Trajectory planning of unmanned aerial vehicle based on the improved biogeography-based optimization algorithm

Zhishuang Xue¹,² and Xiaofang Liu¹,³

Abstract
This paper is concerned with trajectory planning for unmanned aerial vehicle in a three-dimensional complex workspace. Biogeography-based optimization algorithm is widely used in solving practical problems because of its fewer parameters, fast convergence rate, and good global optimization ability. In this paper, some improvements, modifying migration and mutation operations are made on the biogeography-based optimization algorithm to make it suitable for solving the trajectory planning problem. The optimal trajectory obtained by the improved algorithm can be used to generate the reference trajectory. Then, a control scheme of unmanned aerial vehicle based on the Lyapunov theory and radial basis function neural network is formed to track the reference trajectory. The improved trajectory algorithm generates the shortest trajectory and the time consumption is the lowest. Finally, the designed control scheme makes the unmanned aerial vehicle track the different trajectories quite well, the effectiveness of it can be illustrated by algorithm accuracy and electricity consumption.

Keywords
Trajectory planning, improved biogeography-based optimization, Lyapunov theory, neural network

Date received: 29 January 2021; accepted: 26 February 2021

Handling Editor: James Baldwin

Introduction
Unmanned aerial vehicle (UAV) is often seen for reconnaissance, search, and rescue missions in dangerous areas because of its high mobility, low risk, and low cost.¹,² In order to ensure the efficiency and safety of UAV in the task completion, it is usually necessary to estimate an optimal flight trajectory before a mission execution. The trajectory planning for UAV in complex environments to avoid a series of obstacles is really a challenging problem. For these reasons, UAV trajectory planning has become a research hotspot.¹–⁶ Samaniego et al.⁷ regarded the space of UAV flight as a discrete adaptive grid with minimum step size, and used cost response algorithm and recursive calculation to conduct trajectory planning, which reduced the response time. Lupascu et al.⁸ extended the square in two dimensions to the cube in three dimensions for trajectory planning, and demonstrated the feasibility of the algorithm. Contreras-Cruz et al.⁹ trajectory planning in
a static environment using artificial bee colony algorithm was proposed. Firstly, artificial bee colony algorithm was used to explore the local environment, and then evolutionary algorithm was used to obtain the optimal trajectory. Li et al.\textsuperscript{10} used the balance strategy to make an improvement on the artificial bee colony algorithm, and collected the convergence information in the iterative process to control the exploration accuracy, so as to achieve a balance between local and global information. Particle swarm optimization algorithm with dynamic adjustment of inertia weight (WLIPSO) used in Lu et al.\textsuperscript{11} initialized a swarm of particles with a random solution. The cooperation and competition ideas were introduced to update its speed and it found the optimal solution by searching for the personal best and the global best. However, in the late stage of the convergence, the convergence speed becomes slow, all particles tend to be the same and lose diversity. When the algorithm converges to a certain degree, it cannot be further optimized and the accuracy it can achieve is not high. Biogeography-based optimization with covariance matrix based migration (CMM-BBO)\textsuperscript{12} is a novel evolutionary algorithm. The covariance matrix based migration operator combined with the original migration randomly, thus the population diversity is increased. CMM-BBO can get the optimal solution of the shortest trajectory, but it is inefficient because it traverses many nodes and performs many repeated operations. The above algorithms have some disadvantages, with the expansion of the planning area, their computational complexities will increase exponentially, resulting in a “combinatorial explosion.” and they are prone to fall into local optimal solutions, which needs to be avoided. Compared with these algorithms, the improved BBO algorithm can deal with the uncertainty in trajectory planning more effectively and improve the reliability of trajectory planning and the stability of the system.

Biogeography-based optimization (BBO) algorithm, proposed in Simon,\textsuperscript{13} is a global optimization algorithm that studies the distribution of organisms in geographical areas. It uses migration operation to realize information sharing among solutions (habitats) and uses mutation operation to enrich the population diversity in habitats.\textsuperscript{14} It describes the migration, formation, and extinction of species by the mathematical model based on biogeography.\textsuperscript{15} The improved BBO algorithm is similar to particle swarm optimization (PSO) algorithm, ant colony optimization algorithm, and genetic algorithm, but it has fewer designed parameters, low computational complexity, and strong optimization performance.\textsuperscript{3,13,16,17}

According to the grid-based workspace modeling algorithm in Zhang et al.,\textsuperscript{18} Islam et al.,\textsuperscript{19} and He et al.,\textsuperscript{20} the UAV’s workspace is divided into a series of small blocks by using grids through the three-dimensional space, and the vertices of each block are selected as trajectory nodes. Mark these nodes as collision and collision-free nodes by determining whether they are in obstacles or not.\textsuperscript{21,22} Simply select a series of available trajectory nodes (collision-free nodes) in a certain order and put them into a trajectory node list, that list consists of some available trajectories. Each available trajectory can be regarded as a habitat in the biogeography algorithm, and its nodes can be regarded as suitability index variables (SIVs)\textsuperscript{23} of the habitat. The habitat suitability index (HSI)\textsuperscript{24} of each habitat can be obtained by taking its nodes as parameters into the cost function calculations. According to the obtained HSIs, the immigration rate, emigration rate, and mutation probability corresponding to each habitat can be calculated, which can be used for the migration operation and mutation operation.\textsuperscript{25,26} Finally, the optimal trajectory is generated. In particular, an improved migration and mutation operation is proposed in section 2.D for UAV trajectory planning. In addition, WLIPSO and CMM-BBO algorithm are used to complete the task of the UAV trajectory planning in the same workspace, and then compared with the proposed algorithm.

The optimal trajectory generated by the proposed biogeography algorithm can be used to generate the flight reference trajectory for UAV. Then, a fixed-wing UAV mathematical model under external interference is constructed and selected as the research object.\textsuperscript{27} Finally, according to the generated reference trajectory, a robust controller designed based on Lyapunov stability theory and radial basis function neural network (RBFNN)\textsuperscript{28,29} is used for the fixed-wing UAV tracking the reference trajectory.

Our main contributions are summarized as follows:

- In this paper, the algorithm of setting collision nodes is used in the 3D workspace to model the UAV operation environment, which effectively improves the operation efficiency of the trajectory planning algorithm in the 3D workspace.
- The novel trajectory planning algorithm based on biogeography algorithm improves the migration and mutation operations compared with the original biogeography algorithm, making it more suitable for UAV trajectory planning.
- According to the trajectory generated by the improved BBO algorithm proposed in this paper, an interpolation algorithm based on time step is designed, which can convert the discrete points in the workspace to a relatively continuous trajectory, and can effectively realize the UAV’s accurate tracking of the reference trajectory after combining with the designed UAV control scheme.

The arrangement of this paper is made as follows: In section 2, the improved trajectory planning algorithm
based on BBO in the 3D workspace is presented. We give the mathematical model of the UAV in section 3. The control scheme of UAV with the planned trajectory is proposed in section 4. Section 5 presents the experimental details and shows the effectiveness of the improved BBO algorithm and the control scheme. We give a conclusion in section 6.

The improved trajectory planning algorithm based on BBO in three-dimensional environment

For ease of expression, the following notations are exploited in this paper:

1. \( \mathbb{R} \) denotes a real constant space, \( \mathbb{R}^n \) denotes an \( n \)-dimensional real vector space, \( \mathbb{R}^{m \times n} \) represents an \( m \times n \)-dimensional real matrix space.

2. \( \mathbb{C} \) represents the transpose operation of a matrix, \( ||\mathbb{C}|| \) represents the Euclidean norm of a matrix and can be expressed as \( \sqrt{tr(\mathbb{C}^T \mathbb{C})} \), and \( tr(\mathbb{C}) \) represents the trace of a matrix.

3. There exists a pseudo inverse in a non-full rank matrix \( \mathbb{M} \in \mathbb{R}^{m \times l} \), \( m \neq l \), such that \( MM^+ = I_m \).

BBO is a biology evolutionary algorithm using biogeographic dynamic models and ideas that describe the movement of species from habitat to habitat in an ecosystem and how species are created or lost. In BBO, each habitat resembles an island in the sea, is regarded as a possible solution and has a HSI to test the suitability and the priority of the habitat. Other aspects, for example, plant and animal diversity, rainfall capacity, etc., which characterize HSI are called SIVs.

In the following proposed BBO based trajectory optimizing process, the habitat is considered as the trajectory in the trajectory planning, a SIV of each habitat as a route node, the HIS of each habitat as the cost function.

The construction of the UAV’s workspace

The first step of the vehicle’s trajectory planning in three-dimensional space is to construct the vehicle’s work environment in a meaningful way. According to He et al. and Garrido et al., the grid-based workspace modeling algorithm is introduced to describe the vehicle’s workspace. As shown in Figures 1 and 2, this algorithm cuts the UAV’s workspace into \( n_x \times n_y \times n_z \) blocks (\( n_x, n_y, \) and \( n_z \) are the number of cutting layers on the \( x, y, \) and \( z \) axes, respectively), the vertexes of each block are selected as the trajectory node, and the nodes in obstacles are defined as collision nodes, which means that UAV cannot pass through them. On the contrary, UAV can plan a trajectory from start to end by constructing a list only containing collision-free nodes.

A trajectory generation algorithm based on the constructed workspace

Before operating BBO algorithm, some initial trajectories need to be generated randomly so that they can be used as initial habitats in BBO to facilitate the formation of optimal trajectories by immigration, mutation, and other operations.

As shown in Figure 3, according to the construction algorithm of UAV’s workspace described above, the node where the UAV is currently located can be marked as \( node(i) \), the \( i \)-th node in UAV’s constructed
workspace, \(i \in [1, 2, ..., n_x \times n_y \times n_z]\). Where \(n_x, n_y, n_z\) are the number of UAV’s workspace cutting layers on \(x\)-axis, \(y\)-axis, and \(z\)-axis in the inertial coordinate frame, respectively. The adjacent nodes on its \(x\), \(y\), and \(z\) axes can be marked as \(\text{node}(i - 1)\), \(\text{node}(i + 1)\), \(\text{node}(i - n_y)\), \(\text{node}(i + n_y)\), \(\text{node}(i - n_x \times n_y)\), and \(\text{node}(i + n_x \times n_y)\), respectively.

The steps of the trajectory generation algorithm are as follows:

Step 1. A trajectory node list \(\text{PL}\) is initialized as an empty list; the iteration times \(\text{IR}\) is initialized as zero, the maximum iteration times is set as \(\text{IR}_{\text{max}}\); the current node and goal node is initialized as \(\text{node}(i)\) and \(\text{node}(j)\), \(i, j \in [1, 2, ..., n_x \times n_y \times n_z]\), respectively; an available node set \(\text{VS}\) is initialized as an empty set.

Step 2. Firstly, \(\text{node}(i)\) is selected into \(\text{PL}\). Then, judging whether \(\text{node}(i)\) or \(\text{node}(x)\) is a collision node, if it is, return error information, else, obtain the three-dimensional position of \(\text{node}(i)\) and \(\text{node}(j)\), which is \((x_i, y_i, z_i)\) and \((x_j, y_j, z_j)\), respectively. Here, \(x\), \(y\), and \(z\) are \(x\)-axis position, \(y\)-axis position, and \(z\)-axis position in the inertial coordinate frame, respectively. If \(x_j > x_i\), \(\text{node}(i + 1)\) is selected into \(\text{VS}\), else if \(x_j < x_i\), \(\text{node}(i - 1)\) is selected into \(\text{VS}\), else, there is no node on the \(x\)-axis of \(\text{node}(i)\) can be selected. If \(y_j > y_i\), \(\text{node}(i + n_y)\) is selected into \(\text{VS}\), else if \(y_j < y_i\), \(\text{node}(i - n_y)\) is selected into \(\text{VS}\), else, there is no node on the \(y\)-axis of \(\text{node}(i)\) can be selected. If \(z_j > z_i\), \(\text{node}(i + n_x \times n_y)\) is selected into \(\text{VS}\), else if \(z_j < z_i\), \(\text{node}(i - n_x \times n_y)\) is selected into \(\text{VS}\), else, there is no node on the \(z\)-axis of \(\text{node}(i)\) can be selected.

Step 3. Firstly, the collision nodes in \(\text{VS}\) is eliminated; then, one node in \(\text{VS}\) is randomly selected as the next position of UAV; particularly, an adjacent node of \(\text{node}(i)\) that is not collision node is randomly selected as the next position of UAV, when there is no node in \(\text{VS}\). The randomly selected node is set as \(\text{node}(k)\), and \(\text{IR} = \text{IR} + 1\).

Step 4. To avoid duplicate nodes in \(\text{PL}\), \(\text{node}(i)\) is set as a collision node, then the selected node in Step 3 is set as the current node by letting \(i = k\). If three-dimensional position of the current node is equal to goal node or \(\text{IR} > \text{IR}_{\text{max}}\), save the trajectory list \(\text{PL}\) and return success information, else, skip to Step 2.

To illustrate the trajectory generation algorithm visually, all available trajectories generated from the current node to the goal node are shown in Figure 4.

**Cost function**

In the process of UAV trajectory planning, choosing an optimal trajectory is the ultimate goal. Usually, there is a cost function to measure whether the current trajectory is optimal. If the algorithm has a small value of cost function, then the computational complexity is small. Compared with other trajectories, the generated trajectory with the smallest value of cost function will be considered as the optimal trajectory. \(^5\)

The cost function is defined as follows:

\[
F_{\text{cost}} = N_{\text{number}} + P_{\text{length}} + P_{\text{smoothing}}
\]  

In the cost function, \(N_{\text{number}}\) represents the number of nodes in the trajectory’s node list.

The term related to the length of the planned trajectory is represented as follows:

\[
P_{\text{length}} = \sum_{i=1}^{N_{\text{max}}-1} \| \text{pos}(i) - \text{pos}(i + 1) \| 
\]
Where \( \text{pos}(i) = (x_i, y_i, z_i) \), \( i \in [1, 2, \ldots, N_{\text{number}}] \) is three-dimensional position of the \( \text{node}(i) \) in the trajectory node list.

The term related to the smoothness of the planned trajectory is represented as follows:

\[
P_{\text{smoothing}} = \frac{1}{N_{\text{number}} - 1} \sum_{i=1}^{N_{\text{number}}-1} \arccos(-q_i \cdot q_{i+1})
\]

Where \( q_i = (\text{pos}(i) - \text{pos}(i-1))/||\text{pos}(i) - \text{pos}(i-1)||, \) \( i \in [1, 2, \ldots, N_{\text{number}}] \).

### The migration and mutation operation

**Migration.** The \( i \)-th habitat \( PL_i \) (a trajectory node list of the \( i \)-th habitat) and \( j \)-th habitat \( PL_j \) (a trajectory node list of the \( j \)-th habitat) are selected for immigration and emigration operation respectively, \( i, j \in [1, 2, \ldots, N_p] \), where \( N_p \) is the number of habitats or trajectories. Before the migration operation, in order to improve the diversity of species in each habitat, if there is a same node \( \text{node}(k) \), \( k \in [1, 2, \ldots, n_k \times n_1 \times n_2] \), between \( PL_i \) and \( PL_j \), all nodes in \( PL_j \) after \( \text{node}(k) \) will be cleared. Then, \( \text{node}(m) \) is randomly selected from \( PL_j \) and use it as the start node to generate a new trajectory to the goal node by using the trajectory generation algorithm proposed in section 2.B. Similarly, \( \text{node}(p) \) is randomly selected from \( PL_i \) and use it as the start node to generate a new trajectory to the \( \text{node}(m) \). In the end, clear the \( \text{node}(p) \) and all subsequent nodes in \( PL_i \) and put the two new generated trajectories into \( PL_i \).

**Mutation.** The \( i \)-th habitat \( PL_i \) is selected for mutation operation. Firstly, a node \( \text{node}(m) \) that is not a collision code is randomly selected from the UAV’s workspace in Figure 2 and use it as the start node to generate a new trajectory to the goal node by using the trajectory generation algorithm, then, similarly, \( \text{node}(p) \) is randomly selected from \( PL_i \) and use it as the start node to generate a new trajectory to the \( \text{node}(m) \). In the end, clear the \( \text{node}(p) \) and all subsequent nodes in \( PL_i \) and put the two new generated trajectories into \( PL_i \).

### An improved BBO algorithm

Each habitat has its specific emigration rate \( \mu_k \) and immigration rate \( \lambda_k \) in BBO,\(^1,3,33\) which are the functions of the HSIs in habitats and can be represented as follows:

\[
\mu_k = E \cdot \frac{F_k}{F_{\text{maximum}}}, \lambda_k = I \left(1 - \frac{F_k}{F_{\text{maximum}}}ight)
\]

Where \( E \) and \( I \) denote the maximum value of the emigration and immigration rate, respectively. \( F_{\text{cost}}^{k} \) is the \( k \)-th habitat’s cost function, and \( F_{\text{maximum}}^{\text{cost}} = \max [F_{\text{cost}}^1, F_{\text{cost}}^2, \ldots, F_{\text{cost}}^{N_p}] \).

Now, the variable \( P_k \) denotes the probability of the habitat’s population migration, and can be updated as \( P_k = P_k + \hat{P}_k \), and according to Mo and Li\(^32\) and Huang et al.,\(^35\) the updating laws of the probability \( \hat{P}_k \) are defined as:

\[
\hat{P}_k = \begin{cases} 
-(\lambda_k + \mu_k)P_k + \mu_{k-1}P_{k-1}, & k = 0 \\
-(\lambda_k + \mu_k)P_k + \lambda_{k-1}P_{k-1} + \mu_{k-1}P_{k-1}, & 1 \leq k < N_p \\
-(\lambda_k + \mu_k)P_k + \lambda_{k-1}P_{k-1}, & k = N_p 
\end{cases}
\]

The mutation probability \( m_k \) for each habitat’s population is defined as:

\[
m_k = m_i(1 - \frac{P_k}{P_{\text{max}}})
\]

Where \( k \in [1, 2, \ldots, N_p] \), \( P_{\text{max}} = \max [P_1, P_2, \ldots, P_{N_p}] \).

The steps of the improved BBO algorithm are summarized as follows:

Step 1. The trajectory generation algorithm proposed in section 2.2 is used to randomly generate \( N_p \) trajectories, which are used as the initial habitats, between the start node and the goal node. These parameters are initialized: the max emigration rate \( E \), the max immigration rate \( I \), the probability \( P_k \), and the mutation probability \( m_k \) of each habitat. In the end, the iteration times \( NR \) of the improved BBO algorithm is initialized as zero, the maximum iteration times is set as \( NR_{\text{max}} \).

Step 2. The cost function \( F_{\text{cost}}^k \), the immigration \( \lambda_k \), and emigration rate \( \mu_k \), the population migration probability \( P_k \) and the mutation probability \( m_k \) of each habitat are calculated in this iteration process. The optimal trajectory, that is the trajectory with the lowest cost function, is saved.

---

**Figure 5.** The fixed-UAV in three-dimensional space.
Step 3. If \( \lambda_i \sim \text{rand}(0, d) \), \( \mu_j \sim \text{rand}(e, 1) \), \( 0 < d, e, \ i \neq j, i, j \in [1,2,...,N_p] \) are satisfied, the \( i \)-th habitat and the \( j \)-th habitat are selected for migration operation in section 2.D; especially, the \( i \)-th habitat is selected for the immigrant operation and the \( j \)-th habitat is selected for the emigrant operation; besides, the upper limit \( d \) and the lower limit \( e \) are selected to ensure that elites in habitats do not lose. Then, if \( m_i \sim \text{rand}(0,1) \), the \( i \)-th habitat can be selected for mutation operation, and \( N_R = N_R + 1 \).

Step 4. If \( N_R > N_{R_{\max}} \), save the optimal trajectory and return the success information, else, skip to Step 2.

The mathematical model of the fixed-wing UAV

Usually, the UAV’s motion in three-dimensional space needs to be transformed among three coordinate frames: body coordinate frame, inertial coordinate frame, and airflow coordinate frame, expressed in letters \( n, b, w \), respectively.\(^{36-38} \) The relationships between three coordinate frames are shown in Figure 5.

Firstly, some definitions are given as: \( P = (x, y, z)^T \) and \( \dot{P} = (\dot{x}, \dot{y}, \dot{z})^T \) represent the position vector and velocity vector in the inertial coordinate frame, respectively. \( E = (\varphi, \theta, \psi)^T \) and \( \dot{E} = (\varphi, \theta, \psi)^T \) are the attitude angle and attitude angular velocity vectors in the inertial coordinate frame, respectively. \( \varphi, \theta, \psi \) are the roll angle, pitch angle, and yaw angle in the inertial coordinate frame, respectively. \( V^b = (u, v, w)^T \) and \( \Omega^b = (p, q, r)^T \) are the linear velocity and attitude angle velocity vectors in the body coordinate frame, respectively. \( u, v, w \) are linear velocities in the body coordinate frame of \( x \)-axis, \( y \)-axis, and \( z \)-axis, respectively. The relationships between them are given as:

\[
V^b = R^b_n \dot{P}, \quad \Omega^b = C^b_n \dot{E}
\]  
(7)

The coordinate transform matrices \( R^b_n \) and \( C^b_n \) are given in Appendix 1. According to Newtonian mechanics\(^ {27,39} \), the UAV’s mathematical model can be defined as:

\[
\dot{V}^b = \left( G^b + F^b_{\text{engine}} + C^b_w F^w_{\text{aerodynamic}} \right) / d - \Omega^b \times V^b
\]  
(8)

\[
\dot{\Omega}^b = (\dot{P})^{-1}(M^b - \Omega^b \times (\dot{P} \cdot \Omega^b))
\]  
(9)

The coordinate transform matrix \( C^b_w \) is given in Appendix 1. The forces \( G^b \), \( F^b_{\text{engine}} \), and \( F^w_{\text{aerodynamic}} \) acting on the UAV are given in Appendix 2. The moment \( M^b \) and the inertial matrix \( P \) are given in Appendix 3.

Let \( u_v = (F_{u, v}, F_{u, v}, F_{u, w})^T \), \( u_0 = (\delta_v, \delta_s, \delta_1, \delta_2, \delta_3)^T \), \( F_{u, v}, F_{u, v}, F_{u, w} \) are the UAV’s engine force on \( x \)-axis, \( y \)-axis, and \( z \)-axis in the body coordinate frame, respectively.\( \delta_v \) is the controller of the left aileron, \( \delta_s \) is the controller of the right aileron, \( \delta_1 \) is the controller of the left elevator, \( \delta_2 \) is the controller of the right elevator, and \( \delta_3 \) is the controller of the rudder. The UAV’s mathematical model is simplified as:

\[
\dot{V}^b = AV^b + Bu_v + Q_v
\]  
(10)

\[
\dot{\Omega}^b = F\Omega^b + Gu_\Omega + Q_\Omega
\]  
(11)

The complete form of \( A, B, Q_v, F, G, \) and \( Q_\Omega \) are given in Appendix 4.

The external disturbances and model uncertainty issues are considered, equations (10) and (11) are rewritten as:

\[
\dot{V}^b = AV^b + Bu_v + Q_v + f_v
\]  
(12)

\[
\dot{\Omega}^b = F\Omega^b + Gu_\Omega + Q_\Omega + f_\Omega
\]  
(13)

Where \( f_v = (f_{\varphi}, f_{\theta}, f_{\psi})^T \) and \( f_\Omega = (f_{\varphi}, f_{\theta}, f_{\psi})^T \) are the unknown external disturbances or model uncertainties.

The control scheme of UAV with the planned trajectory

According to the trajectory optimization algorithm proposed in section 2.E, an optimal trajectory PL connecting the initial node and the goal node is obtained. In particular, this trajectory consists of a series of trajectory nodes.

The reference trajectory and its derivative required by UAV is as follows:

Step 1. The start node in PL is initialized as the current node \( \text{node}(i) \), the air speed of UAV is set as \( v_s \), and the time step is set as \( h \). The time \( t \) is initialized as zero. The time \( T \) is initialized as zero. According to equation (2), the total length of the trajectory PL is \( P_{\text{length}} \). The time the UAV spends on this trajectory is \( T_{\text{total}} = P_{\text{length}} / v_s \).

Step 2. The time it takes to fly from the current node \( \text{node}(i) \) to the following node \( \text{node}(i + 1) \) is designed as \( T_s = \text{norm}(\text{pos}(i + 1) - \text{pos}(i)) / v_s \), and \( \text{pos}(i + 1) = (x_{i + 1}, y_{i + 1}, z_{i + 1}) \) are three-dimensional positions of the node \( i \) and node \( i + 1 \) respectively. Let \( x_s = 0, y_s = 0, z_s = 0 \), while \( T_s < T + T_s \), do the following:

\[
\dot{x}(t) = (x_{i + 1} - x_s) / T_s, \quad \dot{y}(t) = (y_{i + 1} - y_s) / T_s,
\]

\[
\dot{z}(t) = (z_{i + 1} - z_s) / T_s
\]

\[
x(t) = x_s + h \times \dot{x}(t), \quad y(t) = y_s + h \times \dot{y}(t), \quad z(t) = z_s + h \times \dot{z}(t)
\]

\[
x_s = x(i + 1) + h \times \dot{x}(t), \quad y_s = y(i + 1) + h \times \dot{y}(t), \quad z_s = z(i + 1) + h \times \dot{z}(t)
\]
Step 3. The time $T$ is updated by $T = T + T$, the current node is updated by letting $i = i + 1$, that is transferred from the $i$-th node to the $i + 1$-th node in PL. If node(i) is equal to the goal node in PL, return the reference trajectory $x_i(t)$, $y_i(t)$, $z_i(t)$ and its derivatives $\dot{x}_i(t)$, $\dot{y}_i(t)$, $\dot{z}_i(t)$ between $t = 0$ and $t = T_{all}$, else, skip to Step 2.

Lemma 1\(^{40,42}\): Let $R(t) \in R$ be a continuous positive function with a bounded initial $R(0)$. If the inequality holds $R(t) \leq \beta R(t) + \eta$, where $\beta$ and $\eta$ are positive constants, then the following inequality is obtained:

$$R(t) \leq R(0)e^{-\beta t} + \frac{\eta}{\beta}(1 - e^{-\beta t})$$

Lemma 2\(^{28,42}\): Let $(x(t), y(t)) \in R^m$ are bounded functions with continuous values, the following formula can be obtained by Young’s inequality and Cauchy inequality:

$$x(t)y(t) \leq \|x(t)\| \|y(t)\| \leq \frac{\|x(t)\|^2}{\lambda} + \frac{\|y(t)\|^2}{\gamma}, \lambda, \gamma > 0, \frac{1}{\lambda} + \frac{1}{\gamma} = 1$$

The control scheme for UAV’s attitude and position is designed as follows:

Let $P_r = (x_r, y_r, z_r)^T$, $\dot{P}_r = (\dot{x}_r, \dot{y}_r, \dot{z}_r)^T$, $V_r = R_{b}^\perp \dot{P}_r$, and $P = (R_{b}^\perp)^{-1}V_r$. The error dynamic between the agent and the reference is represented as:

$$e = P - P_r + V_r - V_r^b$$

(14)

A Lyapunov function is given as follows:

$$V_1 = \frac{1}{2} e^T e$$

(15)

According to equation (12), the derivative of $V_1$ is obtained:

$$\dot{V}_1 = e^T e = e^T (A V_r + B e + Q_r e + f_r + f_\delta)$$

(16)

Where $f_\delta = P - P_r - V_r^b$ is a smooth function with bounded area, and satisfies $\|f_\delta\| \leq a$. Here, $a$ is a positive value. According to Lemma 2, the following one is obtained:

$$e^T f_\delta \leq a e^T \leq \frac{a^2}{2} + \frac{\|e\|^2}{2}$$

(17)

Use equation (17), equation (16) is rewritten as:

$$\dot{V}_1 \leq e^T A V_r + e^T B e + e^T Q_r e + e^T f_r + \frac{e^T e}{2} + \frac{a^2}{2}$$

(18)

To complete the UAV’s task, the desired controller $u_e$ is designed as:

$$u_e = -k e - B_v^\perp (A V_r^b + Q_r + f_r)$$

(19)

Let $k = \gamma(1/2 + 1/2\gamma)B_v^\perp$, $\gamma > 0$, and substitute it into equation (16), yields:

$$\dot{V}_1 \leq -\gamma V_1 + \frac{a^2}{2\gamma}$$

(20)

Use Lemma 1, one can obtain that:

$$V_1(t) \leq \frac{a^2}{2\gamma} + (V_1(0) - \frac{a^2}{2\gamma})e^{-\gamma t}$$

(21)

$f_\delta$ is unknown, thus the controller in equation (19) cannot be introduced to equation (12) immediately. According to Luan et al.\(^{41}\) and Tang et al.\(^{42}\) the RBFNN is often used to estimate the unknown function.

$$f_\delta = w^T \eta(z) + e(z), \dot{\hat{w}} = \hat{w}^T \eta(z)$$

(22)

where $w^* \in R^N$ is the optimal weight, $\eta(z)$ is an estimated error, and $\|e(z)\| \leq \delta$, $\hat{w} \in R^N$ is used to estimate the optimal weight, and the error between them can be defined as $\hat{w} = w - w^*$. $\eta(z) = \exp(-\|z - \xi_i\|^2/\zeta_i^2)$ and $\eta(z) = [\eta_1(z), ..., \eta_N(z)]^T \in R^N$. $\xi_i$ is the center in the receptive field and $\zeta_i$ is the width of the receptive field. $z = e \in R^m$ is the input unknown disturbance.

Then, a redesigned Lyapunov function is generated as:

$$V = V_1 + \frac{1}{2\eta} tr(\hat{w}^T \hat{w})$$

(23)

By combining equations (12), (17), and (22), the derivative of equation (23) is obtained as:

$$\dot{V} \leq e^T (A V_r + B e + Q_r + w^* \eta(z) + e(z)) + \frac{e^T e}{2} + \frac{a^2}{2} + \frac{1}{\eta} tr(\hat{w}^T \hat{w})$$

(24)

According to Lemma 2, the following one is obtained:

$$e^T e(z) \leq \delta e^T \leq \frac{\delta^2}{2} + \frac{\|e\|^2}{2}$$

(25)

Use equation (25), equation (24) is rewritten as:

$$\dot{V} \leq e^T (A V_r + B e + Q_r + w^* \eta(z)) + \frac{e^T e}{2} + \frac{\delta^2}{2} + \frac{1}{\eta} tr(\hat{w}^T \hat{w})$$

(26)

To finish the control task, the controller is redesigned as:
\[
\begin{align*}
    u_v &= -ke - B^+(AV^h + Q_v + \hat{w}^T \eta(z)) \\
    \text{Where } k &= \gamma(1/2 + 1/\gamma)B^+, \ \gamma > 0, \text{ and the weight adaptive law of the neural network is designed as:} \\
    \dot{\hat{w}} &= \eta(\eta(z)e^T - \sigma\hat{w}) \\
    \text{Substitute equations (27) and (28) into equation (26), yields:} \\
    \dot{V} &= -\frac{\gamma}{2}e^Te - \sigma\eta \frac{1}{2\eta}tr(\hat{w}^T \hat{w}) + \frac{1}{2}\sigma tr(w^Tw^*) \\
    &+ \frac{\sigma^2 + \delta^2}{2} \\
    &\leq -\beta V + \eta \\
    \text{Where } \beta &= \min[\gamma, \sigma\eta], \eta = \frac{1}{2}\sigma tr(w^Tw^*) + \frac{\sigma^2 + \delta^2}{2}. \\
    \text{Use Lemma 1, the following one can be obtained:} \\
    V(t) &\leq \frac{\eta}{\beta} + (V(0) - \frac{\eta}{\beta})e^{-\beta t} \\
    \text{From it one can conclude that the trajectory tracking error can become very small in the situation of selecting the suitable parameters and choosing RBFNN to estimate the unknown disturbance.} \\
    \text{Similarly, the controller } u_\Omega \text{ in equation (13) is designed as:} \\
    u_\Omega &= -ke_\Omega - G^+(F\Omega^h + Q_\Omega + \hat{w}^T \eta(z)) \\
    \text{Where } k &= \gamma(1/2 + 1/\gamma)G^+, \ \hat{w} = \eta(\eta(z)e^T - \sigma\hat{w}), \ \text{and } e_\Omega = E - E_r + \Omega^h - \Omega^\Omega. \ \text{The elements of the reference attitude vector } E_r = (\phi_r, \theta_r, \psi_r)^T \text{ related to the inertial coordinate frame are follows:} \\
    &\phi_r = \arcsin(u/V_T), \ \theta_r = \arcsin(w/V_T), \ \text{and } \psi_r = \arcsin(v/V_T). \\
    \text{To verify the RBFNN’s performance of estimating the external disturbance, the contrasted controllers without RBFNN of equations (27) and (34) are set as:} \\
    u_v &= -ke - B^+(AV^h + Q_v + \hat{w}^T \eta(z)) \\
    u_\Omega &= -ke_\Omega - G^+(F\Omega^h + Q_\Omega) \\
\end{align*}
\]
can be concluded that the improved BBO based trajectory planning algorithm has good performance, and the trajectories planned by these algorithms in three-dimensional workspace are shown in Figure 6.

As shown in Figures 7 and 8, the improved BBO based trajectory planning algorithm is used to find an optimal trajectory connecting the start node and the goal node in the complex working environment. The trajectory generated by the improved BBO is selected as the reference trajectory. From the two-dimensional plane and the three-dimensional plane, it can be seen that the actual flight trajectory of the UAV almost coincides with the reference trajectory. From Figure 9, it can be seen that the control scheme without RBFNN is bad in following the reference trajectory. The reason why the proposed control scheme has good tracking effect is that it can better compensate for the compound interference terms and eliminate the influence of various interferences in the flight process.

The trajectory tracking effects of different methods can be seen from Figures 10 to 15. By using the proposed control scheme, the UAV can track the trajectory quite well in three different situations. The position deviation and the attitude deviation are small in all
tracking process. The algorithm accuracies of positions generated by WLIPSO, CMM-BBO, and improved BBO are 96.30%, 96.86%, and 96.48%, respectively. The average algorithm accuracy is 96.55%. The electricity consumption is a crucial problem for UAV tracking. The time consumption of UAV in tracking...
the trajectories generated by WLIPSO, CMM-BBO, and improved BBO are 20.40, 18.80, and 18.20 s, respectively. Among the three algorithms, the trajectory length and time consumption of the proposed algorithm is the lowest, and the electricity consumption of it is the lowest.

To evaluate the good tracking performance of the designed controller, controllers with and without RFBNN are compared. The tracking effects of positions and attitudes in three directions are illustrated from Figures 10 to 15. The controllers with neural network proposed in equations (27) and (34) are used to track the different trajectories, the control effects are better than controllers without neural network proposed in equations (35) and (36). That is because the neural network used in this paper can accurately estimate the unknown external disturbance. From Figures 16 to 17, it can be concluded that the estimated values are close to the actual values, the neural network has good estimation ability.

**Conclusions**

This paper proposes an off-line trajectory planning algorithm and a robust trajectory tracking control scheme for fixed-wing UAV. In this paper, an improved trajectory planning algorithm based on BBO is proposed to generate an optimal trajectory connecting the start node and the goal node for obstacle collision avoidance. By comparing with WLIPSO and CMM-BBO algorithm, it is concluded that this algorithm has certain advantages in UAV trajectory planning. A robust controller with RBFNN to estimate unknown disturbances is proposed to track the trajectory generated by three algorithms. Compared with other two algorithms, the improved BBO algorithm and the designed control scheme generate the optimal trajectory in the subjective and objective aspects. In the following work, the reduction of the algorithm complexity will be considered.

**Declaration of conflicting interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.
Funding
The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported in part by Sichuan Science and Technology Program under Grant 2017GZ0303, in part by Fund Project of Sichuan Provincial Academician (Expert) Workstation under Grant 2016YSGZ01, and in part by Special Fund for Training High Level Innovative Talents of Sichuan University of Science and Engineering under Grant B12402005.

ORCID IDs
Zhishuang Xue https://orcid.org/0000-0002-2074-5216
Xiaofang Liu https://orcid.org/0000-0001-7644-5979

References
1. Yan F, Zhu X, Zhou Z, et al. Heterogeneous multi-unmanned aerial vehicle task planning: simultaneous attacks on targets using the Pythagorean hodograph curve. Proc IMechE, Part G: J Aerospace Engineering 2019; 233: 4735–4749.
2. Radmanesh M, Kumar M, Nemati A, et al. Dynamic optimal UAV trajectory planning in the national airspace system via mixed integer linear programming. Proc IMechE, Part G: J Aerospace Engineering 2016; 230: 1168–1682.
3. Guo W, Li W, Zhang Q, et al. Biogeography-based particle swarm optimization with fuzzy elitism and its applications to constrained engineering problems. Eng Optim 2014; 46: 1465–1484.
4. Kim S, Silson P, T sourdos A, et al. Dubins trajectory planning of multiple unmanned airborne vehicles for communication relay. Proc IMechE, Part G: J Aerospace Engineering 2011; 225: 12–25.
5. Zhang B and Duan H. Three-dimensional trajectory planning for uninhabited combat aerial vehicle based on predator-prey pigeon-inspired optimization in dynamic environment. IEEE/ACM Trans Comput Biol Bioinform 2015; 14: 97–107.
6. Aydogdu I. Cost optimization of reinforced concrete cantilever retaining walls under seismic loading using a biogeography-based optimization algorithm with Levy flights. Eng Optim 2017; 49: 381–400.
7. Samaniego F, Sanchis J, García-Nieto S, et al. Recursive Rewarding Modified Adaptive Cell Decomposition (RR-MACD): a dynamic path planning algorithm for UAVs. Electronics 2019; 8: 306–327.
8. Lupascu M, Hustiu S, Burlacu A, et al. Path planning for autonomous drones using 3D rectangular cuboid decomposition. In: 2019 23rd International conference on system theory, control and computing (ICSTCC), Sinaia, Romania, 9–11 October 2019, pp.119–124. IEEE.
9. Contreras-Cruz MA, Ayala-Ramírez V and Hernandez-Belmonte UH. Mobile robot path planning using artificial bee colony and evolutionary programming. Appl Soft Comput 2015; 30: 319–328.
10. Li B, Gong L and Yang W. An improved artificial bee colony algorithm based on balance-evolution strategy for unmanned combat aerial vehicle path planning. ScientificWorldJournal 2014; 14: 1–10.
11. Lu X, Zhang X, Jia S, et al. Design of quadrotor hovering controller based on improved particle swarm optimization. In: 2017 10th International symposium on computational intelligence and design (ISCID), Hangzhou, China, 9–10 December 2017, pp.1–4. IEEE.
12. Chen X, Tianfield H, Du W, et al. Biogeography-based optimization with covariance matrix based migration. Appl Soft Comput 2016; 45: 71–85.
13. Simon D. Biogeography-based optimization. IEEE Trans Evol Comput 2008; 12: 702–713.
14. Guo W, Chen M, Wang L, et al. A survey of biogeography-based optimization. Neural Comput Appl 2017; 28: 1909–1926.
15. Chanda CK, Debлина M and Sumit B. Economic dispatch solution for cogeneration unit assisted by biogeography-based optimization algorithm in competitive environment. In: 2017 Australasian universities power engineering conference (AUPEC), Melbourne, VIC, 19–22 November 2017, pp.1–5. IEEE.
16. Yang G, Zhang Y, Yang J, et al. Automated classification of brain images using wavelet-energy and biogeography-based optimization. Multimed Tools Appl 2016; 75: 15601–15617.
17. Wu X, Wang H, Chen Z, et al. Biogeography-based optimization for cluster analysis. In: Bhutia S, Mishra K, Tiwari S, et al. (eds) Advances in computer and computational sciences. Singapore: Springer, 2017, pp.3–12.
18. Zhang J, Yong Z and Yong Z. Trajectory planning of mobile robot based on hybrid multi-objective bare bones particle swarm optimization with differential evolution. IEEE Access 2018; 6: 44542–44555.
19. Islam MN, Gallardo-Alvarado J, bin Hj Ahmad O, et al. Trajectory planning algorithm for manipulators based on an improved backtracking search algorithm. In: 2015 5th International conference on IT convergence and security (ICITCS), Kuala Lumpur, Malaysia, 24–27 August 2015, pp.1–5. IEEE.
20. He Y, Zeng Q, Liu J, et al. Trajectory planning for indoor UAV based on Ant Colony Optimization. In: 2014 25th Chinese control and decision conference (CCDC), Guiyang, China, 25–27 May 2013, pp.2919–2923. IEEE.
21. Lin Y and Srikanth S. Sampling-based trajectory planning for UAV collision avoidance. IEEE Trans Intell Transp Syst 2017; 18: 3179–3192.
22. Sanchez-Lopez JL, Wang M, Olivares-Mendez MA, et al. A real-time 3D trajectory planning solution for collision-free navigation of multirotor aerial robots in dynamic environments. J Intell Robot Syst 2019; 93: 33–53.
23. Dehghani H and Mahmoudi Z. Crude oil price forecasting: a biogeography-based optimization approach. Energy Sour B Econ Plann Policy 2018; 13: 328–339.
24. Zhang X, Kang Q, Tu Q, et al. Efficient and merged biogeography-based optimization algorithm for global optimization problems. Soft Comput 2019; 23: 4483–4502.
25. Aljarah I, Faris H, Mirjalili S, et al. Training radial basis function networks using biogeography-based optimizer. Neural Comput Appl 2018; 29: 529–533.
Xue and Liu

Appendix

Appendix 1

The coordinate transform matrices $R_n^b$ and $C_n^b$ are exploited to transform the linear velocities and angular velocities from the inertial coordinate frame to the body coordinate frame, respectively. The coordinate transform matrix $C_w^b$ is introduced to transform the aerodynamic forces from the airflow coordinate frame to the body coordinate frame. They are defined as follows:

$$R_n^b = \begin{bmatrix} \cos\theta \cos\phi & \cos\theta \sin\phi & -\sin\theta \\ 
\sin\theta \cos\phi \sin\phi - \sin\phi \cos\phi & \sin\theta \sin\phi \sin\phi + \cos\phi \cos\phi & \cos\theta \cos\phi \\ 
\sin\theta \cos\phi \cos\phi + \sin\phi \sin\phi & \sin\theta \sin\phi \cos\phi - \cos\phi \sin\phi & \cos\theta \sin\phi \end{bmatrix},$$

$$C_n^b = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi \cos\theta \\ 0 & -\sin\phi & \cos\phi \cos\theta \end{bmatrix},$$

$$C_w^b = \begin{bmatrix} \cos\alpha \cos\beta & -\cos\alpha \sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha \cos\beta & -\sin\alpha \sin\beta & \cos\alpha \end{bmatrix}.$$
Appendix 2

The gravity $G^b$ and thrust $F_{\text{engine}}$ of the UAV in body coordinate frame and the aerodynamic force $F_{\text{aerodynamic}}$ in airflow coordinate frame are defined as follows:

\[
G^b = \begin{bmatrix}
-g \sin \theta \\
g \sin \varphi \cos \theta \\
g \cos \varphi \cos \theta
\end{bmatrix}, \quad F_{\text{engine}} = \begin{bmatrix}
F_u \\
F_v \\
F_w
\end{bmatrix}, \quad F_{\text{aerodynamic}} = \begin{bmatrix}
\bar{q} \hat{S}(C_{X1} + C_{X2} \alpha + C_{X3} \alpha^2 + C_{X4} \beta^2) \\
\bar{q} \hat{S} C_{Y1} \beta \\
\bar{q} \hat{S} C_{Z1} + C_{Z2} \alpha
\end{bmatrix}
\]

Where $\bar{q} = \rho V_T^2/2$, $g$ is the acceleration of the earth’s gravity, $F_u$, $F_v$, and $F_w$ are the thrust forces of UAV on body coordinate system. The rest coefficients can be seen in Appendix 5.

Appendix 3

The moment $M^b$ and the inertial matrix $I^b$ in body coordinate frame are defined as:

\[
M^b = \begin{bmatrix}
\bar{q} \hat{S} \bar{b} (C_{X1} \delta a_1 + C_{X2} \delta a_2 + C_{X3} \delta c_1 + C_{X4} \delta c_2 + C_{X5} \beta + C_{X6} \beta + C_{X7}) \\
\bar{q} \hat{S} \bar{c} (C_{M1} + C_{M2} \delta a_1 + C_{M3} \delta a_2 + C_{M4} \delta b_1 + C_{M5} \delta b_2 + C_{M6} \delta c_1 + C_{M7} \delta c_2 + C_{M8} \beta + C_{M9} \beta + C_{M10}) \\
\bar{q} \hat{S} B (C_{N1} \delta r + C_{N2} \delta r + C_{N3} \delta r + C_{N4} \beta)
\end{bmatrix}, \quad I^b = \begin{bmatrix}
I_{xx} \\
I_{yy} \\
I_{zz}
\end{bmatrix}
\]

Where $\bar{q} = \rho V_T^2/2$, $\bar{b} = \beta a_2/2 V_T$, $\bar{c} = \bar{c} q/2 V_T$, $\bar{r} = b r/2 V_T$, $\bar{c}$ denotes UAV’s mean aerodynamic chord and $b$ denotes UAV’s wingspan. The rest coefficients can be seen in Appendix 5.

Appendix 4

The complete form of $A$, $B$, $Q_v$, $F$, $G$, and $Q_\Omega$ are given as follows:

\[
A = \begin{bmatrix}
0 & r & -q \\
0 & p & q \\
q & -p & 0
\end{bmatrix}, \quad B = \begin{bmatrix}
1/m \\
0 \\
0
\end{bmatrix}, \quad Q_v = \frac{C_w F_{\text{aerodynamic}} + G^b}{m}, \quad F = \begin{bmatrix}
\frac{I_{xx} S_{\beta} (I_{yy} - I_{zz} - 2 I_{xy}) - N_1 I_{xx} - N_2 I_{yy} - N_3 I_{zz}}{I_{yy} I_{zz} - I_{xy}^2} \\
\frac{I_{yy} S_{\alpha} (I_{xx} - I_{zz} - 2 I_{yy}) - N_1 I_{yy} - N_2 I_{xx} - N_3 I_{zz}}{I_{xx} I_{zz} - I_{xy}^2} \\
\frac{I_{zz} S_{\alpha} (I_{xx} - I_{yy} - 2 I_{xy}) - N_1 I_{zz} - N_2 I_{xx} - N_3 I_{yy}}{I_{xx} I_{yy} - I_{xy}^2}
\end{bmatrix}, \quad G = \bar{q}
\]

\[
Q_\Omega = \begin{bmatrix}
S_{\beta} \bar{q} (I_{xx} C_{\beta} - I_{yy} C_{\beta}) \\
S_{\alpha} \bar{q} (I_{yy} C_{\alpha} - I_{zz} C_{\alpha})
\end{bmatrix}, \quad S_{\alpha} \bar{q} C_{\alpha} = \frac{S_{\alpha} \bar{q} C_{\alpha}}{I_{xx} I_{yy} - I_{xy}^2}, \quad \bar{q} = \rho V_T^2/2
\]

Where $N_1 = I_{xx} (I_{xx} - I_{yy} - I_{zz})$, $N_2 = I_{yy} (I_{yy} - I_{xx} - I_{zz})$, $N_3 = I_{zz} (I_{zz} - I_{xx} - I_{yy})$. The rest coefficients can be seen in Appendix 5.

Appendix 5

The parameters needed to construct the UAV mathematical model are given as follows.

| Term | Value | Unit | Term | Value | Unit | Term | Value | Unit | Term | Value | Unit |
|------|-------|------|------|-------|------|------|-------|------|------|-------|------|
| $I_{xx}$ | 2.56 | kg⋅m$^2$ | $C_{\alpha\alpha}$ | $-0.091$ | / | $\rho$ | 1.29 | kg/m$^3$ | $C_G$ | 0.0361 | / |
| $I_{yy}$ | 10.9 | kg⋅m$^2$ | $C_{\alpha\beta}$ | $-0.013$ | / | $S$ | 0.8 | m$^2$ | $C_{\alpha\delta}$ | $-0.21$ | / |
| $I_{zz}$ | 11.3 | kg⋅m$^2$ | $C_{\beta\beta}$ | 0.087 | / | $b$ | 1.8 | m | $C_{\beta\gamma}$ | $-0.033$ | / |
| $I_{xx}$ | 0.5 | kg⋅m$^2$ | $m$ | 28 | kg | $\bar{c}$ | 0.58 | m | $C_{\alpha\gamma}$ | $-0.192$ | / |
| $I_{yy}$ | 0.5 | kg⋅m$^2$ | $g$ | 9.8 | m/s$^2$ | $C_{\beta\delta}$ | $-0.027$ | / | $C_{\beta\delta}$ | $-1.55$ | / |
| $C_{\alpha\alpha}$ | 0.0534 | / | $C_{\alpha\beta}$ | $-3.25$ | / | $C_{\alpha\gamma}$ | $-3.25$ | / | $C_{\alpha\delta}$ | $-0.38$ | / |
| $C_{\lambda\mu}$ | $-0.005$ | / | $C_{\alpha\mu}$ | 0.013 | / | $C_{\lambda\nu}$ | $-0.005$ | / | $C_{\mu\nu}$ | 0.038 | / |
| $C_{\alpha\alpha}$ | $-3.25$ | / | $C_{\beta\beta}$ | $C_{\beta\beta}$ | / | $C_{\alpha\gamma}$ | $C_{\alpha\gamma}$ | / | $C_{\beta\delta}$ | $C_{\beta\delta}$ | / |
| $C_{\alpha\gamma}$ | $-0.401$ | / | $C_{\alpha\delta}$ | $0.021$ | / | $C_{\gamma\delta}$ | $-0.401$ | / | $C_{\gamma\delta}$ | $0.2725$ | / |
Appendix 6

The parameters needed for the design of UAV’s control scheme are given as follows.

| Term | Value | Unit | Term | Value | Unit | Term | Value | Unit | Term | Value | Unit |
|------|-------|------|------|-------|------|------|-------|------|------|-------|------|
| $h$  | 0.01  | s    | $v_s$| 5     | m/s  | $\gamma$| 5     | /    | $\eta$| 0.5   | /    |
| $\sigma$ | 0.01 | /    |      |       |      |       |       |      |       |       |      |

Appendix 7

The parameters needed for the proposed BBO based trajectory planning algorithm are given as follows.

| Term | Value | Unit | Term | Value | Unit | Term | Value | Unit | Term | Value | Unit |
|------|-------|------|------|-------|------|------|-------|------|------|-------|------|
| $n_x$| 20    | /    | $n_y$| 50    | /    | $n_z$| 10    | /    | $IR_{\max}$ | 200  | /    |
| $E$  | 1     | /    | $l$  | 1     | /    | $N_p$| 25    | /    | $d$  | 0.9   | /    |
| $e$  | 0.1   | /    | $NR_{\max}$ | 40   |       |       |       |      |       |       |      |