

High-Performance Path Planning of UAV under Complex Environment

Mohammed Abozied Hassan Abozied and Shiyin Qin

Abstract—This paper presents a high-performance path planning algorithm against resource and physical constraints under complex environment based on a modified RRTs algorithm so as to generate a fast and near optimal 3D UAV collision-free path with good smoothness and optimality. The advantages of the proposed algorithm lie in its low computational complexity and small memory to model complicated spaces with a large number of obstacles thus it can meet the requirements of implementation using low cost embedded processors. A series of simulation experiments and comparative analysis are carried out about four benchmark scenarios with VPB-RRT and DRRT algorithms to verify the predominance of our algorithm. The implementation results with hardware in the loop on a low cost embedded board based on Atmel SAM3X8E ARM Cortex-M3 CPU validated its performance effect. Meanwhile, various results demonstrated its robustness and effectiveness to play an important role in the online UAV mission generation.

Index Terms—Path planning, rapidly-exploring random tree (RRT), unmanned aerial vehicle (UAV), Obstacle avoidance

I. INTRODUCTION

UAVs have presented significant interest in the last few years, particularly, with its increased availability and widely growing applications in civil, and military fields. They can be used in several environments over large areas much faster than ground vehicles with the feature of covering from above, especially with recent developments in processors and onboard sensors which provide a tremendous decrease in weight, cost, and size of UAVs. The achievement of increasing demands for special UAVs missions attracts the researchers' interest in realizing the whole UAVs autonomy. Path planning can be considered a key element in developing autonomous systems to accomplish different missions. The term of path planning in robotics field is defined by finding a collision free optimal path from starting to goal points under a given environment and vehicle constraints [1]. Unfortunately, 2D path planning algorithms introduce a weakness when dealing with complex 3D environments and complicated UAV structure constraints. The basic principles and theories of path planning are deeply introduced in [1]-[3]. The recent research over the past decades can be classified into two main approaches: global and local path planning [4]. Global path planning is based on global world knowledge, it requires a good complete map

of terrain, obstacles and non-flying zones which are usually executed before the mission. On the other hand, local path planning is based on immediate sensory horizon detection, so it can respond to dynamic environments which lead to an interaction between the algorithm and the environmental changes generating a new path based on sensors detections. Global path planning techniques can be classified into two categories, classical and heuristic-based methods [5], [6]. In classical techniques, we can find an optimal solution if it exists while heuristic-based algorithms search for a solution with good quality focusing on time reduction over path optimality. The classical methods for path planning are potential field method (PF) [7]-[9], visibility graph [10] and grid methods [11], [12]. Although the potential field method can be considered one of the most important methods for path planning it has some limitations, including trap local minima and oscillation in narrow environments [6]. Grid methods use cells to form the world space environment that is exponentially proportional to its dimensions which need an increasing memory for representing complex environments [13]. Most of the classical methods drawbacks have been solved using sampling based and heuristic methods based on their nature of randomization to find a high-quality solution considering execution time reduction. Sampling based method can be classified into two main categories: single-query and multiple-query variants, the most famous algorithms for these two categories are the rapidly exploring random tree algorithm (RRT) and probabilistic roadmap algorithm (PRM) respectively [1]. The limitations introduced by these techniques regarding online UAV path planning are long execution time in the complicated large environment, the generated path is not optimal and may or may not satisfy the UAV physical and environmental constraints. Heuristic methods were introduced for solving the problems of path optimality based on Bio-inspired algorithms which originated in swarm intelligence field, it mainly consists of simple agents population which interact locally with their environment and with one another as in genetic algorithms (GA) [14], [15], ant colony optimization (ACO) [36], particle swarm optimization (PSO) [16], artificial bee colony algorithms (ABC) [6], [37]. These algorithms deal with a set of feasible solutions that are evaluated each time step according to fitness function, with time the particles are accelerated to converge to an optimal solution or evolving the recent population to next generations according to their fitness, then applying crossover and mutation until reaching pre-set value of iterations number or finding an optimal solution. These algorithms use the principle of mutation for optimizing the path which needs an increasing number of iterations leading to long execution time, thus, these

Manuscript received July 9, 2016; revised October 12, 2016. This work was partly supported by National Natural Science Foundation of China under Grant 61273350 and Grant U1435220).

The authors are with the School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China (e-mail: mah21181@hotmail.com; qsy@buaa.edu.cn).

algorithms are usually used for offline path planning [13].

II. RELATED WORKS

Recent researches on UAVs path planning focus on a subset of desired features for ideal real-time online path planner. UAV ideal path planner must consider path optimality, completeness, execution time, safety, smoothness, running functionally on embedded processor and dynamic constraints [1], [16], [17]. Although RRTs algorithms present a solution for most of the classical approach limitations it introduces weakness because it doesn't satisfy necessary conditions for asymptotic optimality and dynamics constraints. The development of RRTs algorithms goes through, running the RRT multiple times, deleting and rebuilding parts of the tree and running multiple trees concurrently. Many types of research have been carried out by modifying RRTs algorithm to be more convenient for the problem of UAV path planning, Dynamic-Domain RRTs DDRRT [18], Rapidly-exploring Random Graphs (RRGs) [19], Information-rich RRT (IRRT) [20], improved RRT* algorithm which introduces D* Lite to solve path planning problems in 3-D environment for UAV is proposed in [21], adaptive RRT algorithm based on dynamic step (DRRT) for UAV path planning [22], speeding up RRTs algorithm through parallelization on large-scale distributed memory using the message passing [23], multiple trees for team of vehicles path planning [24], Closed Loop Rapidly-exploring Random Trees (CL-RRT) [25], Potential Guided Directional-RRT* [26], RRT-A* algorithm [27], Guided RRT [28], Guiding attraction based random tree (GART) [29], Medial Axis RRT (MARRT) [30], Rapidly explore random tree policy iteration (RRTPI) [31] and Variable probability based bidirectional RRT algorithm (VPB-RRT) [32].

In this paper, we are devoted to the design and implementation of online UAV path planning using low cost embedded board that is compatible with on the shelf open source autopilots. This work presents a modified UAV path planning algorithm based on a modified RRT algorithm considering execution time, obstacle avoidance, smoothing and optimality for the generated path, UAV physical constraints (maximum turning rate) and the required memory for implementing the algorithm, to achieve the following requirements:

- 1) Smoothing procedure based on fillet path geometry considers the UAV physical constraints for selecting a new node to be added to the generated path, it also reduces the algorithm execution time by checking new segment for collision only if it satisfies the UAV dynamic constraints.
- 2) Reduce the algorithm execution time by modifying and integrating the work introduced in the literature [22], [32] and apply virtual world space procedure for reducing the collision check frequency.
- 3) A simple optimization algorithm for fast high-quality path generation.
- 4) Code modification and simplification to meet the implementation memory and processing constraints of the low cost embedded boards.

The algorithm tested using several scenarios for evaluating its performance, also, it's implemented on low cost embedded board based on Atmel SAM3X8E ARM Cortex-M3 CPU core microcontroller and tested using hardware in the loop simulation.

III. MODELING ANALYSIS OF COMPLEX ENVIRONMENT

Modeling the space to be explored and the expected environment obstacles are a key task in the UAV path planner. In this work, we define a starting point S and goal point D. We address the problem of defining the world space based only on the start and end points as shown in Fig.1-a,b where in some complex spaces as in Fig.1-a, the algorithm may be trapped to find the feasible path.

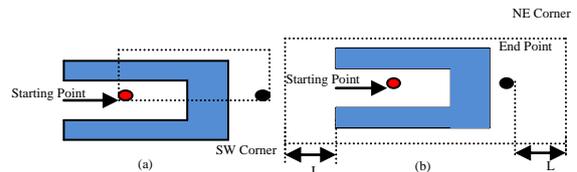


Fig. 1. World space corners specification.

A. Modeling of Feasible Flight Space

The proposed algorithm adaptively chooses the world space southwest and northeast corners as shown in Fig.1-b based on the minimum position of obstacles or the starting point and maximum position of obstacles or end point, biased by distance (L) which satisfies the UAV dynamic constraints, safety, and algorithm capability to find the path. The boundaries of the workspace are modeled as shown in Fig.1-b and (Equ.1,2) where NE is the northeast corner, SW is southwest corner, R_{obs} is the obstacle region.

$$NE = \max(D, R_{obs}) + L \quad (1)$$

$$SW = \min(S, R_{obs}) - L \quad (2)$$

The workspace $W \in R^n$ where $n \geq 2$, is divided into two regions, obstacle region R_{Obs} , and free region R_{Free} , the mission of path planner is to find feasible path from starting point (S) to destination point (D) avoiding obstacles, the path node set (P) is defined as $P \in R_{Free}$ and $P \cap R_{Obs} = \emptyset$ [29].

B. Modeling of Threatening Areas and Obstacles

There are many obstacles modeling methods introduced in details in [1]. Considering all the obstacle boundaries, especially with the problem of uncertainty of obstacle detection we need an extended memory for describing the environment obstacles coordinates which are not suitable for implementation using embedded processors.

The problem of uncertainty is discussed in [29]. The sensors used for detecting the workspace obstacles SAR radar, camera, laser range finder, etc., which can detect the obstacles with different accuracy depending on the state and sensor accuracy. Inaccurate measurements of obstacle position in the workspace lead to unknown boundaries. A policy is described for mapping the workspace obstacles uncertainty as Φ_i where the obstacle and free regions are described by $(R_{Obs} + \Phi_i)$ and $(R_{Free} - \Phi_i)$ respectively.

by the path $\overline{W_{i-1}A} \rightarrow \widehat{AB} \rightarrow \overline{BW_{i+1}}$ which is a fully smooth path, and follows the minimum turning angle constraint for UAV, a new random node will not be processed for collision check unless the angle β generated by this node is greater than the minimum UAV turning angle, this reduces the collision check frequency and reduces execution time.

B. Fast checking of Collision Based on Virtual Dynamic Workspace

The basic principle in RRTs algorithm is to develop a tree from start to end points and checking collision for each new node added to the tree. In the case of complicated environments that contains a high number of obstacles, collision check for new node added to the tree is the main problem that consumes most of the execution time where collision free points must be checked with all the workspace obstacles. We introduce the dynamic virtual workspace idea where we only check obstacles inside the virtual dynamic space to save time and enhance the algorithm performance. The dynamic virtual space is generated as shown in Fig.4 the red rectangle is a virtual dynamic space where the edges are defined by the parent node in the tree and the newly generated node segment.

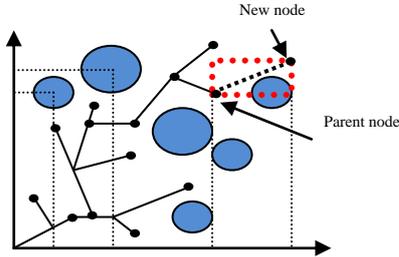


Fig. 4. Virtual dynamic world space for reducing collision check.

C. Improvement and Optimization of Feasible Path

Sampling based techniques introduce a weakness in optimizing the generated path which is an important issue for an ideal UAV path planner in the name of safety and fuel saving. The proposed algorithm adapts the heuristic optimization algorithm introduced in [35] to present an optimization algorithm which consists of three stages with low computation load and can run in real time.

1) Path improvement by pruning

The objective of this procedure is to remove redundant nodes through the generated path, for a given path P consists of n waypoints and a waypoint W_{i+1} is redundant if the local path $\overline{W_i W_{i+2}}$ is collision free then we can remove W_{i+1} as shown in Fig. 5 the path (abcde) if the line (ac) is collision free then waypoint (b) can be removed, then check the line (ad) in this case, it's not collision free so waypoint (c) can't be removed, continue checking the remaining waypoints, the resultant path is (ace) after removing two needless waypoints, for more details see algorithm I.

2) Path optimization with short-cut

Two stages short-cut path optimization applied for generating a high-quality path. In the first stage, we uniformly split $\overline{W_{i-1}W_i}$ and $\overline{W_i W_{i+1}}$ According to Equ.12, 13 where P_1 and P_2 are uniformly distributed points $\in \overline{W_{i-1}W_i}$ and $\overline{W_i W_{i+1}}$ respectively if we find $\overline{P_1 P_2}$ which

is collision free we can replace the $\overline{W_{i-1}W_i} \rightarrow \overline{W_i W_{i+1}}$ by the path $\overline{W_{i-1}P_1} \rightarrow \overline{P_1 P_2} \rightarrow \overline{P_2 W_{i+1}}$ as shown in Fig. 6, the pseudo code is for this procedure is as in algorithm 2.

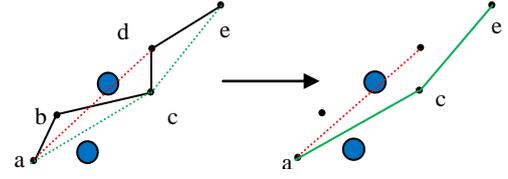


Fig. 5. Path pruning procedure.

Algorithm 1: Pruning Algorithm
 new path(1:M)=prune(Path(1:N),workspace obstacles)

Path original path of N waypoints
 new path optimized path of M waypoints, $M \leq n$
 ptr is a pointer for waypoints in the original path

```

1: new path(1)=Path(1)
2: Ptr=2
3: While ptr<=N-1 do
4:   If (collision(new path(end),Path (ptr+1))
5:     new path=[new path(1);Path(ptr)]
6:   end
7:   Ptr=ptr+1
8: end
    
```

Algorithm 2:Optimization with short-cut-1
 new path(w_1, w_2, \dots, w_m)=short-cut1(Path(w_1, w_2, \dots, w_n), workspace obstacles) where m, n is number of waypoints for the new and old paths respectively.

```

1: Initialization:
   new path (1) = Path(1), tmp=path(2),ptr=1,ptr1=1
2: for i=1:number of path waypoints -2
3:   flag=1
4:   for sigma=0:0.1:1
5:     p1= (sigma)*Path(ptr+1) + (1-sigma)*tmp
6:     p2=(1-sigma)*Path(ptr+2) + (sigma)*P(ptr+1)
7:     if (p1p2 segment is collision free)
8:       if Turning radius is greater than minimum
9:         new path (ptr1+1)=p1
10:        new path (ptr1+2)=p2
11:        tmp=p2
12:        flag=0
13:        GoTo 21
14:      end
15:    else
16:      new path (ptr1+1)=path (ptr+1)
17:      new path (ptr1+2)=path (ptr+2)
18:      tmp=path (ptr,1:3)
19:    end
20:  end
21:  ptr=ptr+1
22:  ptr1=ptr1+2
23:  if flag=1
24:    tmp=Path(ptr);
25:  end
26: end
27: if (new path (ptr1),Path(end) is collision free)
28:   new path(ptr1+1)=path (end)
29: else
30:   new path (ptr1+1)=path (end-1)
31:   new path (ptr1+2)=Path (end)
    
```

32 end

$$P1_{0.1-0.9} = \text{sigma} * W_i + (1 - \text{sigma}) * w_{i-1} \quad (12)$$

$$P2_{0.1-0.9} = \text{sigma} * W_{i+1} + (1 - \text{sigma}) * W_i \quad (13)$$

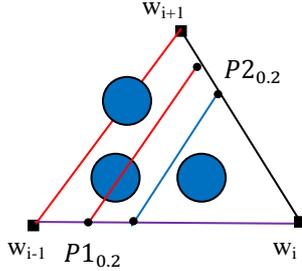


Fig . 6. Path short-cut-I.

The second stage, the algorithm optimizes the generated suboptimal path from the previous stage applying algorithm 3, as shown in Fig. 7 W_{i-1} , w_i , and w_{i+1} are three adjacent waypoints, for any $x \in \overline{W_i W_{i+1}}$ and $\overline{W_{i-1} x}$ is collision free line then w_i is replaced by x , after checking all possible values for x . The algorithm selects the position x that ensures the smallest length for $\overline{W_{i-1} W_i W_{i+1}}$ (position denoted by the blue line in Fig. 7), repeat this procedure for the remaining waypoints through the path which will lead to generating the shortest path from start to end point, the resolution for selecting x depends on the length of $\overline{W_i W_{i+1}}$.

Algorithm 3: Optimization with short-cut-2
 new path(w_1, w_2, \dots, w_m)=short-cut2(Path(w_1, w_2, \dots, w_n), workspace obstacles) where m, n is number of waypoints for the new and old paths respectively.

```

1:  Initialization:
    tmp = Path(2), new path(1)=Path (1), ptr =2 , tmp2=0
    flag=0
2:  while ptr <= n-1
3:      if new path (ptr-1)=Path(ptr)
4:          new path (ptr-1)=new path (ptr-2)
5:      end
6:      for sigma = 0:0.1:1
7:          p1 = sigma*Path(ptr+1,1:3) + (1-
8:          sigma)*tmp;
9:          if (p1,new Path (end) is collision free)
10:             if turning radius is greater than minimum
11:                 tmp2=p1
12:                 flag=1
13:                 GoTo 16
14:             end
15:         end
16:     end
17:     if flag>0
18:         new path (ptr)=tmp2
19:         tmp2=0
20:         flag=0
21:     else
22:         new path (ptr)=Path (ptr)
23:     end
24:     ptr=ptr+1
25:     tmp=Path (ptr)
26: end
27: new path(ptr)=Path (end)
    
```

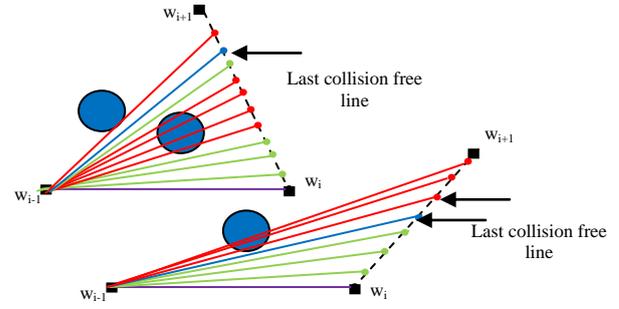


Fig. 7. Path short-cut-II.

D. Dynamic step algorithm

The proposed algorithm adopts dynamic step algorithm introduced in [22] to explore the world space with directed tree by defining dynamic segment length based on the last node added to the growing tree and the end point multiplied by factor α , where α is less than 1 as in Equ.14, it also controls the segment length based on the horizon capacity with obstacles.

$$\text{segment length} = \max(\text{end node} - \text{tree}(\text{end})) * \alpha \quad (14)$$

The coverage capability is also enhanced by selecting a new random point based on the current tree location and the end point which ensures a complete coverage of the extended area in parts of a second. Based on the concept introduced in [32], by using the tree coverage rate to control the goal point deviation probability when it's low, the algorithm devoted to search the unexplored area, on the other hand, when the coverage rate is high the extension will be directed to the goal point, the details of calculating the tree coverage rate and probability is found in [32].

V. SIMULATION EXPERIMENT AND COMPARATIVE ANALYSIS

In this section, we demonstrate the simulation framework applied for evaluating the modified algorithm performance using four benchmark maps for testing the proposed algorithm effectiveness and feasibility. For further prove of the proposed algorithm efficiency and robustness we run the proposed algorithm 100 times for each of the four benchmark maps while changing the obstacles position, start, and goal points through a comparative study of our algorithm and VPB-RRT, and DRRT algorithms. Finally, we implement the proposed algorithm on low cost embedded board based on Atmel SAM3X8E ARM Cortex-M3 CPU core microcontroller and test it though hardware in the loop simulation.

A. Simulation Platform and Environment Conditions

The performance of the proposed algorithm is evaluated through extensive simulation for different scenarios for static obstacles benchmark map simulated using MATLAB programming tool. The environment hardware is personal computer Intel(R) Core i7-4510U CPU @ 2.00GHz, 4 GB of RAM. The characteristics of the generated benchmark maps summarized in Table-I The evaluation parameters for the generated path are path length, execution time, and smoothness which is already guaranteed by using fillet path

geometry. The modified algorithm implemented using the new procedures applied for decreasing the execution time and path smoothing and optimization. The results of the four benchmark maps presented in Fig.8-11. The results illustrate that the modified algorithm based on the dynamic step and random points generation based on coverage probability doesn't considerably improve the first solution, these procedures just made the algorithm able to find the first solution in complex space in short time, the key point in the proposed modified algorithm is to rapidly find the first feasible solution as presented in Table-II, after that apply the optimization procedures which can run in real time because of its simplicity and small execution time. The results also present a significant decrease in collision check, and the number of nodes added to the constructed tree making the algorithm able to rapidly searches the space and calculates the first path as presented in Table-II. The modified algorithm execution time is effected by changing segment step length by tuning the parameter α in Equ.14, running the algorithm for different times in different environments the value of $\alpha = 0.25$ is the best value which makes the space exploration time independent of its dimensions. The results also present the effectiveness of path smoothing procedure where the algorithm is simple and can run in real time compared with other path smoothing techniques, it also makes the generated path satisfy the UAV dynamics constraints and maneuver capabilities.

TABLE I: PARAMETERS OF BENCHMARK MAPS

case	Map	Size(km)	Start point	End point
			[North East Up] km	[North East Up] km
1	Random distributed obstacles	100×100	[0 0 0]	[100 100 1.5]
2	Uniformly distributed obstacles	100×100	[0 0 0]	[100 100 1]
3	Back and forth	80 × 80	[40 0 0]	[80 80 3]
4	Bug trap	800 ×800	[400 400 0]	[370 600 3]

TABLE II: BENCHMARK MAPS SIMULATION RESULTS

	Case-1	Case-2	Case-3	Case-4
Time to find first path(Sec)	0.0617	0.0823	0.2148	0.3047
First path cost (km)	179.215 6	183.6875	254.6458	176.5112
Time to find Optimized path (Sec)	0.1529	0.1728	0.5174	0.6042
Optimal path cost (km)	142.315 9	143.6791	175.9344	131.7231
Number of collision check	98	108	243	1785
Number of nodes in the generated tree	76	72	99	268

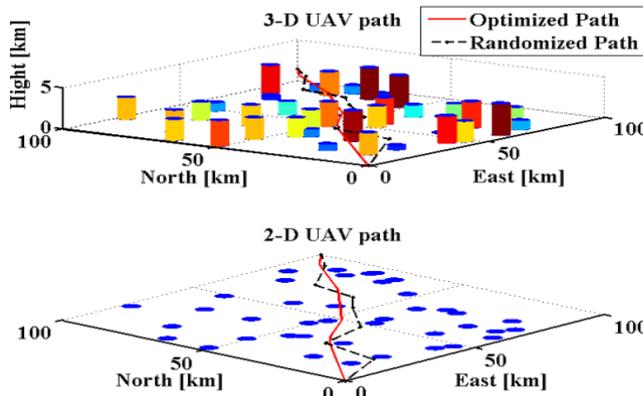


Fig. 8. Random distributed obstacles.

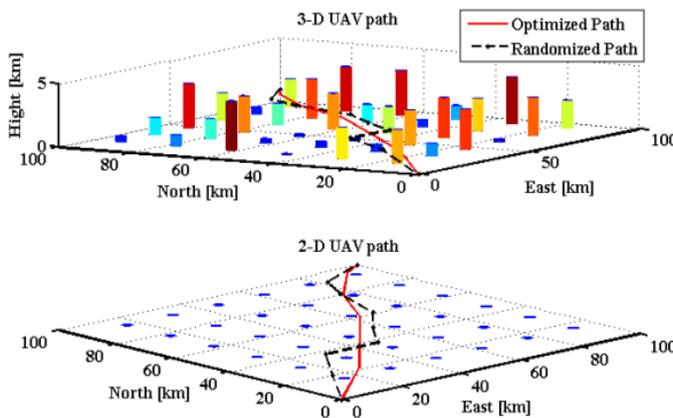
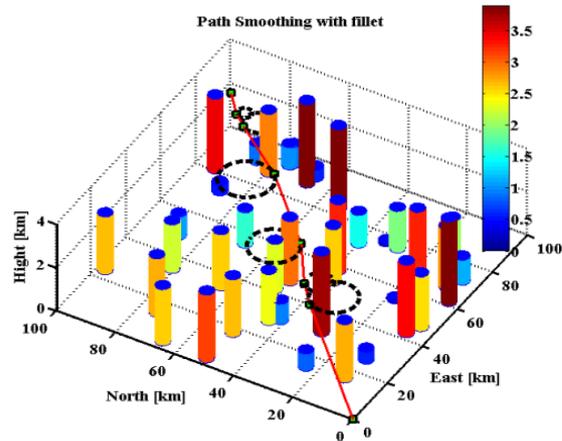
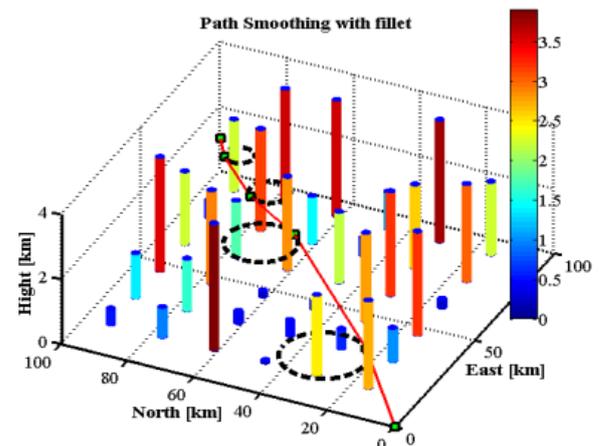


Fig. 9. Uniformly distributed obstacles.



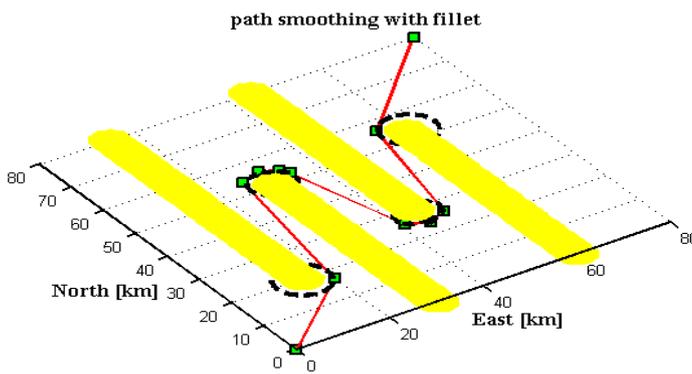


Fig. 10. Back and forth obstacle distribution.

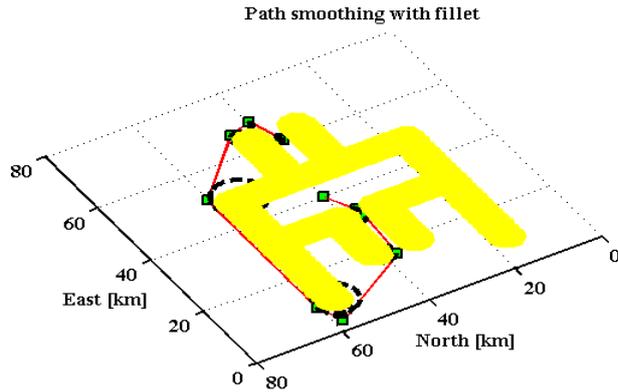
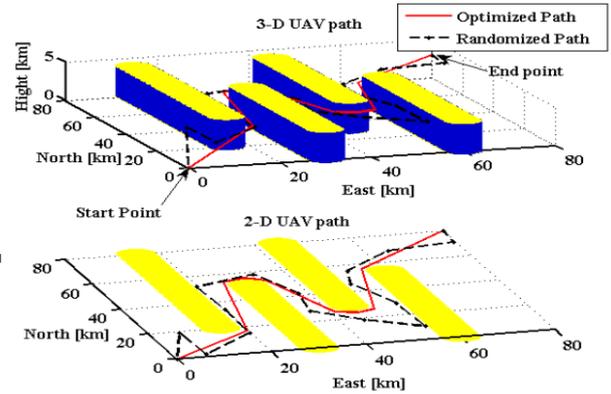
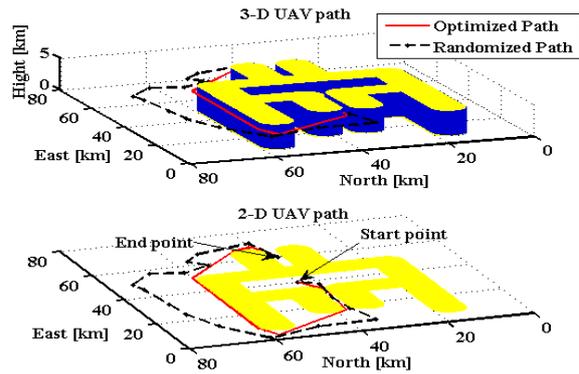


Fig. 11. Bug trap obstacle distribution.



B. Comparative Analysis and Verification of Experiment Results

Comparative analysis has been conducted for evaluating the proposed algorithm effectiveness and robustness. The simulation performed for the previous benchmark maps by running the algorithm for each one 100 times and comparing the generated path length and execution time with VBP-

RRT and DRRT algorithms results as shown in Fig.12-15 and Table-III. The proposed algorithm presents stability for calculating a near optimal path within the margin of 6 [km] for 100 times, its execution time for complex space is less than 0.65 [Sec] making the algorithm effective for online UAV path planning.

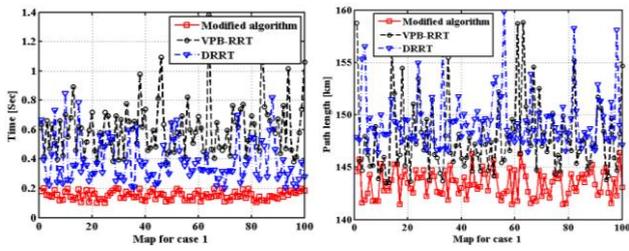


Fig. 12. Comparative analysis for Random distributed obstacles benchmark map.

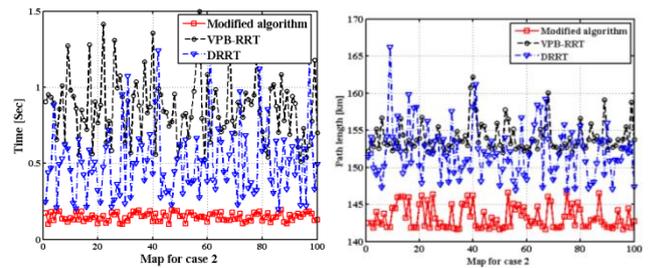


Fig.13. Comparative analysis results for Uniformly distributed obstacles benchmark map.

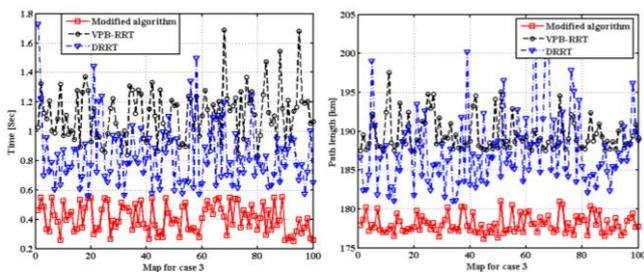


Fig. 14. Comparative analysis results for Back and forth obstacle distribution benchmark map

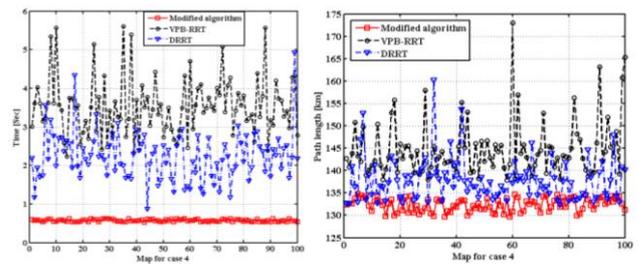


Fig.15. Comparative analysis results for Bug trap obstacle distribution benchmark map.

TABLE III: COMPARATIVE RESULTS OF THE MODIFIED ALGORITHM WITH VPB-RRT, AND DRRT ALGORITHM

Comparative terms Case No.	Method	Path length (km)				Time (Sec)			
		Mean	STD	Max	Min	Mean	STD	Max	Min
Case-1	DDRT	149.3293	2.6766	159.8064	146.4019	0.3775	0.1464	0.8464	0.2015
	VPB-RRT	147.7872	3.5064	158.8403	143.4675	0.6052	0.1834	1.3798	0.3786
	Our algorithm	143.4626	1.2775	146.3832	141.4214	0.1513	0.0286	0.1991	0.1016
Case-2	DDRT	151.9808	3.5944	166.2159	146.8738	0.5024	0.2384	1.3171	0.1997
	VPB-RRT	153.9279	1.9649	162.1891	151.9859	0.8583	0.2110	1.4969	0.5133
	Our algorithm	143.2249	1.5638	146.5884	141.6168	0.1487	0.0271	0.1955	0.1004
Case-3	DDRT	187.3783	4.8276	203.9373	181.0182	0.8287	0.2182	1.7258	0.5605
	VPB-RRT	189.4955	2.1708	197.5501	187.3908	1.1116	0.1758	1.6876	0.8323
	Our algorithm	178.0477	1.1854	181.0105	176.1316	0.4045	0.0944	0.5491	0.2507
Case-4	DDRT	137.7702	4.7823	160.2379	132.4097	2.2283	0.6271	4.9250	0.8767
	VPB-RRT	144.6598	6.2901	173.0370	137.8024	3.5173	0.7457	5.6076	2.2275
	Our algorithm	132.3471	1.4822	134.7425	129.6734	0.5706	0.0302	0.6264	0.5269

VI. HARDWARE IMPLEMENTATION AND VALIDATION

RRT algorithm Computational complexity prevents implementation for direct application to real-time UAV path planning, so an optimization of the memory required for modeling the world space obstacles and dynamic memory allocation, the procedure applied for optimizing the algorithm programming code which gave the ability for implementing the proposed algorithm on Arduino Due microcontroller board based on Atmel SAM3X8E ARM Cortex-M3 CPU. The complete specifications for the proposed board are shown in Table-IV.

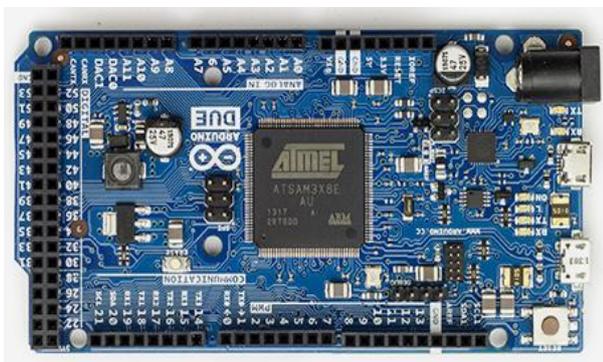


Fig.16 Arduino Due microcontroller board

A. Hardware Architecture and Implementation Conditions

Hardware in the loop (HIL) simulation is applied for evaluating the proposed algorithm as shown in Fig.17. Path planning algorithm is installed on Arduino Due board, a computer used for simulating environment obstacles and

mission selection control and another computer used for UAV flight simulation. The Arduino board interfaced with the two computers through three universal Synchronous asynchronous receiver/transmitter (USART). USART1 is dedicated to receiving obstacles data and directly buffer the received data into obstacle memory buffer in two modes based on the nature of the environment, if the environment is dynamic the newly received data will replace the previous one, but if the environment obstacles is static the newly received obstacle data considered as obstacles detected during mission then the newly received data will be added to the previously received data, the size of obstacles memory buffer is limited only for maximum 100 obstacles. Mission selection control data received through USART2 for controlling the path planning algorithm mode of operation. The generated path is sent to UAV flight simulation computer through USART3.

TABLE IV: EMBEDDED BOARD SPECIFICATIONS

Feature	Specifications
Microcontroller	AT91SAM3X8E
CPU	ARM Cortex-M3
Operating Voltage	3.3V
Input Voltage (limits)	6-16V
Digital I/O Pins	54
Analog Input Pins	12
Flash Memory	512 KB
SRAM	96 KB
Clock Speed	84 MHz
Length	101.52 mm
Width	53.3 mm
Weight	36 g

B. Implementation Results and Validation Analysis

The performance analysis of the proposed algorithm through (HIL) simulation presented an adequate performance for generating high-quality obstacle free UAV path in a complex environment in time less than one second. The results of the HIL simulation presented in Fig.18 for U-section scenario using 6DOF UAV simulation under Matlab environment.

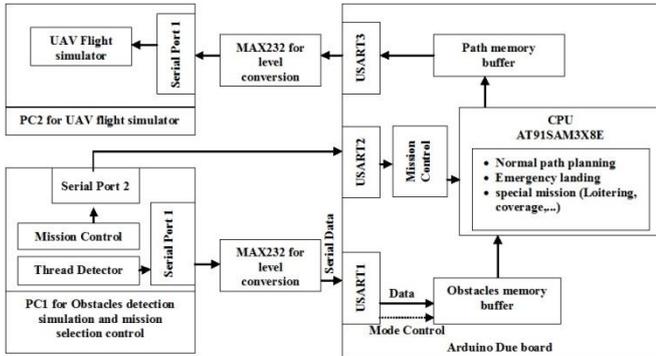


Fig. 17. HIL simulation architecture.

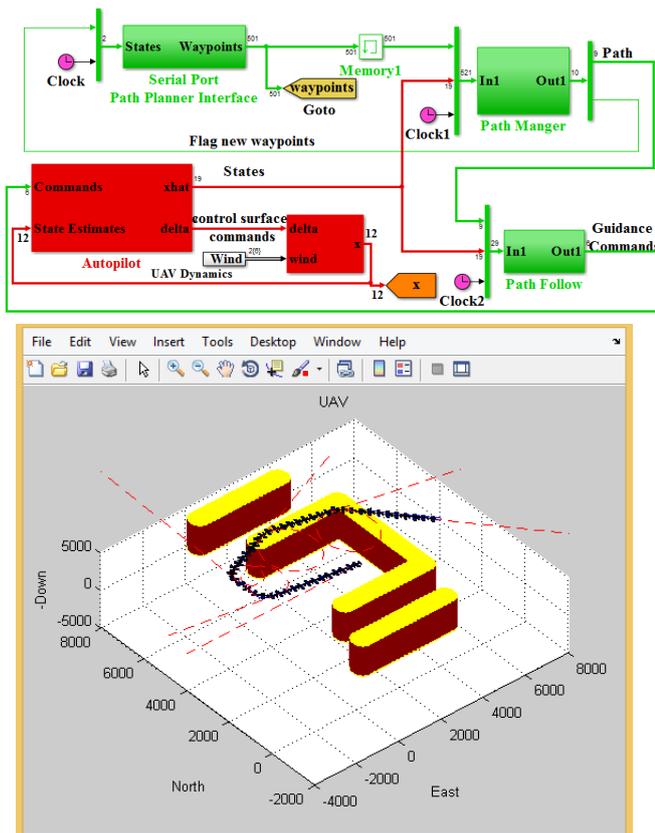


Fig. 18. HIL Simulation results.

VII. CONCLUSION

In view of the resource and physical constraints under complex environment, we present a high-performance UAV path planner based on modified RRT algorithm with virtual dynamic workspace for speeding up the RRTs searching, fillet path geometry for smoothing the generated paths against constraints, effective and well-suited optimization procedure, powerful searching tree based on dynamic step and coverage probability so as to significantly enhance its

performance in execution time and path length compared with the other two algorithms in [22] and [32]. A series of experimental results demonstrate our proposed algorithm is also characterized with efficiency and simplicity to be implemented on low cost embedded boards for real world UAV applications. Furthermore, our presented work can be extended to deal with obstacles detection for online UAV path planning by using a more powerful embedded board. In the further research projects, we will try to use bio-inspired approaches to solving the global optimization problem and enhancing the integrative performance of UAV path planning.

ACKNOWLEDGEMENT

This work was partly supported by National Natural Science Foundation of China under Grant 61273350 and Grant U1435220).

REFERENCES

- [1] S. M. L. Valle, *Planning Algorithms*, New York: Cambridge University Press, 2006.
- [2] J. C. Latombe, *Robot Motion Planning*, Boston, MA: Springer US, 1991.
- [3] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation*, Cambridge, MA: MIT Press, 2005.
- [4] K. Sedighi, K. Ashenayi, T. Manikas, R. Wainwright, and H.-M. Tai, "Autonomous local path planning for a mobile robot using a genetic algorithm," in *Proc. the 2004 Congress on Evolutionary Computation*.
- [5] E. Masehian and D. Sedighzadeh, "Classic and heuristic approaches in robot motion planning - a chronological review," in *Proc. the World academy of Science, Engineering, and Technology*, vol. 23, pp. 101-106, 2007.
- [6] M. Agarwal and P. Goel, "Path planning of mobile robots using bee colony algorithm," *MIT International Journal of Computer Science and Information Technology*, vol. 3, no. 2, pp. 86-89, Aug 2013.
- [7] P. Vadakkepat, T. H. Lee, and L. Xin, "Application of evolutionary artificial potential field in robot soccer system," in *Proc. the Joint 9th IFSA World Congress and 20th NAFIPS International Conference*, vol. 5, pp. 2781-2785, 2001.
- [8] L. C. A. Pimenta, A. R. Fonseca, G. A. S. Pereira, R. C. Mesquita, E. J. Silva, W. M. Caminhas, and M. F. M. Campos, "Robot navigation based on electrostatic field computation," *IEEE Transactions on Magnetics*, vol. 42, no. 4, pp. 1459-1462, Apr. 2006.
- [9] O. Cetin, I. Zagli, and G. Yilmaz, "Establishing obstacle and collision free communication relay for UAVs with artificial potential fields," *Journal of Intelligent and Robotic Systems*, vol. 69, no. 1-4, pp. 361-372, Jan. 2013.
- [10] F. Scholer, A. L. Cour-Harbo, and M. Bisgaard, "Generating approximative minimum length paths in 3D for UAVs," *2012 IEEE Intelligent Vehicles Symposium*, Spain, June 3-7, 2012.
- [11] M. Seda, "Roadmap method vs. cell decomposition in robot motion planning," in *Proc. the 6th WSEAS International Conference on Signal Processing, Robotics and Automation*, Grees, pp. 127-132, 2007.
- [12] C. Goerzen, Z. Kong, and B. Mettler, "A survey of motion planning algorithms from the perspective of autonomous UAV guidance," in *Proc. the Selected papers from the 2nd International Symposium on UAVs*, Reno, Nevada, U.S.A. June 8-10, 2009, pp. 65-100, 2009.
- [13] L. Yang, J. Qi, J. Xiao, and X. Yong, "A literature review of UAV 3D path planning," in *Proc. the 11th World Congress on Intelligent Control and Automation Shenyang*, China, June 29 - July 4, 2014.
- [14] Y. V. Pehlivanoglu, O. Baysal, and A. Hacioglu, "Path planning for autonomous UAV via vibrational genetic algorithm," *Aircraft Engineering and Aerospace Technology*, vol. 79, no. 4, pp. 352-359, Jul. 2007.
- [15] H. Ergezer and K. Leblebicioglu, "Path Planning for UAVs for Maximum Information Collection," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 1, pp. 502-520, Jan. 2013.
- [16] J. L. Foo, J. Knutzon, V. Kalivarapu, J. Oliver, and E. Winer, "Path planning of unmanned aerial vehicles using B-splines and particle swarm optimization," *Journal of Aerospace Computing Information and Communication*, vol. 6, no. 4, pp. 271-290, Apr. 2009.
- [17] United States Air Force, *Unmanned Aircraft Systems Flight Plan 2009-2047*, Technical Report, Headquarters, United States Air Force, UAS, 2009.
- [18] A. Yershova, L. Jaillet, T. Simeon, and S. Lavalle, "Dynamic-domain RRTs: efficient exploration by controlling the sampling domain," in *Proc. the 2005 IEEE International Conference on Robotics and Automation*, 2005.
- [19] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," *Robotics: Science and Systems VI*, 2010.

- [20] G. Hollinger and G. Sukhatme, "Sampling-based motion planning for robotic information gathering," *Robotics: Science and Systems IX*, 2013.
- [21] L. Yang, Z. Wei-Guo, S. Jing-Ping, and L. Guang-Wen, "A path planning method based on improved RRT," in *Proc. 2014 IEEE Chinese Guidance, Navigation and Control Conference*, pp. 564-567, 8-10 August. 2014.
- [22] N. Lin and Y.-L. Zhang, "An adaptive RRT based on dynamic step for UAVs route planning," in *Proc. 2014 IEEE 5th International Conference on Software Engineering and Service Science.*, 2014.
- [23] D. Devaurs, T. Simeon, and J. Cortes, "Parallelizing RRT on large-scale distributed-memory architectures," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 571-579, Apr. 2013.
- [24] A. A. Neto, D. Macharet, L. Chaimowicz, and M. Campos, "Path planning with multiple rapidly-exploring random trees for teams of robots," in *Proc. the 2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 25-29 November. 2013.
- [25] A. H. Qureshi, K. F. Iqbal, S. M. Qamar, F. Islam, Y. Ayaz, and N. Muhammad, "Potential guided directional-RRT* for accelerated motion planning in cluttered environments," in *Proc. the 2013 IEEE International Conference on Mechatronics and Automation*, 2013.
- [26] J. Li, S. Liu, B. Zhang, and X. Zhao, "RRT-A* Motion planning algorithm for non-holonomic mobile robot," in *Proc. the 2014 SICE Annual Conference (SICE)*, 2014.
- [27] D. Lee and D. H. Shim, "RRT-based path planning for fixed-wing UAVs with arrival time and approach direction constraints," in *Proc. the 2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp.317-328, 2014.
- [28] J. Zhang, M. Wisse, and M. Bharatheesha, "Guided RRT: A greedy search strategy for kinodynamic motion planning," in *Proc. the 2014 13th International Conference on Control Automation Robotics & Vision*, pp. 480-485, 10-12 Dec. 2014.
- [29] L. Yang, J. Qi, Z. Jiang, D. Song, J. Han, and J. Xiao, "Guiding attraction based random tree path planning under uncertainty: Dedicate for UAV," in *Proc. the 2014 IEEE International Conference on Mechatronics and Automation*, pp.1182-1187, 3-6 Aug. 2014.
- [30] J. Denny, E. Greco, S. Thomas, and N. M. Amato, "MARRT: Medial Axis biased rapidly-exploring random trees," in *Proc. the 2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 90-97, May 31 2014-June 7 2014
- [31] M. S. Sivamurugan and B. Ravindran, "RRTPI: Policy iteration on continuous domains using rapidly-exploring random trees," in *Proc. the 2014 IEEE International Conference on Robotics and Automation*, pp. 4362-4367, 2014.
- [32] W. Xinggang, G. Cong, and L. Yibo, "Variable probability based bidirectional RRT algorithm for UAV path planning," in *Proc. the 26th Chinese Control and Decision Conference*, pp. 2217-2222, 2014.
- [33] P. Yao, H. Wang, and Z. Su, "Real-time path planning of unmanned aerial vehicle for target tracking and obstacle avoidance in complex dynamic environment," *Aerospace Science and Technology*, vol. 47, pp. 269-279, 2015.
- [34] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*, Princeton, NJ: Princeton University Press, 2012.
- [35] R. Geraerts and M. H. Overmars, "Creating high-quality paths for motion planning," *The International Journal of Robotics Research*, vol. 26, no. 8, pp. 845-863, Aug. 2007.
- [36] C.-T. Cheng, K. Fallahi, H. Leung, and C. K. Tse, "Cooperative path planner for UAVs using ACO algorithm with Gaussian distribution functions," in *Proc. the 2009 IEEE International Symposium on Circuits and Systems*, pp. 173-176, 2009.
- [37] H. Duan and S. Li, "Artificial bee colony-based direct collocation for reentry trajectory optimization of hypersonic vehicle," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 615-626, Jan. 2015.



Mohammed Abozied Hassan Abozied received his bachelor's and a master's degrees in electrical engineering from MTC in 2003 and 2010 respectively, and he is PHD candidate in School of Automation Science and Electrical Engineering, Beihang University, Beijing



Shiyin Qin received his bachelor's and master's degrees in engineering science in automatic controls and industrial systems engineering from Lanzhou Jiaotong University in 1978 and 1984, respectively, and his PhD degree in Industrial Control Engineering and Intelligent Automation from Zhejiang University in 1990. He is now a Professor at the School of Automation Science and Electrical Engineering in Beihang University (Beijing University of Aeronautics and Astronautics).

His current major research topics include image processing and pattern recognition, intelligent control systems of UAVs, intelligent optimising controls of large scale multi-robot hybrid systems, complex systems and complexity science, etc.