technology, Harbin 150001, China.
[2] The 41st Institute Of Fourth Academy Of Aerospace Science And Technology Corporation, Xi'an 710025, China.
(Linpenghong@qq.com, songlin@hit.edu.cn, lccayz@126.com)

Abstract: A novel scheme to solve the trajectory planning problem for quadrotors with model and state constraints is proposed. First the RRT (Rapidly-exploring Random Tree) algorithm is employed to generate an initial route in context of a 3D environment. Then with the model of quadrotoer, the MPC (Model Predictive Control) method is used to construct an inner simulator which can generate the trajectories satisfying the state constraints by following the waypoints exported by the RRT solution. Finally, the trajectory generation of quadrotors with state and input constraints is transformed into a general form of QP problem, which can be solved expediently by common numerical algorithms. The proposed scheme can generates high-quality and feasible trajectories satisfying both state constraints and input constraints, which is verified sufficiently by extensive simulations.

Keywords: quadrotor, state constrained, trajectory planning, MPC, RRT

1. INTRODUCTION

In recent years, due to the applications of new energy and new materials, quadrotor has been extensively studied and popularized because of its compact structure, large thrust, strong hovering ability, high dynamic characteristics and good robustness. It has been widely used in military and civil fields, such as delivery service, aerial photography, disaster rescue and indoor anti-terrorist etc. Due to the unique advantages of quadrotor, a lots of institutes all over the world are carrying out research and development on such aircraft [1−4]. One of main research directions is autonomous flight, which gives quadrotors the ability to fly automatically which are the foundation to complete many tasks.

In the low speed trajectory planning, due to the good maneuverability of the quadrotor, we almost do not need to consider the state constraints and input constraints. In most missions, however, the quadrotor is required to cruise at a speed of more than 10m/s. Thus, the state constraints and input constraints become very important. If these constraints are not taken into account in high speed autonomous navigation, the security of quadrotor can not be guaranteed, which may lead to the failure of the mission.

Trajectory planning is one of the basic problems of quadrotor's autonomous navigation. Trajectory planning is usually composed of two parts. The first part is the path searching, and the second part is the trajectory fitting. A path always consists of a set of waypoints, while a trajectory can be seen as a time parametrization of the path. The goal of path searching is to find a connection from the beginning to the end in the feasible region. There are many algorithms to deal with path searching, for example, A* algorithm [6], RRT algorithm [7] and Dijkstra algorithm [8] etc. The goal of trajectory fitting is to fit the original path with a smooth curve with time parameters and the polynomial interpolation is a general method. Although there already exist some algorithms for trajectory planning for quadrotor, there, to our best knowledge, has yet to emerge a single and effective algorithm that can both deal with state constraints and input constraints. Reference [5] proposed a kind of polynomial-based trajectory planning algorithm for quadrotor that satisfies the terminal constraint. But duo to the lack of the use of model, this algorithm is not able to deal with the state constraints and input constraints.

Aiming to solve the trajectory planning problem for quadrotors with model-related state and input constraints, this paper proposes a novel trajectory planning scheme by combining RRT algorithm and MPC method. In this scheme, for given 3D grid map and terminal constraints, the RRT algorithm is adopted to produce a set of waypoints. With these waypoints and a given mathematical model of quadrotor, An inner simulator based on model predictive control is constructed to generate the trajectory by waypoints tracking. Finally, one can get a feasible trajectory that satisfies the model and state constraints as long as the mathematical model is available. This scheme can extend effectively the ability of quadrotor to execute some special task, such as plant protection and aerial photo.

The arrangement of the paper is as follows. The problem formulation is stated in Section 2. The MPC-based trajectory planning scheme is designed in Section 3. In Section 4 simulations are presented. At last, a brief summary of the full text is given in Section 5.

* Address all correspondence to this author. This work is supported by National Nature Science Foundation of China (61333001).

2. QUADROTOR DYNAMICS AND STATE CONSTRAINTS

2.1 Problem statement

Given a 3D grid map of an environment, as shown in figure 1, the initial state of quadrotor and the terminal state of quadrotor, we wish to compute physically feasible and trackable trajectories that satisfy the state constraints and input constraints.

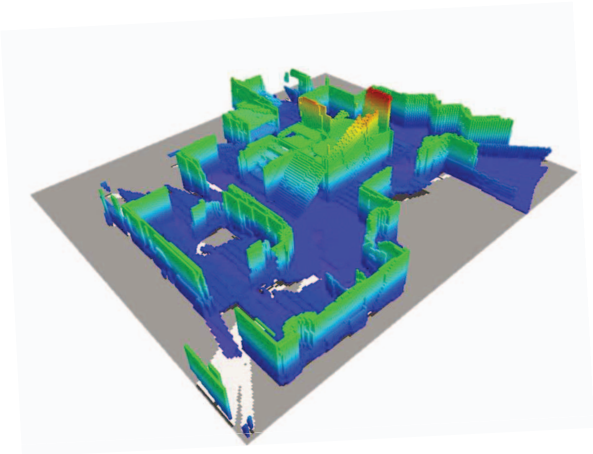

Fig. 1 Diagram of 3D gridmap

2.2 Quadrotor dynamics

In recent years, the dynamic modelling of quadrotor has been widely, and the usual mathematical models include convolution model, linear state space model, fuzzy model, neural network model and chaotic model etc. Considering the computational complexity, we tend to use the linear state space model. The coordinate systems of quadrotor are given in figure 2 and the symbols to be used in this paper are defined in table 1.

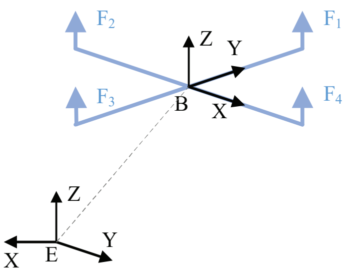

Fig. 2 Definition of coordinate systems

According to Newtons second law and the aircraft dynamics principle, we can obtain the system dynamic equations as follows [9].

Table 1 Parameter Definition

Symbol	Definition
α	Rotation angle around X axis
β	Rotation angle around Y axis
γ	Rotation angle around Z axis
R	Rotation matrix
F_E	Force vector denoted in body coordinate
F_B	Force vector denoted in earth coordinate
m	Mass of quadrotor
J_x	Body inertia around X axis
J_y	Body inertia around Y axis
J_z	Body inertia around Z axis
M	Resultant moment applied to quadrotor
H	Angular momentum
F	Resultant force applied to quadrotor
F_i	Force produced by the ith rotor
L	Length between two rotors on the diagonal
k_f	Length coefficient
V	Vehicle velocity

$$\begin{cases} \ddot{x} = (\cos\alpha sin\beta cos\gamma + sin\alpha sin\gamma)\, U_1/m \\ \ddot{y} = (\cos\alpha sin\beta \sin\gamma - sin\alpha \cos\gamma)\, U_1/m \\ \ddot{z} = -g + \cos\alpha\cos\beta U_1/m \\ \ddot{\alpha} = \left[\dot{\beta}\dot{\gamma}\,(J_y - J_z) + LU_2\right] \Big/ J_x \\ \ddot{\beta} = [\dot{\alpha}\dot{\gamma}\,(J_z - J_x) + LU_3]/J_y \\ \ddot{\gamma} = \left[\dot{\alpha}\dot{\beta}\,(J_x - J_y) + k_f U_4\right] \Big/ J_z \end{cases} \tag{1}$$

where

$$\begin{cases} U_1 = F_1 + F_2 + F_3 + F_4 \\ U_2 = F_1 - F_3 \\ U_3 = F_2 - F_4 \\ U_4 = F_1 - F_2 + F_3 - F_4 \end{cases}$$

As can be seen from the above formula, quadrotor's dynamic model is nonlinear and under-actuated.

2.3 Description of Constraints

In the actual systems, there are many different state constraints. For the quadrotor, the lift force of each motor is limited, so such limitation can be described as follows.

$$F_i \in [0,\ F_{\max}],\ i \in \{1, 2, 3, 4\} \tag{2}$$

where $F_{\max}$ is the maximum thrust provided by each single motor.

Considering the autonomous obstacle avoidance, the position of quadrotor is constrained.

$$(x(t), y(t), z(t)) \in O_{3,\ free} \tag{3}$$

where $O_{3,\ free}$ defines the three dimensional space without obstacles.

Considering the initial condition, the starting state and final state of the trajectory are constrained.

$$\begin{cases} x(0)=x_0,\ y(0)=y_0,\ z_{des}(0)=z_0 \\ \dot{x}(0)=\dot{x}_0,\ \dot{y}(0)=\dot{y}_0,\ \dot{z}_{des}(0)=\dot{z}_0 \\ \ddot{x}(0)=\ddot{x}_0,\ \ddot{y}(0)=\ddot{y}_0,\ \ddot{z}_{des}(0)=\ddot{z}_0 \\ x(T)=x_T,\ y(T)=y_T,\ z_{des}(T)=z_T \end{cases} \tag{4}$$

Considering the aerial photography, the maximum of angular and angular velocity is constrained.

$$\begin{cases} |\alpha_{des}(t)| < \alpha_{\max},\ |\beta_{des}(t)| < \beta_{\max} \\ |\dot{\alpha}_{des}(t)| < \dot{\alpha}_{\max},\ \left|\dot{\beta}_{des}(t)\right| < \dot{\beta}_{\max} \end{cases} \tag{5}$$

3. TRAJECTORY PLANNING BASED ON MODEL PREDICTIVE CONTROL

3.1 Solution outline

In this paper, we first use RRT algorithm to create a set of waypoints. Then we design a model predictive controller based on the mathematical model of quadrotor, and further an inner simulator is constructed via fourth-order Runge-Kutta algorithm. For given 3D grid map and terminal state constraints, the inner simulator can generate the feasible trajectory by tracking those waypoints exported by RRT algorithm. Once the trajectories are generated successfully, all constraints can be satisfied completely. The general framework of the scheme is shown in figure 3.

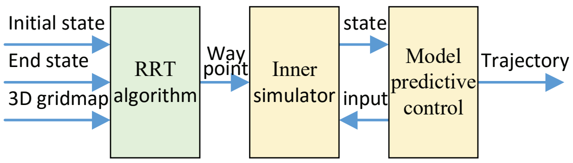

Fig. 3 Trajectory planning system

3.2 Inner simulator

In this paper, we use a inner simulator, which contains a model predictive controller, to generate the desirable trajectory of quadrotor. In order to solve numerical integration equation, the fourth-order Runge-Kutta algorithm [10] is adopted. Given a state space $\dot{x} = f(x)$ and the state x_k at time k, we can compute the state at time $k+1$ by

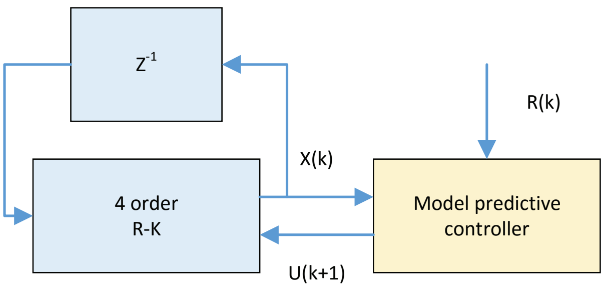

Fig. 4 Inner simulator

$$x_{k+1} = x_k + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{6}$$

where

$$\begin{cases} k_1 = f(x_k) \\ k_2 = f(x_k + \frac{1}{2}hk_1) \\ k_3 = f(x_k + \frac{1}{2}hk_2) \\ k_4 = f(x_k + hk_3) \end{cases}$$

3.3 Model predictive controller

In order to facilitate the of controller design of quadrotor, the control system is divided into two subsystems, i.e. the attitude control system and position control system, as shown in figure 5.

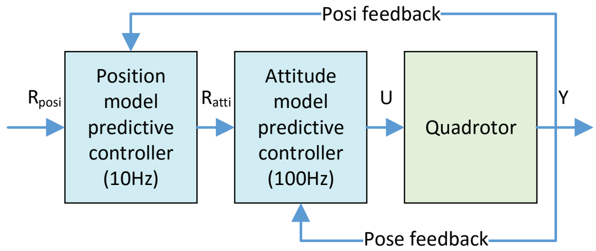

Fig. 5 Subsystems

Model predictive control is a kind of optimization control strategy build in uncertain environments. It can ensure some optimal performance by choosing some particular cost functions. Further more, MPC is good at dealing with many kinds of constraints, including input/output constraints and state constraints, performing the advantage of strong robustness. The general MPC algorithm includes 3 parts: model based prediction, open-loop optimization and feedback correction [11].

First of all, we need to design the inner loop of the controller, namely attitude controller. Formulation (7) is separated from the quadrotor dynamics, representing the state space of attitude subsystem. In order to reduce the complexity, we further linearise the model by Taylor expansion. Thus, the attitude subsystem can be seen as a linear time-varying system.

$$\begin{cases} \dot{x_a} = A_t x_a + B_t u_a \\ y_a = C_a x_a \end{cases} \quad \text{where}$$

$$\tilde{A}_a = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & A_{8,10} & 0 & A_{8,12} \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & A_{10,8} & 0 & 0 & 0 & A_{10,12} \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & A_{12,8} & 0 & A_{12,10} & 0 & 0 \end{bmatrix}$$

$$\tilde{B}_a = \begin{bmatrix} 0 & 0 & 0 \\ \frac{L}{J_x} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{L}{J_y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{L_z}{J_z} \end{bmatrix}$$

$$\tilde{C}_a \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

$$x_a = \begin{bmatrix} x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} \triangleq \begin{bmatrix} \alpha \\ \dot{\alpha} \\ \beta \\ \dot{\beta} \\ \gamma \\ \dot{\gamma} \end{bmatrix}$$

$$u_a = \begin{bmatrix} 0 \\ U_2 \\ 0 \\ U_3 \\ 0 \\ U_4 \end{bmatrix}$$

$$y_a = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

$$\begin{array}{l} A_{8,10} = \left(\frac{J_y - J_z}{J_x}\right) x_{12},\ A_{8,12} = \left(\frac{J_y - J_z}{J_x}\right) x_{10}, \\ A_{10,8} = \left(\frac{J_z - J_x}{J_y}\right) x_{12},\ A_{10,12} \left(\frac{J_z - J_x}{J_y}\right) x_8, \\ A_{12,8} = \left(\frac{J_x - J_y}{J_z}\right) x_{10},\ A_{12,10} \left(\frac{J_x - J_y}{J_z}\right) x_8 \end{array}$$

In the above formulation, parameters in $\tilde{A}_a$ can be seen as time-varying variables, being updated from feedback at each control period. Then discretizing the linearised model and modifying the input as incremental form, we get

$$\left\{ \begin{array}{l} \chi_{a,k+1} = A(\chi_{a,k})\chi_{a,k} + B(\chi_{a,k})\Delta u_{a,k} \\ y_{a,k} = C(\chi_{a,k})\chi_{a,k} \end{array} \right. \tag{7}$$

where

$$A(\chi_{a,k}) = \begin{bmatrix} A(x_{a,k}) & B(x_{a,k}) \\ 0 & I \end{bmatrix},$$

$$B(\chi_{a,k}) = \begin{bmatrix} B(x_{a,k}) \\ I \end{bmatrix},$$

$$C(\chi_{a,k}) = \begin{bmatrix} C(x_{a,k}) & 0 \end{bmatrix},$$

$$\chi_{a,k+1} = \begin{bmatrix} x_{a,k} \\ u_{k-1} \end{bmatrix},\ \Delta u_k = u_k - u_{k-1}.$$

Set m as the number of controlling horizon and p as the prediction horizon, let

$$\left\{ \begin{array}{l} X_{a,k+1} = [\chi_{a,k+1}^T, ..., \chi_{a,k+p}^T]^T \\ R_{a,k+1} = [r_{a,k}^T, ..., r_{a,k+p}^T]^T \\ Y_{a,k} = [y_{a,k+1}^T, ..., y_{a,k+p}^T]^T \\ \Delta U_{a,k} = [\Delta u_{a,k}^T, ..., \Delta u_{a,k+m-1}^T]^T \end{array} \right. \tag{8}$$

Considering the tracking error and the changing rate of inputs, we choose the following cost function

$$\begin{array}{l} J_{a,k} = \Delta U_{a,k}^T \Gamma_{a,u} \Delta U_{a,k} \\ \quad +(R_{a,k+1} - Y_{a,k+1})^T \Gamma_{a,y} (R_{a,k+1} - Y_{a,k+1}) \end{array} \tag{9}$$

where

$$Y_{a,k+1} = \Phi_{a,k} A_{a,k} \chi_{a,k} + S_{a,k} \Delta U_{a,k},$$

$$\Phi_{a,k} = \Theta_{a,k} \Omega_{a,k},$$

$$S_{a,k} = \Theta_{a,k} \Psi_{a,k},$$

$$\Theta_{a,k} = diag(C_{k+1}, C_{k+2}, ..., C_{k+p}),$$

$$\Omega_{a,k} = \begin{bmatrix} {}_1^0A_{k+i}^T & {}_1^1A_{k+i}^T & \cdots & {}_1^{p-1}A_{k+i}^T \end{bmatrix},$$

$$\Psi_{a,k} =$$
$$\begin{bmatrix} {}_1^0\coprod_k & 0 & \cdots & 0 \\ {}_1^1\coprod_k & {}_2^1\coprod_{k+1} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ {}_1^{m-1}\coprod_k & {}_2^{m-1}\coprod_{k+1} & \cdots & {}_m^{m-1}\coprod_{k+m-1} \\ \vdots & \vdots & \cdots & \vdots \\ {}_1^{p-1}\coprod_k & {}_2^{p-1}\coprod_{k+1} & \cdots & {}_m^{p-1}\coprod_{k+m-1} \end{bmatrix},$$

$${}_r^v\coprod_j = {}_r^vA_{k+i}B_j,$$

$${}_l^hA_{k+i} \triangleq \left\{ \begin{array}{l} A_{a,k+h}A_{a,k+h-1} \cdots A_{a,k+l},\ if\ l \leq h \\ I,\ if\ l > h \end{array} \right. .$$

In the attitude subsystem, the state constraints include input constraints, angular constraints and angular velocity constraints. The following formulation gives the general form of model predictive control problem.

From the above formulation, we can attribute this MPC problem to a quadratic optimization problem. The analytical solution can be easily obtained by mathematics derivation. However, if considering the state constraints, it's hard to find the analytical solution. So it's necessary to find the numerical solution. In this paper, we use the quadratic programming to get the numerical solution satisfying the constraints. The general form of QP problem [12] is given as

$$\min_{\Delta U} J_k = \Delta U_k^T H \Delta U_k + G^T \Delta U_k \tag{10}$$

s.t.

$$A_{t,k} \Delta U_k \leq b_{t,k}$$

Transform the cost function into the general QP form

$$J_k = E_p^T \Gamma_y E_p + \Delta U_k^T H \Delta U_k + G^T \Delta U_k \tag{11}$$

with $\left\{ \begin{array}{l} H \triangleq S_k^T \Gamma_y S_k + \Gamma_u \\ G \triangleq -2S_k^T \Gamma_u^T E_p \end{array} \right.$.

Then transform the constraints into general QP form.

From formulation (2), we can get the input constraints

$$\begin{bmatrix} L \\ -L \end{bmatrix} \Delta U_k \le \begin{bmatrix} u_{k,\max} - u_{k-1} \\ \vdots \\ u_{k+m-1,\max} - u_{k-1} \\ u_{k-1} - u_{k,\min} \\ \vdots \\ u_{k-1} - u_{k+m-1,\min} \end{bmatrix} \quad (12)$$

with $L = \begin{bmatrix} I_{nu \times nu} & 0 & 0 \\ \vdots & \ddots & 0 \\ I_{nu \times nu} & \cdots & I_{nu \times nu} \end{bmatrix}_{m \times m}$.

And from (4) and (5), we can get state constraints

$$\begin{bmatrix} S_{u,k} \\ -S_{u,k} \end{bmatrix} \Delta U \le \begin{bmatrix} Y_{t\max} - S_{x,k}\chi_k \\ S_{x,k}\chi_k - Y_{t\min} \end{bmatrix} \quad (13)$$

with

$$S_{u,k} \triangleq \begin{bmatrix} C & \cdots & C \\ \vdots & \ddots & \vdots \\ C & \cdots & C \end{bmatrix}_{p \times p} \Psi_{a,k},$$

$$S_{x,k} \triangleq \begin{bmatrix} CA_k \\ CA_{k+1}A_k \\ \vdots \\ CA_{k+p-1} \cdots A_k \end{bmatrix},$$

$$Y_{t\max} \triangleq \begin{bmatrix} y_{t\max,k+1} \\ \vdots \\ y_{t\max,k+p} \end{bmatrix},$$

$$Y_{t\min} \triangleq \begin{bmatrix} y_{t\min,k+1} \\ \vdots \\ y_{t\min,k+p} \end{bmatrix}.$$

Thus, we have transformed the model predictive control problem into a QP problem. There are many ways to solve QP problems, for examples, interior point method and active set method. After the attitude controller is designed, the position controller can be obtained in similar way. And finally we obtain a complete controller for quadrotor which is used to generate the expected trajectory.

4. RESULTS

In this section, the simulations are conducted with MATLAB and the quadprog.m is used to solve the quadratic programming problems and generate feasible trajectories. And we also test our approach in the off-line trajectory planning for real quadrotor system.

In simulation, a 3D grid map is provided, which is composed of $100 \times 100 \times 100$ cubes with 0.1m length of side of each cube. Here we have a task of reaching a specified location, so the terminal state and initial state of the quadrotor is also provided.

Table 2 Simulation Parameters

Parameter	Value
Initial state	zeros(12,1)
Simulator step size	0.001s
Simulation time length	10s
Position control period	0.2s
Position prediction step	10
Position control step	2
Position error weight $\Gamma_{p,y}$	eye(60)
Position input weight $\Gamma_{p,u}$	0.1*eye(6)
Attitude control period	0.02s
Attitude prediction step	20
Attitude control step	20
Attitude error weight $\Gamma_{a,y}$	eye(60)
Attitude input weight $\Gamma_{a,u}$	0.1*eye(60)

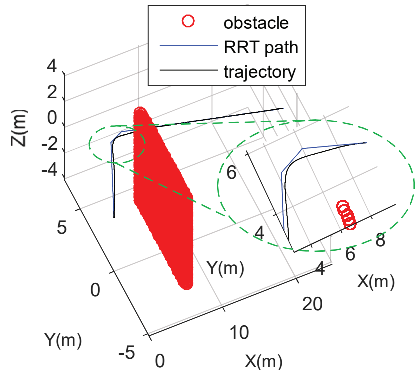

Fig. 6 Path to trajectory

Figure 6 shows the simulation result of trajectory planning. We first use RRT algorithm to find a path that connect the start state to the end state. Then we use the inner simulator to track the path, generating a physical feasible trajectory that satisfying state constraints. Figure 7 shows the transformation of quadrotor's attitude when tracking the path generated by the proposed scheme. From figure 8, we can see that the quadrotor's angular velocity changes within the constraints during tracking the path.

Figure 9 shows the off-line trajectory planning results for real system. The 3D grid map is given. The coordinate in the figure is represents the current state of a quadrotor, and the red ball represents the target state. Applying our approach, the white trajectory is planned, with the considering of state and input constraints.

5. SIMULATION CONCLUSION

This paper proposed an innovative trajectory planning scheme based on RRT and MPC in order to solve trajectory planning problem of quadrotor with model-related state and input constraints which emerge in some applications. In this scheme, the RRT algorithm is adopted to create waypoints that connects the start point and ending point while avoiding the obstacles in a give 3D map.

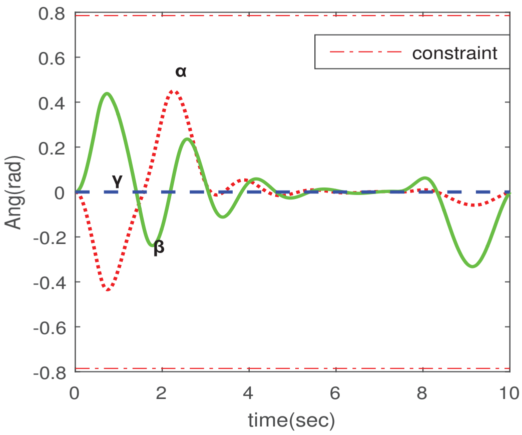

Fig. 7 Attitude changes

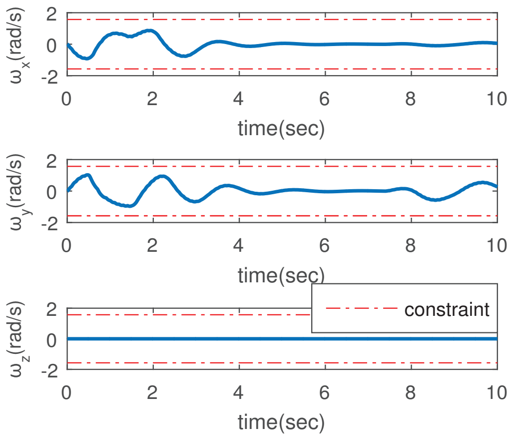

Fig. 8 Angular velocity changes

And a MPC-based controller is designed to generate the feasible trajectories that satisfies given constraints by using an inner simulator to track those waypoints. To our best knowledge, it is first time that the MPC method is employed to generate trajectory satisfying model-related state and input constraints. The simulation results show the effectiveness of the proposed scheme and the potential of MPC in terms of multi-constraint trajectory planning. This scheme will extend greatly the ability of quadrotor to execute some special task. To facilitate the online application of this scheme, the future work will focus on the improvement of realtime performance of this scheme.

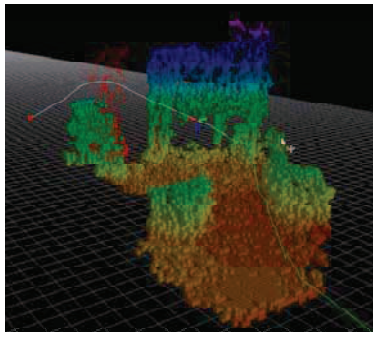

Fig. 9 Off-line planning for real system

REFERENCES

[1] Steven L. Waslander and Gabriel M. Hoffmann, "Multi–Agent Quadrotor Testbed Control Design: Integral Sliding Mode vs. Reinforcement Learning," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 468–473, 2005.

[2] Jonathan P. How, Breet Bethke, Adrian Frank, Daniel Dale, and John Vian, "Real–time indoor autonomous vehicle test environment," *IEEE Control System Magazine*, 51–64, 2008.

[3] Paul Pounds, Robert Mahony and Peter Corke, "Modelling and Control of a Quad-Rotor Robot," *Robert E. Mahony*, 2006.

[4] Kok–Mend Lee, Zhi Zhou, Robert Blenis and Erik Blasch, "Real–time vision-based tracking control of an unmanned vehicle," *Mechatronics*, 1995.

[5] Richter, Charles, Adam Bry, and Nicholas Roy, "Polynomial trajectory planning for quadrotor flight," *International Conference on Robotics and Automation*, 2013.

[6] Hart, Peter E., Nils J. Nilsson, and Bertram Raphael, "A formal basis for the heuristic determination of minimum cost paths." *IEEE Transactions on Systems Science and Cybernetics*, (1968): 100-107.

[7] Kuffner, James J., and Steven M. LaValle, "RRT-connect: An efficient approach to single-query path planning." *IEEE International Conference on Robotics and Automation*, 2000.

[8] Skiena, S, "Dijkstra's algorithm." *Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica, Reading, MA: Addison-Wesley*, (1990): 225-227.

[9] Penghong, Lin, Chen Songlin, and Wang Xiaochen, "An improved backstepping based controller with integrators for the quadrotor." *Control Conference (CCC), 2015 34th Chinese*, IEEE, 2015.

[10] Jameson, Antony, Wolfgang Schmidt, and Eli Turkel, "Numerical solutions of the Euler equations by finite volume methods using Runge-Kutta time-stepping schemes," *AIAA paper*, 1259 (1981): 1981.

[11] Mayne, David Q., et al, "Constrained model predictive control: Stability and optimality," *Automatica*, 36.6 (2000): 789-814.

[12] Gould, N. and P. L. Toint. "Preprocessing for quadratic programming." *Math. Programming*, Series B, Vol. 100, pp. 95C132, 2004.