2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Congress Center Hamburg
Sept 28 - Oct 2, 2015. Hamburg, Germany

Generation of Dynamically Feasible and Collision Free Trajectory by Applying Six-order Bezier Curve and Local Optimal Reshaping

Liang Yang[1,3], Dalei Song[1], Jizhong Xiao[2], Jianda Han[1], Liying Yang[1] and Yang Cao[3]

Abstract— **This paper considers the problem of generating dynamically feasible and collision free trajectory for unmanned aerial vehicles(UAVs) in cluttered environments. General random-based searching algorithms output piecewise linear paths, which cause big discrepancy when used as navigation reference for UAVs with high speed. Meanwhile, the disturbance may also occur to lead the UAVs into danger. In order to obtain agile autonomy without potential dangers, this paper introduces a three-step method to generate feasible reference. In the first step, a six-order Bezier curve, which uses Tuning Rotation to decrease the curvature, is introduced to smooth the output of the path planner. Then a forward simulation is implemented to find the potential dangerous regions. Finally, the path is reshaped by local optimal reshaping planner to eliminate residual dangers. The three steps form a circulation, the reshaped path sent to the first step again to check dynamic feasibility and safety. The method combining Six-order Bezier curve, Tuning Rotation, and local optimal reshaping is proposed by us for the first time, where the Tuning Rotation is able to meet various curvature requirements without violating the previous path, local optimal reshaping obtains both temporal and spatial reshaping with high time efficiency. The method addresses the system dynamics to achieve agile autonomy, which provides the geometry reference as well as the low level control. The effectiveness of the proposed method is demonstrated by the simulations.**

I. INTRODUCTION

Vehicle Autonomy consists of Environment Perception and Modeling[1], [2], Path planning and smoothing [3], [4], [5], and On-line Navigation[6], [7]. Path planning aims to find a feasible path from the initial position to the goal position in the configuration space while avoiding the obstacles or dangerous regions. The goal of the path planners is to support a creditable reference for on-the-flight navigation, which contains geometry constraints as well as system states, called kinodynamic planning.

Traditionally, UAVs are embedded with open loop path planning heuristic where UAVs are thought to ensure accurate tracking without un-modeled disturbance. Therefore general path planning methods simply introduce the kinematic constraints into the whole planning procedure [8] to achieve a possible reference path, where the path only holds continuity with way points or velocity [9]. UAVs can only follow this kind of paths with relative slow speed and hovering at the discrete points, while the discrepancies increase for agile autonomous vehicles.

Numerous approaches for UAV path planning have been proposed, and our previous effort has been demonstrated that Rapidl-exploring Random Tree(RRT) and its improved versions have good performance in path planning in cluttered environments, and RRT is able to cope with robot path planning with high degree of freedom [10]. But still piecewise linear path is generated after exploration, where the discrete way points are not easy to follow without discrepancy.

For random search algorithms, another problem is that the planned path may lead to danger if disturbances happen[11]. In order to avoid dangers, a replanning with much stronger constraints is supposed to be executed. However, the time needed for replanning is greatly increased with such strong constraint[3], thus path reshaping is more useful[11].

This paper solves these problems by introducing a closed-loop path planning method. Based on kinematic planner, this paper first introduces a kinematic smoother, which is based on six-order Bezier curve and Tuning Rotation, to generate second-order continuous trajectory. Then, it executes forward simulation to check dynamic feasibility and potential dangers. Finally, the method introduces a local optimal reshaping planner to reshape the path to avoid potential dangers. The method outputs dynamically feasible and collision free trajectory as navigation reference.

Dubins curve [12] and Bezier curve [13] are two most famous methods for path smoothing. Dubins curve calculates the shortest curve between two oriented states, such curve with curvature constraint is decided by expected speed and the maximum acceleration. This kind of method requires the shortest connection under the maximum curvature constraints, the path generated is only connected by straight lines and arcs without parameterization. Thus the method is not appropriate for designed discrete way points, which may generate so many arcs without curvature continuity. [14] combined Bezier curve with genetic algorithm to generate way-points continuous path, which does not consider curvature or velocity continuity. [13] introduces a control points interpolating Bezier curve for RRT. Eight control points are needed to achieve second-order continuity. However, the method cannot accurately follow the path if it is zigzag like.

B-spline is a variation of Bezier curve, which allows more controls than Bezier curve. [15] introduced this method to smooth the path achieved by RRT planner. B-spline can generate constant acceleration flight path, and if collision is detected, it returns feedback to the replanning planner

*This work was by NSFC under Grant #61433016.

[1]Liang Yang, Dalei Song, Liying Yang and Jianda Han is with Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China. Liang Yang is also with University of Chinese Academy of Sciences, Beijing 100049, China yangliang1@sia.cn

[2]Jizhong Xiao, who is Corresponding Author, is with Department of Electrical Engineering, The City College, City University of New York, New York, USA jxiao@ccny.cuny.edu

[3]Yang Cao is with School of Software Engineering, USTC, China. youngcaoustc@outlook.com

to adjust the previous control points. [16] proposed a cubic spline based path smoothing strategy for Voronoi. The cubic spline has the ability of ensuring threat and fuel cost minima. G^2-splines [17] and G^3-splines[18] were introduced to solve car-like vehicles path planning and smoothing problem. But, these methods much depend on the constraints of the turning angle and velocity, they can not guarantee accurate tracking of the way-points. Also these methods sometimes need too many computational resources for iterative step adjusting.

Path reshaping consists of spatial and temporal reshaping [11]. [19] first presented a detailed work of path reshaping for obstacle avoidance. It introduced an elastic band which is represented as a finite series of configurations with bubbles, the elastic band is constrained by potential force that supported by the obstacles. The deformation is stopped until equilibrium points reached. However, the method is never tested to be length minimal, or able to jump out of narrow corridor. A probabilistic spatial reshaping is introduced in [15], where the random exploring heuristic ensures local deforming rational, but the method is time consuming. Temporal reshaping first proposed to solve the problem of tracking delay [20]. The paper introduced a changeable look-ahead distance chosen strategy to ensure tracking accuracy without changing the spatial distribution of previous path. However, the method does not guarantee continuity of speed or acceleration.

II. Planning Framework and Preliminary Materials

The framework of our proposed method is illustrated in Fig.1. It generates collision free trajectory with position and speed reference for UAVs from the initial conditions, which is feasible to low level controllers.

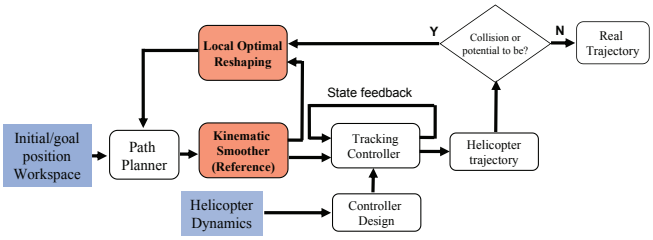


Fig. 1. Framework of proposed method.

The proposed method contains four parts(see in Fig.1), which are, path planner, kinematic smoother, tracking controller and local optimal reshaping. Path planner is our former proposed random-based path planner[10], which can find a near optimal path quickly. The kinematic smoother is designed to ensure control continuity. The tracking controller for our UAV[21] is designed based on the method in [7], which is used to check the dynamic feasibility of the trajectory. If the trajectory is detected with potential dangers, local optimal reshaping is implemented to avoid such dangers. For path smoother, the dynamic constraints can be represented as,

$$\dot{x} = f(x,u) \tag{1}$$

Where $x \in X \in R^n$ denotes the robot state, $u \in R^m$ denotes the control input. To guarantee dynamic feasibility, the states and control inputs are under the following constraints,

$$X = \{x_i | x_i^{min} \le x_i \le x_i^{max}\}, U = \{u_i | u_i^{min} \le u_i \le u_i^{max}\} \tag{2}$$

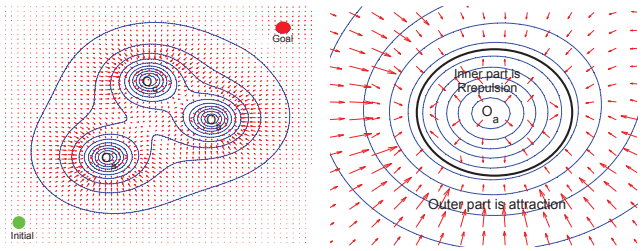
General path planners only try to meet the maximum bounds (2), however the actual actuators are limited with the constraints that,

$$\begin{aligned} x(t_0+\Delta t) &\in \{x(t_0)-\Delta t\cdot\varepsilon, x(t_0)+\Delta t\cdot\varepsilon\} \\ u(t_0+\Delta t) &\in \{u(t_0)-\Delta t\cdot\zeta, u(t_0)+\Delta t\cdot\zeta\} \end{aligned} \tag{3}$$

Where ε and ζ are constant factors. Equation (3) means the designed path must within the actuators' reachability.

A. Path Planner with Kinematic Constraints

The path planner uses Guiding Attraction based Random Tree(GART), which is formerly proposed in[10]. The pseudocode of the proposed method is presented in [10], which consists of the following steps: 1)GART first samples a random node, 2)selects the best parent node to connect from, 3)tries to find a reachable node to connect to, 4)finds the obstacles surrounded, and introduces the attraction from obstacles and goal to redistribute the node generated by 3), 5)evaluates the feasibility. If the node is feasible, GART prunes the tree and outputs the piecewise linear path if the goal is reached.



(a) Global guiding that supported by obstacles and the goal. (b) Illustration of obstacles bi-directional potential.

Fig. 2. Presentation of GART guiding potential that proposed. Where O_a, O_b and O_c are obstacles, the red arrow indicates the resultant potential.

GART differs from general RRT*[15] in several ways, including: the obstacles around the node generated by general *steer* [3] are selected and used to calculate the resultant potential to guide the sampling nodes. The *Guidng_attraction* [10] (Fig.2(a)) is the obstacles and goal resultant potential which aims to ensure fast convergence and environmental adaptability. The obstacle is assumed to provide bidirectional potential, that is, repulsion in the inner part of the obstacle region, attraction in the outer part of the obstacle region (Fig.2(b)).

III. Bezier Curve based Path Smoother and Local Optimal Reshaping

This section discusses the key parts of the method which are illustrated with red boxes in Fig.1. These are the main contributions of this paper, which aim to design a trackable and collision free trajectory for UAV.

A. Bezier Curve based path smoother

1) Six-order Bezier Curve for G^2 Continuity: The path generated by the path planner is piecewise linear, which can be followed by UAVs at a low speed. However, the UAVs can not manage the turning at small and discontinuous radii, which is caused by the fact that path planner negotiates only with global kinematic constraints, and may lead the UAVs into dangerous region. This paper introduces a six-order Bezier curve to smooth the path, that is,

$$P_{[t_0,t_1]}(t) = \sum_{i=0}^{6} B_i^6(t) P_i \tag{4}$$

Where $P_i, i \in \{0,1,...,6\}$ are control points, $(t_0 - t_1)$ is the time needed to cover current local path. $B_i^6(t)$ is Bernstein polynomial and has the following form,

$$B_i^6(t) = \begin{bmatrix} 6 \\ i \end{bmatrix} (\frac{t_1 - t}{t_1 - t_0})^{6-i} (\frac{t - t_0}{t_1 - t_0})^i, i \in \{0,1,...,6\} \tag{5}$$

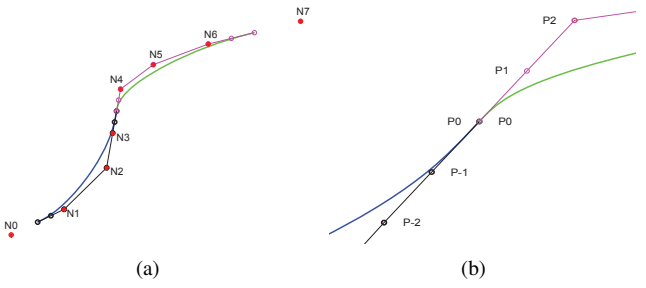
Bezier curve has the advantage of always passing through P_0 to P_6, meanwhile the whole curve always lies within the convex hull that is constructed by the control points. The first derivatives of Bezier curve at t_0 and t_1 are,

$$\dot{P}_{[t_0,t_1]}(t_0) = \frac{6}{t_1 - t_0}(P_1 - P_0), \dot{P}_{[t_0,t_1]}(t_1) = \frac{6}{t_1 - t_0}(P_6 - P_5) \tag{6}$$

The second derivatives are:

$$\begin{aligned} \ddot{P}_{[t_0,t_1]}(t_0) &= \frac{30}{(t_1 - t_0)^2}(P_0 - 2P_1 + P_2) \\ \ddot{P}_{[t_0,t_1]}(t_1) &= \frac{30}{(t_1 - t_0)^2}(P_4 - 2P_5 + P_6) \end{aligned} \tag{7}$$

In order to achieve G^2 continuity, this paper introduces an interpolation method to ensure second derivative continuity by interpolating three points at the joint edge (see in Fig.3). The proposed Bezier curve consists of three discrete points generated by the path planner and four interpolating points which are used to achieve second derivative continuity. Fig.3(b) shows the details of the joint part of Fig.3(a),



(a) (b)

Fig. 3. An illustration of proposed G^2 continuity six-order Bezier curve. (a) shows two six-order Bezier curves which decided by seven discrete control points, and the unlabeled nodes are interpolating points. (b) shows the details of joint between two Bezier curves, where $P_{-2}, P_{-1}, P_0, P_1, P_2$ are interpolating points.

$$\begin{aligned} P_{-2} = N_3, P_{-1} = N_3 \cdot \frac{3}{4} + N_4 \cdot \frac{1}{4} \\ P_0 = N_3 \cdot \frac{1}{2} + N_4 \cdot \frac{1}{2}, P_1 = N_3 \cdot \frac{1}{4} + N_4 \cdot \frac{3}{4}, P_2 = N_4 \end{aligned} \tag{8}$$

Therefore, substitute interpolating condition (8) into (6) and (7), then we get the first and second derivatives at P_0,

$$\dot{P}_L(t_0) = \frac{3(N_4 - N_3)}{2(t_1 - t_0)} = \dot{P}_R(t_1), \quad \ddot{P}_L(t_0) = 0 = \ddot{P}_R(t_1) \tag{9}$$

Where P_L denotes the left blue curve of Fig.3 and P_R denotes the right green curve. Thus the proposed interpolation method ensures the second-order continuity along the trajectory, if given $P_{[t_0,t_1]} = [x(t), y(t)]$, the curvature along the Bezier curve is,

$$K(t) = \frac{1}{R(t)} = \frac{\ddot{x(t)}\dot{y(t)} - \ddot{y(t)}\dot{x(t)}}{(\dot{x(t)}^2 + \dot{y(t)}^2)^{3/2}} \tag{10}$$

Equation (9) illustrates that our interpolation method (8) ensures the second-order continuity, that is, curvature continuity (10).

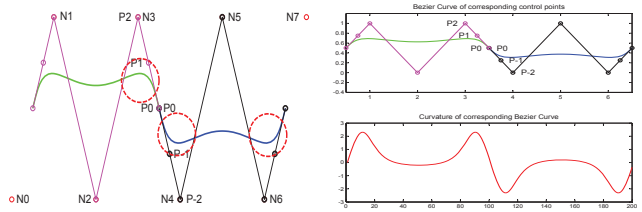
2) Tuning Rotation: The proposed interpolation method ensures curvature continuity. However, a big turning at the joint edge of the two Bezier curves may exist(see in Fig.4(a)). This is caused by: 1) the path planner obtains no smooth techniques; 2) the simple interpolation method that compels the Bezier curves to approach the boundary linear line. These two problems both have the probability of generating a surge of curvature as shows in Fig.4(b), where such a big change in curvature may cause discontinuity of trajectory tracking and lead UAV into dangerous situation.

Let's first explain a concept called path consistency defined as follows:

Definition 1 (Path consistency): Given a set of path nodes, if the sum of the angles between all neighbor piecewise linear lines is small enough, then path is regarded as obtaining consistency. Or can be expressed as,

$$\sum_{i=0}^{m} cos(\theta_i) > b \tag{11}$$

where $\theta_i, i \in \{1,2,...,m\}$ are the angles between two neighbor edges(see in Fig.5(a)). b denotes the minimum value that can be tolerated.

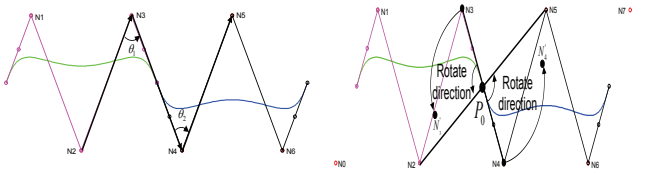


(a) (b)

Fig. 4. Presentation of extreme situation of our interpolating method. (a) shows intuitive image of interpolating method weakness, (b) shows the curvature of corresponding curve.

This paper introduces a method called Tuning Rotation (TR) which aims to increase the consistency of the local connection part. As presented in Fig.5(b), four nodes directly lead sharp turning at the connection part between two curves, and the inconsistency of the four nodes can be simply reflected by the angle θ. Tuning Rotation tries to rotate the inner two nodes, which are N_3 and N_4, to decrease the angle θ between the two cross lines, where P_0 is the middle node and fixed. TR first tries to find the relative direction between $(N_4 - P_0)$ and $(N_5 - P_0)$, then decides the rotational direction (clockwise or anticlockwise). Finally, iterative rotation with a certain small angle α will be implemented, that is,

$$\begin{aligned} N_3' &= \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} (N_3 - P_0) + P_0 \\ N_4' &= \begin{bmatrix} cos\alpha & -sin\alpha \\ sin\alpha & cos\alpha \end{bmatrix} (N_4 - P_0) + P_0 \end{aligned} \tag{12}$$



(a) (b)

Fig. 5. Consistency definition and Tuning rotation. (a) Illustration of path consistency, (b) Tuning rotation tries to rotate the inner nodes to ensure path consistency.

TR stops when the maximum curvature of this part gets no bigger than a certain constant, which means the Bezier curve is smooth enough for the UAV to track without exceeding the actuator limitation.

B. Local Optimal Reshaping

Local Optimal Reshaping (LOR) tries to eliminate residual danger along the kinematic smoothed path. UAV tracking accuracy is sensitive to the speed, which decreases with speed increasing and even worse when disturbance occurs. UAV simulations with dynamics show that the dangerous regions are with the nearest parts around the obstacles region. However, simply enlarge the safe margin will lead to exponentially increasing of the exploring time. The reasons are: 1) it decreases the Lebesgue measurement (i.e. the area or volume) of collision free region, which in turn leads to increasing of probability of sampling in collision region. 2) it decreases the step extending length [3], which also increases the exploring time when compared to the exploration in the same workspace.

LOR first analyzes the feedback returned by forward simulation which is the second step of our path planning framework. LOR holds the processes that:1) It first captures the dangerous regions along the planned path, where these regions are selected by forward simulation with disturbance and large speed. 2) Then for each local dangerous region, LOR picks up the discrete nodes within dangerous distance. 3) Finally, LOR reshapes the local path by minimizing the length and curvature cost of each local path, and the cost is subject to the reshaping direction and safe distance. The reshaping criterion is,

$$\begin{aligned} J_{cost} = \int_0^1 \sqrt{dx^2 + dy^2} dt + \int_0^1 (\frac{\dddot{x}\ddot{y} + \dddot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \\ + \frac{-3 \cdot (\dot{x}\ddot{y} + \dot{y}\ddot{x})(\dot{x}\ddot{x} + \dot{y}\ddot{y})}{(\dot{x}^2 + \dot{y}^2)^{3/2}}) \end{aligned} \tag{13}$$

Subject to,

$$\begin{cases} \sqrt{(x - x_j)^2 + (y - y_j)^2} > r_j + d_{safe} & (1) \\ P_i(1,t) \cdot sgn[cos(atan(\frac{P_i(2) - y_j}{P_i(1) - x_j}))] > P_i(1) & (2) \\ P_i(2,t) \cdot sgn[sin(atan(\frac{P_i(2) - y_j}{P_i(1) - x_j}))] > P_i(2) & (3) \\ P_t = \sum_{i=0}^{n} \begin{bmatrix} n \\ i \end{bmatrix} \cdot t^i \cdot (1-t)^{n-1} P(i) & (4) \end{cases} \tag{14}$$

Where $P(t) = [x(t), y(t)] = [P(1), P(2)]$ denotes the Bezier curve in 2D. The cost function consists of length and the average curvature along the Bezier curve that is formed by the local control nodes.

The constraints contain three parts(see (14)). The first constraint (14.1) is to guarantee minimal safe distance, i.e., $r_j + d_{safe}$, r_j denotes the radius of the jth dangerous region, d_{safe} is the minimal safe distance to ensure safety according to the speed and disturbance. (x_j, y_j) denotes the center coordinate of the jth dangerous region. Constraint (14.2) and (14.3) stipulate the spatial reshaping direction and region for each local nodes that selected(see in Fig.6). For each dangerous node P_i, LOR first calculate the angle of vector $\vec{R}$ which starts from center coordinate of dangerous region to node P_i. Then LOR defines that P_i can only be redistributed to the quadrant according to previous calculated angle, which is the blue region of Fig.6. Constraint (14.4) requires the cost to be calculated must according to the Bezier curve generated. This Bezier curve is controlled by the reshaped nodes and two fixed nodes of two sides.

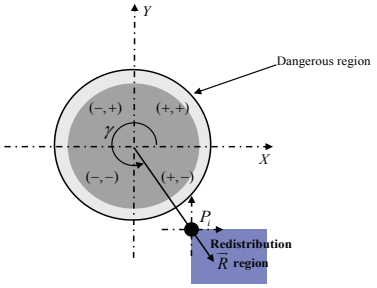
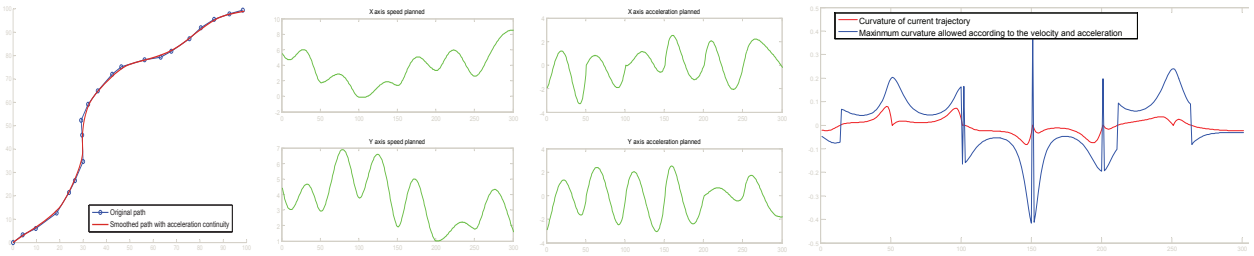


Fig. 6. Each node redistribution range defined by LOR with equation 14.2 and 14.3

IV. Simulation Results and Analysis

This paper designs four sets of experiments to prove the efficiency of the algorithms for path planning algorithm, six-order Bezier curve, TR and LOR, and tracking controller, respectively. All the simulations are programmed in MATLAB



(a) Smoothed trajectory generated by six-order Bezier curve.

(b) Speed generated by taking the first derivative of Bezier curve.

(c) Acceleration by taking the second derivatives.

(d) Comparative results of the curvature along the path.

Fig. 7. Results after sxi-order Bezier curve smoothing. (d)The blue line denotes the maximum curvature that allowed with $3m/s^2$ acceleration, the red line denotes the planned curvature.

2013B, and the laptop platform has a Core I5 processor and a 2G RAM. For all the simulations, the 2D or 3D maps are randomly generated.

A. *Comparison of path planning algorithm*

GART is discussed discussed in Section II(A) and [10] in detail, a set of carefully designed experiments in 2D and 3D are presented in [10]. The results illustrate the efficiency of the proposed algorithm.

B. *Path Smoother*

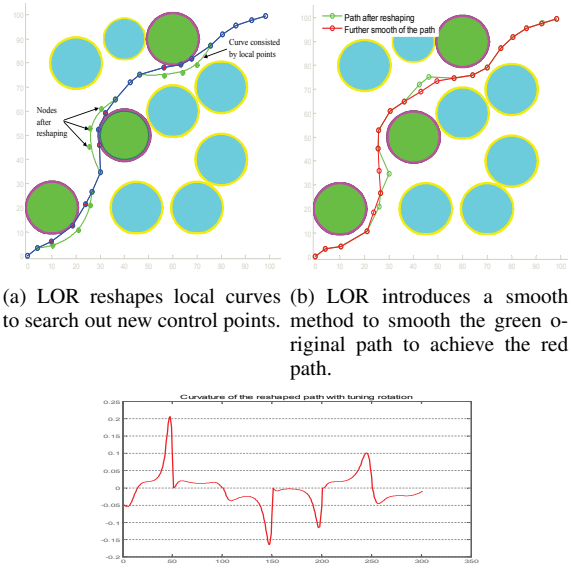
Smoothed results of six-order Bezier curve are presented in Fig.7, where the red line (in Fig.7(a)) is the smoothed path. This paper plans the speed and acceleration by taking the first the second derivatives along the trajectory, SERVOHELI-40 is limited with $3m/s^2$ acceleration.

Fig.7(c) shows that the planned acceleration is within $3m/s^2$, which is continuous globally. The speeds (in Fig.7(b)) guarantee no bigger than $10m/s$, which is suitable for our UAV. The curvature along the smoothed trajectory is calculated based on (15). But it should be noted that the curvature must obey the constraints with planned speed and maximum acceleration constraints, that is, $m\frac{v_{planned}^2}{r} = ma_{max}$.

Thus, the maximum curvature can be derived. The comparative results of real curvature along the planned path and the maximum curvature constraints are presented in Fig.7(d). We can see that the planned curvature (red line) is with smaller magnitude than the maximum curvature(blue line).

C. *Path Reshaping and Tuning Rotation*

It is illustrated in Fig.8 that LOR is introduced to find the probable dangerous regions(green circles), meanwhile LOR labels the discrete nodes within dangerous distance which are red nodes shown in Fig.8(a). Then LOR executes optimal reshaping to adjust local control points to new safer position. The paper sets the safe distance as 4 meters, and the green nodes are redistributed nodes. The green curves are Bezier curves formed by the local control points. LOR further introduces a path-even method to delete big turning along the path, which interpolates new nodes to smooth the path. The final results of LOR are the red line as shown in Fig.8(b), and the red small circles are control nodes.



(a) LOR reshapes local curves to search out new control points.

(b) LOR introduces a smooth method to smooth the green original path to achieve the red path.

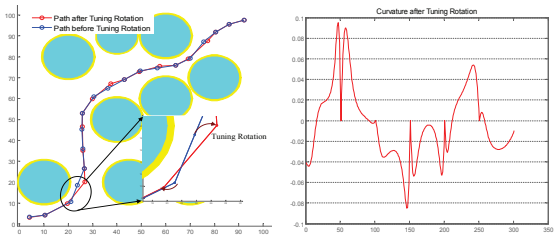
(c) The curvature along the trajectory after reshaping by applying the six-order Bezier curve.

Fig. 8. Illustration of local nonlinear reshaping.

Path smoothing is still executed for path generated by LOR, the curvature along the path is presented in Fig.8(c). Although it is no bigger than the maximum bound(in Fig.7(d)), and it is still puts a heavy load on the actuator. This paper sets the maximal curvature as 0.1, and each time we set the rotation with one fifth of the angle between two pair nodes. It can be seen in Fig.9(a), the red line is the results after tuning rotation. Final curvature of TR is presented in Fig.9(b), it can be seen that no curvature is bigger than 0.1.

D. *Final simulation based on system model with disturbance*

Velocity disturbances are added to verify the performance of our method, which are white Gaussian noise whose magnitude is $0.2m/s$. We test the tracking ability of both tracking method, and the results are illustrated in Fig.10. Position tracking result is shown in Fig.10(a), where the red line is the tracking trajectory, and the blue line in the reference trajectory. The figure shows that the disturbance causes big tracking error when turning occurs, while it

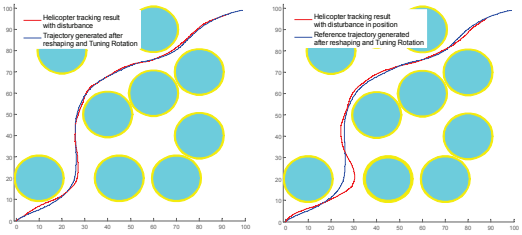


(a) TR rotate the connecting nodes by considering the trend of the path to generate a new path with expected curvature.

(b) A comparative curvature result generated after TR, which means that the trajectory is smoother than before.

Fig. 9. The path generated after TR and its curvature.

obtains safety. Speed reference tracking result is illustrated in Fig.10(b), where the red line is the tracking result, and the blue line is the reference trajectory. LOR can increase the safe distance to resist bigger disturbance.



(a) Position tracking result.

(b) Speed tracking result.

Fig. 10. Two tracking tests with position and speed reference respectively, both with speed disturbance, where the disturbance is white Gaussian noise with $0.2m/s$ magnitude.

V. CONCLUSIONS

In this paper, we have presented a method which generates dynamically feasible and collision free reference for UAVs. The method is closed-loop which divides path planning as kinematic planning and dynamic adjusting. The kinematic planning is a synthesis of path planer, path smoother. Path smoother contains Six-order Bezier curve and TR. In order to decrease the tracking discrepancy caused by our GART path planner, Six-order Bezier curve and TR are introduced to ensure G^2 continuity and curvature minimal, respectively. LOR is proposed to use the dynamics feedback to find the dangerous region for local reshaping, thus to achieve safety under the dynamic requirements. The proposed method successfully solves the problem of supporting dynamically feasible and collision free reference. Simulation results show that the framework is robust to disturbance and feasible for actuators.

Currently, this paper does not embed dynamics in the path planner, and the obstacles are simply modeled as ellipse in 2D and ellipsoid in 3D. For further work, we want to conduct a more detailed research in environment modeling, and also try to embed the dynamics into the path planner.

REFERENCES

[1] Gutmann.J.S, Fukuchi.M, Fujita. M, 3D perception and environment map generation for humanoid robot navigation[J], The International Journal of Robotics Research, vol.27, no.10, pp:1117-1134, 2008.

[2] Sud.A, Andersen.E, Curtis.S, et al, Real-time path planning in dynamic virtual environments using multiagent navigation graphs[J], IEEE Transactions on Visualization and Computer Graphics,vol.14, no.3, pp:526-538, 2008.

[3] Karaman.S, Frazzoli.E, Sampling-based algorithms for optimal motion planning[J], The International Journal of Robotics Research, vol.30, no.7, pp:846-894, 2011.

[4] Yang.K, Gan.S.K, Sukkarieh.S, An efficient path planning and control algorithm for RUAV's in unknown and cluttered environments[J], Journal of Intelligent and Robotic Systems, vol.57, no.1-4, pp:101-122, 2010.

[5] Neto.A.A, Macharet.D.G, Campos.M.F.M, Feasible RRT-based path planning using seventh order Bzier curves[C], IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 1445-1450, 2010.

[6] Hoffmann.G.M, Waslander.S.L, Tomlin.C.J, Quadrotor helicopter trajectory tracking control[C], AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii, pp:1-14, 2008.

[7] Mahmoud.M.S, Koesdwiady.A.B, Improved digital tracking controller design for pilot-scale unmanned helicopter[J], Journal of the Franklin Institute, vol.349, no.1, pp:42-58, 2012.

[8] Roberge.V, Tarbouchi.M, Labont.G, Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning[J], IEEE Transactions on Industrial Informatics, vol.9, no.1, pp:132-141, 2013.

[9] Kuwata.Y, Karaman.S, Teo.J, et.al, Real-time motion planning with applications to autonomous urban driving[J], IEEE Transactions on Control Systems Technology, vol.17, no.5, pp:1105-1118, 2009.

[10] Yang.L, Qi.J, Jiang.Z, et.al, Guiding attraction based random tree path planning under uncertainty: Dedicate for UAV[C], IEEE International Conference on Mechatronics and Automation(ICMA), Tianjin, China, pp:1182-1187, 2014.

[11] Yoshida.E, Esteves.C, Belousov.I, et.al, Planning 3-d collision-free dynamic robotic motion through iterative reshaping[J]. IEEE Transactions on Robotics, vol.24, no.5, pp:1186-1198, 2008.

[12] Anderson.E.P, Beard.R.W, McLain.T.W, Real-time dynamic trajectory smoothing for unmanned air vehicles[J], IEEE Transactions on Control Systems Technology, vol.13, no.3, pp:471-477, 2005.

[13] Yang.K, Sukkarieh.S, An analytical continuous-curvature path-smoothing algorithm[J], IEEE Transactions on Robotics, vol.26, no.3, pp:561-568, 2010.

[14] Sahingoz.O.K, Generation of Bezier Curve-Based Flyable Trajectories for Multi-UAV Systems with Parallel Genetic Algorithm[J]. Journal of Intelligent & Robotic Systems, vol.74, no.1-2, pp:499-511, 2014.

[15] Koyuncu.E, Inalhan.G, A probabilistic b-spline motion planning algorithm for unmanned helicopters flying in dense 3d environments[C], IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), Nice, France, pp:815-821, 2008.

[16] Judd.K.B, McLain.T.W, Spline based path planning for unmanned air vehicles[C], AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, pp:6-9.2001

[17] Piazzi.A, Guarino.Lo.Bianco.C, Bertozzi.M, et.al, Quintic G2-splines for the iterative steering of vision-based autonomous vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, vol.3, no.1, pp:27-36, 2002.

[18] Piazzi.A, Bianco.C.G.L, Romano.M, B-Splines for the Smooth Path Generation of Wheeled Mobile Robots[J], IEEE Transactions on Robotics, vol.23, no.5, pp:1089-1095, 2007.

[19] Quinlan.S, Real-time modification of collision-free paths[D], PHD thesis, Stanford University, 1994.

[20] Gupta.R.A, Masoud.A.A, Chow.M.Y, A delay-tolerant potential-field-based network implementation of an integrated navigation system[J]. IEEE Transactions on Industrial Electronics, vol.57, no.2, no.769-783, 2010.

[21] Song.D, Qi.J, Dai.L, et.al. Modelling a small-size unmanned helicopter using optimal estimation in the frequency domain[J], International journal of intelligent systems technologies and applications, vol.8, no.1, pp:70-85, 2010.