Two Approaches for Path Planning of Unmanned Aerial Vehicles with Avoidance Zones

Chuangchuang Sun*, Yen-Chen Liu† Ran Dai‡

Iowa State University, Ames, IA, 50011, USA

David Grymin §

Wright Patterson Air Force Research Lab, Ohio, 45433, USA

I. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have been widely used in the battle field for military operations, such as target attacking, surveillance and reconnaissance and replace the roles of manned aircraft in a more efficient and less risky fashion. The advances in autonomous control, signal processing, communication, etc. have enabled the UAVs to make onboard decisions in certain operations without human interactions. However, the requirement for autonomy and decision making will be more challenging when operations are performed in hostile environments, i.e., areas with radar systems. In such scenarios, the susceptibility of UAVs in hostile environments raises requirements for flight path planning to avoid or reduce the possibility of detection by adversarial radars.

*Graduate Student, Aerospace Engineering Department, Iowa State University, Ames, IA, 50011 Email: ccsun@iastate.edu
†Graduate Student, Aerospace Engineering Department, Iowa State University, Ames, IA, 50011 Email: clarkliu@iastate.edu
‡Assistant Professor, Aerospace Engineering Department, Iowa State University, Ames, IA, 50011 Email: dairan@iastate.edu
§Engineer, Autonomous Control Branch, Wright Patterson Air Force Research Lab, Ohio, 45433. Email: david.grymin.1@us.af.mil
Due to the importance of UAV path planning in achieving mission success, many methodologies have been developed, especially for problems with obstacle avoidance constraints.\textsuperscript{1–3} Early work in path planning, such as artificial potential function\textsuperscript{4,5} and randomized sampling algorithms,\textsuperscript{6–9} focuses on searching a feasible path without optimizing the desired path performance, i.e., path length or travel time. Due to the nonconvex nature of obstacle avoidance constraints, the constrained path planning problem to optimize the desired path performance is generally formulated as a nonlinear programming (NLP) problem which can be solved via an NLP solver.\textsuperscript{10} However, the solution of an NLP problem is dependent on the initial guess and convergence to a local optimum is generally not guaranteed. When multiple avoidance zones are considered, it is even difficult to find a feasible solution using an NLP. On the other hand, an optimized heuristic method, such as the optimized Rapidly-Exploring Random Tree algorithm, named as RRT*, has been developed to search for obstacle avoiding paths with asymptotic optimality property.\textsuperscript{11,12} However, the RRT* algorithm itself does not include the flight kinematic constraints.

Inspired by the computational efficiency of convex optimization techniques\textsuperscript{13} and the robustness of RRT* algorithm, two approaches are proposed to search for the minimum-time path for a UAV flying through a specified area with fixed starting and ending points and multiple avoidance zones, including circular and elliptical. One approach is to reformulate the path planning problem as a general/nonconvex Quadratically Constrained Quadratic Programming (QCQP) problem where an iterative rank minimization (IRM) method proposed in our previous work\textsuperscript{14,15} is applied to solve the path planning problem. The IRM method with each iteration formulated as a convex optimization problem has guaranteed linear convergence to local optimum. Different from the NLP approach, an initial guess of the planned path is not required in IRM. A second approach based on heuristic search integrates the flight kinematic constraints to generate smooth paths by proposing a refined RRT* algorithm.

Contribution of this numerical optimization approach is the unique QCQP formulation for the path planning problem and the associated IRM method with guaranteed linear con-
vergence. Moreover, the refined RRT* algorithm is the first one which integrates flight
kinematic in a sampling-based heuristic search. In addition, simulation results from both
approaches and NLP solver are provided to analyze performance of the proposed approaches
in terms of convergence, objective value, and computational time.

The paper is organized as follows. In §II, the path planning problem with avoidance
zones are introduced. The numerical optimization approach is described in §III including the
general QCQP formulation and the IRM method. The refined RRT* approach is described
in §IV, followed by the simulation results and analysis in §V. We conclude the paper with a
few remarks about the two proposed methods in §VI.

II. PROBLEM FORMULATION

The problem of solving a single UAV passing through hostile environments with avoidance
zones is illustrated in Fig. 1, where the UAV has assigned starting and ending points, denoted
as triangles. The avoidance zones have pre-defined locations and shapes. Depending on
the requirements of flight mission, the performance index can be assigned accordingly, i.e.,
minimum flight time.

The flight kinematics of a UAV in two-dimensional space is represented by a single control
model in the form of

\[
\begin{align*}
\dot{x} &= V \cos \theta \\
\dot{y} &= V \sin \theta \\
\dot{\theta} &= u \\
|u| &\leq u_{max},
\end{align*}
\]

where \(x\) and \(y\) are the coordinates, \(V\) is the specified cruise speed, \(\theta\) is the heading angle,
and \(u_{max}\) is the maximum rate of change of the heading angle. The starting and ending
points are specified as \([x_0, y_0]\) and \([x_f, y_f]\). The avoidance zones are represented by ellipses
Avoidance zone $j$ at $[c_j, d_j]$

Figure 1. Illustration of UAV path planning problem with avoidance zones.

formulated as,

$$
\left( \frac{x - c_j}{a_j} \right)^2 + \left( \frac{y - d_j}{b_j} \right)^2 \geq 1, \quad \forall \ j = 1, \ldots, m',
$$

(2.2)

where $[c_j, d_j]$ represents the center of avoidance zone $j$, $j = 1, \ldots, m'$, $(a_j, b_j)$ are pre-defined parameters, i.e., the semi-major axes of the elliptical zones, and $m'$ is the number of avoidance zones. The above function can create different elliptical shapes (including rotated ones, for they are still quadratic) modeling a range of avoidance zones with specified centers. A special case is a circular-shaped zone by setting $a_j = b_j$. For minimum time of flight, the performance index is $J = \int_{t_0}^{t_f} 1 \, dt$. Consequently, the minimum time path planning problems
can be formulated as

\[ J = \min_u \int_{t_0}^{t_f} 1 \, dt \]

s.t. \[ x(t_0) = x_0, y(t_0) = y_0, x(t_f) = x_f, y(t_f) = y_f, \]

\[ \dot{x} = V \cos \theta \]

\[ \dot{y} = V \sin \theta \]

\[ \left( \frac{x - c_j}{a_j} \right)^2 + \left( \frac{y - d_j}{b_j} \right)^2 \geq 1, \ \forall \ j = 1, \ldots, m', \ \forall x, y, t_0 \leq t \leq t_f \]

\[ \dot{\theta} = u \]

\[ |u| \leq u_{\text{max}}. \] (2.3)

The above path planning problems formulated by nonlinear equations are difficult to solve. The indirect method requires deriving the necessary conditions for optimality based on Hamiltonian and Euler Lagrange equations. In most cases, the difficulty of guessing the initial adjoint variables and the trigonometric equations involved in the problem formulation make the indirect method infeasible. The direct method, such as using collocation and an NLP cannot guarantee fast convergence to a local optimal solution, or even convergence to a feasible solution, when highly nonlinear equations are included in the constraints and/or an initial guess of the solution is randomly selected. Therefore, two distinct approaches are proposed below to solve the UAV path planning problems posed in (2.3) to improve the computational performance.

### III. Numerical Optimization Approach

The first step in the numerical optimization approach is to convert the nonlinear optimization problem formulated in (2.3) into a general QCQP problem, where the objective is a quadratic function and the constraints are quadratic equalities or inequalities. An iterative convex optimization method is then introduced to solve the general QCQP problem. The novelty of QCQP formulation and its associated iterative method is that it does not in-
olve linearization procedures in the formulation and optimization approach such that errors generated from linearization of a highly nonlinear model are inevitable.

III.A. Reformulation of the Path Planning Problems as QCQP Problems

The single control model described in (2.1) for UAV path planning includes trigonometric functions, which are highly nonlinear and may generate singular matrices in computational operations. The first step is to reformulate the above nonlinear optimization problems as general QCQP problems via a discretization method.

A continuous flight path can be discretized into a series of segments represented by coordinates \([x_h, y_h], h = 1, \ldots, H\), at each node, where \(H\) is the number of discrete nodes. By discretization, the change rate of the coordinates can be approximately determined by two adjacent nodes,

\[
\begin{align*}
\dot{x} &= \frac{x_{h+1} - x_h}{\Delta t} = V \cos \theta, \ h = 1, \ldots, H, \\
\dot{y} &= \frac{y_{h+1} - y_h}{\Delta t} = V \sin \theta, \ h = 1, \ldots, H, 
\end{align*}
\]

where \(\Delta t\) is the uniform time interval between two adjacent nodes. The above two equations can be synthesized as

\[
(x_{h+1} - x_h)^2 + (y_{h+1} - y_h)^2 = V^2(\Delta t)^2. \tag{3.6}
\]

Differentiating (3.4)-(3.5) leads to

\[
\begin{align*}
\ddot{x} &= \frac{x_{h+2} + x_h - 2x_{h+1}}{(\Delta t)^2} = -V \dot{\theta} \sin \theta, \ h = 1, \ldots, H, \\
\ddot{y} &= \frac{y_{h+2} + y_h - 2y_{h+1}}{(\Delta t)^2} = V \dot{\theta} \cos \theta, \ h = 1, \ldots, H.
\end{align*}
\]
Combining with $\dot{\theta} = u$ and $|u| \leq u_{\text{max}}$, the above equations can be synthesized as

$$
\dot{\theta}^2 = \left( \frac{x_{h+2} + x_h - 2x_{h+1}}{V(\Delta t)^2} \right)^2 + \left( \frac{y_{h+2} + y_h - 2y_{h+1}}{V(\Delta t)^2} \right)^2 \leq u_{\text{max}}^2.
$$

By introducing an additional variable $t' = \Delta t^2$, the above constraint can be reformulated as a quadratic inequality in the form of

$$
(x_{h+2} + x_h - 2x_{h+1})^2 + (y_{h+2} + y_h - 2y_{h+1})^2 \leq V^2 u_{\text{max}}^2 t'^2, \ h = 1, \ldots, H - 2.
$$

Meanwhile, the avoidance zones are originally formulated as quadratic inequalities. Based on the above reformulation, the path planning problems can be generalized as a nonconvex QCQP problem in the form of

$$
J = \min_{x, y, \Delta t, t'} (H - 1)\Delta t
$$

s.t. 

\begin{align*}
(x_{h+1} - x_h)^2 + (y_{h+1} - y_h)^2 &= V^2(\Delta t)^2, \ h = 1, \ldots, H - 1, \\
(x_{h+2} + x_h - 2x_{h+1})^2 + (y_{h+2} + y_h - 2y_{h+1})^2 &\leq V^2 u_{\text{max}}^2 t'^2, \ h = 1, \ldots, H - 2, \\
\left( \frac{x_h - c_j}{a_j} \right)^2 + \left( \frac{y_h - d_j}{b_j} \right)^2 &\geq 1, \ \forall \ j = 1, \ldots, m', \ h = 1, \ldots, H, \\
t' &= (\Delta t)^2.
\end{align*}

Based on the above reformulation, the path planning problem with avoidance zones can be generalized as a nonconvex QCQP problem in the form of

$$
J = \min_x x^T Q_0 x + a_0^T x
$$

s.t. 

\begin{align*}
x^T Q_j x + a_j^T x &\leq c_j, \ \forall \ j = 1, \ldots, m \\
l_x &\leq x \leq u_x,
\end{align*}

where $x \in \mathbb{R}^n$ is the unknown vector to be determined, $Q_j \in \mathbb{S}^{n \times n}$, $j = 0, \ldots, m$, is an
arbitrary symmetric matrix, \( c_j \in \mathbb{R}, j = 1, \ldots, m \), and \( a_j \in \mathbb{R}^n, j = 0, \ldots, m \). Moreover, \( l_x \in \mathbb{R}^n \) and \( u_x \in \mathbb{R}^n \) are the lower and upper bounds on \( x \), respectively. Since \( Q_j (j = 0, \ldots, m) \) is not necessarily a positive definite matrix, the problem in (3.10) is classified as NP-hard.

### III.B. An Iterative Approach for General QCQP Problems

Efforts toward solving general QCQP problems have been focused on finding the bounds on the optimal values by linear or semidefinite relaxation.\(^{16}\) Although randomization and linearization have been used to find an approximate solution, neither approach guarantees optimality of the approximate solution with a determined convergence rate.\(^{17}\)

The above QCQP problem with inhomogeneous quadratic function can be transformed into a homogeneous one by introducing a new variable \( \alpha \in \mathbb{R} \) and a new quadratic constraint \( \alpha^2 = 1 \) by the following formulation

\[
J = \min \begin{bmatrix} x^T & \alpha \end{bmatrix} \begin{bmatrix} Q_0 & a_0/2 \\ a_0^T/2 & 0 \end{bmatrix} \begin{bmatrix} x \\ \alpha \end{bmatrix} \tag{3.11}
\]

\[
\text{s.t.} \quad \begin{bmatrix} x^T & \alpha \end{bmatrix} \begin{bmatrix} Q_j & a_j/2 \\ a_j^T/2 & 0 \end{bmatrix} \begin{bmatrix} x \\ \alpha \end{bmatrix} + c_j \leq 0, \forall j = 1, \ldots, m, \\
l_x \leq x \leq u_x, \\
\alpha^2 = 1.
\]

Then \( x^*/\alpha^* \) will be solution of the original problem stated in (3.10) while \( (x^*, \alpha^*) \) is the solution pair of (3.11). In addition, linear constraints in (3.10) can be rewritten in the above quadratic form as well by setting the corresponding matrix \( Q = 0 \). The homogeneous QCQP problem is formulated as

\[
J = \min x^T Q_0 x \tag{3.12}
\]

\[
\text{s.t.} \quad x^T Q_j x \leq c_j, \forall j = 1, \ldots, m.
\]
Based on this fact, any inhomogeneous QCQP can be transformed into a homogeneous one. Without loss of generality, the following approach to solve nonconvex QCQP problems focuses on homogeneous QCQPs.

After introducing a rank one positive semidefinite matrix, \( X = xx^T \), the positive semidefinite relaxation method is formulated as

\[
J = \min_X \langle X, Q_0 \rangle 
\]

\[
s.t. \quad \langle X, Q_j \rangle \leq c_j, \, \forall \, j = 1, \ldots, m \\
X \succeq 0,
\]

where \( \langle \cdot \rangle \) denotes the inner product of two matrices, i.e., \( \langle A, B \rangle = \text{trace}(A^TB) \). By reformulating the QCQP problem as a relaxed SDP problem, one can obtain a lower bound on its optimal value. However, the relaxation method will not yield an optimal solution of the unknown variables \( x \). The only difference between the SDP relaxation and the equivalent conversion is that the rank-one constraint, \( X = xx^T \), is excluded in SDP relaxation. In order to obtain the optimal solution of \( x \), work in\textsuperscript{14,15} reconsider the rank-one constraint on matrix \( X \) and propose an alternative approach to gradually approach the rank constraint.

A matrix with rank-one has only one nonzero eigenvalue. Therefore, instead of putting a constraint on the rank, we focus on constraining the eigenvalues of \( X \) to enforce the \( n - 1 \) smallest eigenvalues of \( X \) are all zero. It has been proven that, when \( X \) is a nonzero positive semidefinite matrix, \( X \) is a rank-one matrix if and only if \( rI_{n-1} - V^TXV \succeq 0 \), where \( V \in \mathbb{R}^{n \times (n-1)} \) are the eigenvectors corresponding to the \( n - 1 \) smallest eigenvalues of \( X \), \( I_{n-1} \in \mathbb{R}^{(n-1) \times (n-1)} \) is an identity matrix, and \( r \) is a significantly small positive number. In other words, a substitute for the rank one constraint on \( X \) is the semidefinite constraint \( rI_{n-1} - V^TXV \succeq 0 \), where \( r \to 0 \).\textsuperscript{18} However, until we can find the optimal solution of \( X \), we cannot obtain the exact \( V \) matrix that is dependent on \( X \). In our recent work,\textsuperscript{14} an IRM method was proposed to gradually minimize the rank of \( X \). At each step, we try to optimize the original objective function while at the same time minimizing the weighted parameter \( r \).
such that, when \( r = 0 \), the rank one constraint on \( X \) will be satisfied. The relaxed problem solved at each iteration step, \( k \), of IRM is formulated as a convex optimization problem in the form of

\[
J = \min_{X_k, r_k} \langle X_k, Q_0 \rangle + w^k r_k \tag{3.14}
\]

\[
s.t. \quad \langle X_k, Q_j \rangle \leq c_j, \quad \forall \ j = 1, \ldots, m
\]

\[
X_k \succeq 0
\]

\[
r_k I_{n-1} - V_{k-1} X_k V_{k-1} \succeq 0,
\]

where \( w > 1 \) is the weighting factor for \( r_k \). Obviously, the solution at the converged point satisfies the rank one constraint on \( X \) as well as the other constraints described in the equivalent QCQP problem. Through the Karush-Kuhn-Tucker conditions, we then have attained linear convergence to at least a local optimum of the proposed IRM method. A more detailed derivation including a proof of linear convergence can be found in.\textsuperscript{14,15}

Different from the NLP method which can be applied to solve general nonlinear optimization problems, the IRM algorithm above does not require an initial guess of the unknown variables. The IRM algorithm is initiated by solving problem (3.13) without considering the rank constraint to obtain \( V_0 \) from \( X_0 \) for \( k = 0 \). The method then iteratively solves problem (3.14) to obtain \( X_k \) and update \( V_k \) until \( r_k \leq \delta \), where \( \delta \) is a stopping threshold. Furthermore, except the newly introduced variable \( r_k \), there are no extra introduced unknown variables in the formulation. This simple procedure can be easily implemented for any general QCQP problem. With the avoidance zones originally formulated as quadratic constraints, together with the quadratic discretized formulation of UAV flight kinematics, the IRM method is applied here to solve the path planning problem with avoidance zones.
IV. Heuristic Search Approach

This section attempts to approach the path planning problem from an incremental sampling and exploring aspect. The RRT* algorithm in\textsuperscript{11,12} has been modified for the specific UAV path planning problem with avoidance zones. Before discussing the algorithm, it is important to introduce a few necessary procedures in the optimized RRT algorithm, named as RRT*.

IV.A. RRT* Algorithm

Let $S$ represent the unexplored domain and $S \subset \mathbb{R}^n$. The obstacle region and obstacle-free region are denoted as $S_{\text{obs}}$ and $S_{\text{free}}$, respectively, where $S_{\text{free}} = S \setminus S_{\text{obs}}$.

Given the initial position, $z_{\text{initial}} = [x_0, y_0]$, and goal position, $z_{\text{goal}} = [x_f, y_f]$, the RRT* algorithm solves the motion planning problem by searching through the $S_{\text{free}}$ area to yield a tree, $T$. Furthermore, an iterative process examines $T$ from $z_{\text{initial}}$ to $z_{\text{goal}}$ and outputs a feasible path, $P$, that satisfies all the given constraints.

Sampling: The \texttt{sample} function returns a state, $z_{\text{rand}}$, that randomly samples from the desired field. The sampling process is assumed to be uniformly distributed among the selected nodes in the field such that $z_{\text{rand}} \in S_{\text{free}}$, $p_1 \in P$, as well as $p_2 \in P$.

Collision Evaluation: By introducing a set of nodes, the \texttt{Collision\_eval} function forms an edge among two nodes and returns true if the entire edge occurs on the obstacle-free zone.\textsuperscript{20}

Cost: The \texttt{cost} computation procedure evaluates the summation of distance from $z_{\text{rand}}$ to $z_{\text{goal}}$ and the distance from $z_{\text{rand}}$ to any nearby nodes. When seeking for the minimum total distance during exploring, the edges of $T$ tend to construct toward $z_{\text{goal}}$ to accelerate the process of solving for a feasible solution.

Path Connecting: Given a tree with a feasible solution, i.e., $P \subset \mathbb{R}^m$ and $P \subset T$, the \texttt{Path\_Connect} function backtracks the path from $z_{\text{goal}}$ to each corresponding parents until it reaches $z_{\text{initial}}$. 
In the RRT* algorithm, the tree begins with \( z_{\text{initial}} \) and an empty path and the steps are summarized as:

1) \( T \leftarrow \{ z_{\text{initial}} \}; \ P \leftarrow \{ \} \)
2) Iteration: While isempty(\( P \)) Do
   2.1) A random position \( z_{\text{rand}} \) is generated by sampling the obstacle-free zone, \( z_{\text{rand}} \leftarrow \text{Sample}(S_{\text{free}}) \).
   2.2) It then determines its potential parent \( z_{\text{min}} \) and tests for collision before extending the tree, \( z_{\text{min}} \leftarrow \text{Cost}(z_{\text{rand}}, T) \).
   2.3) If the two points do not contact with obstacles in between, \( z_{\text{rand}} \) becomes \( z_{\text{new}} \) and a link is made between two points, denoted by \( T \leftarrow T \cup \{(z_{\text{min}}, z_{\text{new}})\} \) for \( \text{Collision\_eval} (z_{\text{rand}}, z_{\text{min}}) \) being not true.
   2.4) If \( z_{\text{rand}} \) collides with obstacles on the way to \( z_{\text{min}} \), it returns to sample a new random state and continue the algorithm, denoted by \( z_{\text{new}} \leftarrow z_{\text{rand}} \).
   2.5) With sufficient iteration, the expansion of the tree should obtain at least one \( z_{\text{new}} \) that is close to \( z_{\text{goal}} \) while satisfying the condition that the distance between \( z_{\text{new}} \) and \( z_{\text{goal}} \) is less than \( d_{\text{min}} \), where \( d_{\text{min}} \) represents a defined minimum distance within which \( T \leftarrow T \cup \{(z_{\text{new}}, z_{\text{goal}})\} \).
3) The path \( P \) starts building up from \( z_{\text{goal}} \) and traces back to \( z_{\text{initial}} \) via the remarks of related parent from trees, denoted by \( P \leftarrow \text{Path\_Connect}(T) \).

**IV.B. Refined RRT***

The RRT* algorithm stated above does not include the flight kinematics described in (2.1). In addition, there is more room to shorten the entire path after connecting the feasible sampling trees. Two refinement steps are considered to further improve the performance index and smooth the path \( P \) obtained from RRT* by integrating the flight kinematic constraints.

*Path Optimization:* The \texttt{Path\_Opt} function randomly samples two points on \( P \) and checks for any possible obstacle collision. If \texttt{Collision\_eval} returns true, the new nodes \( p_1 \) and \( p_2 \) are added to \( P \) to replace any intermediate node. Meanwhile, a new edge between \( p_1 \) and \( p_2 \) is created as a substitution resulting in shortening the entire path length.
Path Refinement: In order to satisfy the limitation of maximum heading angle change rate, $u_{\text{max}}$, the \texttt{Steering_eval} function verifies the heading angle change rate, defined as $\Delta \dot{\phi}$, between two connecting edges added in \texttt{Path_{Opt}} function. The rate of change of the heading angle is determined by the three ending points of two connecting edges located on a circular arc, as shown in Fig. 2. The heading angle change rate is approximated by $\Delta \phi$ divided by the traveling time along the circular arc. The \texttt{Steering_eval} function returns true when all of the approximated heading angle change rates are no greater than $u_{\text{max}}$.

For step 3 in RRT* algorithm, the refinement steps are stated below:

4) Set $\text{Path}_{\text{Opt}}(P) \leftarrow \text{RRT}^*(P)$,

5) Iteration: while $\text{Steering_eval}(P)$ is not true, $(p_1, p_2) \leftarrow \text{Sample}(P)$, If $\text{Collision_eval}(p_1, p_2)$, then $P \leftarrow P \cup \{(p_1, p_2)\}$.

The replaced edges stated in the refinement step reduce the travel time and smooth the sharp turns by eliminating the violation of threshold $\phi$. The loop is terminated once $\text{Path}_{\text{Opt}}(P)$ is no longer empty. The refinement procedure is completely heuristic as well.
as the RRT* algorithm.

V. SIMULATION EXAMPLES

In this section, a group of simulation cases are presented to demonstrate the advantages and limitations of the two proposed approaches. In the first case, one avoidance zone is considered and the path planning problem for this simple case has an analytical solution as a reference to verify the accuracy of proposed approaches. Other cases consider a group of cluttered avoidance zones, which makes the path planning problem more difficult. In both types of cases, the parameters in the IRM method are set as \( [x_0, y_0] = [0, 0]^T, [x_f, y_f] = [10, 10]^T, V = 1, u_{max} = 0.2, w = 1.5, \) and \( \epsilon = 1e - 4 \) as the threshold for the convergence of \( r \). All cases use 25 discrete nodes to represent the planned path. The corresponding parameters in the refined RRT* method are set as \( d_{min} = 0.3 \) and \( \theta = 150^\circ \). All of the simulation cases are run on a desktop computer with a 3.50 GHz processor and 16 GB RAM.

V.A. Case with One Avoidance Zone

The single avoidance zone considered here is a circle centered at \([5, 5]\) with radius of 5. The analytical solution for this case yields a minimum flight time of \( f^{*}_{benchmark} = 17.85 \). The optimal solution from IRM is \( f^{*}_{IRM} = 17.84 \). The relative error of the solution from IRM is only 0.066% compared to the benchmark, which verifies the high accuracy of IRM. In addition, results obtained from the NLP solver in MATLAB is provided. Groups of initial guesses are randomly generated and a feasible path which yields a flight time of \( f^{*}_{NLP} = 17.84 \) is demonstrated in Fig. 3(a). Therefore, the comparative results indicate that IRM and NLP both work in solving this type of benchmark problems when converging to a local minimizer. The paths obtained from IRM and NLP are demonstrated in Fig. 3(a). It takes 10 iterations for the IRM with threshold \( \epsilon = 1e - 4 \) converging to the optimal solution. The overall computation time for IRM is 202.90 seconds.
The planned paths from RRT* and refined RRT* are shown in Fig. 3(b). The original path from RRT* includes sharp turns at connections of sampling trees. Through refinement steps, the refined path with flight time of $f_{\text{refined}}^* = 18.00$ is very close to the planned path generated from IRM. It takes 0.95 seconds to generate the coarse path from RRT*. The refinement steps take additional 25.15 seconds to refine the coarse path. Although the refined RRT* does not yield the same high accuracy compared to IRM, it has significant reduction on computation time in this simple example.

Moreover, as the dynamics equations are discretized, we present the results from different $N$ in Table 1. In this example, as we have $N = 20$ for it is already accurate enough.

![Figure 3. Comparative results of planned path for case one.](image)

(a) Planned paths generated from IRM (dash) and NLP (solid) for case one
(b) Planned paths generated from RRT* (dash) and refined RRT* (solid) for case one

<table>
<thead>
<tr>
<th>N</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{\text{benchmark}}$</td>
<td>17.85</td>
<td>17.85</td>
<td>17.85</td>
<td>17.85</td>
</tr>
<tr>
<td>$f_{\text{IRM}}$</td>
<td>17.80</td>
<td>17.86</td>
<td>17.84</td>
<td>17.85</td>
</tr>
<tr>
<td>IRM Rel. Error/%</td>
<td>-0.27</td>
<td>0.063</td>
<td>-0.066</td>
<td>0.0029</td>
</tr>
</tbody>
</table>

**Table 1.** Comparative results for different $N$.

V.B. **Cases with Multiple Avoidance Zones**

A group of cluttered avoidance zones in the shape of circles and ellipses are considered in the second case. The optimal solution from IRM is $f_{\text{IRM}}^* = 14.75$ while the one from NLP is $f_{\text{NLP}}^* = 16.02$. Figure 4(a) shows the planned paths from IRM and NLP for this case. The
IRM converges to the optimal solution within 15 iterations and the overall computational time is 1065.77 seconds. Due to the multiple avoidance zones, it is more difficult to find a feasible path when using a random initial guess with the NLP method. Different from case one where the paths from refined RRT* lead to a semi-uniform result with significantly small differences, paths generated for case two fall into several branches, as shown in Fig. 4(b). Among 350 simulation runs, only 5.43% simulations lead to a shortest path with $f^*_{\text{refined}} = 14.93$, which is consistent with the one obtained from IRM. However, most of the simulations, with the probability of 79.71%, generate flight paths yielding $f^*_{\text{refined}}$ around 17.96 with ±0.001 difference. The remaining 14.85% simulations generate results with even larger $f^*_{\text{refined}}$. Although the random process is assumed to sample the distribution uniformly, it is observed that the RRT is biased to generate more samples in larger spaces.ʻ Thus the sampling trees are not likely to distribute vertices in narrow spaces, leading to the low opportunity of producing the shortest outcome in case two. It takes an average of 10.88 seconds to find the coarse path from RRT* and the average computation time to obtain the final path considering refinement steps is 193.29 seconds. Compared with the simulation time of IRM, the refined RRT* requires much less computation time without guarantee of optimality. For the convenience for comparison, we listed the result from the three methodologies in Table 2.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Parameter</th>
<th>time/sec</th>
<th>$f^*$</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLP</td>
<td>SNOPT</td>
<td>20.76</td>
<td>16.02</td>
<td>Least time</td>
</tr>
<tr>
<td>IRM</td>
<td>$N = 25, w = 1.5, \epsilon = 1e - 4$</td>
<td>1065.77</td>
<td>14.75</td>
<td>(Equally best $f^*$)</td>
</tr>
<tr>
<td>RRT*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Refined RRT*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

To further verify advantages of the proposed methods, results of four additional cases are demonstrated in Fig. 5, where overlap between avoidance zones are considered. For each case, the refined RRT* execute 15 simulation runs and the path yielding best performance is provided in the corresponding plot to compare with those obtained from IRM. Among the
four cases, two of them generate very similar results and the remaining two lead to different paths where flying time resorting from the IRM is slightly shorter than the corresponding results from the best result of the refined RRT*.

All example cases indicate that the numerical optimization approach using the IRM method achieves high performance with guaranteed convergence. However, at each iteration of IRM, the additional semidefinite constraint, $r_kI_{n-1} - V_{k-1}X_kV_{k-1} \succeq 0$ in (3.14), introduces $(n-1) \times (n-1)$ linear constraints, which results in significant increases of computational time compared to the semidefinite relaxation formulated in (3.13). The heuristic search based on refined RRT*, on the other hand, has faster convergence without guarantee of optimality. In real applications, according to the mission definition, one of the approaches can be selected to meet specific mission planning requirement.

VI. CONCLUSIONS

This paper introduces two approaches for UAV path planning problems with avoidance zones. The numerical optimization approach converts the path planning problems in nonlinear form into Quadratically Constrained Quadratic Programming (QCQP) problems. An iterative rank minimization method is applied to gradually approach the optimal solution of QCQPs. The heuristic search method refines the paths obtained from the optimized Rapidly-
Figure 5. Planned paths generated from IRM (solid) and best result of refined RRT* (solid) for cases three to six

Exploring Random Tree algorithm to satisfy constraints and reduce the cost. Examples are presented to demonstrate the feasibility and effectiveness of the two approaches in solving path planning problems. Both of our proposed algorithms outperform the general-purposed NLP solvers in the sense of the objective value.

References


