UAV Path Planning Framework Under Kinodynamic Constraints in Cluttered Environments

Liang Yang[1,2], Juntong Qi[1(✉)], Yang Cao[1,2,3], Yuqing He[1,2,3], Jianda Han[1], and Jizhong Xiao[3]

[1] State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, People's Republic of China
{yangliang1,qijt,jdhan}@sia.cn, youngcaoustc@outlook.com, jxiao@ccny.cuny.edu
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Department of Electrical Engineering, The City College, City University of New York, New York, USA

Abstract. A novel kinodynamic planning framework, which covers path panning, smoothing, tracking and emergency threat managing, is proposed. The framework is proposed based on sampling-based algorithm, which is improved to ensure dynamics feasibility as well as emergency threat management ability by applying Bezier curve and Extending Forbidden respectively. The Bezier curve guarantees both G^1 and G^2 continuity to decrease the tracking error of our LQI based tracking controller, where two Bezier curves with different continuity order are discussed. Extending Forbidden is firstly proposed by us to enable generating multiple paths of sampling-based algorithms, thus support on-line switching to avoid emergency threats. Our main contribution is that the proposed framework is a combination of path planning with emergency threat managing, where a time compromised moving obstacle avoiding method is proposed. Results proves the efficiency of the proposed algorithm in generating feasible trajectory for SERVOHELI-40, which not only guarantees kinematic feasible of avoiding obstacles, but also can ensure dynamics feasibility.

1 Introduction

Unmanned aerial vehicles (UAVs) are now widely used as platforms in cluttered, unstructured environments, where they are required to perform sophisticated tasks and should obtain agile autonomous performance. As the complexity of the environment and task increases, the need for efficient path planning, smoothing and navigation methods is urgent nowadays. Although there exist so many

L. Yang—It is high acknowledged that this work is partially supported by National Science and technology support under Grant #2013BAK03B02. Yang Cao is with School of Software Engineering, USTC, China.

H. Liu et al. (Eds.): ICIRA 2015, Part II, LNAI 9245, pp. 248–259, 2015.
DOI: 10.1007/978-3-319-22876-1_22

methods that are proposed to manage these problems, when facing the limitation of the on-board sensors, actuators, and unstructured environments, there still exists no common solution for this NP-hard problem[1].

UAV path planning typically focuses on time efficiency of finding a collision free path. Famous path planning methods include Visibility Graphs[2] which plan via connecting the visible nodes of the dangerous region, Rapidly-exploring Random Tree [3] which samples the whole configuration space randomly to guarantee probabilistic completeness, A* [4] algorithm which is successful in handling path planning problems by introducing heuristic graph search algorithm, Evolutionary algorithms[5] which are inspired by biological behaviour, and mathematic model based method [6] which tries to find the best path under kinodynamic constraints with absolute smooth path etc. However, classical methods output discrete way points which are unsuitable for UAVs to follow with a certain high speed. Therefore, the problem of designing a well smoothed navigation reference is needed.

Path smoothing is a classical topic of computer geometry, which aims to represent the local segments with parametric curves. The curves have the ability of guaranteeing continuity[7]. Among all the smoothing methods, Bezier curve[8] and B-spline[9] are the most famous and successful methods. Bezier curve describes the local discrete nodes by applying parametric curve, it can connect the local nodes. B-spline is a variation of Bezier curve, which allows more control to meet the requirement. All these methods are able to guarantee the continuity of the path, while they do not consider kinodynamic constraints.

This paper also tries to provide a common method of dealing with moving obstacles and pop-up threats. Unlike general methods of applying the replanning [10] or local reshaping [11] to deal with such problems with a certain designed method, the proposed method is a compromise with time, which can avoid emergency threat flexibly with two kinds of methods.

The main contribution of this paper is that it proposes an efficient path planning framework for UAV to obtain agile autonomous in cluttered environments where moving obstacles and pop-up threat exist. The framework contains a sampling-based path planner, a Bezier curve based path smoother, a LQI based tracking controller and a GART[12] based on-line emergency threat manager. Several comparative simulations are implemented, which prove the efficiency of the proposed framework.

2 Preliminary Material

2.1 Problem Statement

For path planning, the environment constraints are prior problems that need to be discussed. The general methods of tackling environmental constraints are to distinguish the dangerous region with safe region. In order to guarantee safety, a feasible path is designed to obtain the following properties,

$$P_j = \{p_{j,i} | p_{j,1} = start, p_{j,n} = goal, i = i, ..., n\} \tag{1}$$

Subject to,

$$\begin{cases} P_j \subset R_{free} \\ P_j \cap R_{obstacle} = \oslash \\ C = R_{obstacle} \cup R_{free} \end{cases} \tag{2}$$

Where P_j is the path nodes set which contains the start node *start* and the goal node *goal*. C denotes the configuration space, and $R_{obstacle}$ denotes the obstacle region and R_{free} denotes the safe region within the configuration space. The path is planned under the principle of reaching certain requirement. Such principles contain energy minimal, threat minimal, robots tracking error minimal etc, that is,

$$Cost(P_j) = C_{energy}(P_j) + C_{threat}(P_j) + C_{threat\_error}(P_j) \tag{3}$$

Thus, the final output is the path which ensures cost minimum based on current cost function to meet the requirement of certain robot. However, the output of the path planner is sometimes in a piecewise linear form, which is distributed depending on the distribution of the obstacles. UAV is not able to track this kind of path under kinematic and dynamic constraints. Thus the problem of how to smooth the path and parameterize the path to support more accurate reference for the planner is needed to be solved.

2.2 System Presentation and Tracking Controller

Our helicopter is a rotor-wing unmanned aerial vehicle with 40kg take-off weight, called SERVOHELI-40 [13]. The parameterized model of the helicopter system has also been identified already in the hovering model, and it is used as the controlled object. This paper regards that the yaw angle is the time integral of the angular rate, that is, $\psi \approx \int r dt$, to ensure model linearity. Therefore the above 13 states model is expanded to 14 states model, $X^F = (u, v, w, p, q, r, \varphi, \theta, a, b, w, r, \psi, r_{fb}, c, d)$ and

$$\dot{X^F} = A^F X^F + B^F u \tag{4}$$

Where X is the system state with thirteen states. u, v, w denote the longitudinal, lateral, and vertical velocity. p, q, r denote the roll, pitch and yaw rate. φ, θ denote the roll and pitch angles. a, b denote the first harmonic flapping angle of main rotor. c, d denote the first harmonic flapping angle of stabilizer bar. r_{fb} is the feedback control value of the angular rate gyro. The control inputs are lateral δ_{lat}, longitudinal δ_{lon}, yawing δ_{ped}, vertical control input δ_{col}.

The measurement states are $y^{aug} = (u, v, w, p, q, r, \varphi, \theta, \psi, a_x, a_y, a_z)$. Here a_x, a_y, a_z are the coordinates of each axis in body-axis.

This paper designs a LQI controller for trajectory tracking [16], which is an optimal controller, to achieve desired dynamic and steady-state performances. LQI differs from general LQR controller in that LQI eliminating steady state error by augmenting the state with integral error between reference and output. Thus the augmented state is,

$$X^{aug} = [X^F; X_I] \tag{5}$$

Where,

$$X_I(n+1) = X_I(n) + Ts \cdot (X_{ref}(n) - y^{'}(n)) \tag{6}$$

X^{aug} denotes the augmented state vector, X_I denotes the integral of error between the reference and output, TS denotes the sampling time for discrete system, X_{ref} denotes the reference. $y^{'}$ denotes the outputs corresponding to the references. Because the system state does not contain position, the position reference need to be converted to velocity reference by using PID controller.

The LQI control law $u_{LQI} = -KX^{aug}$ is generated by minimizing the cost function,

$$J(u_{LQI}) = \sum_{n=1}^{\infty} \{(X_{aug}^{T} Q X^{auq} + u_{LQI}^{T} R u_{LQI}) \tag{7}$$

Where Q, R are weighting matrices, and Q is none negative symmetric matrix, R is positive definite symmetric matrix.

3 Path Planning Framework

This section illustrates a novel framework for UAV path planning in cluttered environment (see in Fig. 1). The framework consists of a random based path planner, a Bezier curve based path smoother, a tracking controller and an emergency threat manager.

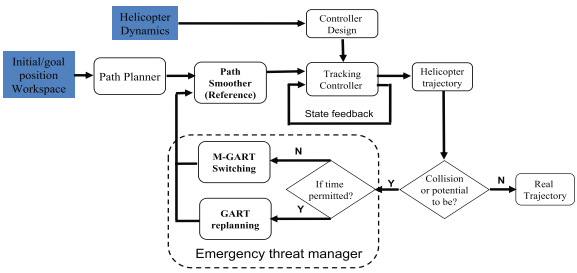
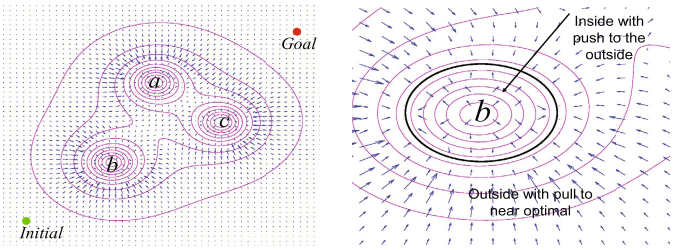


Fig. 1. Path planning framework that proposed.

3.1 Path Planner

The path planner is based on Guiding Attraction Based Random Tree (GART) which is also a kind of RRT based algorithm and still performs in a random exploring way to generate feasible path for robots. But general RRT or RRT* [14] can not guarantee global optimal, and the time required to achieve the asymptotic optimal still can not be guaranteed due the fact that it ignores the information of the configuration space.



(a) Global guiding that supported by obstacles and goal.

(b) Details of obstacles bidirectional potential.

Fig. 2. Illustration of GART guiding potential.

GART shares the same merits as artificial potential field(APF) [15] to obtain the local or global environmental information to decrease the random characteristic of RRT*, while it differs in: 1) GART introduces a bidirectional potential for random generated nodes(Fig. 2), that is, repulsion is achieved if the random node is in the obstacle region, and attraction when the random node outside the obstacle region. 2) The force supported by an obstacle is inverse proportional to the distance from the current node to the obstacle surface, and goal always support attraction. GART holds the following steps: 1) vFirst, it takes a sample from the configuration space. 2)Then it tries to connect to the nearest node with a limited distance to achieve the reachable sampling node $P_{sample}^{current}$ by adding obstacle and goal attraction.Details of the planning procedures can be seen in[12]

3.2 Path Smoother

Path smoother aims to generate geometrically continuous curves to stitch the local segments along the path. Let $P = \{p_1, p_2, ..., p_n\}$ denote the output of the path planner, for each single point, it obtains the form $p_i = (x_i, y_i)$ in a 2D space and $p_i = (x_i, y_i, z_i)$ in a 3D space. This paper adopts the Bezier curve, and Bezier curve with $n+1$ control points is [8],

$$P_{[t_0,t_1]}(t) = \sum_{i=0}^{n} \begin{bmatrix} n \\ i \end{bmatrix} (\frac{t_1 - t}{t_1 - t_0})^{n-i} (\frac{t - t_0}{t_1 - t_0})^{i} p_i, i \in \{0, 1, ..., n\} \tag{8}$$

Where $p_i, i = \{0, 1, ..., n\}$ are discrete control points generated by the path planner, $(t_1 - t_0)$ is the duration for UAV to cover current segment, where t_0 is initial time and t_1 is the final time. $B_i^n(t)$ are Bernstein polynomials.

Thus the whole path can be represented by parametric curves. The speed and acceleration can be calculated by taking the first and second derivative of the parametric curve. Bezier curve has the ability of balancing the speed and acceleration by adjusting the duration. The curve segments abut, which is able to agree with m parametric derivative continuity, called G^m or mth order parametric continuity [8].

Fourth Order Bezier Curve for G^1 Continuity. G^1 continuity means the continuity with the first derivative of the parametric curves, which ensures the tangent continuity at the joint of the curves. Let first discuss (8) and (9), assume $t_0 = 0$ and $t_1 = 1$, then the initial and final first order derivatives are,

$$P^{(1)}_{[0,1]}(0) = n(p_1 - p_0), P^{(1)}_{[0,1]}(1) = n(p_n - p_{n-1}) \tag{9}$$

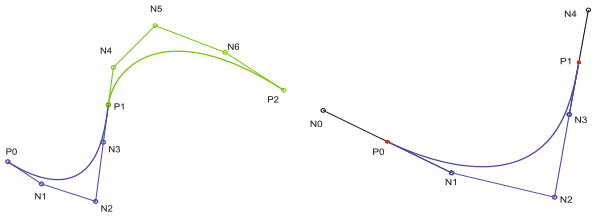


Fig. 3. Node interpolation for velocity continues. (a) A G^1 continuous curve between two curves by applying interpolation at the joint. (b) In order to ensure velocity continuity at the joint, P_0 is the middle point of N_0 and N_1, P_1 is the middle point of N_3 and N_4.

Let take Fig. 3(a) for example, two curves which are controlled by $\{P_0, N_1, N_2, N_3, P_1\}$ and $\{P_1, N_4, N_5, N_6, P_2\}$ respectively. The first-order derivatives of each segment at the joint node are,

$$P^{(1)}_L(1) = 4(P_1 - N_3), P^{(1)}_R(0) = 4(N_4 - P_1) \tag{10}$$

Where $P^{(1)}_L(1)$ denotes the final derivative of the left curve, and $P^{(1)}_R(0)$ denotes the initial derivative of the right curve. In order to keep the tangent continuity between the two curves, two parts of (10) are required to be equal, that is,

$$4(P_1 - N_3) = 4(N_4 - P_1) \tag{11}$$

Thus,

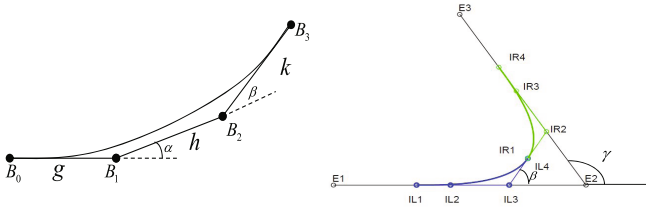
$$P_1) = (N_4 + N_3)/2 \tag{12}$$

It means that P_1 must be the middle point to ensure G^1 continuity. Thus, this paper introduces the simple middle interpolation method to generate first order smooth path(see in Fig. 3(b)).

Cubic Spline with Eight Node Interpolation for G^2 Continuity. Second order continuity of two adjacent curves ensures the continuity of the curvature. This paper adopts the method that was proposed by [7], which designs the cubic Bezier spiral to generate G^2 continuous path by applying interpolation in the

adjacent piecewise linear segments. The method introduces a cubic Bezier curve which is controlled by four control points. The cubic spline still obeys the form of equation (3) with $n = 3$, and let $t_0 = 0$ and $t_1 = 1$, then the initial and final second order derivatives are,

$$P^{(2)}_{[0,1]}(0) = 6(p_1 - 2p_2 + p_3), P^{(2)}_{[0,1]}(1) = 6(p_0 - 2p_1 + p_2) \tag{13}$$



(a) (b)

Fig. 4. The cubic spiral method for G2 continuous curve. (a) Cubic Bezier curve that can represented by angles and distances between the control points. (b) Final solution of G2 continuous method which proposed by [7] and solved by [8].

In [7], the proposed Beizer spiral algorithm is assumed to be composed by translation and rotation (see in Fig. 4(a)). B_0, B_1, B_2, B_3 are the control points of the cubic curve, α is the angle between $(B_1 - B_0)$ and $(B_2 - B_1)$, β is the angle between $(B_2 - B_1)$ and $(B_3 - B_2)$, $g = ||B_1 - B_0||$, $h = ||B_2 - B_1||$, $k = ||B_3 - B_2||$. If let $B_0 = (0, 0)$, the Bezier curve can be represented as,

$$\boldsymbol{C(t)} = (x(t), y(t)) \tag{14}$$

where,

$$\begin{aligned} x(t) = 3g(1-t^2)t + 3(g + hcos\alpha)(1-t)t^2 + (h + hcos\alpha + kcos(\alpha + \beta)) \\ y(t) = 3hsin\alpha(1-t)t^2 + (hsin\alpha + ksin(\alpha + \beta))t^3 \end{aligned} \tag{15}$$

In order to ensure curvature continuity at the joint of two curves, the parameters of the cubic Bezier can be adjusted. Analytical solution was given for this method in [8] which is illustrated in Fig. 4(b), where E_1, E_2, E_3 are original control points for the local segment. $IL1$, $IL2$, $IL3$, $IL4$ and $IR1$, $IR2$, $IR3$, $IR4$ are the designed control points of the proposed method, where

$$\begin{aligned} IL1 = E_2 - d \cdot U_1, IL2 = IL1 + g_L \cdot U_1, IL3 = IL2 + h_L \cdot U_1 \\ IL4 = IL3 + k_L \cdot U_M, IR4 = E_2 + d \cdot U_2, IR3 = IR4 - g_R \cdot U_2 \\ IR2 = IR3 - h_g \cdot U_2, IR1 = IR2 - k_g \cdot U_M \end{aligned} \tag{16}$$

Where,

$$\begin{aligned} d = 1.1228 \cdot sin\beta / K_{max}(cos\beta)^2, h_L = h_g = 0.346d \\ g_L = g_R = 0.58h_L, k_L = k_R = 1.31h_l cos\beta \\ U_1 = \frac{(E_2 - E_1)}{||E_2 - E_1||}, U_2 = \frac{(E_3 - E_2)}{||E_3 - E_2||}, U_M = (IR_2 - IL_3)/||IR_2 - IL_3|| \end{aligned} \tag{17}$$

The angle $\beta = \gamma/2$, K_{max} is the maximum curvature that allowed for the current robot. Thus the second derivatives at the joint of the two curves, which are consisted by $IL1$, $IL2$, $IL3$, $IL4$ and $IR1$, $IR2$, $IR3$, $IR4$ respectively, are,

$$P_{IL}^{(2)}(0) = 6(-h_L \cdot U_1 + k_L \cdot U_M), P_{IR}^{(2)}(1) = 6(h_R U_2 - k_R U_M) \tag{18}$$

As $h_L = h_g$ and $g_L = g_R$, thus $||P_{IL}^{(2)}(0)|| = ||P_{IR}^{(2)}(1)||$. Substitute them into the above equation then,

$$P_{IR}^{(2)}(1) - P_{IL}^{(2)}(0) = -h_L(U_1 + U_2) + k_L \cdot U_M = (1.71cos\beta - 1)h_L(U_1 + U_2) \tag{19}$$

Hence, the angle β can be well picked to ensure direction consistency, that is, the continuity of the curvature.

3.3 Tackling Emergency Threats

This paper proposes two methods to manage moving and pop-up obstacles: 1) One is the replanning method that tries to find a new path from current position to the goal, which is shown in Fig. 5(a). UAV is able to find a path without noticing the moving obstacle. After a finite time duration, UAV noticed that the previous planned path will collide with the moving obstacle. Then the our planner is able to replan a new path from current position to the goal (see in Fig. 5(b)). 2)The other method is called multiple-GART (M-GART), which is able to plan several possible paths simultaneously. M-GART generates multiple

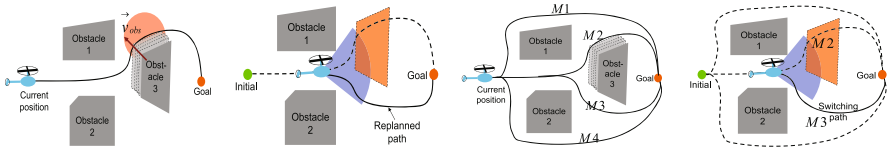


Fig. 5. Illustration of GART replanning and on-line switching methods. (a) Path planner planned a path for UAV without noticing of the moving obstacle. (b) UAV on board sensors detect the moving obstacle, where the fanshaped region is the detection range. Then path planner plans a new path for UAV, which is the solid line. (c)M-GART can plan several possible paths simultaneously. (d)If collision is detected by the sensors, then UAV picks a feasible from the previous planned paths.

pathes by disabling the extending of the branches which have already reached the goal. It is shown in Fig. 5(c), M-GART planned four paths simultaneously. If obstacle is detected to collide with previous planned reference path, then UAV switches to a nearby planned safe path (see in Fig. 5(d), $M3$ is chosen for switching).

This paper proposes the strategy that if the distance from UAV current position to the potential dangerous region is big enough, then GART tries to replan a new path to support collision avoiding. If the distance is close that leaves not enough time to replan a path, then execute on-line switching to pick a previous planned safe path to avoid collision.

4 Simulation and Analysis

This Section illustrates the performance of the proposed method, where each segment of the framework is illustrated with simulation testes to prove its efficiency. Let the two-dimension (2D) space as $100 \times 100m^2$, the obstacles are generated randomly. The initial position is $(0, 0)$, the goal position is $(99, 99)$ (see in Fig. 6(a). It is shown in Fig. 6(a), GART finds a feasible path within 3.5 seconds after 752 iterations. G^1 and G^2 path smoothing methods are tested, which are illustrated in Fig. 6(b) and Fig. 6(c). The results prove that our methods have good performance in generating smoothed reference for UAV.

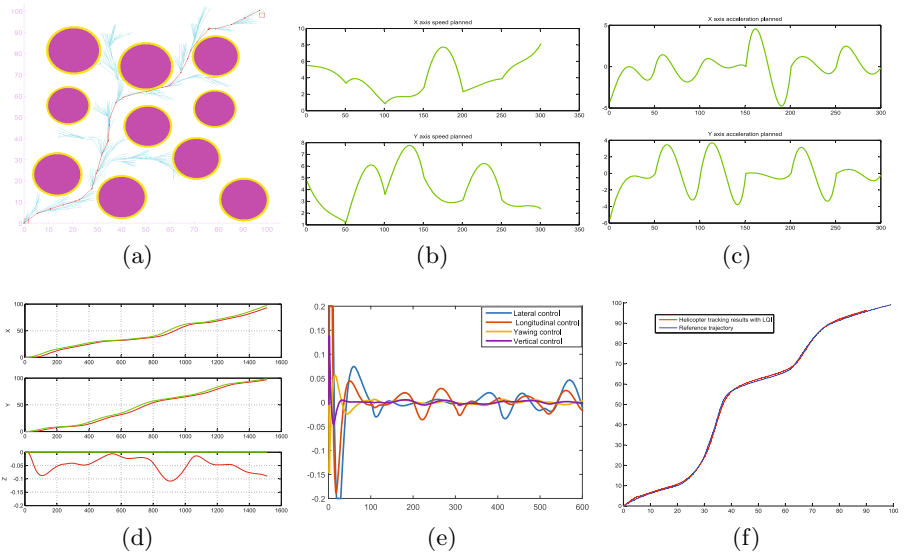
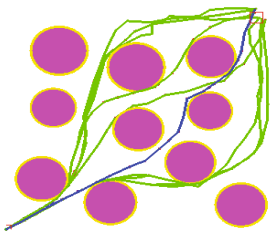


Fig. 6. Simulations for path planner, path smoother, and tracking controller. (a) Exploring tree generated by GART. (b)G^1 smoother with velocity continuity. (c)G^2 smoother with acceleration continuity. (d)The tracking results of position reference tracking controller, reference is the green line, tracking result is the red line. (e)The control inputs corresponding to the reference. (f)The velocity reference tracking results.

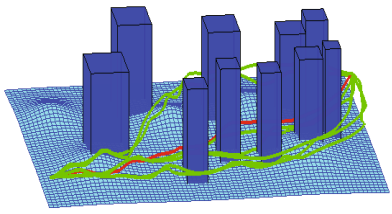
Then we test the efficiency of our proposed tracking controller and path smoother. The SERVOHELI-40[13] is used as the system to be tested which is discussed in Section 2.2. The reference for position reference controller is generated by G^1, and the tracking results are shown in Fig. 6(d). The control inputs generated by G^2 smoother are illustrated in Fig. 6(e) which are continuous and the tracking errors are proved to be small(see in Fig. 6(f)).

Our M-GART algorithm is able to generate multiple paths for on-the-flight switching, the on-the-flight safe switching is with better performance than replanning in time efficiency. 2D and 3D examples are represented in Fig. 7. The blue (Fig. 7(a)) and red(Fig7(b)) lines are minimal cost paths. This paper uses the method to generate 10 paths in both 2D and 3D environments. The iterations, time and cost of each path are supported in Fig. 7(c). It is illustrated that the time and iterations increase fast for finding more feasible pathes. However, because GART holds the near optimal property, the first generated path obtains small cost.

For emergency threat manager, it can be seen in Fig. 8, the simulation tests are intuitively illustrated in 2D environment. The UAV is assumed to be unknown of the moving obstacle in prior, where the obstacle's moving speed $V_{ostacle}$ is $\sqrt{2}m/s$ (see in Fig. 8(b)). In Fig. 8(a), UAV first chooses a path (the red line) which has minimal cost to the goal. Then UAV finds the moving obstacle with on-board sensors which detection range is $10m$. As the UAV have a relative high speed, the distance to the obstacle does not alow path replanning. Thus UAV at postion P_{UAV} tries to find a feasible switching path within



(a) 2D M-GART example.

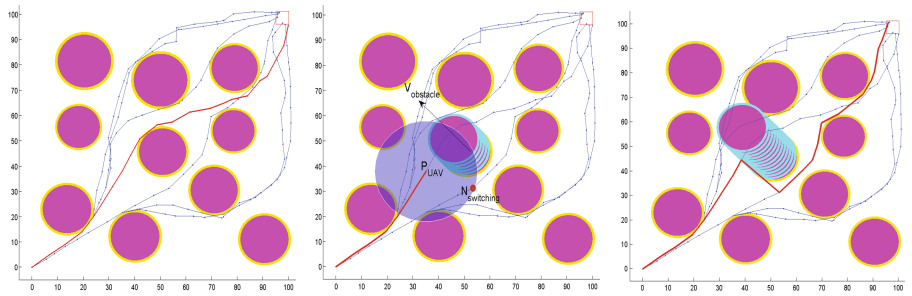


(b) 3D M-GART example.

		Number of Path Generated							
		1	2	3	4	5	6	7	8
2D Scenario	Iterations	310	934	1136	1482	2217	4147	4560	4892
	Cost	115.97	116.03	127.78	127.80	117.55	126.71	127.89	130.28
	Times(s)	1.287	5.555	7.33	10.66	20.89	49.25	65.83	78.24
3D Scenario	Iterations	752	1518	2121	2319	3058	4040	4811	6500
	Cost	147.43	146.88	156.54	169.90	170.86	146.09	159.40	164.14
	Times(s)	3.456	9.115	15.393	17.869	28.391	41.247	45.274	59.692

(c) Details of M-GART on planning multiple path.

Fig. 7. The results generated by proposed M-GART.



(a) (b) (c)

Fig. 8. Emergency threat management. (a) The path planner first find the best as reference. (b) UAV finds the moving obstacle with on-board sensors, then it tries to switch to other paths as UAV is too near to the obstacle. (c) The red path is the final navigation path.

detection range of $20m$ (the blue circle in Fig. 8(b)). UAV calculates the cost of each path, then chooses $N_{swtiching}$ as entry node which obtains minimal cost and the corresponding path as the switching path. The final safe path without collision is the red path that shows in Fig. 8(c).

5 Conclusion

This paper proposed a novel framework for UAV which requires agile autonomous in cluttered environments, while the proposed framework is not only suitable for UAV. Four parts are covered in this paper which include path planner, path smoother, trajectory tracking controller and emergency threat manager. The advantage of the proposed planner is that it obtains the ability of taking both kinematic and dynamic constraints into consideration, where the dynamics is treated as constraint to adjust the path smoother to meet the requirement of the actuator. This paper proposes a new way of dealing with emergency threats, where both replanning and multiple planning are introduced to support on-line switching to avoid collision. The method is based on RRT*, which is proved to be probabilistic complete. Four sets of simulations are implemented, results show that the framework works well with all kinds of environments, but time efficiency can not be guaranteed. Thus, for further work, GART will be improved to speedup the planning process.

References

1. Frazzoli, E., Dahleh, M.A., Feron, E.: Real-time motion planning for agile autonomous vehicles. Journal of Guidance, Control, and Dynamics **25**(11), 116–129 (2002)

2. Scholer, F., la Cour-Harbo, A., Bisgaard, M.: Generating Configuration Spaces and Visibility Graphs from a Geometric Workspace for UAV Path Planning. Autonomous Robots. Springer (2012)
3. Yang, K., Sukkarieh, S.: Real-time continuous curvature path planning of UAVs in cluttered environments. In: Proc. of IEEE 5th International Symposium on Mechatronics and its Applications, pp. 1–6. IEEE Press, Amman (2008)
4. De Filippis, L., Guglieri, G., Quagliotti, F.: Path planning strategies for UAVS in 3D environments. Journal of Intelligent and Robotic Systems **65**, 247–264 (2012)
5. Hasircioglu, I., Topcuoglu, H.R., Ermis, M.: M.: 3-D path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms. In: Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation, pp. 1499–1506. IEEE Press, Atlanta (2008)
6. Miller, B., Stepanyan, K., Miller, A., et al.: 3D path planning in a threat environment. In: Proc. of 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 6864–6869. IEEE Press, Orlando (2011)
7. Walton, D.J., Meek, D.S., Ali, J.M.: Planar G2 transition curves composed of cubic Bzier spiral segments. Journal of Computational and Applied Mathematics **157**(2), 453–476 (2003)
8. Yang, K., Sukkarieh, S.: An analytical continuous-curvature path-smoothing algorithm. IEEE Transactions on Robotics **26**(3), 561–568 (2010)
9. Koyuncu, E., Inalhan, G.: A probabilistic B-spline motion planning algorithm for unmanned helicopters flying in dense 3d environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008), pp. 815–821. IEEE Press Nice, France (2008)
10. Van Den Berg, J., Ferguson, D., Kuffner, J.: Anytime path planning and replanning in dynamic environments. In: Proc. of IEEE International Conference on Robotics and Automation(ICRA 2006), pp. 2366–2371. IEEE Press, Orlando (2006)
11. Yoshida, E., Esteves, C., Belousov, I., et al.: Planning 3-d collision-free dynamic robotic motion through iterative reshaping. IEEE Transactions on Robotics **24**(5), 1186–1198 (2008)
12. Yang, L., QI, J., Jiang, Z., Xiao, J., et al.: Guiding attraction based random tree path planning under uncertainty: dedicate for UAV. In: Proc. of IEEE International Conference on Mechatronics and Automation, pp. 1182–1187. IEEE Press, Tianjin (2014)
13. Song, D., Qi, J., Dai, L., et al.: Modelling a small-size unmanned helicopter using optimal estimation in the frequency domain. International Journal of Intelligent Systems Technologies and Applications **8**(1), 70–85 (2010)
14. Karaman, S., Frazzoli, E.: Sampling-based algorithms for optimal motion planning. The International Journal of Robotics Research **30**(7), 846–894 (2011)
15. Su, H., Wang, X., Lin, Z.: Flocking of multi-agents with a virtual leader. IEEE Transactions on Automatic Control **54**(2), 293–307 (2009)
16. Mahmoud, M.S., Koesdwiady, A.B.: Improved digital tracking controller design for pilot-scale unmanned helicopter. Journal of the Franklin Institute **349**(1), 42–58 (2012)