29th Congress of the International Council of the Aeronautical Sciences
St. Petersburg, Russia; September 7-12, 2014

A REINFORCEMENT LEARNING BASED UAVS AIR COLLISION AVOIDANCE

Jian Yang, Dong Yin, Haibin Xie
Department of Mechatronics and Automation, National University of Defense Technology, Changsha, China
tekesixinjiang@163.com; yindong@nudt.edu.cn;xhb2575_sx@sina.com

Keywords: *reinforcement learning, conflict resolution, model-based planning, optimal policy search*

Abstract

In this paper, we propose to deal with the UAV airspace conflict resolution problem. We propose to search near optimal conflict free policies in virtue of the model-based reinforcement learning. We first analyze the UAV airspace conflict problem and the basic conditions in ensuring collision-free planning, and then discuss the features that effect the optimal action. We then propose the reinforcement learning based conflict resolution algorithm. In the model-based learning structure, we consider the simplified dynamics of the UAVS in the model, and employ the heuristic method to estimate the state-action value. In the multi-dimension, continuous space, the optimal policy search method is utilized to find the near optimal policy. The experience from the real environment is used to criticize the model-based learning policy. In the end, we apply simulation experiments to demonstrate the proposed algorithm.

1 Introduction

With the rapid increase of UAV applications, there will be a great number of UAVs in the air in the near future. The UAV airspace management problem is becoming more and more crucial in terms of airspace safety[1]. The primary problem for UAV airspace management is airspace conflict resolution[2]. In this paper, we propose to study the UAV conflict resolution problem in virtue of model-based reinforcement learning.

Since there are no pilots in the cabins, additional measures should be taken to ensure the safety of the UAVs. As there will be plenty of UAVs in the airspace, the global centralized management is impractical. In this paper, it is proposed to search for the near optimal policy for the conflict resolution problem by local centralized coordination. In the local airspace, the involved planes would be limited, which would facilitate the processing. However, the local perspective may also induce some problems such as the horizontal restriction problem.

In the reinforcement learning, the agents often know nothing about the environment, and they learn the environment by trial and error. It will be inefficient in dynamic environment. The conflict resolution scenery includes more than one UAV in the environment, which means the state space is multi-dimensional and continuous. It is impractical to apply the trial and error training method in the whole space. On the other hand, in the real environment, the experiences acquired by interacting with the environment are limited. So we propose the model-based reinforcement learning method. The Dyna architecture can improve the learning efficiency of real knowledge. The architecture integrates the model learning, planning and direct learning. We build the model for participant UAVs and try to search a near optimal policy by model learning, and then modify the control parameters by learning the real experiences.

In continuous space, the tabular method cannot store all the states. We use the approximation method to estimate the approximate state-action value. To find the optimal policy, we consider several essential features about the state-action value function, namely safety, planned routes, and cost. In addition, as different UAVs are differently prioritized, the costs of route change are also different. All these factors will be taken into account for the near optimal policy.

In the Dyna structure, the learning efficiency depends on the learning method and the model itself. In this paper we use the approximate policy search method. The dynamic constraints are considered to ensure the approximation of the real environment. We establish the airspace conflict model based on the real circumstance, and plan the optimal policy by interacting with this model. In addition, the optimal policy search algorithm is designed in the model based learning.

The remainder of this paper is organized as follows. Section 2 scrutinizes related works; the model based reinforcement learning conflict resolution algorithm is presented in Section 3. In Section 4 we demonstrate the algorithm by experiments; and conclusions on our works are presented in section 5.

2 Related works

2.1 Study on conflict resolution

The literature that deals with the conflict detection and resolution problem is rapidly growing [3] in volume. The hybrid fuzzy potential field method is proposed for autonomous mobile robot motion planning with dynamic environments. It overcomes the local minimum problem by setting fuzzy rules and using Adaptive Neuron Fuzzy Inference System (ANFIS) [4]. An approach for navigation and collision detection based on the kinematic equations is introduced in [5]. This approach employs the notion of collision cones (CCs). The concept of velocity obstacles is introduced in [6], which takes the velocity of moving obstacles into account. By using the local observer, the method in [7]constructs the virtual plane, which is an invertible transformation equivalent to the workspace. The conflictions detection process is performed based on this virtual plane.

In the Reciprocal collision avoidance method, both robots are assumed to select a velocity outside the RVO induced by the other robot [8]. Each robot takes half of the responsibility for collision avoidance.

The reinforcement learning is an unsupervised learning method. There are many classical reinforcement learning methods, e.g. Q-learning, TD learning, and Sarsa-learning [9]. Numerous researches have been done in path planning and conflict resolution by integrating the reinforcement learning [10][11][12][13]. In paper [10], the Cell-mapping method is integrated with reinforcement learning to find the optimal path. It uses the online learning method to improve the control policy. In paper [11], the potential method and reinforcement learning method are integrated to generate an optimal maneuver policy to avoid the obstacle. The Dyna-Q based method is presented for navigation problem in unknown environments [14].

2.2 Model based reinforcement learning

Sutton proposes the architecture of Dyna-Q learning to improve the efficiency of learning, which is model based learning. In the following researches, the dyna-H(heuristic) algorithm is proposed by Matilde Santos et.al[15], and a version of Dyna based on approximations in the form of dynamic Bayes networks and decision trees is developed by Sigaud and Wuillemin (2006). Paper [16] studies the Dyna architecture. In the UAVs conflict resolution problem, the UAVs are in a dynamic environment. In the real environment the safety of each UAV should be guaranteed, and it is impractical to learn the optimal policy by the trial and error method. In this paper the model-based reinforcement learning method is proposed. At every step, the system searches the neat optimal policy by model based learning. The most frequently used model-based learning architecture is Dyna architecture, as shown in Fig 1.

As illustrated in Fig 1, the Dyna structure combines direct learning, model learning and planning together.

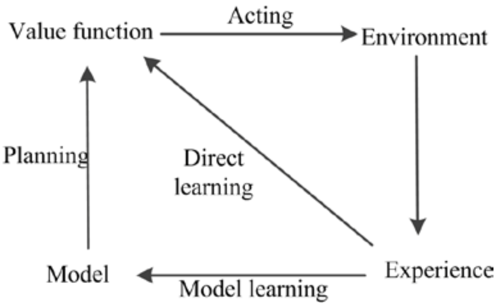


Fig 1 Dyna learning model

In case of any airspace conflicts, the system first builds the conflict model which concern on the states and performances of involved UAVs, and then the model begins to learn the optimal policy h_t for confliction resolution. By executing the policy h_t we have the next state s_{t+1} and get the reward r_{t+1}. The tuple $(s_t, a_t, s_{t+1}, r_{t+1})$ is the experience from the real environment, which can be learned by the system to modify the model. Furthermore, the experience can be used to update other state values.

Tab 1 Dyna-Q algorithm [9]

```
Algorithm 1：Dyna-Q algorithm
1: Initialize Q(s,a),Model(s,a) $\forall s \in S, a \in A$
2: repeat
3:     s←current(non terminal)state
4:     a← $\varepsilon - greedy(s,Q)$
5:     execute a; observe s and r
6:     $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
7:     $Model(s,a) \leftarrow s',r$
8:     for i = 1 to N do
9:         s←random previously observed state
10:        a←random action previously taken in s
11:        $s',r \leftarrow Model(s,a)$
12:        $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max Q(s',a') - Q(s,a)]$
13:    end for
14:until $s'$ is terminal
```

The Dyna-Q algorithm is shown as Tab 1, it shows the possible relationship between experience, model and values for Dyna-Q. In Algorithm 1, Model(s,a) denotes the contents of the model (predicted next state and reward, respectively) for state-action pair (s,a). Direct reinforcement learning, model-learning, and planning are implemented by steps 6, 7 and 8, respectively. The Dyna-Q learning is tabular-based learning method, which carry on model-based learning by recording and backward propagation. The Dyna-Q structure is a referable model, we consider the inner structure of the model when dealing with the practical problem. In this paper, we deal with the airspace conflict resolution problem by using heuristic search method[15]; which combines with domain specific knowledge, so we first focus on the theoretical analysis of UAV conflict resolution problem.

3 UAVs conflict resolution

3.1UAV airspace conflict analysis

In this paper, the coordination method is used to deal with multi-UAV conflict detection and resolution. The involved UAVs may get rid of the dangerous situation in a cooperative manner. Suppose that there is a ground station, which manages the territorial airspace traffic and surveillances the UAVs that are close to each other, when they are too close and may collide in the mid-air, the station will take coordination measures to eliminate the risks.

As shown in Fig 2, when one UAV is flying in the air, there should be a safe region around it based on its dynamic characteristics. In this space, any kind of invasion is extremely dangerous for the safety of the UAV. The region can be defined as:

$$D_i(r) = \{p \,|\, \| p_x - x_o, p_y - y_o \| < r_i, \left\| p_y - z_o \right\| < z_i\} \quad (1)$$

Although conflict resolution maneuvers contain attitude modifications and horizontal maneuvers, in this paper, we consider the conflict resolution by horizontal modification of the direction of UAVs velocities.

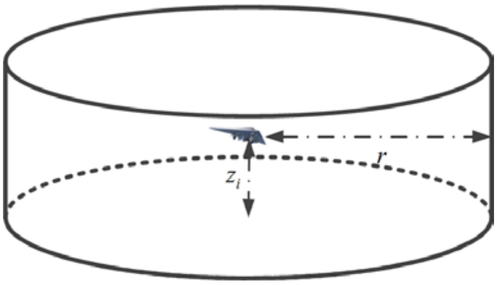
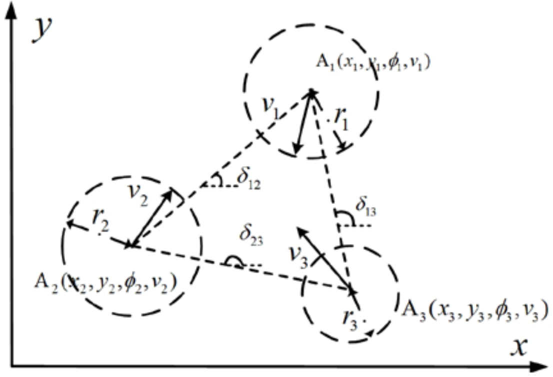


Fig 2 UAV safe region

We first describe the dynamic multi-UAV environment, as illustrated in Fig 3. The UAVs in Fig 3 move according to the following equations:

$$\begin{aligned} \dot{x}_i &= v_i \cos\phi_i \\ \dot{y}_i &= v_i \sin\phi_i \\ \dot{v}_i &= a_i \\ \dot{\phi}_i &= w_i \end{aligned} \tag{2}$$

Where ϕ_i is the motion direction of UAV A_i and v_i and w_i represent the linear and angular velocities respectively.



y
$A_1(x_1,y_1,\phi_1,v_1)$
v_1
r_1
δ_{12}
δ_{13}
v_2
r_2
δ_{23}
v_3
$A_2(x_2,y_2,\phi_2,v_2)$
r_3
$A_3(x_3,y_3,\phi_3,v_3)$
x

Fig 3 Multi-UAV environment

When there is a conflict between two UAVs, the crises is resolved by changing the magnitude or direction of the velocity. In the airspace, the change of speed may lead to unstability of the involved UAVs. In this paper, we mainly consider the direction modification of UAV velocity.

We study the conflict detection and resolution between different UAVs by using related velocity and position between UAVs, as shown in Fig 4

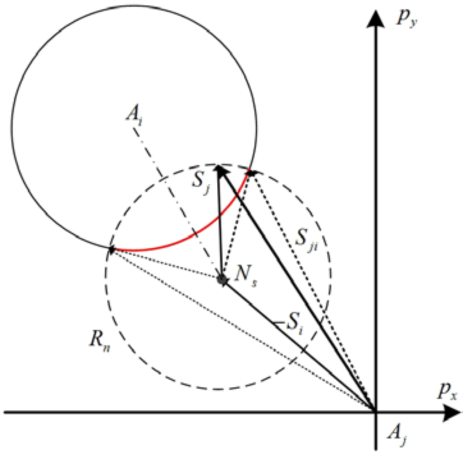


Fig 4 relative displacement and velocity

As shown in Fig 4, for the analysis on the relationship between UAVs A_i and A_j, we take the position of A_j as the original point in the local coordinates frame and suppose A_i is static. The relative position and orientation of A_i are:

$$P_{ij} = (x_i - x_j, y_i - y_j) \tag{3}$$

The velocity and velocity attitude of A_j relative to A_i are given as:

$$\vec{v}_{ji} = \vec{v}_j - \vec{v}_i = (\dot{x}_j - \dot{x}_i, \dot{y}_j - \dot{y}_i) \tag{4}$$

The related safe region can be expressed as:

$$\begin{gathered} D_{i|j}(r) = \{p \,||\, \| p_x - x_o, p_y - y_o \| < r_{\max}, \left\| p_y - z_o \right\| < z_{\max}\}, \\ r_{\max} = \max\{r_i, r_j\}, z_{\max} = \max\{z_i, z_j\} \end{gathered} \tag{5}$$

When A_j is supposed to enter the related safe region $D_{i|j}(r)$ in time window $(0-\tau)$, there is a potential conflict between these two participants.

3.2Conflict resolution

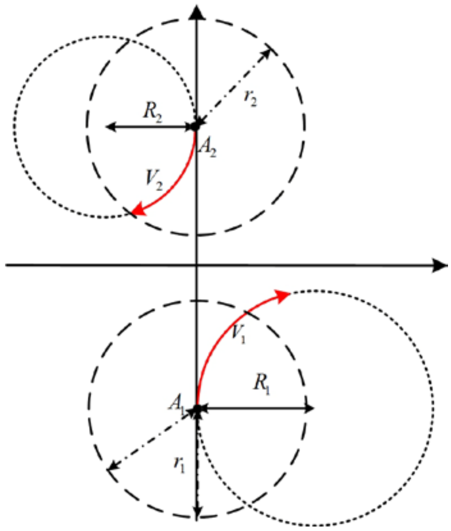


Fig 5 UAV motion modification

In a multi-UAV coordination confliction resolution problem, all the involved UAVs would take measures to avoid collision, as shown in Fig 5. The UAVs turn in circles when they change the direction of their velocities, and the radius is ε, which can be expressed as

$$\varepsilon = \frac{V}{R} \tag{6}$$

If the angular velocity is determined, the turn radius will be determined based on (6).

Supposing A_2 is static relate to A_1 , the velocity of A_1 is the vector sum of $\vec{\boldsymbol{v}}_1$ and $\vec{\boldsymbol{v}}_2$. By rotating the X-Y coordinate system, the position of A_2 and A_1 can be altered onto Y axis, as shown in Fig 5. Supposing that the motion angular of A1 is ϕ_1 and A2 is ϕ_2, then the motion differential units can be expressed as (7):

$$\begin{aligned}\Delta x_i &= \cos(\phi_i - \varepsilon_i\tau)v_i d\tau \\ \Delta y_i &= \sin(\phi_i - \varepsilon_i\tau)v_i d\tau \qquad i=1,2\end{aligned} \tag{7}$$

Therefore, the differential unit position of A1 relative to A2is

$$\begin{aligned}\Delta x' &= (\cos(\phi_1 - \varepsilon_1\tau)v_1 - \cos(\phi_2 - \varepsilon_2\tau)v_2)d\tau \\ \Delta y' &= (\sin(\phi_1 - \varepsilon_1\tau)v_1 - \sin(\phi_2 - \varepsilon_2\tau)v_2)d\tau\end{aligned} \tag{8}$$

The motion of the UAV is integrable, and the displacement can be expressed as:

$$\begin{aligned}x' &= \int_0^t (\cos(\phi_1 - \varepsilon_1\tau)v_1 - \cos(\phi_2 - \varepsilon_2\tau)v_2)d\tau \\ &= -\frac{v_1}{\varepsilon_1}(\sin(\phi_1 - \varepsilon_1\tau) - \sin(\phi_1)) + \frac{v_2}{\varepsilon_2}(\sin(\phi_2 - \varepsilon_2\tau) - \sin(\phi_2))\end{aligned} \tag{9}$$

$$\begin{aligned}y' &= \int_0^t (\sin(\phi_1 - \varepsilon_1\tau)v_1 - \sin(\phi_2 - \varepsilon_2\tau)v_2)d\tau \\ &= \frac{v_1}{\varepsilon_1}(\cos(\phi_1 - \varepsilon_1 t) - \cos(\phi_1)) - \frac{v_2}{\varepsilon_2}(\cos(\phi_2 - \varepsilon_2 t) - \cos(\phi_2))\end{aligned} \tag{10}$$

Therefore we can draw a conclusion that the solution to UAV conflict $(\varepsilon_1, \varepsilon_2)$ should meet the following conditions.

1. $d_t(A_iA_j) = \sqrt{(x' - x_i)^2 + (y' - y_i)^2} > R \quad t \in (0,\tau)$ (11)

2. After τ time, the modified velocities may guarantee it is collision free.

The (11) can be translated into the solution to the following equation:

$$\sqrt{(x'(t) - x_i(t))^2 + (y'(t) - y_i(t))^2} = R, t \in (0,\tau) \tag{12}$$

If $\sqrt{(x'(0) - x_i(0))^2 + (y'(0) - y_i(0))^2} > R$ is guaranteed and there is no solution for (12) in time period $(0,\tau)$, then the angular velocity pair $(\varepsilon_1, \varepsilon_2)$ is feasible for conflict resolution.

For the second condition, we can calculate the finish point A1 by (9) and (10):

$$\begin{aligned}E_x &= -\frac{v_1}{\varepsilon_1}(\sin(\phi_1 - \varepsilon_1\tau) - \sin(\phi_1)) + \frac{v_2}{\varepsilon_2}(\sin(\phi_2 - \varepsilon_2\tau) - \sin(\phi_2)) \\ E_y &= \frac{v_1}{\varepsilon_1}(\cos(\phi_1 - \varepsilon_1 t) - \cos(\phi_1)) - \frac{v_2}{\varepsilon_2}(\cos(\phi_2 - \varepsilon_2 t) - \cos(\phi_2))\end{aligned} \tag{13}$$

And the velocity A1 relate to A2 is

$$\begin{aligned}V'_x &= v_1\cos(\phi_1 + \varepsilon_1\tau) - v_2\cos(\phi_2 + \varepsilon_2\tau) \\ V'_y &= v_1\sin(\phi_1 + \varepsilon_1 t) - v_2\sin(\phi_2 + \varepsilon_2 t)\end{aligned} \tag{14}$$

By using (13) and (14), we can determine whether the angular pair $(\varepsilon_1, \varepsilon_2)$ is feasible. As shown in Fig 6, if two UAVs still come into collision at t_r, we define this kind of motion as safety hazard. The fatalness of the safety hazard maneuver can be weighed by t_r.

3.3 Application of reinforcement learning

To return the ways back to the planned waypoints with least cost, rather than to prevent UAVs from collision is the ultimate objective of conflict resolution. In addition, the policy should consider the fairness of costs of all the participants. In this paper, a near optimal policy in local airspace is obtained in virtue of reinforcement learning.

A. States

The state in the system describes the features such as position, orientation and post-modification of UAVs at the same time. The state-space is bounded and holds all possible combinations of these features of all involved UAVs. A state is represented as a 4n-dimensional vector

$$x = [x_1^{pos}, y_1^{pos}, \phi_1, \Phi_1^{post} \ldots, x_n^{pos}, y_n^{pos}, \phi_n, \Phi_n^{post}] \tag{15}$$

Where n stands for the number of related vehicles, x_i^{pos} and y_i^{pos} are Cartesian coordinates, ϕ_1 denotes the direction of motion, and Φ_i^{post} represents the post modification of each UAV. This paper does not concern the whole state space of UAVs, and the searching space is mainly the area around the initial state of these UAVs, which is defined by the searching depth K.

B. Action

Actions describe the behavior which the system may choose in a specific situation. In our work, each UAV has the minimum turn radius constraint, which determines the maximum angular velocity of each UAV:

$$-\frac{v_1}{r_1^{min}} \le \varepsilon_i \le \frac{v_1}{r_2^{min}} \tag{16}$$

Therefore, one action for the system can be described as an action tuple $a = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n)$. The action space is an n dimensional continuous space. In order to reduce the computation complexity, we use the heuristic policy search method, which will be discussed later.

C. Reward

A reward function may be used to define the goal in reinforcement learning. It maps perceived states (or state-action pairs) of the environment to a single number, a *reward*, which indicates the intrinsic desirability of the state. A reinforcement-learning agent's sole objective is to maximize the total reward that the agent receives in the long run. The reward function defines the good and the bad events for the agent. As UAVs are to take detours to avoid collision, and then return to their projected paths if the crisis is removed, the goals are states where both vehicles are on the projected paths with appropriate orientations. The closer the state is to the goal, the higher reward it will receive. The forbidden states are those leading to airspace collision so that they have a major negative reward. We bring forth the heuristic state-action value function that can appropriately help obtain the near optimal policy. In the conflict resolution scene, we take the time to the goal state t_g and time to collision t_c as the parameters of the value function. The purpose is to make the UAVs approach their goals collision-free respectively. If two UAVs are too close to each other, the conflict would be unavoidable due to their mobile abilities. In our design, the tangential method is proposed to estimate the collision risk between involved UAVs.

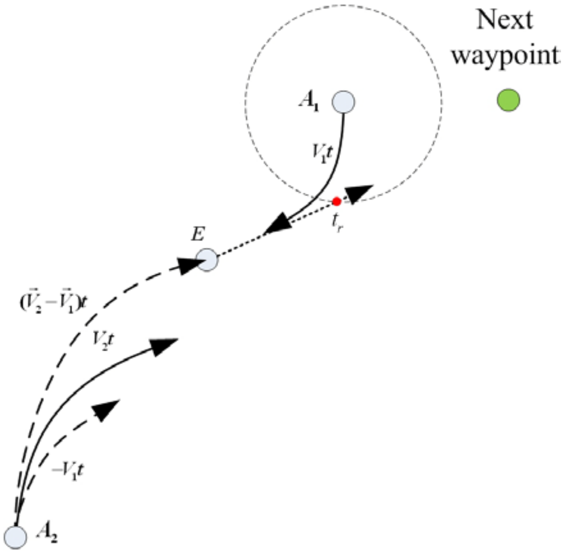


Fig 6 Action prediction and value evaluation

When we evaluate each action for two involved UAVs, the new state could be determined by the policy, as shown in Fig 6. The distance between UAVs and their next goals may vary, and we try to evaluate the policy in making the UAVs approach the goal. We use the heuristic method to estimate the time from the current state to the goal state in virtue of their current speeds.

The estimated time can be expressed as (17), which neglects the current direction of the velocities of UAVs.

$$t_{goal} = \frac{\left\| S_i^t - Goal \right\|}{\left\| V_i \right\|} \tag{17}$$

D. Approximation value function

For each policy $\{\varepsilon_1, ..., \varepsilon_n\}$, the next state is given as:

$$\begin{aligned} x_i^{n+1} &= x_i^n + \frac{v_i}{\varepsilon_i}(\sin(\phi_i) - \sin(\phi_i - \varepsilon_i \tau)) \\ y_i^{n+1} &= y_i^n - \frac{v_1}{\varepsilon_1}(\cos(\phi_i) - \cos(\phi_i - \varepsilon_i \tau)) \end{aligned} \tag{18}$$

For the non-goal states, their value depends on their reward and the discounted value of the best following states. With the heuristic method, we could estimate the non-goal state values. The range of the state-action value is set to be [0, 1], and the Q function can be achieved by the iteration equation (19).

$$Q_{k+1}^{\pi}(x) = r_{k+1} + \gamma \max_a [Q_k(f(\pi))] \tag{19}$$

Where γ is the discount factor. The state-value can be approximated by iteration. In the online planning phase, it is impractical to obtain the global optimal policy. The infinite-horizon return is approximated by truncating each simulated trajectory after K steps. Supposing that the distance between two waypoints is D_W and the speed of each UAV is V_i, K should meet the following condition:

$$K > \max\{\frac{D_w}{V_i}, i \in n\} \tag{20}$$

The equation (20) indicates that K should guarantees the UAVs reach the next waypoints from the current states in K steps.

Because the course of the UAVs is made up by many waypoints, if one UAV is far from the current target, then it will choose the next waypoint as the new goal state at the price of a greater loss, and its cost will be higher if it maneuvers to escape from collision in the future.

3.4 Model-based reinforcement learning algorithm

3.4.1 The problem in conflict resolution

In the above sections we analyzed the airspace conflict resolution problem and defined the constraints about collision-free maneuvers. The goal states of conflict resolution courses are to return to the projected paths rather than to get rid of the possible collisions. What's more, the time window may induce some unsuspected problems. As shown in Fig 7, A_1 and A_2 are about to collide at P, if the local coordinator only considers the temporal crisis elimination, it can be seen that these two planes will get into a more difficult situation after the indicated maneuvers. As the conflict would become more reluctant, each of them would be closer and closer. What worse, one of the participants may be far away from its projected path. This kind of problems cannot be eliminated even if the time window is extended. The reinforcement learning method can deal with this problem with the heuristic method, because this method would take the goal into consideration and estimate possible conflicts beyond time window limitation.

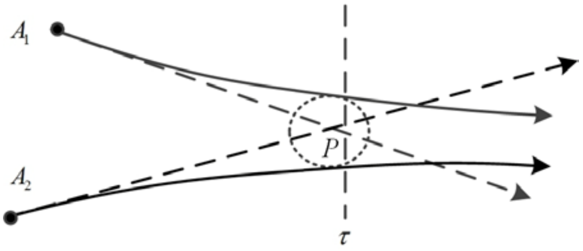


Fig 7 Problem in time window restriction

3.4.2 Algorithm design

The model-based reinforcement learning structure is shown in Fig 1. It is hard to apply the trial and error method in conflict resolution in the real environment. In this paper, we propose the online planning method to search the near optimal policy by model-based reinforcement learning.

1. Value approximation

In the high dimension continuous space, it is impractical to search the whole space. As for the conflict resolution problem, the initial state of the system is fixed in each case, so it is possible for us to search around the initial state for the near optimal policy. As mentioned above, we could search for the near optimal policy in K steps. Each state-action value is approximated by approximation function. In this paper, we consider the linear approximation method, in which state-action pairs are mapped first to feature vectors and then to value in a linear way with learned parameters [17].

The features of the linear approximation should make use of the prior knowledge of the problem. The form of the feature vector can be expressed as:

$$\varphi = \{\varphi_1(x), \varphi_2(x), ..., \varphi_n(x)\} \qquad (21)$$

Suppose θ is the parameter vector, so the state-action value can be expressed as:

$$V_t(\pi) = \vec{\theta}_t^T \vec{\varphi}_s \qquad (22)$$

Intuitively, the features should correspond to the natural features of the task, along which generalization is most appropriate [9]. Four features are considered for each state. Feature 1 reflects the conflict probability between vehicles, which helps to filter out motions that lead to collisions. Feature 2 reflects the overall distance to the projected path, which helps vehicles move closer to the temporary goal state. The third feature is regarding orientation differences from agents' projected goal. It is helpful to ensure them return to their intended path in the appropriate orientation. In the whole flying course, one UAV may encounter with airspace conflicts more than once, so we should consider the post cost of each UAV. Therefore, the last feature concerns fairness. Our aim is to balance the cost of conflict resolution between each participant

2. Heuristic learning

In the real environment, the UAVs often take several actions to avoid collision, e.g. turn left in 40 degree, turn left in 20 degree, go straight, turn right in 20 degree, and turn left in 40 degree. We also discretize the continuous action of each UAV into discrete action subsets.

Because of the high dimension continuous space, particularly when the amount of UAVs becomes large, the search space would be too large in a search for the whole tree. In fact, the change of the motion direction of one specific UAV may influence only UAVs around it. Therefore, we could reduce the search space by considering the real situation information. As shown in Fig 8, each UAV only conflicts with two other UAVs around it. Therefore, the whole space can be decoupled into several subspaces when the algorithm searches for the optimal solution, which saves both CPU time and memory.

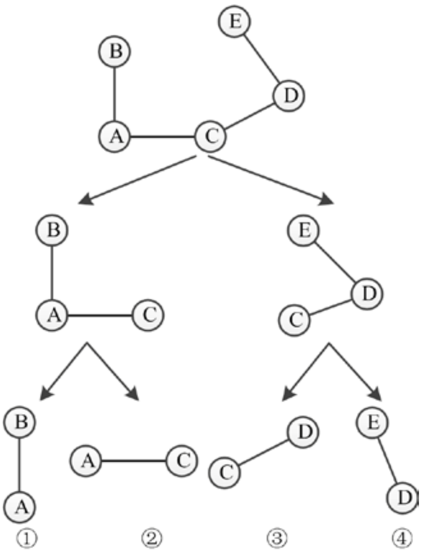


Fig 8 Decomposition of conflict connection

In Fig 8, it is supposed that each UAV has n selectable actions. The original action space is n^5, so we can divide the whole UAVs into four subsets by cutting the chain, and then the action space could be reduced to $4n^2$.

In a heuristic search, for each state encountered, a large tree of possible alternatives is considered. The approximate value function is applied to the leaf nodes, and then backed up at the previous state towards the root. The backing up in the search tree is just the same as in the max-backups. It stops at the state-action nodes of the current state. Once the backed-up values of these nodes are computed, the best of them is chosen as the current action, and the rest of the values are discarded. In conventional heuristic search no effort is made to save the backed-up values and the value function, which once designed, never changes as a result of the search. However, it would be reasonable to allow the value function to be improved over time using either the backed-up values computed during the heuristic search or by any other method.

3. Direct learning from the experience

Each time the system chooses the near optimal policy by model-based planning, after it executes this policy, it will get reward from the environment. In the airspace conflict resolution problem, the system cannot go back to retry from the beginning. Therefore the updating of the predecessors is useless in the learning. The experience should be used to facilitate the decision making in successive states, such as to improve the model. As the model is a simplified expression of the real environment, it cannot model the uncertainty of the environment and may have some errors. We can criticize the model-based learning policy by real results, as shown in Fig 9.

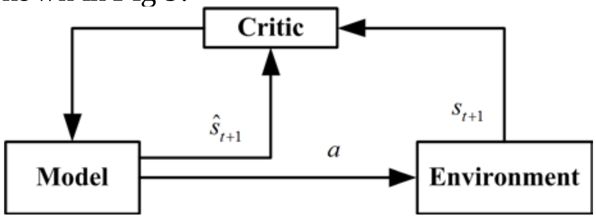


Fig 9 Model-based learning architecture

The algorithm is shown in Tab 2.

Tab 2 Model based online planning algorithm

```
Algorithm: Model based reinforcement learning algorithm
Initialize Q(s,a),Model(s,a),Goal state(S_g), initial state(S_0) $\forall s \in S, a \in A$
repeat
  s←current(non terminal)state
  initial the search tree i=1
  while i<k
     $s_{node} \leftarrow greedy(s,Q)$        find the expand node
     if depth( $s_{node}$ )==i
        i=i+1;
     end
    For j=1: Discrete action
        $a_j \in Actionset$
        $s',r = Model(s,a_j)$
        $Q(s,a) \leftarrow r + \gamma \max_a Q(s',a')$    update the value
     End for
   End for
   $a = \max_{u \in actionset}\{Q(s,a)\}$
  execute a in the real environment, observe $s'$ and r
until $s'$ is Goal state
```

4 Experiments

This section focuses on two experiments. We use the real dynamic model of the UAVs in the environment, and add the noise into the input/output information. The noise follows Gaussian distribution with zero mean. We use the time variables to estimate the state-action value function, which can be expressed as (23). Where $t_{collision}$ is the collision time if the maneuver result in collision in $0-\tau$, t_g^i refers to the estimated time to the goal state, t_r refers to the time collision after time window τ , t_a^i refers to the estimated time from current direction to the planned motion direction, U_i refers to the modification angle of each UAV during the flight.

$$R=\begin{cases}\dfrac{t_{collision}}{t_{step}}\theta_1 & collision\\ \theta_1+\dfrac{\theta_2}{e^{t_g^1+t_g^2}}+\dfrac{\theta_3}{t_r}+\theta_4^T\begin{bmatrix}e^{t_a^1}\\ e^{t_a^2}\end{bmatrix}+\theta_5\left\|U_1-U_2\right\| & nocollision\end{cases}\tag{23}$$

The state-action value should be positive. When the involved UAVs are about collision, the state-action value should be small enough to guarantee the search algorithm not to choose this policy. Other wise, the state-action value is determined by t_g^i, t_r, t_a^i and U_i .

In the first experiment, there are two UAVs in the environment, which fly in a rounded trace. One flies clockwise and the other anticlockwise. The waypoints are set at every other km, and the UAV velocities are set as 0.1km/s. The designed traces determine there would be airspace conflicts now and then, as shown in Fig 10(a). In each conflict resolution course, the system takes the next waypoint of each UAV as the temporary goal state and carries on the near optimal policy search. The result of the experiment is shown in Fig 10.

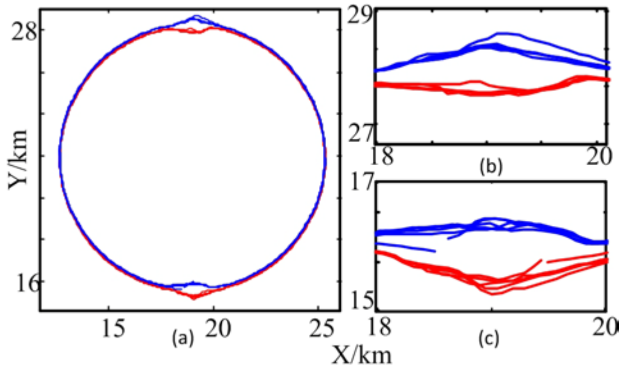


Fig 10 Two-UAV conflict resolution

Fig 10 (a) illustrates the result of conflict resolution all around the experiment. (b) and (c) demonstrate the effect in detail. As shown in Fig 10, the system can deal with the conflict in the long period simulation. The system only takes about 0.07s to find a near optimal policy. The safe distance is 0.6km, and the minimum distance between these two UAVs in the simulation is 0.635km. From Fig 10 we know that the modifications of these two UAVs are almost the same.

In the second experiment, the environment is a square of which the area is 100km^2. There are 25 agents in the environment, which are uniformly spaced in the square. Their line velocities are all 0.1km/s and their motion directions are random. Their maximum angular velocities are all 0.5 radians/s. The UAVs in the environment are flying based on their dynamics and kinematics. The controller receives the motion information of the UAVs, and predicts if there exist airspace conflicts. The radius of the safe region is 0.3km, and the experiment runs 250s. Fig 11 shows the minimum distance between each pair of these agents.

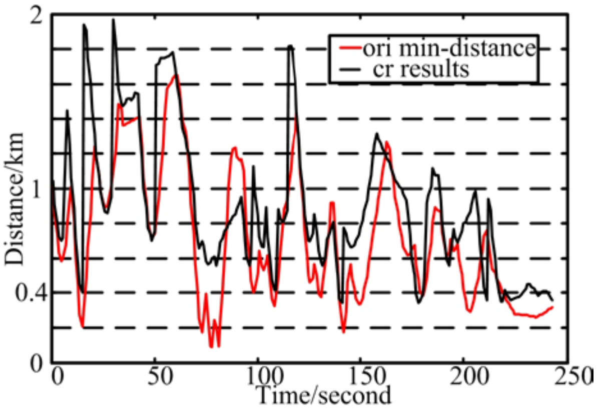


Fig 11. Minimum distance between agents.

As shown in Fig 11, the original min distance curve shows that there are many space conflicts between these UAVs during the experiment because the red line is often close to 0.3 km. After the conflict resolution, all the conflicts are resolved by multi-agent coordination. The minimum distance between agents is more than 0.3km. The data demonstrate the ability of the multi-agent coordination method in dealing with multi-UAV problems.

5 Conclusions

This paper deals with UAVs conflict detection and resolution in dynamic environments. The UAVs are supposed to be managed by the ground stations, so they can take cooperative maneuvers to eliminate the airspace conflictions.

The dynamics of UAVs are considered when dealing with conflict resolution problems.

In this paper, we present a model-based reinforcement learning method for online conflict resolution planning. We firstly study the UAV airspace conflict problem, and then bring forth several constraints in keeping off midair collision, as expressed in math inequalities. Since the goal of the online conflict resolution is to return to the projected path, we propose to search the near optimal policy by reinforcement learning, and then approximate the value of the state-action pair in the large and continuous space. Furthermore, to reduce the computation complexity, we propose to apply the heuristic method. The proposed algorithm exhibits smooth and convincing behavior in our experiments.

Reference

[1] Consiglio M, Chamberlain J, Muñoz C, et al. Concept of Integration for UAS Operations in the NAS. 28th International Congress of the Aeronautical Sciences, Brisbane, Australia. 2012.

[2] Dalamagkidis K, Valavanis K P, Piegl L A. On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the National Airspace System. Progress in Aerospace Sciences, 2008, 44(7): 503-519.

[3] Emami H, Derakhshan F. An overview on conflict detection and resolution methods in air traffic management using multi agent systems. Artificial Intelligence and Signal Processing (AISP), 2012 16th CSI International Symposium on. IEEE, 2012: 293-298.

[4] Jaradat M A K, Garibeh M H, Feilat E A. Autonomous mobile robot dynamic motion planning using hybrid fuzzy potential field. Soft Computing, 2012, 16(1): 153-164.

[5] Chakravarthy A, Ghose D. Obstacle avoidance in a dynamic environment: A collision cone approach. Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on, 1998, 28(5): 562-574.

[6] Fiorini P, Shiller Z. Motion planning in dynamic environments using velocity obstacles. The International Journal of Robotics Research, 1998, 17(7): 760-772.

[7] Belkhouche F. Reactive path planning in a dynamic environment. Robotics, IEEE Transactions on, 2009, 25(4): 902-911.

[8] Van Den Berg J, Guy S J, Lin M, et al. Reciprocal n-body collision avoidance. Robotics research. Springer Berlin Heidelberg, 2011: 3-19.

[9] Sutton & Barto A G. Reinforcement learning: An introduction. MIT press, 1998.

[10] Gómez M, González R V, Martínez-Marín T, et al. Optimal motion planning by reinforcement learning in autonomous mobile vehicles. Robotica, 2012, 30(02): 159-170.

[11] Viet H, Choi S, Chung T. Dyna-QUF: Dyna-Q based univector field navigation for autonomous mobile robots in unknown environments. Journal of Central South University, 2013, 20(5): 1178-1188.

[12] Wang Q, Phillips C. Cooperative collision avoidance for multi-vehicle systems using reinforcement learning. Methods and Models in Automation and Robotics (MMAR), 2013 18th International Conference on. IEEE, 2013: 98-102.

[13] Cooperation in a distributed hybrid potential-field/reinforcement learning multi-Agent-based autonomous path planning in Dynamic Time varying unstructured environment. IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support, New Orleans, LA.2012

[14] Viet H H, Kyaw P H, Chung T C. Simulation-based evaluations of reinforcement learning algorithms for autonomous mobile robot path planning. IT Convergence and Services. Springer Netherlands, 2011: 467-476.

[15] Santos M, Martín H J A, López V, et al. Dyna-H: A heuristic planning reinforcement learning algorithm applied to role-playing game strategy decision systems. Knowledge-Based Systems, 2012, 32: 28-36.

[16] Hwang K S, Jiang W C, Chen Y J, et al. Model-Based Indirect Learning Method Based on Dyna-Q Architecture. Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on. IEEE, 2013: 2540-2544.

[17] Sutton R S, Szepesvári C, Geramifard A, et al. Dyna-style planning with linear function approximation and prioritized sweeping. arXiv preprint arXiv:1206.3285, 2012.

Copyright Statement