International Journal of Engineering

Journal Homepage: www.ije.ir

Any-time Randomized Kinodynamic Path Planning Algorithm in Dynamic Environments with Application to Quadrotor

E. Taheri*

Electrical Engineering Department, Malek Ashtar University of Technology, Tehran, Iran

PAPER INFO

Paper history:
Received 24 May 2021
Received in revised form 30 August 2021
Accepted 05 September 2021

Keywords:
Kinodynamic Path Planning
Rapidly-exploring Random Trees
Quadrotor
Collision Avoidance
Dynamic Obstacles

ABSTRACT

Kinodynamic path planning is an open challenge in unmanned autonomous vehicles and is considered an NP-Hard problem. Planning a feasible path for vertical take-off and landing quadrotor (VTOL-Q) from an initial state to a target state in 3D space by considering the environmental constraints such as moving obstacles avoidance and non-holonomic constraints such as hard bounds of VTOL-Q is the key motivation of this study. To this end, let us propose the any-time randomized kinodynamic (ATRK) path-planning algorithm applicable in the VTOL-Q. ATRK path-planning algorithm is based on the Rapidly-exploring random trees (RRT) and consists of three main components: high-level, mid-level, and low-level controller. The high-level controller utilizes a randomized sampling-based approach to generate offspring vertices for rapid exploring and expanding in the configuration space. The mid-level controller uses the any-time method to avoid collision with moving obstacles. The low-level controller with a six-DOF dynamic model accounts for the kinodynamic constraints of VTOL-Q in the randomized offspring vertices to plan a feasible path. Simulation results on three different test-scenario demonstrate the kinodynamic constraints of the VTOL-Q are integrated into the randomized offspring vertices. Also, in presence of moving obstacles, the ATRK re-plans the path in the local area as through an any-time approach.

doi: *10.5829/ije.2021.34.10a.17*

NOMENCLATURE

$\eta = [\eta_1\ \eta_2]$	Quadrotor position and orientation in the E-frame.	M_B	System inertia matrix.
$\nu = [\nu_1\ \nu_2]$	Quadrotor linear and angular velocity in the B-frame.	C_B	Coriolis-centripetal matrix.
$J: R^6 \to R^{6\times 6}$	Coordinate transformation matrix relating the two frames.	$\Lambda = \left[f_B \quad \tau_B \right]^T$	Quadrotor force and torque.

1. INTRODUCTION

In the last decade, a dramatic interest has appeared in vertical take-off and landing quadrotors. In fact, these vehicles are already becoming a portion of everyday reality, especially in modern societies. Several huge potential domains for stakeholders such as surveillance and inspection, search and rescue, military and defence, civil, agriculture, payload delivery, environmental protection, etc. require these vehicles to perform special missions in unpredictable and time-variable situations [1-6]. Current status and future perspectives on the development of quadrotors were studied in literature [7, 8]. One of the standing problems in the field of quadrotor autonomy is the achievement of real-time guidance, navigation, and control (GNC) system. To meet the need for autonomy in the quadrotor, many public and private attempts are made with regard to the path planning issues in these vehicles. Path planning is defined as a vital requirement in autonomous robotics which is described by algorithms that transform high-level demand (mission) into low-level demand (movement) [9, 10]. Thanks to robotic science, the path planning problem has important applications in areas such as drug design, animation design, assist surgeons, graphics, games theory, virtual and augmented reality, very-large-scale

*Corresponding Author Institutional Email: *taheri.ehsan@mut-es.ac.ir* (E. Taheri)

Please cite this article as: E. Taheri, Any-time Randomized Kinodynamic Path Planning Algorithm in Dynamic Environments with Application to Quadrotor, International Journal of Engineering, Transactions A: Basics Vol. 34, No. 10, (2021), 2360-2370

integration (VLSI) design, etc. In this context, various studies conducted and number of methods have been proposed. Detailed articles survey were accomplished to define the current state of the art of path planning problem in literature [11-13]. For motion planning and autonomy subjects, numerous researches exist in different manuscripts and number of methods are being presented. The intelligent path planning method based on the artificial neural network for the quadrotor is proposed by Khosravian and Maghsoudi [14]. The position and attitude control with considering the under actuated constraints through the recurrent neural network based on non-linear PID is the main evolution of this study. Also, the path planning module is proposed in this paper as a high-level controller. The decentralized fuzzy adaptive sliding mode algorithm for the multi-quadrotor is proposed by Manouchehri et al. [15]. In this approach, affine nonlinear agents with nonlinear control gain are considered. The proposed method is used to plan the formation between a groups of quadrotors. Performance of this method is evaluated through the numerical simulation. Parameters of a target tracker robot are identified through the particle swarm optimization (PSO) algorithm by Sangdani and Tavakolpour-Saleh [16]. Motion equations of robot are described with considering flexible joints. For this purpose a non-linear spring and damper system are defined. Parameters of this system is identified and evaluated though the experimental results. A variable-step RRT* path planning algorithm is proposed for a quadrotor to increase the convergence rate and efficiency in a below-canopy environment [17]. A cost function is used in the variable-step RRT* to optimize the offspring vertices connection and the Bezier curve is utilized to smooth the path. Allen and Pavone [18] proposed a real-time path planning algorithm that is applied on a physical quadrotor operating in an indoor area. The proposed algorithm utilizes an offline-online computation procedure, a machine-learning approach, an optimal cost distance metric function, and trajectory smoothing to plan a path. A chaos-based logistic map is integrated with the particle swarm optimization algorithm to improve the path optimality for an autonomous unmanned aerial vehicle (UAV) [19]. The optimal controller is proposed by Mashadi et al. [20] for tracking the predefined path with application to the integrated driver/vehicle system. Controller parameters are designed through the LQR and GA algorithm. Hence, controller performance is appropriated in the various driving styles, road conditions and initial errors of vehicle position and orientation. The integrated driver/DYC control system is presented to overcome the vehicle path tracking problems with considering the vehicle active safety [21]. In this controller, the heading error and the lateral deviation between the real and desired path are received as inputs and then the steering angle and a direct yaw moment are designed as outputs. PID coefficients are optimized through the heuristic approach. Hence, the performance of the driver/DYC controller is acceptable on lane-change maneuvers, double-lane changes, J-turn and other desired tracks. Finally, Monte-Carlo simulation is utilized to assess the performance of this method. Gao et al. [22] introduced the path planning problem which is divided into front-end path searching and back-end path filtration.

The path is planned by the sampling-based informed method and the planned path is optimized through the penalty functions. This approach is implemented in a lightweight quadrotor platform and is evaluated in unknown cluttered indoor and outdoor workspaces. In, The optimal path planning algorithm was proposed by Heidari and Saska [23] for a quadrotor by considering various objective functions, such as energy efficiency. The necessary conditions of optimality are formulated here as a normal form of a two-point boundary-value problem using Pontryagin's minimum principle. The control forces were added into the artificial potential field (APF) to plan an optimal path for a quadrotor [24]. The slack variables in this paper are used to change the constrained optimization problem to an unconstrained problem. However, there are many various criteria to classify the path planning algorithms such as 1) complete algorithms, 2) probabilistic-complete algorithms, 3) deterministic algorithms, 4) randomized algorithms, 5) optimal algorithms, 6) suboptimal or asymptotically optimal algorithms, 7) global algorithms (map-based), 8) local algorithms (sensor-based), 9) real-time algorithms, 10) kinodynamic algorithms, etc. [25-30]. These path planning algorithms are developed through the three main groups of classical, heuristic, andmeta-heuristic [31]. Figure 1 illustrates classical algorithms including family of bug-based (Bug 1 & 2, I-Bug, K-bug, and Tangent-bug), roadmap-based, graph-based, cell decomposition-based, sampling-based, potentialfield-based, etc. Most of the classical algorithms are complete and deterministic. Hence, these are sensitive to parameter tuning and computational-complexity of them increase in the order O (n^2) by

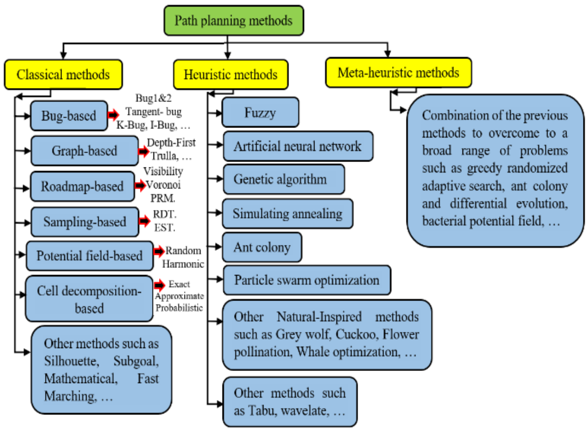

Path planning methods
Classical methods
Heuristic methods
Meta-heuristic methods
Bug-based
Bug1&2
Tangent- bug
K-Bug, I-Bug, …
Graph-based
Depth-First
Trulla, …
Roadmap-based
Visibility
Voronoi
PRM.
Sampling-based
RDT.
EST.
Potential field-based
Random
Harmonic
Cell decomposition-based
Exact
Approximate
Probabilistic
Other methods such as Silhouette, Subgoal, Mathematical, Fast Marching, …
Fuzzy
Artificial neural network
Genetic algorithm
Simulating annealing
Ant colony
Particle swarm optimization
Other Natural-Inspired methods such as Grey wolf, Cuckoo, Flower pollination, Whale optimization, …
Other methods such as Tabu, wavelate, …
Combination of the previous methods to overcome to a broad range of problems such as greedy randomized adaptive search, ant colony and differential evolution, bacterial potential field, …

Figure 1. Classification of path planning methods

increasing the density of obstacles, workspace dimensions, numbers of kinodynamic constraints, etc. While this can be proved the most heuristic and meta-heuristic algorithms are probabilistic-complete.

Remark 1. Probabilistic-complete means that a probability of designing a path goes to one if one path exists and time goes to infinity.

Among the most path planning algorithms mentioned above, the external (kinematic) constraints are considered in the path planning procedure but the internal (differential) constraints are seldom considered. Thus they usually cannot be able to plan a feasible path. This challenge becomes harder when unknown or moving obstacles have appeared in the kinodynamic path planning problem. Also, computational and time complexity of the kinodynamic path planning problem increase in the order O (n^2) by increasing numbers of moving obstacles, kinodynamic constraints, degrees of freedoms, and C-space dimension. These challenges of real-time kinodynamic planning can be formulated into two questions that serve as the key motivation of this study.

Motivating Question 1. Can we propose an any-time path planning algorithm that takes into account both kinematics (moving obstacles avoidance) and dynamics (velocity and acceleration) constraints for VTOL-Q in a simultaneous manner?

Motivating Question 2. Given such a generalized method to kinodynamic path planning, can we utilize a randomized sampling-based approach to reduce the computational and time complexity of the problem?

In response to such motivating questions, this article proposes the any-time randomized kinodynamic (ATRK) path-planning algorithm here as the state-of-the-art in this field applicable in the VTOL-Q. In a nutshell, the main contributions of this article can be summarized as follows:

1) In the proposed ATRK path planning algorithm the reachability of each offspring vertex which is planned through the rapidly-exploring random trees (RRT) method is assessed to assure the quadrotor kinodynamic (kinematic and dynamic) constraints are considered in the planned path. Hence, both dynamic and kinematic (kinodynamic) constraints are integrated into the designed path and it's followed through the quadrotor properly.

2) In the proposed ATRK the randomized sampling-based approach is utilized to reduce the computational and time complexity of the problem. The search tree is expanded in the configuration space in a random manner here. In this way, the total number of vertices is reduced in the search tree compare to the conventional methods. It is worth to be mentioned that the computational and time complexity in the complete path planning algorithm increase in the order O (n^2) by increasing numbers of moving obstacles, kinodynamic constraints, degrees of freedoms, and C-space dimension.

3) The any-time reactive event-based method re-plans the path in the local areas to avoid collision with moving obstacles and prevents trapping the path in the local minima problem.

The rest of this manuscript is organized as follows: a brief background of the quadrotor dynamic model and sampling-based path planning algorithm is presented in section 2. The any-time randomized kinodynamic (ATRK) path-planning algorithm consists of three main components that are presented in section 3. These components are: high-level, mid-level, and low-level controller. Performance and effectiveness of the proposed path planning algorithm in presence of moving and unknown obstacles are assessed through the two different scenarios in section 4, and finally, the conclusion is given in section 5.

2. OVERVIEW OF QUADROTOR DYNAMIC AND SAMPLING-BASED ALGORITHM

2. 1. Quadrotor Kinematic and Dynamic

Typically, the structure of a quadrotor is rigid and symmetric, in which four rotors are attached at the ends of arms in a plus or cross-configurations. The desired forces and moments are generated on the quadrotor through the counter-clockwise rotating in front and back propellers (numbers 1 and 3 in Figure 2) and clockwise rotating in right and left propellers (numbers 2 and 4 in Figure 2). The Euler-Lagrange and Newton-Euler are conventional methods that are utilized to explain dynamics behaviours for a quadrotor [32-35]. Before explaining the dynamic model of a quadrotor, it is necessary to explain the reference frames and kinematic equations. Two reference frames are used to derive the

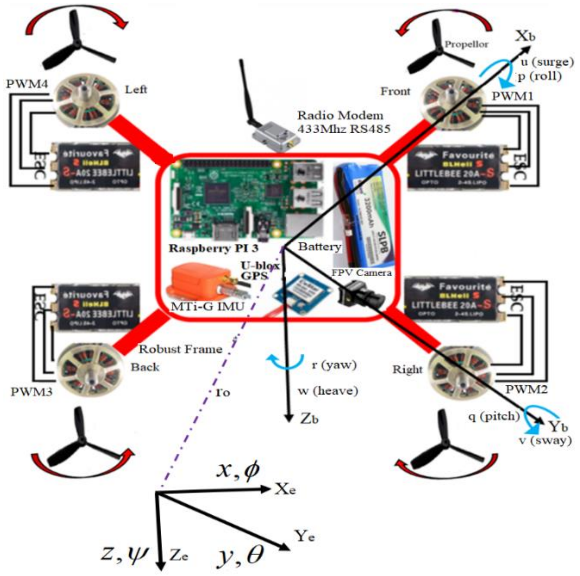


Figure 2. Quadrotor structure and reference frames

dynamic model (Figure 2): The fixed (Earth) frame (E-frame), the mobile (Body) frame (B-frame).

Six DOF are required in 3D-space to explain the quadrotor translation and rotation movement. The quadrotor position and orientation ($\eta \in R^6$) in the E-frame is presented by:

$$\begin{aligned}&\eta = [\eta_1\ \eta_2] = (x, y, z,\\ &\phi \in [-\pi, \pi], \theta \in [-\pi/2, \pi/2], \psi \in [-\pi, \pi]) \in R^6\end{aligned} \quad (1)$$

and the quadrotor linear and angular velocity ($v \in R^6$) in the B-frame is presented through the following vector:

$$v = [v_1\ v_2] = (u, v, w, p, q, r) \in R^6 \quad (2)$$

Quadrotor kinematic motion equations by considering the geometric aspects of motion are defined in the following form where J is $R^6 \to R^{6\times 6}$:

$$\dot{\eta} = J(\eta)v \to \begin{bmatrix}\dot{\eta}_1\\ \dot{\eta}_2\end{bmatrix} = \begin{bmatrix}J_1(\eta_2) & 0_{3\times 3}\\ 0_{3\times 3} & J_2(\eta_2)\end{bmatrix}\begin{bmatrix}v_1\\ v_2\end{bmatrix} \quad (3)$$

where, $\eta_1 \& \eta_2 \in R^3$ presents the position and orientation, and vectors $v_1 \& v_2 \in R^3$ presents the linear and angular velocities in the body coordinate system. Relation between the position and linear velocities is explained through the $J_1(\eta_2) = C_z(\psi).C_y(\theta).C_x(\phi) \in SO(3)$:

$$\begin{aligned}&J_1(\eta_2) =\\ &\begin{bmatrix}c\psi & -s\psi & 0\\ s\psi & c\psi & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}c\theta & 0 & s\theta\\ 0 & 1 & 0\\ -s\theta & 0 & c\theta\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & c\phi & -s\phi\\ 0 & s\phi & c\phi\end{bmatrix} =\\ &\begin{bmatrix}c\psi c\theta & -s\psi c\phi + c\psi s\theta s\phi & s\psi s\phi + c\psi s\theta c\phi\\ s\psi c\theta & c\psi c\phi + s\psi s\theta s\phi & -c\psi s\phi + s\psi s\theta c\phi\\ -s\theta & c\theta s\phi & c\theta c\phi\end{bmatrix}\end{aligned} \quad (4)$$

and the correlation between the orientation and angular velocities is defined through the $J_2(\eta_2)$:

$$J_2(\eta_2) = \begin{bmatrix}1 & s\phi t\theta & c\phi t\theta\\ 0 & c\phi & -s\phi\\ 0 & s\phi/c\theta & c\phi/c\theta\end{bmatrix} \quad (5)$$

Consequently, the quadrotor kinematic motion is defined as below:

$$\dot{x} = w[s\phi s\psi + c\phi c\psi s\theta] - v[c\phi s\psi - c\psi s\phi s\theta] + u[c\psi c\theta]\ \dot{y} = v[c\phi c\psi + s\phi s\psi s\theta] - w[c\psi s\phi - c\phi s\psi s\theta] + u[c\theta s\psi]\ \dot{z} = w[c\phi c\theta] - u[s\theta] + v[c\theta s\phi]\ \dot{\phi} = p + r[c\phi t\theta] + q[s\phi t\theta]\ \dot{\theta} = q[c\phi] - r[s\phi]\ \dot{\psi} = r\frac{c\phi}{c\theta} + q\frac{s\phi}{c\theta} \quad (6)$$

Using the Newton-Euler equation, and by making the following assumptions: 1- The quadrotor frame is supposed to be rigid and symmetrical, 2- The centre of mass and the body-fixed frame origin are assumed to coincide, 3- The propellers are supposedly rigid, 4- Thrust and drag are proportional to the square of the propellers speed, 5- The dynamics of the rotors and propellers are negligible. 6- The inertia matrix (I) is diagonal, in other words, the axes of the B-frame coincide with the body principal axes of inertia, the dynamic model is written in the following form.

Newton's law and Euler's law are stated to explain the total forces and the total torques that are applied to the quadrotor:

$$\begin{aligned}&Newton's\,law \Rightarrow m(v_2 \wedge v_1 + \dot{v}_1) = f_B\\ &= [f_x f_y f_z]^T \in R^3\\ &Euler's\,law \Rightarrow I.\dot{v}_2 + v_2 \wedge (I.v_2) = \tau_B\\ &= [\tau_x \tau_y \tau_z]^T \in R^3\end{aligned} \quad (7)$$

where m=2.1kg is the quadrotor mass,^ is the cross product,

$I = [I_{xx} = 0.00612kg\,m^2, I_{yy} = 0.00612kg\,m^2,$
$, I_{zz} = 0.0137kg\,m^2]^T$

is the inertia matrix, f_B is the quadrotor force, and τ_B is the quadrotor torque. These equations are defined in the state-space in the following form:

$$\begin{bmatrix}mI_{3\times 3} & 0_{3\times 3}\\ 0_{3\times 3} & I\end{bmatrix}\begin{bmatrix}\dot{v}_1^B\\ \dot{v}_2^B\end{bmatrix} + \begin{bmatrix}v_2^B \times (mv_1^B)\\ v_2^B \times (Iv_2^B)\end{bmatrix} = \begin{bmatrix}f_B\\ \tau_B\end{bmatrix} \quad (8)$$

The dynamic model is described in a matrix form:

$$\begin{aligned}&M_B\dot{v} + C_B(v)v = \Lambda = [f_B\ \ \tau_B]^T\\ &M_B = \begin{bmatrix}mI_{3\times 3} & 0_{3\times 3}\\ 0_{3\times 3} & I\end{bmatrix}, C_B = \begin{bmatrix}0_{3\times 3} & -mS(v_1^B)\\ 0_{3\times 3} & S(Iv_2^B)\end{bmatrix}\end{aligned} \quad (9)$$

where M_B is the diagonal and constant system inertia matrix, C_B is the Coriolis-centripetal matrix which is tabulated in Table 1. S (.) is the skew-symmetric operator which is defined as below:

$$S(k) = -S^T(k) = \begin{bmatrix}0 & -k_3 & k_2\\ k_3 & 0 & -k_1\\ -k_2 & k_1 & 0\end{bmatrix} \quad (10)$$

Inertia and Coriolis-centripetal matrix are defined as bellow:

TABLE 1. Quadrotor Parameters of dynamic model

$m = 2.1kg$	$b = 3.21\times 10^{-5} kg\,m^2$
$I_{xx} = I_{yy} = 0.00612kg\,m^2$	$d = 6.45\times 10^{-7} kg\,m^2$
$I_{zz} = 0.0137kg\,m^2$	*l=0.2m*
$J_r = 1.23\times 10^{-5} kgm^2$	*g=9.81* $m\,s^{-2}$

$$M_B = \begin{bmatrix} 2.1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2.1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.00612 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.00612 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0137 \end{bmatrix}$$
$$C_B = \begin{bmatrix} 0 & 0 & 0 & 0 & 2.1w & -2.1v \\ 0 & 0 & 0 & -2.1w & 0 & 2.1u \\ 0 & 0 & 0 & 2.1v & -2.1u & 0 \\ 0 & 0 & 0 & 0 & 0.0137r & -0.00612q \\ 0 & 0 & 0 & -0.0137r & 0 & 0.00612p \\ 0 & 0 & 0 & 0.00612q & -0.00612p & 0 \end{bmatrix} \quad (11)$$

$\Lambda = \left[f_B \quad \tau_B \right]^T$ in the quadrotor (Equation (9)) consists of three components:

$$\begin{aligned} &\Lambda = G_B(\eta) + O_B(v)\Omega + E_B\Omega^2 \\ &\Omega^{Overall} = -\Omega_1^{front} + \Omega_2^{right} - \Omega_3^{back} + \Omega_4^{left} \\ &\Omega = [\Omega_1, \Omega_2, \Omega_3, \Omega_4]^T \end{aligned} \quad (12)$$

where $\Omega(rad\,s^{-1})$ is the propellers speed, G_B is the gravitational force that is generated due to the gravitational acceleration,

$$G_B(\eta) = \left[mgs\theta, -mgc\theta s\phi, -mgc\theta s\phi, 0, 0, 0 \right]^T \quad (13)$$

O_B is the gyroscopic torque that is produced due to the propeller rotation (overall rotors speed is an imbalance or the quadrotor has the roll or pitch),

$$O_B(v) = J_{TP} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ q & -q & q & -q \\ -p & p & -p & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (14)$$

$J_{TP}[Nms^2] \rightarrow total\ inertia\ of\ rotors$

and E_B is the main forces and torques that the $l(m)$ is the distance between the centre of the quadrotor mass and the centre of any propeller, the $d(kg\,m^2)$ is drag factor, and the $b(kg\,m^2)$ is thrust factor.

$$E_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \\ 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ -d & d & -d & d \end{bmatrix} \quad (15)$$

From Equations (9) and (12) it is possible to write:

$$\dot{v} = M_B^{-1}(-C_B(v)v + G_B(\xi) + O_B(v)\Omega + E_B\Omega^2) \quad (16)$$

The quadrotor dynamic in the body frame is defined as follows:

$$\begin{aligned} &\dot{u} = (vr - wq) + gs\theta \\ &\dot{v} = (wp - ur) - gc\theta s\phi \\ &\dot{w} = (uq - vp) - gc\theta s\phi + \frac{U_1}{m} \\ &\dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{J_{TP}}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \\ &\dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{J_{TP}}{I_{yy}} p\Omega + \frac{U_3}{I_{yy}} \\ &\dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}} pq + \frac{U_4}{I_{zz}} \end{aligned} \quad (17)$$

The control signal $U = \left[U_1, U_2, U_3, U_4 \right]^T \in R^4$ is defined as below:

$$\begin{aligned} &U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ &U_2 = lb(-\Omega_2^2 + \Omega_4^2) \\ &U_3 = lb(-\Omega_1^2 + \Omega_3^2) \\ &U_4 = d(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2) \end{aligned} \quad (18)$$

By Equations (17) and (4) the dynamic model in hybrid-frame (linear equations in the E-frame and angular equations in the B-frame) is possible to write:

$$\begin{aligned} &\ddot{x} = (s\psi s\phi + c\psi s\theta c\phi) \frac{U_1}{m} \\ &\ddot{y} = (-c\psi s\phi + s\psi s\theta c\phi) \frac{U_1}{m} \\ &\ddot{z} = -g + (c\theta c\phi) \frac{U_1}{m} \\ &\dot{p} = \frac{I_{yy} - I_{zz}}{I_{xx}} qr - \frac{J_{TP}}{I_{xx}} q\Omega + \frac{U_2}{I_{xx}} \\ &\dot{q} = \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{J_{TP}}{I_{yy}} p\Omega + \frac{U_3}{I_{yy}} \\ &\dot{r} = \frac{I_{xx} - I_{yy}}{I_{zz}} qp + \frac{U_4}{I_{zz}} \end{aligned} \quad (19)$$

Remark 2. If the rotation movement in the quadrotor is achieved with the small angles, the dynamic model (Equation (19)) can be more simplified. That is to say, the dot product between the angular velocities and the inertia of each rotor are negligible.

2. 2. Sampling-based Path Planning Algorithms

Sampling-based path planning algorithms have been proposed since late 90s for NP-hard problems in the high-dimensional workspace and for high-degree-of-freedom robots that are out of reach for deterministic, complete methods [36-38]. The Rapidly-exploring random trees (RRT) is one of the most famous and most studied algorithms for sampling-based path planning issue. The RRT is able to apply the robot's dynamics in the path planning procedure. Also, due to the randomized nature of RRT, the computational and time complexity of the problem are reduced. In a nutshell, RRT is a simple and fast method which is appropriate for real-time and kinodynamic applications.

Remark 3. Here, the kinodynamic term means that both kinematics and dynamics (velocity and acceleration) constraints are taken into account during the path planning procedure in a simultaneous manner. Hence, the planned path through this method is feasible (flyable) for the VTOL-Q. Also, the any-time term means that the quality of initial path improves during the planning time remains. Hence, the VTOL-Q can be avoid collision by unknown or moving obstacles in the local area.

Algorithm 1: The pseudo-code of the standard RRT

```
$(x, x_{free}) \leftarrow$ Configuration/Free space
$V \leftarrow \{q_{ini}, q_{goal}\}$
$E \leftarrow \{\emptyset\}$
$R(V, E) \leftarrow$ Search Tree
$(N, \beta) \leftarrow$ Max. number of iteration and threshold of goal vertex.
$R_c \leftarrow L^2 norm(q_{goal}, q_{new})$
while $i < N$ and $R_c \geq \beta$ do
    $q_{rand} \leftarrow$ The random sampling is generated through the function number 1 ( $x_{free}$ ).
    $V_{near} \leftarrow$ The best candidate parent vertices are determined through the function # 3.
    $q_{parent} \leftarrow$ The parent vertex is determined through the function number 2. $\rhd$ $byEq.(20)$.
    $[q_{new}, \Im] \leftarrow$ The edge and new vertex are generated through the function # 4.
    if $(q_{new}, \Im)$ located in $x_{free}$ then $\rhd$ $byFun.\#5$
        $V \leftarrow \{V \cup q_{new}\}$
        $E \leftarrow \{E \cup \Im\}$
        $R \leftarrow$Update(R)
    endif
    $R_c \leftarrow$ Update $(R_c)$
endwhile
```

Standard RRT framework consists of five main components: 1) Sampling function, 2) Metric function, 3) Branch function, 4) Neighbors function and 5) Collision detection function which are introduced briefly as follows [31]:

1- Sampling function: Random offspring vertices (q_{rand}) are generated by this function to expanding the search tree $R(V, E)$ in the configuration space. Where, (R) is the search tree in free space $C_{free} = C / C_{obstacle}$, (E) is the set of tree edges, and (V) is the set of tree vertices. The random offspring vertices can be guided toward the goal state by adding bias, bridge test, M-Line axis, boundary layer, or other techniques.

2- Metric function: The performance of this function can have a significant influence on the path planning ability. The various strategies are defined: Manhattan, Chessboard, Weighted Euclidean, Adaptive, Diagonal, etc. The metric function (γ) is defined as follows:

$$\gamma(q_{rand}) = g(q_{\text{int}}, q_{parent}) + h(q_{parent}, q_{rand}) \tag{20}$$

3- Neighbors function: This function receives the (q_{rand}) and $\gamma(q_{rand})$ as inputs and determines a subset of the best candidate vertices (Vnear⊂V) among the all available vertices of the search tree (V).

4- Branch function: This function receives the (q_{rand}) and (q_{parent}), then plans an edge $(\Im)$ as an output. If the random offspring vertex is out of reach for the search tree, the new vertex is generated.

5- Collision detection function: The output of the sampling (q_{rand}) and branch functions $(q_{new}, \Im)$ are assessed through this function for located in the free configuration space (c_{free}). The pseudo-code of the standard RRT is presented in Algorithm 1 and also the concept of this is illustrated in Figure 3 [9].

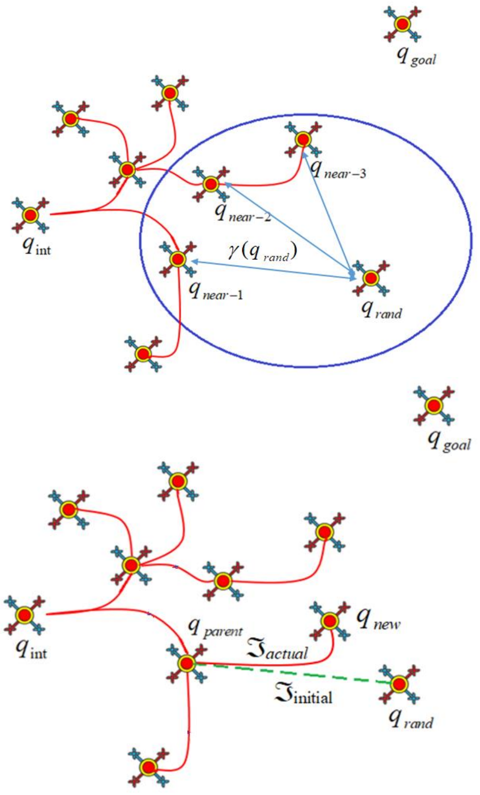


Figure 3. Conceptual representation of RRT path planning algorithm

3. PROPOSED ANY-TIME RANDOMIZED KINODYNAMIC (ATRK) PATH-PLANNING

Here, the ATRK path planning algorithm is proposed to increase the autonomy level in the considered VTOL-Q. The ATRK utilizes the randomized sampling-based method to reduce the time complexity of the path planning problem. By reducing the time complexity, the ATRK is able to rapid re-plan the path in the local areas to avoid collision with moving obstacles. Furthermore, the ATRK integrates both kinematic and dynamic constraints of VTOL-Q in the generated vertices. So, the planned path is feasible for the VTOL-Q. ATRK consists of three main components: high-level, mid-level, and low-level controller which are introduced as follows.

3. 1. High-level Controller The high-level controller utilizes the RRT approach to generate offspring vertices for rapid exploring and expanding in the configuration space. For this purpose, in each iteration, the high-level controller receives several initial parameters such as the configuration space, mission requirements, terminate condition and etc. and then generated a set of vertex and edge in a random manner. In this way, the search tree is expanded in the configuration space until the VTOL-Q reaches the target state threshold. The feasibility of each vertex and edge in the search tree for the VTOL-Q is assessed through the low-level controller as follows, (lines 8-11 Algorithm 2).

3. 2. Low-level Controller The Low-level controller with a six-DOF dynamic model integrates the kinodynamic constraints of VTOL-Q in the randomized offspring vertices to plan a feasible path. The offspring vertices in the low-level controller are converted to the desired references and the control signal is computed to apply to the rotors, (lines 15-16 Algorithm 2). A discrepancy between the output of the high-level controller $\left[q_{new}, \Im\right]$ and the output of the low-level controller $\left[q_{Flyable}, \Im_{Flyable}\right]$ is appeared due to either motion constraints or insufficient time.

3. 3. Mid-level Controller The mid-level controller uses the any-time reactive method to avoid collision with moving obstacles. For this purpose, first, the collision point (CP) and local goal (LG) are determined, then the local search tree is expanded from the current location of VTOL-Q to the LG. The local search tree in comparison to the initial search tree has high priority.

4. TEST-CASES

Performance and effectiveness of the proposed path planning algorithm are evaluated through the three different scenarios as follows:

Scenario 1: It is assumed that the configuration space in the first test-case is a narrow passage in, Figure 4(a) and is a maze-shaped space in, Figure 4(b).

Algorithm 2: Proposed ATRK

```
1: $(x, x_{free}) \leftarrow$ Configuration/Free space
2: $V \leftarrow \{q_{ini}, q_{goal}\}$
3: $E \leftarrow \{\emptyset\}$
4: $R(V,E) \leftarrow$ Search Tree
5: $(N, \beta) \leftarrow$ Max. number of iteration and threshold of goal vertex.
6: $R_c \leftarrow L^2 norm(q_{goal}, q_{new})$
7: while $i < N$ and $R_c \geq \beta$ do
8:     $q_{rand} \leftarrow$ The random sampling is generated through the function number 1 ( $x_{free}$ ).
9:     $V_{near} \leftarrow$ The best candidate parent vertices are determined through the function number 3.
10:    $q_{parent} \leftarrow$ The parent vertex is determined through the function number 2. ⊳ $byEq.(20)$.
11:    $\left[q_{new}, \Im\right] \leftarrow$ The edge and new vertex are generated through the function # 4.
12:    if $(q_{new}, \Im)$ located in $x / x_{free}$ then
13:        go to line 8
14:    else
15:        $[U_1, U_2, U_3, U_4] \leftarrow Low-level\ controller(q_{parent}, q_{new}, \Im)$
16:        $\left[q_{Flyable}, \Im_{Flyable}\right] \leftarrow 6DoF\ \mathrm{mod}\,el\left[U_1, U_2, U_3, U_4\right]$ ⊳ $byEq.(19)$
17:    endif
18:    if $(q_{Flyable}, \Im_{Flyable})$ located in $x_{free}$ then
19:        $V \leftarrow \{V \cup q_{Flyable}\}$
20:        $E \leftarrow \{E \cup \Im_{Flyable}\}$
21:        $R \leftarrow$ Update(R)
22:    endif
23:    $R_c \leftarrow$ Update $(R_c)$
24: endwhile
```

Scenario 2: It is assumed that the configuration space in the second test-case is a cluttered space with two unknown static obstacles.

Scenario 3: It is assumed that the configuration space in the third test-case is a cluttered space with four unknown moving obstacles.

Remark 4. The results are performed on MATLAB software, which is fitted with an ASUS laptop comprising a Core i7 CPU with 16 GB of RAM and a 256 GB SSD.

The following results are summarized from Figures 4-6: In the first scenario position of all obstacles are known for VTOL-Q. However, two different test-case with different complexity (narrow passage and a maze-shaped space) are considered to path planning.

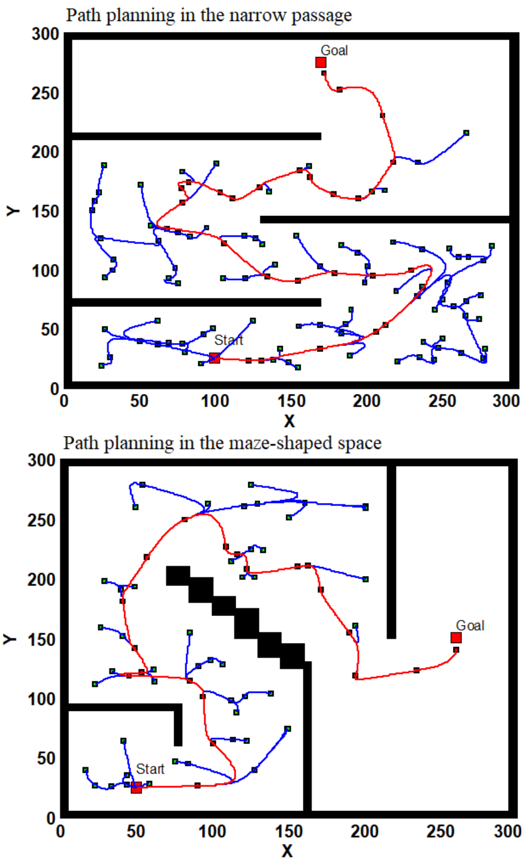


Figure 4. VTOL-Q path planning through the ATRK algorithm in the narrow passage and a maze-shaped space. It is assumed that the all obstacles position are known in the scenario #1

The ATRK path planning algorithm plans a path in the both configuration space by considering kinodynamic constraints of VTOL-Q, (lines 15-16 Algorithm 2).

Hence the planned path is feasible and VTOL-Q tracks the path. In the second scenario, the VTOL-Q detects pre-unknown obstacles during the planned path. In order to avoid collision with them, the collision point (CP) and local goal (LG) are determined. Then the local search tree is expanded from the current location of VTOL-Q to the LG. The VTOL-Q re-plans the path to avoid collision by the first unknown obstacle in the local area. Nevertheless, in the presence of the second unknown obstacle, the VTOL-Q could not reach the first local goal; hence the second local goal is planned. In the third scenario, four moving obstacles are considered which are depicted by a-d in Figure 6. The VTOL-Q predicts collision point by the moving obstacle #b, (lines 12-13 Algorithm 2). Hence re-plans the path in the local area. In a nutshell, path planning through the proposed any-time randomized kinodynamic (ATRK) path planning algorithm in the two types of static known environment with different complexities: 1-narrow passage, and 2-maze-like space is considered in the first scenario and their results are shown in Figure 4 a & b.

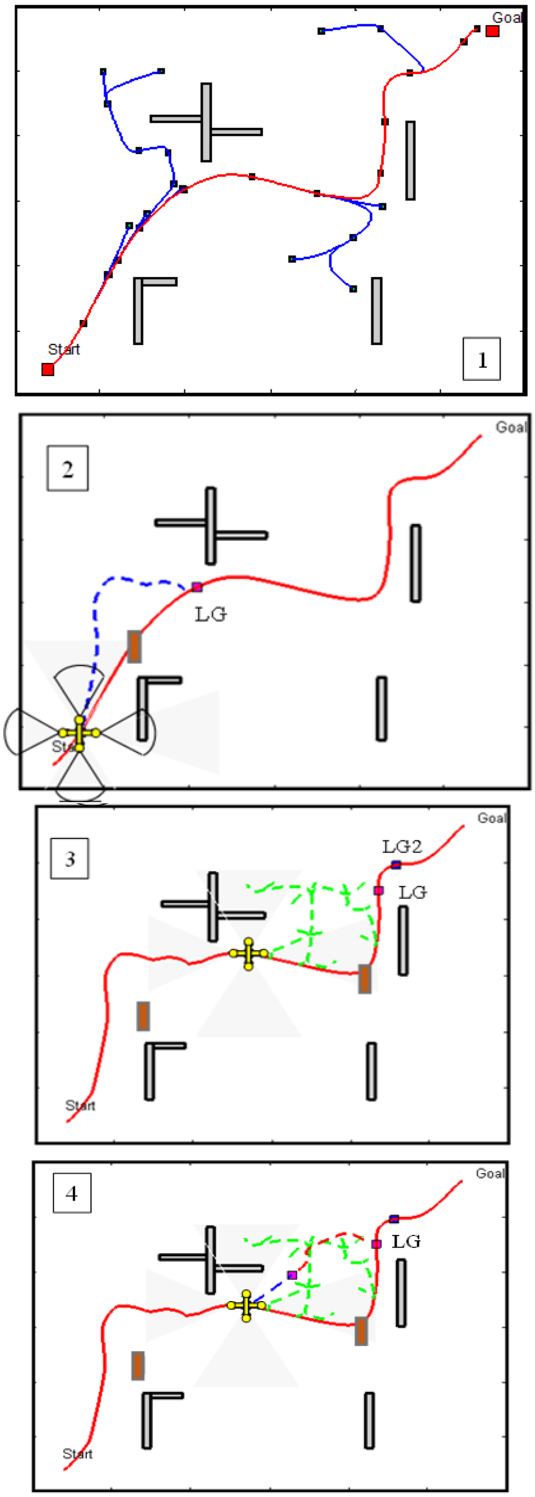
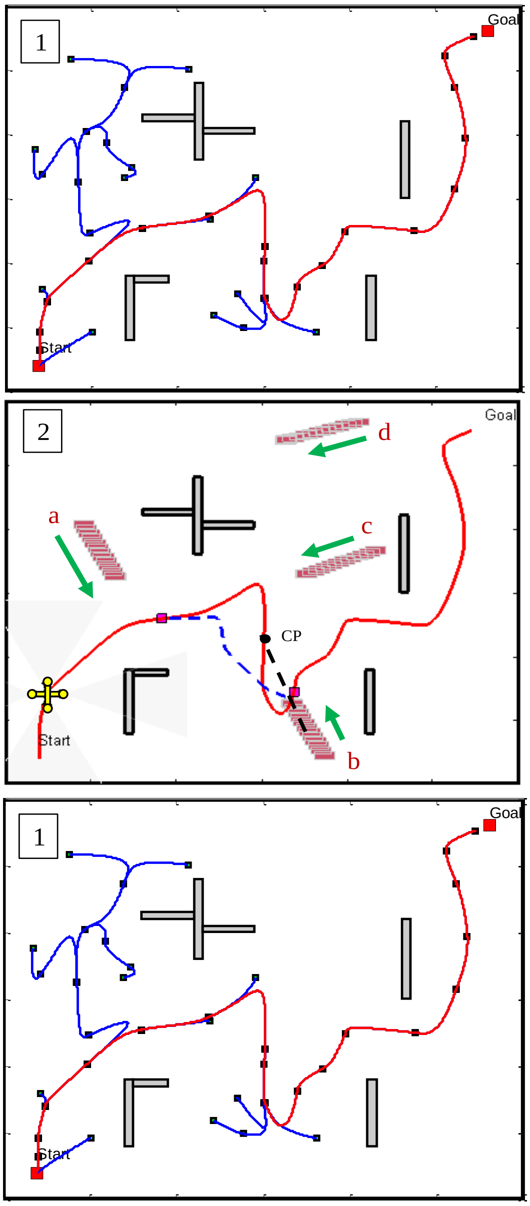


Figure 5. VTOL-Q path planning through the ATRK algorithm in presence pre-unknown static obstacles, (scenario #2)

In both configuration space, the initially planned path is illustrated through the blue line and the optimized path with the triangular inequality is illustrated through the red line. The number of the vertices (waypoints) in the final path (red path) with the triangular inequality is reduced in comparison with the initial path (blue path). Due to the randomized nature of the proposed ATRK the optimized path is sub-optimal. Path planning through the proposed ATRK in the unknown static configuration space is considered in the second scenario. Its related results are shown in Figure 5. The local search tree with green edges is shown in this scenario to avoid collision by the unknown static obstacles. In the third scenario, four unknown moving obstacles are considered in the simulations. Moving obstacles #b has collided with the initially planned path. Hence, the initial path is re-planed in the local area that is shown by the blue edges in Figure 6. Furthermore, the features of proposed ATRK are compared with the standard RRT in Table 2.



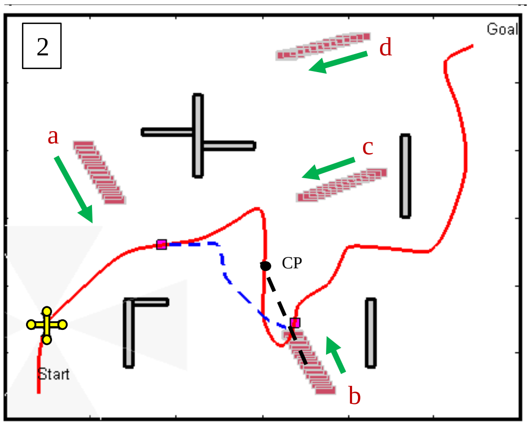


Figure 6. VTOL-Q path planning through the ATRK algorithm in presence pre-unknown moving obstacles, (scenario #3)

TABLE 2. Comparison of the proposed ATRK with the standard RRT

Features	Standard RRT	Proposed ATRK
Execution time and Memory consumption	High	Low
Path optimality	Non-optimal	Sub-optimal
Type of Constraints	Kinematic	Kinodynamic
Training and query phases	Offline-Offline	Offline- Online
Type of Obstacles	Known- Static	Unknown-Moving
Total number of vertices in the search tree	High	Low

5. CONCLUSION

This paper focuses on the study of an any-time kinodynamic path planning framework to plan a feasible path for a VTOL-Q by considering both kinematic and dynamic constraints in the presence of pre-unknown moving obstacles. To this end, let us propose the any-time randomized kinodynamic (ATRK) path-planning algorithm applicable in the VTOL-Q, (Algorithm #2). ATRK path-planning algorithm is based on the RRT and consists of three main components: high-level, mid-level, and low-level controller. The high-level controller utilizes a randomized sampling-based approach to generate offspring vertices for rapid exploring and expanding in the configuration space, (lines 8-11 Algorithm #2). The mid-level controller uses the any-time method to avoid collision with moving obstacles. Low-level controller with a six-DOF dynamic model accounts for the kinodynamic constraints of VTOL-Q in the randomized offspring vertices to plan a feasible path, (lines 15-16 Algorithm #2). In some cases, a discrepancy between the output of the high-level controller and the output of the low-level controller is appeared due to either motion constraints or insufficient time. Simulation results on three different scenarios demonstrate the kinodynamic constraints of the VTOL-Q are integrated into the randomized offspring vertices, hence the proposed ATRK path-planning algorithm is able to plan a feasible path, and the VTOL-Q tracks the planned path appropriately. Also, in presence of moving obstacles, the ATRK re-plans the path in the local areas through the any-time approach. The local search tree in comparison to the initial search tree has a high priority to execute. In a nutshell, the ATRK in all three scenarios is able to integrate the kinodynamic constraints of VTOL-Q in the randomized offspring vertices. Furthermore, the local search trees are expanded in the local areas through the any-time method to avoid collision by moving obstacles.

6. REFERENCES

1. Campana, S., "Drones in archaeology. State-of-the-art and future perspectives", ***Archaeological Prospection***, Vol. 24, No. 4, (2017), 275-296, https://doi.org/10.1002/arp.1569.
2. Yang, K., Keat Gan, S. and Sukkarieh, S., "A gaussian process-based rrt planner for the exploration of an unknown and cluttered environment with a uav", ***Advanced Robotics***, Vol. 27, No. 6, (2013), 431-443, https://doi.org/10.1080/01691864.2013.756386.
3. Pizetta, I.H.B., Brandão, A.S. and Sarcinelli-Filho, M., "Avoiding obstacles in cooperative load transportation", ***ISA Transactions***, Vol. 91, (2019), 253-261, https://doi.org/10.1016/j.isatra.2019.01.019.
4. da Silva, M.F., Honorio, L.M., Marcato, A.L.M., Vidal, V.F. and Santos, M.F., "Unmanned aerial vehicle for transmission line inspection using an extended kalman filter with colored electromagnetic interference", ***ISA Transactions***, Vol. 100, (2020), 322-333, https://doi.org/10.1016/j.isatra.2019.11.007.
5. Shobeiry, P., Xin, M., Hu, X. and Chao, H., "Uav path planning for wildfire tracking using partially observable markov decision process", in AIAA Scitech 2021 Forum. (2021), 1677, https://doi.org/10.2514/6.2021-1677.
6. Mohammed, H., Romdhane, L. and Jaradat, M.A., "Rrt*n: An efficient approach to path planning in 3d for static and dynamic environments", ***Advanced Robotics***, Vol. 35, No. 3-4, (2021), 168-180, https://doi.org/10.1080/01691864.2020.1850349.
7. Floreano, D. and Wood, R.J., "Science, technology and the future of small autonomous drones", ***Nature***, Vol. 521, No. 7553, (2015), 460-466, https://doi.org/10.1038/nature14542.
8. Villasenor, J., "“Drones” and the future of domestic aviation [point of view]", ***Proceedings of the IEEE***, Vol. 102, No. 3, (2014), 235-238, DOI: 10.1109/JPROC.2014.2302875.
9. Taheri, E., Ferdowsi, M.H. and Danesh, M., "Closed-loop randomized kinodynamic path planning for an autonomous underwater vehicle", ***Applied Ocean Research***, Vol. 83, (2019), 48-64, https://doi.org/10.1016/j.apor.2018.12.008.
10. Barraquand, J., Langlois, B. and Latombe, J.-C., "Numerical potential field techniques for robot path planning", ***IEEE Transactions on systems, Man, and Cybernetics***, Vol. 22, No. 2, (1992), 224-241, DOI: 10.1109/21.148426.
11. Cabreira, T., Brisolara, L. and Ferreira Jr, P.R., "Survey on coverage path planning with unmanned aerial vehicles", ***Drones***, Vol. 3, No. 1, (2019), https://doi.org/10.3390/drones3010004.
12. Rubí, B., Pérez, R. & Morcego, B., "A survey of path following control strategies for uavs focused on quadrotors", ***Journal of Intelligent & Robotic Systems***, Vol. 98, No. 2, (2020), 241–265, https://doi.org/10.1007/s10846-019-01085-z.
13. Heidari, H. and Saska, M., "Trajectory planning of quadrotor systems for various objective functions", ***Robotica***, Vol. 39, No. 1, (2021), 137-152, https://doi.org/10.1017/S0263574720000247.
14. Khosravian, E. and Maghsoudi, H., "Design of an intelligent controller for station keeping, attitude control, and path tracking of a quadrotor using recursive neural networks", ***International Journal of Engineering, Transactions B: Applications***, Vol. 32, No. 5, (2019), 747-758. doi: 10.5829/ije.2019.32.05b.17
15. Manouchehri, P., Ghasemi, R. and Toloei, A., "Distributed fuzzy adaptive sliding mode formation for nonlinear multi-quadrotor systems", ***International Journal of Engineering, Transactions B: Applications***, Vol. 33, No. 5, (2020), 798-804, DOI: 10.5829/IJE.2020.33.05B.11.
16. Sangdani, M. and Tavakolpour-Saleh, A., "Particle swarm optimization based parameter identification applied to a target tracker robot with flexible joint", ***International Journal of Engineering, Transactions C: Aspects***, Vol. 33, No. 9, (2020), 1797-1802, DOI: 10.5829/IJE.2020.33.09C.14.
17. Liu, B., Feng, W., Li, T., Hu, C. and Zhang, J., "A variable-step rrt* path planning algorithm for quadrotors in below-canopy", ***IEEE Access***, Vol. 8, (2020), 62980-62989, DOI: 10.1109/ACCESS.2020.2983177.
18. Allen, R.E. and Pavone, M., "A real-time framework for kinodynamic planning in dynamic environments with application to quadrotor obstacle avoidance", ***Robotics and Autonomous Systems***, Vol. 115, (2019), 174-193, https://doi.org/10.1016/j.robot.2018.11.017.
19. Shao, S., Peng, Y., He, C. and Du, Y., "Efficient path planning for uav formation via comprehensively improved particle swarm optimization", ***ISA Transactions***, Vol. 97, (2020), 415-430, https://doi.org/10.1016/j.isatra.2019.08.018.
20. Mashadi, B., Mahmoodi-K, M., Kakaee, A.H. and Hosseini, R., "Vehicle path following control in the presence of driver inputs", ***Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics***, Vol. 227, No. 2, (2013), 115-132, https://doi.org/10.1177/1464419312469755.
21. Mashadi, B., Mahmoudi-Kaleybar, M., Ahmadizadeh, P. and Oveisi, A., "A path-following driver/vehicle model with optimized lateral dynamic controller", ***Latin American Journal of Solids and Structures***, Vol. 11, No. 4, (2014), 613-630, https://doi.org/10.1590/S1679-78252014000400004.
22. Gao, F., Lin, Y. and Shen, S., "Gradient-based online safe trajectory generation for quadrotor flight in complex environments", in 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), IEEE. (2017), 3681-3688, DOI: 10.1109/IROS.2017.8206214.
23. Heidari, H. and Saska, M., "Trajectory planning of quadrotor systems for various objective functions", ***Robotica***, Vol. 39, No. 1, (2020), 137-152, https://doi.org/10.1017/S0263574720000247.
24. Chen, Y.-b., Luo, G.-c., Mei, Y.-s., Yu, J.-q. and Su, X.-l., "Uav path planning using artificial potential field method updated by optimal control theory", ***International Journal of Systems Science***, Vol. 47, No. 6, (2016), 1407-1420, https://doi.org/10.1080/00207721.2014.929191.
25. Frazzoli, E., Dahleh, M.A. and Feron, E., "Real-time motion planning for agile autonomous vehicles", ***Journal of Guidance, Control, and Dynamics***, Vol. 25, No. 1, (2002), 116-129, https://doi.org/10.2514/2.4856.
26. Gong, W., "Probabilistic model based path planning", ***Physica A: Statistical Mechanics and its Applications***, (2021), 125718, https://doi.org/10.1016/j.physa.2020.125718.

27. Le, A.V., Arunmozhi, M., Veerajagadheswar, P., Ku, P.-C., Minh, T.H.Q., Sivanantham, V. and Mohan, R.E., "Complete path planning for a tetris-inspired self-reconfigurable robot by the genetic algorithm of the traveling salesman problem", ***Electronics***, Vol. 7, No. 12, (2018), 344, https://doi.org/10.3390/electronics7120344.

28. Zheng, S. and Liu, H., "Improved multi-agent deep deterministic policy gradient for path planning-based crowd simulation", ***IEEE Access***, Vol. 7, (2019), 147755-147770, DOI: 10.1109/ACCESS.2019.2946659.

29. Kimmel, A., Shome, R. and Bekris, K., "Anytime motion planning for prehensile manipulation in dense clutter", ***Advanced Robotics***, Vol. 33, No. 22, (2019), 1175-1193, https://doi.org/10.1080/01691864.2019.1690207.

30. Wang, W., Xu, X., Li, Y., Song, J. and He, H., "Triple rrts: An effective method for path planning in narrow passages", ***Advanced Robotics***, Vol. 24, No. 7, (2010), 943-962, https://doi.org/10.1163/016918610X496928.

31. Taheri, E., Ferdowsi, M.H. and Danesh, M., "Fuzzy greedy rrt path planning algorithm in a complex configuration space", ***International Journal of Control, Automation and Systems***, Vol. 16, No. 6, (2018), 3026-3035, https://doi.org/10.1007/s12555-018-0037-6.

32. Bouabdallah, S., Noth, A. and Siegwart, R., "Pid vs lq control techniques applied to an indoor micro quadrotor", in 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566). Vol. 3, (2004), 2451-2456 vol.2453, DOI: 10.1109/IROS.2004.1389776.

33. Bresciani, T., "Modelling, identification and control of a quadrotor helicopter", ***M. Sc. Thesis***, Vol., No. 0280-5316, (2008), http://lup.lub.lu.se/student-papers/record/8847641.

34. Sabatino, F., "Quadrotor control: Modeling, nonlinearcontrol design, and simulation", KTH, School of Electrical Engineering (EES), Automatic Control., (2015), http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-175380.

35. Das, A., Lewis, F. and Subbarao, K., "Dynamic inversion with zero-dynamics stabilisation for quadrotor control", ***IET Control Theory & Applications***, Vol. 3, No. 3, (2009), 303-314, DOI: 10.1049/iet-cta:20080002.

36. Kuffner, J.J.a.L., Steven M, "Rrt-connect: An efficient approach to single-query path planning", Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065), Vol. 2, (2000), 995-1001, DOI: 10.1109/ROBOT.2000.844730.

37. Wang, J., Chi, W., Li, C., Wang, C. and Meng, M.Q.H., "Neural rrt*: Learning-based optimal path planning", ***IEEE Transactions on Automation Science and Engineering***, Vol. 17, No. 4, (2020), 1748-1758, DOI: 10.1109/TASE.2020.2976560.

38. Richter, C., Bry, A. and Roy, N., Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments, in ***Robotics Research***, 2016.649-666, https://doi.org/10.1007/978-3-319-28872-7_37.

Persian Abstract

چکیده

طراحی مسیر سینودینامیکی برای ربات های بدون سرنشین خودمختار به عنوان یک چالش باز تحقیقاتی و یک مسئله NP_Hard شناخته می‌شود. هدف از این تحقیق طراحی یک مسیر قابل اجرا برای یک ربات عمود پرواز از موقعیت و سرعت ابتدایی به موقعیت و سرعت نهایی با درنظر گرفتن همزمان قیدهای سینماتیکی همچون موانع متحرک و قیدهای دینامیکی همچون محدودیت‌های حرکت در یک فضای کاری سه بعدی می‌باشد. برای این منظور الگوریتم طراحی مسیر تصادفی سینودینامیکی هر لحظه زمان، برای ربات مد نظر در این مقاله پیشنهاد می‌گردد. الگوریتم طراحی مسیر پیشنهادی از خانواده الگوریتم‌های طراحی مسیر درخت جستجویی سریع تصادفی می‌بوده و از سه زیر سیستم اصلی تشکیل می‌گردد. سه زیرسیتم اصلی به ترتیب عبارتند از: کنترل‌کننده سطح بالا، کنترل کننده میانی و کنترل کننده سطح پایین. کنترل کننده سطح بالا وظیفه نمونه برداری تصادفی به منظور تولید گره‌های فرزند تصادفی در فضای کاری مد نظر و هدایت و گسترش درخت جستجو را برعهده دارد. کنترل کننده میانی به کمک الگوریتم هر لحظه وظیفه رشد درخت‌های جستجوی محلی جهت اجتناب از برخورد با موانع از پیش ناشناخته متحرک را برعهده دارد. همچنین وظیفه حرص شاخه‌های نامتناسب درخت جستجو و طراحی یک مسیر زیربهینه نیز با این واحد می‌باشد. کنترل کننده سطح پایین به کمک مدل دینامیکی شش درجه آزادی ربات وظیفه اعمال قیدهای حرکتی ربات در فرآیند رشد شاخه‌ها و گره‌های درخت جستجو را بر عهده خواهد داشت. بدین ترتیب درخت مشتمل بر شاخه‌ها و گره‌هایی خواهد شد که پیشتر قیدهای حرکتی در انها لحاظ شده و از قابل رهگیری بودن انها توسط ربات اطمینان حاصل شده است. الگوریتم طراحی مسیر پیشنهادی در طی سه سناریو تست متفاوت ارزیابی می‌گردد. نتایج نشان می‌دهد که الگوریتم پیشنهادی نه تنها قادر به طراحی مسیر با قابلیت رهگیری توسط ربات می‌باشد بلکه حجم محاسبات نیز به خاطر عملکرد تصادفی روش کاهش چشمگیری داشته و از برخورد با موانع متحرک نیز به کمک رشد درخت‌های جستجوی محلی اجتناب شده است.