A CONTROL PARAMETRIZATION BASED PATH PLANNING METHOD FOR THE QUAD-ROTOR UAVS

JINLONG GUO1, BIN LI2 AND YUANDONG JI2,*

1College of Electrical Engineering
Sichuan University, Sichuan 610065, China
2School of Aeronautics and Astronautics
Sichuan University, Sichuan 610065, China

(Communicated by Kok Lay Teo)

Abstract. A time optimal path planning problem for the Quad-rotor unmanned aerial vehicles (UAVs) is investigated in this paper. A 3D environment with obstacles is considered, which makes the problem more challenging. To tackle this challenge, the problem is formulated as a nonlinear optimal control problem with continuous state inequality constraints and terminal equality constraints. A control parametrization based method is proposed. Particularly, the constraint transcription method together with a local smoothing technique is utilized to handle the continuous inequality constraints. The original problem is then transformed into a nonlinear program. The corresponding gradient formulas for both of the cost function and the constraints are derived, respectively. Simulation results show that the proposed path planning method has less tracking error than that of the rapid-exploring random tree (RRT) algorithm and that of the A* star algorithm. In addition, the motor speed has less changes for the proposed algorithm than that of the other two algorithms.

1. Introduction. Unmanned aerial vehicles (UAVs), which are able to vertically take-off and land, are widely used as the platforms for both civil and military applications. As the UAV missions are becoming more complicated, path planning for the UAVs has received tremendous attention.

1.1. Literature review. The objective of 3D path planning for the UAVs is to find an optimal and collision free path in a 3D environment by considering the geometric constrains from the environment and the physical constraints from the UAVs. The existing path planning algorithms can be generally classified into the online algorithms and the off-line algorithms[11].

Generally, the computational complexity of the online algorithms is low. For example, the sampling-based algorithms [9, 14, 15] and the node-based algorithms [17, 5, 2]. The idea of the sampling-based algorithms is to sample the environment...
and then find an optimal path by random searching the sampled environment or generate a road net map of the sampled environment. Rapidly-exploring Random Tree (RRT) algorithm [9] is one of the classical sampling-based algorithms. The node-based algorithms are, in fact, are developed based on the principle of the dynamic programing. For example, Dijkstra’s algorithm and A star [17] algorithm. This type of algorithms requires the node information on the map and hence it is usually used with the sampling-based algorithms.

Off-line algorithms has also been intensively studied. The mathematical model based algorithms [13, 3, 1] and the bio-inspired algorithms [4, 10, 6] are such type of algorithms. The computational complexity for the off-line algorithms is normally higher than that of the online algorithms. However, a better performance can be achieved. Mathematical model based algorithms formulate the path planning problem into mathematical programs, such as mixed-integer linear program [13], binary linear program [3] and convex program [1]. The same problem can also be solved by using the bio-inspired algorithms. For example, evolutionary algorithm [4] and neural network [10] and particle swarm optimization [6].

1.2. Contributions. Different from most of the exiting works in the literature, an off-line path planning problem is formulated as an optimal control problem in this paper. The advantage of this formulation is that the dynamics of the UAV is considered. Therefore, the planned path can be tracked with less tracking error in real world applications.

More specifically, the 3D path planning problem is formulated as a time optimal control problem subject to continuous state inequality constraints and terminal equality constraints. Particularly, obstacle avoidance is also considered in this problem, and two different types of 3D obstacles are investigated. The shape of the first type of obstacles is similar to the urban building while that of the second type is more like a mountain in the rural areas. Hence, many real world working conditions can be covered. Both types of the obstacles are formulated as continuous state inequality constraints. In addition, some physical limits of the UAV can be modeled into such constraints as well. For example, the limits on the position, the velocity and certain angles.

Continuous state inequality constraints [16, 12, 20], which are also referred to as path constraints, are difficult to handle. This is because there are infinite number of constraints on the time horizon. Such problems are called semi-infinite program in operations research. In this paper, these constraints are tackled by utilizing a constraint transcription technique [16] together with a local smoothing technique. Thereby, each the continuous state inequality constraint can be transformed into a canonical constraint.

Further, by applying the control parametrization technique [16, 12, 20, 19], the original time optimal control problem is transformed into an optimal parameters selection problem subject to canonical constraints [12, 20, 19]. The transformed problem can be treated as a nonlinear program, and there are many existing computational method for solving such problems. For example, the sequential quadratic program (SQP). However, gradients information are required since gradient-based methods are used. Thus, the corresponding gradient formulas for the objective function and the constraints are derived, respectively. Some off-the-self control parametrization based optimal control software packages are available. For example, VISUAL MISER [18] and MISER 3.2 [8]. To demonstrate the effectiveness of
the proposed algorithm, 4 numerical examples are carried out by considering different working environments. Particularly, the paths generated by the proposed method, the RRT algorithm and the A star algorithm are tracked with a closed loop control system. Simulation results show that the proposed algorithm yield less tracking error and the path generated is more smooth than the RRT algorithm and the A star algorithm. The speed of the motor by using the proposed method has also shown to have less changes than that of the other two algorithms. The contributions of this paper are summarized as follows: (1) A control parametrization based method is developed for the path planning problem, and the dynamics of the UAV is considered; (2) Two general types of obstacles are modeled into continuous inequality constraints which can cover many scenarios in both urban and rural applications; (3) The corresponding gradient formulas are derived for the proposed method; (4) The proposed method has shown to have less tracking error that of the RRT algorithm and that of the A star algorithm by being tested in a closed loop control system. In addition, there are less changes on motor speed trajectories by using the proposed method than that of the other two algorithms.

1.3. Organization. The rest of this paper is organized as follow: The problem is stated in Section 2; A control parametrization method is developed in Section 3; 4 numerical experiments are conducted in Section 4; Finally, Section 5 concludes this paper.

2. Problem statement.

2.1. The dynamic model of the UAV. A quad-rotor UAV is considered in this paper. Here, we choose the Earth-frame and the fixed-body frame to investigate the dynamics of the UAV, which are illustrated in Fig. 1. Thus, the dynamic model
of the UAV can be described by the following nonlinear differential equations [21]:

\[
\begin{align*}
\ddot{x} &= u_1 (\cos(\varphi) \sin(\theta) \cos(\psi) + \sin(\varphi) \sin(\psi)) - \frac{K_1 \dot{x}}{m} \\
\ddot{y} &= u_1 (\sin(\varphi) \sin(\theta) \cos(\psi) - \cos(\varphi) \sin(\psi)) - \frac{K_2 \dot{y}}{m} \\
\ddot{z} &= u_1 \cos(\varphi) \cos(\psi) - g - \frac{K_3 \dot{z}}{m} \\
\dot{\theta} &= u_2 - \frac{L K_4 \dot{\theta}}{I_y} \\
\dot{\psi} &= u_3 - \frac{L K_5 \dot{\psi}}{I_x} \\
\dot{\phi} &= u_4 - \frac{L K_6 \dot{\phi}}{I_z}
\end{align*}
\]

(1)

where \(x, y, \) and \(z\) are inertial horizontal, lateral, and vertical positions in the Earth-frame, respectively; \(\theta, \psi, \phi\) denote the angle of pitch, angle of roll and the angle of yaw, respectively; \(g\) is the acceleration of gravity; \(L\) represents the arm length from motor to mass center; \(m\) is the mass of the UAV and \(I_i, i = x, y, z\) are the moments of inertia; \(K_i, i = 1, 2, \cdots, 6\) are the air resistance coefficients; \(u_i, i = 1, 2, 3, 4\) are the virtual control inputs which are defined as follows:

\[
\begin{align*}
u_1 &= \frac{F_1 + F_2 + F_3 + F_4}{m} \\
u_2 &= \frac{L (F_2 - F_4)}{I_y} \\
u_3 &= \frac{L (F_3 - F_1)}{I_x} \\
u_4 &= \frac{C (F_1 - F_2 + F_3 - F_4)}{I_z}
\end{align*}
\]

(2)

\[F_i = K_v \omega_i^2, \ i = 1, 2, 3, 4\]  

(3)

where \(F_i\) is the thrust generated by the \(i\)th rotor, \(C\) is the thrust-to-moment scaling factor, \(\omega_i (i = 1, 2, 3, 4)\) is the rotation angular velocity of the \(i\)th rotor, and \(K_v\) is the propeller thrust coefficient.

To derive the state-space model, the state vector and the control vector are chosen as

\[
x = \begin{bmatrix} \theta & \psi & \varphi & \dot{\theta} & \dot{\psi} & \dot{\phi} & x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T
\]

(4)

and

\[
u = \begin{bmatrix} u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T
\]

(5)

respectively, where \([\cdot]^T\) denotes the transpose. Considering (4) and (5), (2.1) can be rewritten as:
A PATH PLANNING METHOD FOR THE QUAD-ROTOR UAVS

\[
\begin{align*}
\dot{x}_1(t) &= x_4(t) \\
\dot{x}_2(t) &= x_5(t) \\
\dot{x}_3(t) &= x_6(t) \\
\dot{x}_4(t) &= u_2(t) - \frac{L K_1 x_4(t)}{I_y} \\
\dot{x}_5(t) &= u_3(t) - \frac{L K_2 x_5(t)}{I_z} \\
\dot{x}_6(t) &= u_4(t) - \frac{L K_3 x_6(t)}{I_z} \\
\dot{x}_7(t) &= x_{10}(t) \\
\dot{x}_8(t) &= x_{11}(t) \\
\dot{x}_9(t) &= x_{12}(t) \\
\dot{x}_{10}(t) &= u_1(t) \left(\cos(x_3(t))\sin(x_1(t))\cos(x_2(t)) \right. \\
&\quad \left. + \sin(x_2(t))\sin(x_3(t)) - \frac{K_4 x_{10}(t)}{M}\right) \\
\dot{x}_{11}(t) &= u_1(t) \left(\sin x_3(t)\sin(x_1(t))\cos(x_2(t)) \right. \right. \\
&\quad \left. - \cos(x_3(t))\sin(x_2(t)) - \frac{K_5 x_{11}(t)}{M}\right) \\
\dot{x}_{12}(t) &= u_1(t) \cos(x_3(t))\cos(x_2(t)) - g - \frac{K_6 x_{12}(t)}{M} \\
\end{align*}
\]

For the notation simplicity, (6) is denoted as
\[
\dot{x}(t) = f(x(t), u(t))
\]

2.2. Modelling of the obstacles. Two classical types of obstacles are considered in this paper. The first type is called \( p \) norm obstacle [7] and the second type of obstacles is in a mountainous shape. The obstacle avoidance is formulated as continuous state inequality constraints. More specifically, the UAV will not clash with the obstacles as long as these constraints are satisfied. For the first type of obstacles, the corresponding continuous state inequality constraint is given as
\[
\left( \frac{x(t) - x_c}{a} \right)^{p_x} + \left( \frac{y(t) - y_c}{b} \right)^{p_y} + \left( \frac{z(t) - z_c}{c} \right)^{p_z} \geq 1
\]
where \((x_c, y_c, z_c)\) is the center of the obstacle, \(a, b,\) and \(c\) denote the radii of an obstacle in the directions of \(x, y,\) and \(z,\) respectively, and \(p_x, p_y,\) and \(p_z\) are even integers which determine the shape of the obstacle. For example, it is an ellipsoid if \(p_x = p_y = p_z = 2\). The obstacle is in the box-like shape if \(p_x = p_y = p_z \geq 6\).

The continuous state inequality constraint for the second type of obstacles is defined as
\[
z(t) - \sum_{i=1}^{n} h_i \exp \left[ \left( \frac{x(t) - x_i}{x_{si}} \right)^2 + \left( \frac{y(t) - y_i}{y_{si}} \right)^2 \right] \geq 0
\]
where \((x_i, y_i)\) is the central coordinate of the \(i\)th mountain, \(h_i\) is the topographic parameter used to represent the height of the mountain, \(x_{si}\) and \(y_{si}\) are the attenuation of the \(i\)th mountain along the \(x\)-axis and \(y\)-axis, respectively, and \(n\) denotes the total number of mountains.
Remark 1. In fact, these two types of obstacles are quite general. Many other shapes of obstacles can be covered by these types. The first type of obstacles can be treated as the buildings in urban areas, while shape of the second type of obstacles is similar to the mountains in rural areas.

2.3. Constraints. Besides the obstacles avoidance, there are other factors to be taken into account. For example, the limits of the velocity and the control inputs. In this section, these factors are modeled as three different types of constraints.

(I) Terminal constraints

\[ x(T_f) = x_f \]  

where \( T_f \) is the terminal time. To reach the destination, terminal equality constraint (10) has to be imposed.

(II) Continuous state inequality constraints

The altitude constraint of the UAV can be formulated as

\[ 0 \leq z(t) \leq z_{\text{max}} \]  

To limit the velocity of the UAV, the following constraints are imposed.

\[ |\dot{x}(t)| \leq V^x_{\text{max}}, \quad |\dot{y}(t)| \leq V^y_{\text{max}}, \quad |\dot{z}(t)| \leq V^z_{\text{max}} \]  

Consider the limits of the angles, we impose the following constraints:

\[ \theta_{\text{min}} \leq \theta(t) \leq \theta_{\text{max}}, \quad \psi_{\text{min}} \leq \psi(t) \leq \psi_{\text{max}}, \quad \varphi_{\text{min}} \leq \varphi(t) \leq \varphi_{\text{max}} \]  

(III) Control constraints

The physical limit of the motors can be formulated as

\[ u_{\text{min}} \leq |u_i(t)| \leq u_{\text{max}}, \quad i = 1, 2, 3, 4 \]  

2.4. Optimal control problem. The path planning problem can be formulated as the following time optimal control problem. We refer to it as Problem \( P \). Given the dynamical system (6) and an initial condition \( x_0 \), choose a control input \( u(t) \) such that \( T_f \) is minimized and the continuous state inequality constraints (8)/(9) and (11) - (13), the terminal state constraint (10), and control constraints (14) are satisfied.

3. A control parametrization based method. Problem \( P \) is an optimal control problem with continuous state inequality constraints. Particularly, the continuous state inequality constraints, which is also referred as path constraints, are difficult to be satisfied. This is because there is a constraint at each time point. Thus, there are infinite number of constraints to be satisfied. To tackle this challenge, we first use the control parametrization technique [16, 12, 20, 19] to transform optimal control problem into an optimal parameter selection problem. Then, a constraint transcription method together with a local smoothing technique is introduced to convert the continuous state inequality constraints into the constraints in the canonical form [12, 20]. The converted problem is, in fact, a nonlinear program which can be solved by many standard computational methods. For example, the sequential quadratic program (SQP) method.
3.1. Control parametrization. To proceed, we partition the time horizon \([0, T_f]\) into \(p\) equal sub-intervals with the following \(p + 1\) knots:

\[
0 = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_{p-1} < \tau_p = T_f
\]  

(15)

As shown in Fig. 2, for each \(i\), the control variable \(u_i(t)\) is approximated by the following piece-wise constant function:

\[
u_p^i(t) = \sum_{k=1}^{p} \sigma_k^i \chi_{[\tau_{k-1}, \tau_k)}(t), \quad i = 1, 2, 3, 4
\]

(16)

where \(\tau_k = kT_f/p, \quad k = 0, 1, \cdots, p,\) and

\[
\chi_I(s) = \begin{cases} 
1, & \text{if } t \in I, \\
0, & \text{others}.
\end{cases}
\]

(17)

Then, we define

\[
\sigma = [\sigma_1^T, \sigma_2^T, \sigma_3^T, \sigma_4^T]^T \in \mathbb{R}^{4p}
\]

where, for each \(i\),

\[
\sigma_i = [\sigma_1^i, \sigma_2^i, \cdots, \sigma_p^i]^T \in \mathbb{R}^p, \quad i = 1, 2, 3, 4
\]

In the rest of the paper, we shall use \(\sigma\) to replace \(u(t)\). Thus, (7) becomes

\[
\dot{x}(t) = f(x(t), \sigma)
\]

(18)

In a similar manner, the control constraints (14) can be rewritten as

\[
u_{\min} \leq |\sigma_i^p| \leq u_{\max}, \quad i = 1, 2, 3, 4
\]

(19)

Another challenge is that the terminal time of Problem \(P\) is free. To tackle this challenge, we apply a time scaling transform to map the original time horizon \([0, T_f]\) to a fixed time horizon \([0, 1]\). This can be achieved by the linear transform in (20).

\[
\frac{dt}{ds} = T_f
\]

(20)

By applying chain rule to (18) and considering (20), it follows that

\[
\dot{x}(s) = \frac{dx}{ds} = \frac{dx}{dt} \cdot \frac{dt}{ds} = f(x(s), \sigma) \cdot T_f
\]

(21)

The terminal constraint (10) becomes

\[
x(1) = x_f
\]

(22)

Now Problem \(P\) can be restated as follows. We refer to this new problem as Problem \(P(p)\). Given the dynamical system (21) and an initial condition \(x_0\), choose a vector
\( \sigma \) and a parameter \( T_f \) such that \( T_f \) is minimized and the continuous state inequality constraints (8)/(9) and (11) - (13), the terminal state constraint (22), and (19) are satisfied. Problem \( P(p) \) is an optimal parameter problem subject to continuous state inequality constraints. The problem is still challenging due to the continuous state inequality constraints. In the next section, a constraint transcription method together with local smoothing technique will be utilized to handle these constraints.

**Remark 2.** Problem \( P(p) \) is, in fact, a sub-optimal solution of Problem \( P \). The optimal solution of Problem \( P(p) \) converges to the optimal solution of Problem \( P \) as \( p \to +\infty \), which has been proven in Theorem 8.5.1 of [16].

3.2. **Constraint approximation.** In this section, we shall use the constrain transcription method together with a local smoothing technique \([12, 20]\), to transform the continuous inequality constraints (8)/(9) and (11) - (13) into the canonical form. For convenience, the continuous state inequality constraints (8)/(9) and (11) - (13) are denoted as the following general form:

\[
g_i(x(t)) \geq 0, \ i = 1, 2, \cdots, 15 \tag{23}
\]

The idea of the constraint transcription technique is to convert the infinite number of constraints into a finite number of constraints, and this can be achieved by introducing the following integral equality constraint:

\[
\int_0^{T_f} \min \{g_i(x(t)), 0\} \, dt = 0, \ i = 1, 2, \cdots, 15 \tag{24}
\]

Obviously, (24) is equivalent to (23). More importantly, it is not continuous state inequality constraints.

However, (24) is non-differentiable, which implies that we cannot derive the gradients for (24). To smooth (24), for each \( i \), we replace (24) with a differentiable function:

\[
L_{i,\varepsilon}(g_i(x(t))) = \begin{cases} 
0, & g_i(x(t)) > \varepsilon \\
-(g_i(x(t)) - \varepsilon)^2/4\varepsilon, & -\varepsilon \leq g_i(x(t)) \leq \varepsilon \\
g_i(x(t)), & g_i(x(t)) < -\varepsilon 
\end{cases} \tag{25}
\]

as illustrated in Fig. 3, where \( \varepsilon > 0 \). Clearly, (25) is a smooth function.

Nevertheless, (25) is an approximation to (24) and this approximation is more accurate as \( \varepsilon \to 0 \). Another difficulty caused by this approximation is that (25) fail to satisfy the constraint qualification.

To overcome these difficulties, for each \( i \), a new constraint

\[
g_{i,\varepsilon,\gamma} = \gamma + \int_0^{T_f} L_{i,\varepsilon}(g_i(x(t))) \, dt \geq 0 \tag{26}
\]

is used to approximate (24) rather than (25). Here, \( \gamma > 0 \). After introducing \( \gamma \), (26) satisfies the constraint qualification. In addition, \( \gamma \) can also compensate the approximation error introduced from (25) as shown in Fig. 3.

By replacing (8)/(9) and (11) - (13) with (26) in Problem \( P(p) \), an optimal parameter selection problem subject to canonical constraints is obtained, which is referred to as Problem \( P_{\varepsilon,\gamma}(p) \).

In fact, Problem \( P_{\varepsilon,\gamma}(p) \) can be treated as a nonlinear program according to [16], and it is able to be solved by standard computational methods, such as SQP method. Furthermore, there are off-the-shelf optimal control software packages available for solving such problems. For example, VISUAL MISER [18] and MISER 3.2 [8].
A PATH PLANNING METHOD FOR THE QUAD-ROTOR UAVS

-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

min(gi, 0)
ε = 0.1
ε = 0.2
ε = 0.5

Figure 3. An Illustration of the Local Smoothing Technique

Table 1. Algorithm 1: an iteration algorithm for solving Problem P(p).

| Step | Description |
|------|-------------|
| 1.   | Solve Problem $P_{\varepsilon, \gamma}(p)$ for the optimal solution $K_{\varepsilon, \gamma}^*$. |
| 2.   | For each $i$, check the feasibility of $g_i(x(t)) \geq 0$ with $K_{\varepsilon, \gamma}^*$. |
| 3.   | If all the constraints in Step 2 are satisfied, then go to the Step 5. Otherwise, go to the Step 4. |
| 4.   | Set $\gamma = \gamma/2$ and go to Step 1. |
| 5.   | Set $\varepsilon = \varepsilon/10$, $\gamma = \gamma/10$, and go to Step 1. |
|      | Stopping criterion: Algorithm 1 stops when $\varepsilon \leq \varepsilon_{\text{min}}$. |

Since Problem $P_{\varepsilon, \gamma}(p)$ is an approximation of Problem $P(p)$, $\varepsilon$ and $\gamma$ have to be adjusted to find an optimal solution. This procedure is summarized in Table 1.

Algorithm 1 can be regarded as an algorithm of providing an optimal solution for Problem $P(p)$ by solving a number of Problem $P_{\varepsilon, \gamma}(p)$ with different $\varepsilon$ and $\gamma$. $\gamma$ is used for ensuring the feasibility and $\varepsilon$ is utilized to guarantee the accuracy. More specifically, there are two loops in Algorithm 1. In the inner loop, for each given $\varepsilon$, a series of Problem $P_{\varepsilon, \gamma}(p)$ is solved by decreasing $\gamma$ until a feasible solution is found. In the outer loop, $\varepsilon$ is decreased until it is ‘small’ enough, which implies an accurate solution is obtained for Problem $P(p)$.

Remark 3. As $\varepsilon \to 0$ the solution generated by Algorithm 1 converges to the optimal solution of Problem $P(p)$, which has been proven in Theorem 8.4.1 in [16].

3.3. Gradient formulas. In this section, we shall derive the gradient formulas for the cost function and the constraints of Problem $P_{\varepsilon, \gamma}(p)$. This is because gradient-based algorithms are adopted for solving the nonlinear program.
Theorem 3.1. The gradient formulas of the cost function $J = T_f$ with respect to $\sigma$ are

$$\frac{\partial J}{\partial \sigma} = T_f \int_0^1 \left[ \frac{\partial f(x(s), \sigma)}{\partial \sigma} \right]^T \lambda_0(s) ds$$  \hspace{1cm} (27)

where

$$\frac{\partial f(x(s), \sigma)}{\partial \sigma} = \begin{bmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\cos x_3 \sin x_1 \cos x_2 + \sin x_2 \sin x_3 & 0 & 0 & 0 \\
\sin x_3 \sin x_1 \cos x_2 - \sin x_2 \cos x_3 & 0 & 0 & 0 \\
\cos x_2 \cos x_3 & 0 & 0 & 0 \\
\end{bmatrix}$$ \hspace{1cm} (28)

$$\frac{\partial J}{\partial T_f} = 1 + \int_0^1 \left[ \lambda_0^T(s)f(x(s), \sigma) \right] ds$$ \hspace{1cm} (29)

where $\lambda_0$ is the solution of the following co-state equation

$$\dot{\lambda}_0(s) = -T_f \left[ \frac{\partial f(x(s), \sigma)}{\partial x} \right]^T \lambda_0(s)$$ \hspace{1cm} (30)

where

$$\frac{\partial f(x(s), \sigma)}{\partial x} = \begin{bmatrix}
0 & E & 0 & 0 \\
0 & M_1 & 0 & 0 \\
0 & 0 & 0 & E \\
M_2 & 0 & 0 & M_3 \\
\end{bmatrix}$$ \hspace{1cm} (31)

Dimension of the matrix above is $12 \times 12$. $0$ is a null matrix, $E$ is an identity matrix and they are all matrices of $3 \times 3$. $M_1, M_2, M_3$ are as follows.

$$M_1 = \begin{bmatrix}
-\frac{L_1 K_1}{I_y} & 0 & 0 & 0 \\
0 & -\frac{L_2 K_2}{I_x} & 0 & 0 \\
0 & 0 & -\frac{L_3 K_3}{I_z} & 0 \\
\end{bmatrix}$$ \hspace{1cm} (32)

$$M_2 = \begin{bmatrix}
\sigma_1^T \cos x_1 \cos x_2 \cos x_3 & \sigma_1^T (\cos x_2 \sin x_3 - \sin x_1 \sin x_2 \cos x_3) \\
\sigma_1^T \cos x_1 \cos x_2 \sin x_3 & -\sigma_1^T (\sin x_1 \sin x_2 \sin x_3 + \cos x_2 \cos x_3) \\
0 & -\sigma_1^T \cos x_3 \sin x_2 \\
-\sigma_1^T (\sin x_2 \cos x_3 - \sin x_1 \cos x_2 \sin x_3) & \sigma_1^T (\sin x_1 \cos x_2 \cos x_3 + \sin x_2 \sin x_3) & -\sigma_1^T \sin x_3 \cos x_2 \\
\end{bmatrix}$$ \hspace{1cm} (33)

$$M_3 = \begin{bmatrix}
-\frac{K_4}{M} & 0 & 0 & 0 \\
0 & -\frac{K_5}{M} & 0 & 0 \\
0 & 0 & -\frac{K_6}{M} & 0 \\
\end{bmatrix}$$ \hspace{1cm} (34)

with terminal condition $\lambda_0(1) = 0$ \hspace{1cm} (35)
Proof. See Appendix A.

**Theorem 3.2.** The gradient formulas for the terminal equality constraint $h = x(1) - x_f = 0$ are

$$\frac{\partial h}{\partial \sigma} = T_f \int_0^1 \left[ \frac{\partial f(x(s), \sigma)}{\partial \sigma} \right]^T \lambda(s) ds$$

$$\frac{\partial h}{\partial T_f} = \int_0^1 \left[ \lambda^T(s) f(x(s), \sigma) \right] ds$$

where $\lambda(s)$ is the solution of the following co-state equations

$$\dot{\lambda}(s) = -T_f \left[ \frac{\partial f(x(s), \sigma)}{\partial x} \right]^T \lambda(s)$$

with terminal condition

$$\lambda(1) = \frac{\partial h(x(1))}{\partial x}$$

The proofs of Theorem 3.2 and Theorem 3.3 are omitted since they can be proved in a similar manner as Theorem 3.1.

**Theorem 3.3.** The gradient formulas for the continuous state inequality constraint function $g_{i, \varepsilon, \gamma}$, $i = 1, 2, \ldots, 15$, are:

$$\frac{\partial g_{i, \varepsilon, \gamma}}{\partial \sigma} = T_f \int_0^1 \left[ \frac{\partial f(x(s), \sigma)}{\partial \sigma} \right]^T \tilde{\lambda}_i(s) ds$$

$$\frac{\partial g_{i, \varepsilon, \gamma}}{\partial T_f} = \int_0^1 \left[ L_i(x(s)) + \tilde{\lambda}_i^T(s) f(x(s), \sigma) \right] ds$$

where $\tilde{\lambda}_i(s)$ is the solution of the following co-state equation

$$\dot{\tilde{\lambda}}_i(s) = -T_f \left\{ \frac{\partial L_i(x(s))}{\partial x} + \left[ \frac{\partial f(x(s), \sigma)}{\partial x} \right]^T \tilde{\lambda}_i(s) \right\}$$

with terminal condition

$$\tilde{\lambda}_i(1) = 0$$

4. **Numerical experiments.** To demonstrate the effectiveness of the proposed algorithm, 4 different numerical experiments are conducted in this section. Particularly, different types of obstacles are considered in these examples. In addition, the path generated by the proposed algorithm and that by the RRT algorithm are used as reference inputs in a closed-loop control system in order to test whether or not they are able to be tracked in real world applications.

We set $p = 20$, $\varepsilon_0 = 0.1$, $z_{\text{max}} = 5000$ m, $V_{x_{\text{max}}} = 25$, $V_{y_{\text{max}}} = 25$, $V_{z_{\text{max}}} = 20$, $\theta_{\text{min}} = -\frac{2\pi}{3}$, $\theta_{\text{max}} = \frac{2\pi}{3}$, $\psi_{\text{min}} = -\frac{\pi}{3}$, $\psi_{\text{max}} = \frac{\pi}{3}$, $\phi_{\text{min}} = -1.8\pi$, $\phi_{\text{max}} = 1.8\pi$, $u_{\text{min}} = [10, -0.005, -0.005, -0.005]^T$, $u_{\text{max}} = [12, 0.005, 0.005, 0.005]^T$. The parameters for the quad-rotor UAV are given in Table 2. MISER 3.2 [8] within the Matlab environment is used for solving Problem $P_{z, \gamma}(p)$. 


Figure 4. Example 1: the planned path of the UAV with Algorithm 1

Figure 5. Example 1: the optimal states
A PATH PLANNING METHOD FOR THE QUAD-ROTOR UAVS

Figure 6. Example 1: the optimal control inputs

(a) The 3D path
(b) The 2D path

Figure 7. Example 2: the planned path of the UAV with Algorithm 1

Table 2. Parameters of the UAV.

| Parameter | Value |
|-----------|-------|
| $L$       | 0.2 m  |
| $I_x$     | 0.0075 kg $\cdot$ m$^2$ |
| $K_1, K_2$| 0.06 N/m/s |
| $K_6$     | 0.1 N/m/s |
| $M$       | 1.5 kg |
| $I_y$     | 0.0075 kg $\cdot$ m$^2$ |
| $K_3$     | 0.09 N/m/s |
| $I_z$     | 0.013 kg $\cdot$ m$^2$ |
| $K_4, K_5$| 0.002 N/m/s |
| $g$       | 9.8 m/s$^2$ |
| $K_v$     | $1.5 \times 10^{-5}$ N/m/s |
| $C$       | $10^{-7}$ |
4.1. Example 1: An obstacle-free environment. In this example, Algorithm 1 is implemented in an obstacle-free environment. The coordinate of the target is set as \((500, 500, 500)\), and the initial condition is set as \(x_0 = [0; 0; 0; 0; 0; 0; 100; 100; 100; 10; 10; 1]^T\). The generated path is shown in Fig. 4. The corresponding optimal
states and the optimal control inputs are plotted in Fig. 5 and Fig. 6, respectively. The minimum flight time is 25.4 s. As shown in Fig. 5 and Fig. 6, all types of the constraints are satisfied. More importantly, the continuous state inequality constraints are satisfied.

4.2. Example 2: An environment with the first type of obstacles. In this example, the proposed algorithm are implemented in an environment with the first type of obstacles (8). We set the coordinate of the target as (400, 400, 400). x₀ is the same as that in Example 1. Here, pₓ = pᵧ = pᶻ = 8. In other words, the obstacles are in the box-like shape. The trajectory of the UAV is shown in Fig. 7,
4.3. **Example 3: An environment with the second type of obstacles.** An environment with the second type of obstacles (9) is considered in this section. The coordinate of the target is set as \((80, 20, 15)\) and the initial condition is set as \(x_0 = [0; 0; 0; 0; 5; 70; 10; 3; -3; -1]^T\). By using Algorithm 1, the optimal path is plotted in Fig. 10. The corresponding optimal state and control are given in Fig. 11 and Fig. 12, respectively. The minimum flight time is 15.92 s. As expected, the
A PATH PLANNING METHOD FOR THE QUAD-ROTOR UAVS

Figure 14. Example 3: The planned path of UAV with A star

Figure 15. The structure of PD controller

Figure 16. Example 4: the tracking trajectory of the UAV with Algorithm 1

UAV arrives at the target without hitting the obstacles as illustrated in Fig. 10, and all the constraints are satisfied as shown in Fig. 11 and Fig. 12.

The path generated by the sampling-based RRT algorithm is plotted in Fig. 13. As shown in Fig. 13, there is a ‘sharp corner’ on this trajectory. This is because the dynamics of the UAV is ignored for RRT. In Fig. 14, we plot the path generated by A star algorithm. Similarly, two ‘sharp corners’ are observed on the planned path.
4.4. Example 4: Trajectory tracking. In this example, we utilize a closed loop control system to track the paths generated in Example 3. The PID controller is adopted since it is the most used controller in real world applications. Here, a PD controller is used and the structure of the feedback control system is illustrated in Fig. 15. Further, a particle swarm optimization (PSO) method is applied to tune the parameters of PD controller.

The planned paths and the tracking paths generated by Algorithm 1, RRT algorithm and A star are plotted in Fig. 16, Fig. 17, and Fig. 18, respectively. From Fig. 16 we observe that the tracking trajectory follows the planned path almost everywhere for Algorithm 1. In contrast, there are big tracking errors as shown in Fig. 17 and Fig. 18. This is because the dynamics of the UAV is not considered for the RRT algorithm and A star algorithm.

We also plot the speed of the motors with the proposed algorithm, the RRT algorithm and the A start algorithm in Fig. 19. As shown in Fig. 19(a), the motor speed of the proposed algorithm has less changes than that of the RRT algorithm.
and that of the A star algorithm. The changes of the motor speed in Fig. 19(b) and Fig. 19(c) are due to the sharp corners on the planned path, which are marked on Fig. 17(b) and Fig. 18(b).

5. Conclusions. A time optimal path planning method was developed based on the control parametrization. By modeling the physical limits of UAV and the obstacles into the continuous inequality constraints, the problem was formulated as an optimal control problem subject to continuous inequality constraints. This problem was solved by the control parametrization method. Particularly, the continuous inequality constraints were handled by the constraint transcription method. The corresponding gradient formulas for the cost function and the constraints were derived, respectively. Simulation results showed that the proposed algorithm has less tracking error than that of the RRT algorithm and that of the A star algorithm. In addition, motor speed has less changes for the proposed algorithm than that of the RRT algorithm and that of the A star algorithm.
Appendix A. Proof of Theorem 3.1. To begin, we consider the following optimal parameter selection problem.

Problem P1. Given the following system:

\[
\begin{align*}
\dot{x}(t) &= f(t, x(t), \zeta) \\
x(0) &= x_0(\zeta)
\end{align*}
\]  
(A.44)

find a system parameter \( \zeta \in \mathbb{R}^n \) such that the cost functional:

\[
g_0(\zeta) = \Phi_0(x(T_f|\zeta), \zeta) + \int_0^{T_f} L_0(t, x(t|\zeta), \zeta) dt
\]  
(A.45)

is minimized and subject to the equality constraints:

\[
g_i(\zeta) = \Phi_i(x(T_f|\zeta), \zeta) + \int_0^{T_f} L_i(t, x(t|\zeta), \zeta) dt = 0, \ i = 1, 2, \cdots, N_c
\]  
(A.46)

and inequality constraints:

\[
g_i(\zeta) = \Phi_i(x(T_f|\zeta), \zeta) + \int_0^{T_f} L_i(t, x(t|\zeta), \zeta) dt \geq 0, \ i = N_c + 1, 2, \cdots, N
\]  
(A.47)

Then, we give the gradient formulae for the cost functional and the constraint functionals with the following lemma.

Lemma 1. (Theorem 5.2.1 of [16]) Consider Problem P1. For each \( i = 1, 2, \cdots, N \), the gradient of the functional is given as follows:

\[
\frac{\partial g_i(\zeta)}{\partial \zeta} = \frac{\Phi_i(x(T_f|\zeta), \zeta)}{\partial \zeta} + (\lambda_i(0|\zeta))^T \frac{\partial x^0(\zeta)}{\partial \zeta} + \int_0^{T_f} \frac{\partial H_i(t, x(t|\zeta), \zeta, \lambda_i(t|\zeta))}{\partial \zeta} dt
\]  
(A.48)

where, for each \( i = 1, 2, \cdots, N \)

\[
H_i(t, x(t|\zeta), \lambda, \lambda_i(t|\zeta)) = L_i(t, x, \zeta) + \lambda^T f(t, x, \zeta)
\]  
(A.49)

is the corresponding Hamiltonian and \( \lambda_i(t) \) is the corresponding co-state variable that satisfies the following differential equations

\[
(\dot{\lambda}_i(t))^T = -\frac{\partial H_i(t, x(t|\zeta), \lambda_i(t|\zeta))}{\partial x}, \quad t \in [0, T_f)
\]  
(A.50)

with

\[
(\lambda_i(T_f))^T = \frac{\partial \Phi_i(x(T_f|\zeta), \zeta)}{\partial \zeta}
\]  
(A.51)

In order to prove Theorem 3.1, we define the following Hamiltonian

\[
H_0(x, \lambda, \sigma, T_f) = T_f \lambda^T(s) f(x(s), \sigma)
\]  
(A.52)

Since \( \Phi_0(x(T)) = J = T_f \) and \( x^0 \) does not depend on \( \sigma \) and \( T_f \), then it follows that

\[
\frac{\partial \Phi_0}{\partial \sigma} = 0, \quad \frac{\partial \Phi_0}{\partial T_f} = 1
\]  
(A.53)

and

\[
\frac{\partial x^0}{\partial \sigma} = 0, \quad \frac{\partial x^0}{\partial T_f} = 0
\]  
(A.54)

Considering (A.52) and differentiating \( H_0 \) with respective to \( \sigma \) and \( T_f \), we obtain

\[
\frac{\partial H_0}{\partial \sigma} = T_f \left[ \frac{\partial f(x(s), \sigma)}{\partial \sigma} \right]^T \lambda_0(s)
\]  
(A.55)
and
\[
\frac{\partial H_0}{\partial T_f} = \left[ f(x(s), \sigma) \right]^T \lambda_0(s) \tag{A.56}
\]
Applying Lemma 1 to the cost function \(J = T_f\), it follows that
\[
\frac{\partial J}{\partial \sigma} = \Phi_0 + (\lambda_0(0))^T \frac{\partial x_0(\zeta)}{\partial \zeta} + \int_0^{T_f} \frac{\partial H_0}{\partial \sigma} \, ds \tag{A.57}
\]
By substituting (A.53), (A.54) and (A.55) into (A.57), (27) can be obtained straight away. (29) can be derived in the similar manner. This completes the proof.

REFERENCES

[1] X. Cheng, H. Li and R. Zhang, Autonomous trajectory planning for space vehicles with a Newton Kantorovich/convex programming approach, Nonlinear Dynamics, 89 (2017), 2795–2814.
[2] K. Daniel, A. Nash, S. Koenig and A. Felner, Theta*: Any-angle path planning on grids, Journal of Artificial Intelligence Research, 39 (2010), 533–579.
[3] A. Dehghan and M. Shah, Binary quadratic programming for online tracking of hundreds of people in extremely crowded scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence, 40 (2018), 568–581.
[4] H. Ergezer, and K. Leblebiciolu, 3D path planning for multiple UAVs for maximum information collection, Journal of Intelligent and Robotic Systems, 73 (2014), 737–762.
[5] L. D. Filippis, G. Guglieri and F. Quagliotti, Path planning strategies for UAVs in 3D environments, Journal of Intelligent and Robotic Systems, 65 (2012), 247–264.
[6] 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, 6 (2009), 271–290.
[7] B. T. Gatzke, W. Kang and H. Zhou, Trajectory optimization for helicopter unmanned aerial vehicles (UAVs), NPS Thesis, (2010).
[8] L. Jennings, K. L. Teo, M. Fisher and C. J. Goh, MISER3 version 2, optimal control software, theory and user manual, Department of Mathematics. The University of Western Australia, Australia, (1997).
[9] T. Ju, S. Liu, J. Yang and D. Sun, Rapidly exploring random tree algorithm-based path planning for robot-aided optical manipulation of biological cells, IEEE Transactions on Automation Science and Engineering, 11 (2014), 649–657.
[10] V. Kroumov and J. Yu, 3D path planning for mobile robots using annealing neural network, International journal of innovative computing information and control, 6 (2009).
[11] Y. Liang, J. T. Qi, J. Z. Xiao and Y. Xia, A literature review of UAV 3D path planning, Proceeding of the 11th World Congress on Intelligent Control and Automation, (2014), 2376–2381.
[12] C. Y. Liu, Z. H. Gong, K. L. Teo, J. Sun and L. Caccetta, Robust multi-objective optimal switching control arising in 1,3-propanediol microbial fed-batch process, Nonlinear Analysis Hybrid Systems, 25 (2017), 1–20.
[13] J. Omer and J. Farges, Hybridization of nonlinear and mixed-integer linear programming for aircraft separation with trajectory recovery, IEEE Transactions on Intelligent Transportation Systems, 14 (2013), 1218–1230.
[14] B. Oommen, S. Iyengar, N. Rao and R. Kashyap, Robot navigation in unknown terrains using learned visibility graphs. Part I: The disjoint convex obstacle case, IEEE Journal on Robotics and Automation, 3 (1987), 672–681.
[15] P. Pharpatara, B. Hriss and Y. Bestaoui, 3-D trajectory planning of aerial vehicles using RRT, IEEE Journal on Robotics and Automation, 25 (2017), 1116–1123.
[16] K. L. Teo, C. J. Goh and K. H. Wong, A unified computational approach for optimal control problems, Pitman Monographs and Surveys in Pure and Applied Mathematics, John Wiley and Sons, Inc., New York, 1991.
[17] J. Votion and Y. Cao, Diversity-based cooperative multivehicle path planning for risk management in costmap environments, IEEE Transactions on Industrial Electronics, 66 (2019), 6117–6127.
[18] F. Yang, K. L. Teo, R. Loxton, V. Rehbock, B. Li, C. Yu and L. Jennings, Visual Miser: An efficient user-friendly visual program for solving optimal control problems, *Journal of Industrial and Management Optimization (JIMO)*, 12 (2016), 781–810.

[19] C. Yu, K. L. Teo, L. Zhang, and Y. Bai, On a refinement of the convergence analysis for the new exact penalty function method for continuous inequality constrained optimization problem, *Journal of Industrial and Management Optimization (JIMO)*, 8 (2012), 485–491.

[20] J. Yuan, C. Liu, X. Zhang, J. Xie, E. Feng, H. Yin and Z. Xiu, Optimal control of a batch fermentation process with nonlinear time-delay and free terminal time and cost sensitivity constraint, *Journal of Process Control*, 44 (2016), 41–52.

[21] B. Zhao, B. Xian, Y. Zhang and X. Zhang, Nonlinear robust adaptive tracking control of a quadrotor UAV via immersion and invariance methodology, *IEEE Transactions on Industrial Electronics*, 62 (2015), 2891–2902.

Received July 2020; revised October 2020.

*E-mail address*: guo_jinlong@aliyun.com

*E-mail address*: bin.li@scu.edu.cn