RESEARCH ON FLIGHT CONTROL METHOD OF MORPHING UCAV BASED ON REINFORCEMENT LEARNING

YAO Zong-xin
Shenyang Aircraft Design and Research Institute, China

Keywords: *morphing UCAV, flight control, reinforcement learning*

Abstract

This paper presents a Reinforcement Learning control methodology to the problem of unmanned air vehicle morphing. An innovative morphing UCAV with the capabilities in creating the two wing changes of configuration and shape is induced to develop the RL control method, and the control vector of the morphing UCAV is defined as the rotary angle displacements of those smart joints in two wings that changes the morphing vehicle shape towards the optimal one, and the control vector is associated with those aerodynamic force and moment parameters by a neural network, all of which may not have been experienced before. The Reinforcement Learning module of the morphing UCAV is implemented by Q-Learning method. The RL module is composed of the states, actions, Q-matrix, Q-learning update, reward, and control police, and so on. The RL methodology is demonstrated with a numerical example of a hypothetical 3-D smart unmanned air vehicle that can morph in the configuration and shape of wings, to track a specified trajectory by running the RL control module. Results presented in the paper show that this methodology is capable of learning the required morphing into it, and accurately tracking the reference trajectory.

1 Introduction

It has been interested to develop an aircraft which has been designed to both effectively and efficiently complete a multi-disparate mission. An example of such a mission is one which requires an aircraft to have both long loiter endurance as well as supersonic fight capabilities. A solution to achieve the required performance for this scenario is an aircraft that could radically change both its size and shape, or morph.

In the context of flight vehicles, Morphing for Mission Adaptation is a large scale, relatively slow, in-flight shape change to enable a single vehicle to perform multiple diverse mission profiles, and is defined as the Wing Configuration Change, such as the wing's length extending or shrinking, and the wing's area augmenting or dwindling; Conversely, Morphing for Control is an in-flight physical or virtual shape change to achieve multiple control objectives, and is defined as the Wing Shape Change, such as the maneuvering, flutter suppression, load alleviation and active separation control, and so on.

This paper develops an Reinforcement Learning Control methodology by using a Aerodynamic Calculating Neural Network, the Six Freedom Equation, and the Q-learning Update Module, to more efficiently and accurately generalize the knowledge gained from iterative experiences.

The design and use of distributed shape-change devices to provide low-rate maneuvering capability for a tailless aircraft is considered in [1]. An improved Adaptive-Reinforcement Learning control methodology to the problem of unmanned air vehicle morphing is presented in [2]. Reinforcement Learning is utilized with an antenna model to demonstrate that antenna elements equipped with SMA actuators in [3]. A novel UAV upset recovery system is developed that combines the benefits of robust control with the benefits of intelligent learning techniques in

[4]. It is more beneficial to learn the voltage position relationship in order to control Shape Memory Alloy wires using Reinforcement Learning in [5].

The paper is organized as follows: Section 2 explains in detail the configuration and control regulation of an conceptual morphing UCAV. The flight control method of the morphing UCAV based on Reinforcement Learning is presented in Section 3. Section 4 talks about the numerical example. Section 5 summarizes the conclusions.

2 Configuration and Control Regulation of an Conceptual Morphing UCAV

2.1 Translating Fashion of the Morphing UCAV's Configuration

For making an aircraft to have both long loiter as well as rapid dive fight capabilities, it must be depended on that the three-dimensional morphing wing technology.

Thereby, a hypothetical example of morphing UCAV is shown in Fig.1. The aircraft is able to achieve both the three configuration changes of straight, small-forward-bend and large-forward-bend wing in Fig.1 and the three wing shape changes of bend, warp and twist by controlling those smart joints in two wings. The six joints in the left wing are defined as L_1, L_2, L_3, L_4, L_5, and L_6 in Fig.2, while the joints of the right wing are same. Each joint is designed to turn respectively round the body x, y, and z axis for achieving the all configuration and shape changes of the wings.

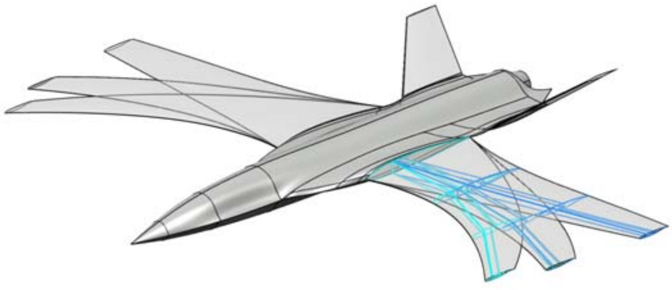

Fig.1. The Conceptual Morphing UCAV

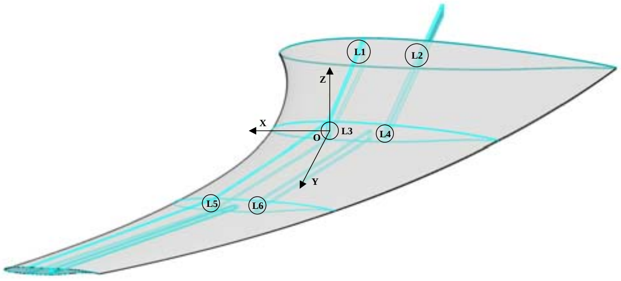

Fig.2. The Approach of Achieving Morph

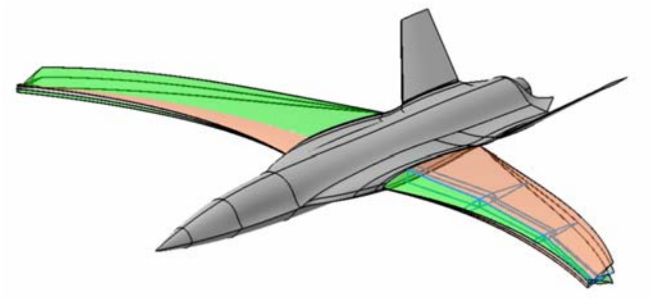

Fig.3. The Morphing Panorama of the Aileron Function

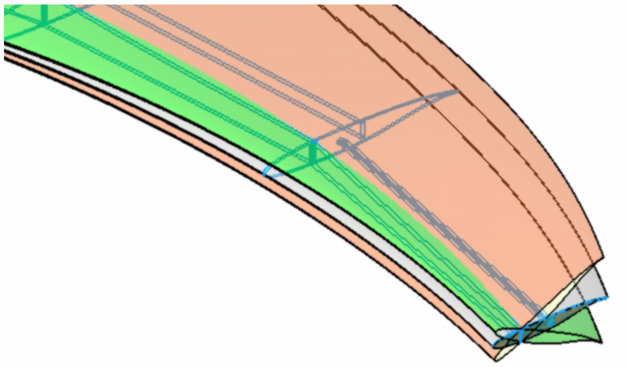

Fig.4. The Morphing Details of the Aileron Function

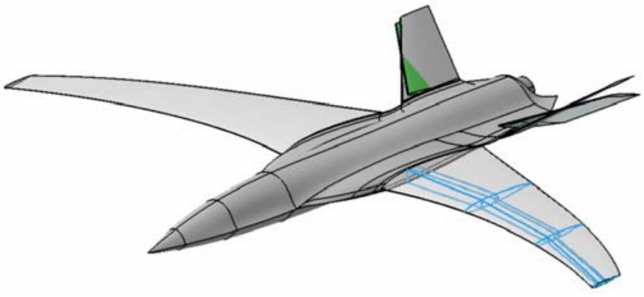

Fig.5. The Morphing of the Tail-wing Function

2.2 Definition of the Control Vectors of the Morphing UCAV

The vectors of controlling shape changes are defined as follows:

Let $AL = (ALx_1, ALy_1, ALz_1, \cdots, ALx_i, ALy_i, ALz_i,$
$\cdots, ALx_m, ALy_m, ALz_m, ALt)$

$AR = (ARx_1, ARy_1, ARz_1, \cdots, ARx_i, ARy_i, ARz_i,$
$\cdots, ARx_m, ARy_m, ARz_m, ARt) \qquad m = 6$

ALx_i , ALy_i ,and ALz_i are the rotary angle displacements of the L_i joint turning respectively round the body x , y , and z axis and ALt is deflection of the left tail-wing , while the meaning of ARx_i , ARy_i , ARz_i , and ARt is similar. AL and AR are used to describe the shape–change driving states of the morphing UCAV. Each shape–change driving state induces actually one only aerodynamic force and moment state of the morphing aircraft. Let

$A_k = (a_1, a_2, \cdots, a_n) = (AL, AR, q_c, \alpha, \beta) =$
$(a_1, \cdots, a_{3m}, a_{3m+1}, a_{3m+2} \cdots, a_{3m+3m+2}, a_{n-2}, a_{n-1}, a_n)$

$AL = (a_1, a_2, \cdots, a_{3m}, a_{3m+1})$

$AR = (a_{3m+2}, a_{3m+3}, \cdots, a_{3m+3m+2})$

$q_c = (a_{n-2}), \qquad \alpha = (a_{n-1}), \qquad \beta = (a_n)$

$B_k = (b_1, b_2, \cdots, b_p)$

$C_k = (c_1, c_2, \cdots c_q) = (L, D, Y, M, \bar{L}, N), q = 6$

L is the lift force, D is the drag force, Y is the side force, M is pitching moment, $\bar{L}$ is rolling moment, and N is yawing moment; q_c is the dynamic press, α is the attack angle, β is the side-slip angle.

A three-layer neural network is created and shown in Fig.3. The input, hidden, and output layer vectors are defined as A_k (n =41), B_k , and C_k (q =6), respectively.

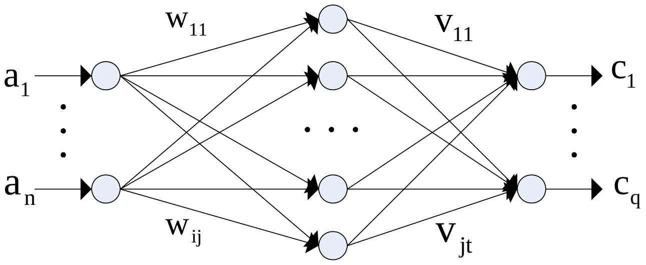

Fig.6. The Neural Network Architecture

The activation function is $f(x) = \tanh(x)$.

The forward calculating process is shown as:

$$b_j = f(\sum_{i=1}^{n} w_{ij} a_i - \theta_j), c_t = f(\sum_{j=1}^{p} v_{jt} b_j - \gamma_t)$$

The approach of acquiring the Neural Network samples is presented as follows:

When the aircraft flights by keeping a certain Euler angles such as hovering with a small roll angle, the control vectors of AL and AR are measured from the all joints in two wings and tail-wings of a three-dimensional morphing UCAV model built by employing the CATIA software, and the all aerodynamic forces and moments of L , D , Y , M , $\bar{L}$, and N are calculated by employing the FLUENT software. The three samples are shown as follows:

$A_{1,2,3} = (a_1, a_2, \cdots, a_{41}) = (AL, AR, q_c, \alpha, \beta) =$
$(a_1, \cdots, a_{19}, a_{20} \cdots, a_{38}, a_{39}, a_{40}, a_{41})$

$AL_{1,2,3} = (a_1, a_2, \cdots, a_{19}) =$
$(ALx_1, ALy_1, ALz_1, ALx_2, ALy_2, ALz_2, ALx_3, ALy_3, ALz_3,$
$ALx_4, ALy_4, ALz_4, ALx_5, ALy_5, ALz_5, ALx_6, ALy_6, ALz_6,$
$ALt)$
$= (0.0, 5.0, 0.0, 0.0, 5.0, 0.0, 0.0, 10.0, 0.0,$
$0.0, 10.0, 0.0, 0.0, 15.0, 0.0, 0.0, 15.0, 0.0, 15.0)$

$AR_{1,2,3} = (a_{20}, a_{21}, \cdots, a_{38}) =$
$(ARx_1, ARy_1, ARz_1, ARx_2, ARy_2, ARz_2, ARx_3, ARy_3, ARz_3,$
$ARx_4, ARy_4, ARz_4, ARx_5, ARy_5, ARz_5, ARx_6, ARy_6, ARz_6,$
$ARt)$
$= (0.0, -5.0, 0.0, 0.0, -5.0, 0.0, 0.0, -10.0, 0.0,$
$0.0, -10.0, 0.0, 0.0, -15.0, 0.0, 0.0, -15.0, 0.0, 15.0)$

$q_{c1,2,3} = (a_{39}) = 9433.6 N / m^2$

$\beta_{1,2,3} = (a_{41}) = 0^\circ$, $\alpha_1 = (a_{40}) = 2^\circ$

$C_1 = (c_1, c_2, \cdots c_6) = (L, D, Y, M, \bar{L}, N) =$
$(90019, 17709, 906, 896, 511901, -89963)$

$\alpha_2 = (a_{40}) = 5^\circ$

$C_2 = (c_1, c_2, \cdots c_6) = (L, D, Y, M, \bar{L}, N) =$
$(152618, 26472, 2757, 7637, 454141, -99041)$

$\alpha_3 = (a_{40}) = 8^\circ$

$C_3 = (c_1, c_2, \cdots c_6) = (L, D, Y, M, \bar{L}, N) =$
$(194606, 39241, 5886, 15864, 399838, -84669)$

where the unit of L, D and Y is (N), and the unit of M, $\bar{L}$ and N is ($N \cdot m$).

3 Flight Control Method of the Morphing UCAV Based on Reinforcement Learning

3.1 Mathematical Model for the Dynamic Behavior of the Morphing UCAV

The dynamic behavior of the morphing air vehicle is modeled by the nonlinear six degree of freedom equations. The dynamic equations are partitioned into 'kinematic level' and 'acceleration level' equations. The kinematic level variables are d_x, d_y, d_z, ϕ, θ and ψ. The states d_x, d_y, d_z are the positions of the center of mass of the morphing vehicle along the inertial X_N, Y_N, and Z_N axis. The states ϕ, θ, and ψ are the 3-2-1 Euler angles which give the relative orientation of the body axis. The acceleration level states are the body-axis linear velocities u, v, w and the body-axis angular velocities p, q, r. Let

$$p_c = [d_x \quad d_y \quad d_z]^T, v_c = [u \quad v \quad w]^T$$

$$\sigma = [\phi \quad \theta \quad \psi]^T, \omega = [p \quad q \quad r]^T$$

The kinematic states and acceleration states are related by the differential equations $\dot{p}_c = J_l v_c, \dot{\sigma} = J_\alpha \omega$. where

$$J_l = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix}$$

$$J_a = \begin{bmatrix} 1 & S_\phi \tan(\theta) & C_\phi \tan(\theta) \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi \sec(\theta) & C_\phi \sec(\theta) \end{bmatrix}$$

$C_\phi = \cos(\phi)$, $S_\theta = \sin(\theta)$, and so on.

The acceleration level differential equations are

$$m\dot{v}_c + \tilde{\omega} m v_c = F + F_d$$

$$I\dot{\omega} + \dot{I}\omega + \tilde{\omega} I \omega = M_k + M_d$$

where m is the mass of the morphing air vehicle, F is the control force, F_d is the drag force, I is the body axis moment of inertia, M_k is the control torque, M_d is the drag moment, and $\tilde{\omega}V$ is the matrix representation of the cross-product between vector ω and vector V.

$$\tilde{\omega} = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix}$$

$$F = F(L, D, Y, T), \quad M_k = M(M, \bar{L}, N)$$

where T is the engine push force.

The shape–change driving states and the position states of the morphing UCAV are related by the calculating process from neural network to six degree of freedom equations.

3.2 Implementation of the Reinforcement Learning Module of the Morphing UCAV

Reinforcement Learning (RL) is a branch of Artificial Intelligence in which the system learns to recognize situations and appropriately adapt to achieve a goal without external supervision. This technique is suited for situations in which little or nothing is known about how to achieve the desired outcome. The agent is responsible for learning based on its encounters with various environmental, or external conditions. Possible conditions for the agent itself are called states s, and the agent is capable of pursuing certain actions a to change its state. The control policy π is a mapping of states to actions; this mapping is used by the agent to select the best action while at a particular state. It is the ultimate goal of the agent to learn the optimal control policy. The desirability of a certain behavior is determined by rewards r, which are used to update the control policy.

- *The Q-Learning Update*

Reinforcement Learning (RL) algorithms are based on estimating value functions. The most general one is the action-value function $Q^\pi(s,a)$, which estimates how good it is, under policy π, for the agent to perform action a in state s. It is defined as the expected return starting from s, taking action a, and thereafter following policy π. The process of computing $Q^\pi(s,a)$ is called policy evaluation. π can be improved to a better π' that, given a state, always selects the action, of all possible actions, with the best value based on $Q^{\pi}(s,a)$. This process is called policy improvement. $Q^{\pi'}(s,a)$ can then be computed to improve π' to an even better π''. The ultimate goal of RL is to find the optimal policy π^* that has the optimal action-value function, denoted by $Q^*(s,a)$ and defined as $Q^*(s,a)=\max_{\pi} Q^{\pi}(s,a)$. This recursive way of finding an optimal policy is called policy iteration.

In this paper, the RL module uses a 1-step Q-learning method, which is illustrated as follows:

```
Q-Learning( )
  • Initialize $Q(s,a)$ arbitrarily
  • Repeat (for each episode)
    - Initialize $s$
    - Repeat ( for each step of the episode)
        * Choose $a$ from $s$ using policy derived from $Q(s,a)$
        * Take action $a$, observe $r$, $s'$
        * $Q(s,a) \leftarrow Q(s,a)+\alpha\{r+\gamma \max_{a'} Q(s',a')-Q(s,a)\}$
        * $s \leftarrow s'$
    - until $s$ is terminal
  • return $Q(s,a)$
```

The discount factor γ, dictates how much weight is to be given to long-term, rather than immediate rewards. The rate of learning parameter α, is related to the probability of state transferring (the number of times a specific state is encountered).

In this paper, γ=0.5 and α=1.0.

- *The Possible States*

The thing it interacts with, comprising everything outside the agent, is called the environment. The agent interacts with it's environment at each instance of a sequence of discrete time steps. At each time step, the agent receives some representation of the environment state. All possible environment states may be defined as a set of the flight condition which the vehicle is flying in.

The flight condition of the morphing vehicle is actually composed of many sorts maneuver actions. However, for reducing complexity of the research in this paper, the state is designed to only consist of four typical flight conditions, and state is defined as:

$$S=(s_0,s_1,s_2,s_3)$$

where s_0, s_1, s_2 and s_3 represent the flight conditions of cruise, dive, climb and hover, respectively.

- *The Possible Actions*

The environment changing and state transferring are driven by selecting a special action in the RL module. In this paper, the action of the morphing vehicle should be defined as the configuration and shape changes of two wings formed by controlling all twelve smart joints (L_1, L_2, L_3, L_4, L_5, L_6, R_1, R_2, R_3, R_4, R_5 and R_6 defined in Fig.1) and the deflection changes of two tail-wings, theoretically.

However, the parameters describing the action will be too much to employ directly in the RL module. For simplifying the research in this paper, the middle action is designed by introducing five parameters:

WC, LSA, LST, RSA and RST

where WC represents Wing Configuration with three values of SW (Straight Wing), SFB (Small-Forward-Bend), and LFB (Large-Forward-Bend); LSA represents the Left wing Shape of Aileron function with five values of 0(no deflection),±1(up and alow narrow-angle deflection), and ±2(up and alow wide-angle deflection); LST represents the Left Tail-wing Shape with five values of 0, ±1 and ±2 (meaning of deflection is similar to LSA); The definition of RSA and RST is similar to LSA and LST.

It is apparent that the amount of all possible middle actions is also very large, thereby ten middle actions ($a_0 \sim a_9$) continually employed in this paper's states are distilled for simplifying the research and shown in Table 1.

Based on above analysis, in this paper, the action set of the RL module is defined as:

$$A=(a_0,a_1,a_2,a_3,a_4,a_5,a_6,a_7,a_8,a_9)$$

The corresponding relation between the control vectors (AL and AR) and those typical actions such as (a_1, a_2, a_7 and a_8) is shown in Table 2 and Table 3.

Table 1. The Definition of the Actions

A	WC	LSA	LST	RSA	RST
a_0	SW	0	0	0	0
a_1	SW	+1	-1	+1	+1
a_2	SFB	0	0	-1	-1
a_3	LFB	0	0	-2	-2
a_4	LFB	0	0	-1	-1
a_5	LFB	0	0	0	0
a_6	LFB	+1	-1	+1	+1
a_7	LFB	+2	-2	+2	+2
a_8	SFB	0	0	+2	+2
a_9	SFB	0	0	+1	+1

Table 2. The Corresponding Relation between the Control Vector (AL) and those Typical Actions

Unit: degree

A'	a_1	a_2	a_7	a_8
ALx_1	0.00	0.00	0.00	0.00
ALy_1	5.00	0.00	10.00	0.00
ALz_1	0.00	9.00	15.00	9.00
ALx_2	0.00	0.00	0.00	0.00
ALy_2	5.00	0.00	10.00	0.00
ALz_2	0.00	9.00	15.00	9.00
ALx_3	0.00	0.00	0.00	0.00
ALy_3	10.00	0.00	20.00	0.00
ALz_3	0.00	20.50	34.00	20.50
ALx_4	0.00	0.00	0.00	0.00
ALy_4	10.00	0.00	20.00	0.00
ALz_4	0.00	20.50	34.00	20.50
ALx_5	0.00	0.00	0.00	0.00
ALy_5	15.00	0.00	30.00	0.00
ALz_5	0.00	33.20	48.70	33.20
ALx_6	0.00	0.00	0.00	0.00
ALy_6	15.00	0.00	30.00	0.00
ALz_6	0.00	33.20	48.70	33.20
ALt	15.00	-15.00	30.00	30.00

Table 3. The Corresponding Relation between the Control Vector (AR) and those Typical Actions

Unit: degree

A'	a_1	a_2	a_7	a_8
ARx_1	0.00	0.00	0.00	0.00
ARy_1	-5.00	0.00	-10.00	0.00
ARz_1	0.00	9.00	15.00	9.00
ARx_2	0.00	0.00	0.00	0.00
ARy_2	-5.00	0.00	-10.00	0.00
ARz_2	0.00	9.00	15.00	9.00
ARx_3	0.00	0.00	0.00	0.00
ARy_3	-10.00	0.00	-20.00	0.00
ARz_3	0.00	20.50	34.00	20.50
ARx_4	0.00	0.00	0.00	0.00
ARy_4	-10.00	0.00	-20.00	0.00
ARz_4	0.00	20.50	34.00	20.50
ARx_5	0.00	0.00	0.00	0.00
ARy_5	-15.00	0.00	-30.00	0.00
ARz_5	0.00	33.20	48.70	33.20
ARx_6	0.00	0.00	0.00	0.00
ARy_6	-15.00	0.00	-30.00	0.00
ARz_6	0.00	33.20	48.70	33.20
ARt	15.00	-15.00	30.00	30.00

- *The Q-Matrix Defining Action-Value Function*

The agent observes the consequences of actions as they are taken and it uses this data to anticipate future outcomes of various state-action pairs. In this paper, the number of the states and actions of the RL module is not large due to the simplifying work illustrated above. Thereby, $Q(s,a)$ can be represented using a table, where the action-value for each state-action pair is stored in one entity of the table.

The table is also shown as Q-Matrix. the probability that a particular state-action pair is the best choice, regardless of the goal, is dictated by the Q-Matrix. Each row of the Q-Matrix represents an action, while each column represents a state.

$$Q(s,a) = \begin{array}{c|cccc} & s_0 & s_1 & s_2 & s_3 \\ \hline a_0 & 7.00 & 5.50 & 5.00 & 4.50 \\ a_1 & 6.5 & 6.0 & 3.5 & 8.0 \\ a_2 & 7.0 & 9.0 & 2.5 & 1.0 \\ a_3 & 6.0 & 8.5 & 1.5 & 0.5 \\ a_4 & 5.0 & 8.0 & 2.0 & 0.8 \\ a_5 & 4.5 & 5.0 & 4.0 & 5.5 \\ a_6 & 7.0 & 4.5 & 3.5 & 8.0 \\ a_7 & 6.5 & 4.0 & 2.5 & 8.5 \\ a_8 & 5.5 & 2.0 & 8.0 & 4.5 \\ a_9 & 6.0 & 2.2 & 7.5 & 5.0 \end{array}$$

- *The Reward Function*

Let $d = [(d_x - d_{x0})^2 + (d_y - d_{y0})^2 + (d_z - d_{z0})^2]^{0.5}$ Where d_x, d_y, and d_z are the coordinates value of the aircraft's space position; d_{x0}, d_{y0}, and d_{z0} are the coordinates value of the base point on the reference trajectory. The selected base point should ensure that d is the shortest one.

The reward function is defined as:

$$r = \begin{cases} 0.2d & 0 < d \le 0.2e \\ 0.1d & 0.2e < d \le 0.5e \\ 0 & 0.5e < d \le e \\ -0.1d & e < d \le 2e \\ -0.2d & d > 2e \end{cases}$$

where e is the allowable error of tracking flight trajectory.

- *The Control Policy*

This paper's option is an ε – greedy policy, where ε is a small value. The action a with the maximum $Q(s,a)$ is selected with probability $1-\varepsilon$, otherwise a random action is selected.

For the present research, the smart morphing vehicle attempts to minimize the total amount of reward over the entire flight trajectory. To reach this goal, it endeavors to learn, from its interaction with the environment, the optimal policy that, given the specific. Flight condition, commands the optimal rotary angle displacements of those joints in two wings (short for rotary angle displacements infra) that changes the morphing vehicle shape towards the optimal one. The environment is the flight conditions which the vehicle flying in. We assume that the RL module has no prior knowledge of the relationship between rotary angle displacements and the dimensions of the morphing vehicle, as defined by morphing control vectors AL and AR. Also it does not know the relationship between the flight conditions, rewards and the optimal shapes. However, the RL module does know all possible rotary angle displacements that can be applied. It has accurate, real-time information of the morphing vehicle shape, the present flight condition, and the current reward provided by a variety of sensors.

4 Numerical Example

4.1 Purpose and Scope

The purpose of the numerical example is to demonstrate the performance of the RL flight control method for tracking trajectory. For learning purposes, a reference trajectory that the vehicle is required to track is specially designed to simulating a typical dive-bomb flight process and shown in Fig.7. The total flight path is divided into thirteen segments marked by the points of d0-d12. The space position of d0-d12 is shown in the Table 4.

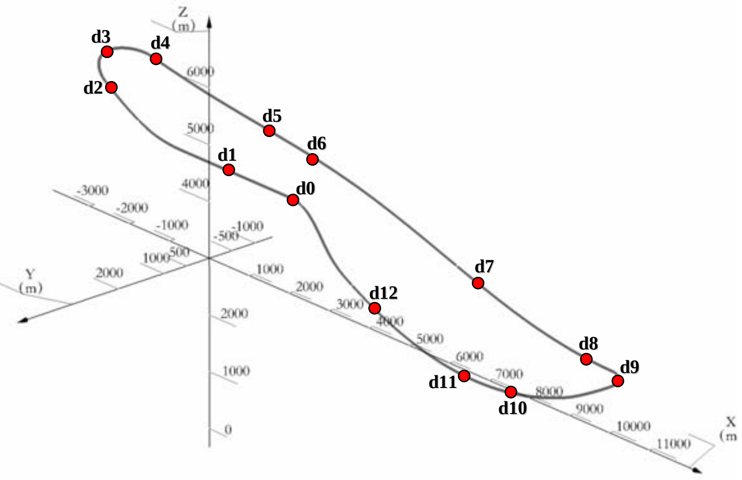

Fig.7. The Reference Trajectory

Table 4. The Space Position of the Reference Trajectory

Unit: meter

D	X	Y	Z
d_0	2500	500	5000
d_1	0	500	5000
d_2	-2000	500	5700
d_3	-2500	0	5900
d_4	-2000	-500	5800
d_5	0	-500	5500
d_6	1600	-500	5300
d_7	6000	-500	4400
d_8	9500	-500	3800
d_9	10000	0	3800
d_{10}	9500	500	3600
d_{11}	6400	500	3200
d_{12}	4000	500	4000

The reference trajectory is designed as a combination of two smooth curves (from d2 to d10 counter-clockwise, and from d4 to d8 clockwise) and two semicircular curves (from d2 to d4, and from d8 to d10, clockwise). The two smooth curves form the two planes, respectively. The distance along to the Y axis between the two planes is 1000 meters; The distance along to the X axis between d3 and d9 is 12500 meters; The distance along to the Z axis between d3 and d9 is 2700 meters.

In the numerical example, the morphing UCAV is required to not only flight through all points of d0, d1, d2, d3, d4, d5, d6, d7, d8, d9, d10, d11, and d12 in sequence, but also has a initial velocity of Mach 0.5 at the point d0 and the least of the spending time. Hereby, the morphing UCAV must increase flying velocity by adaptively changing the configuration and shape of the wings, and besides, it need accurately tracking trajectory.

4.2 Reinforcement Learning Flight Control Architecture

The relation between all parameters and function modules in section 2 and section 3 is shown in the Reinforcement Learning Flight Control Architecture, Fig.8 .

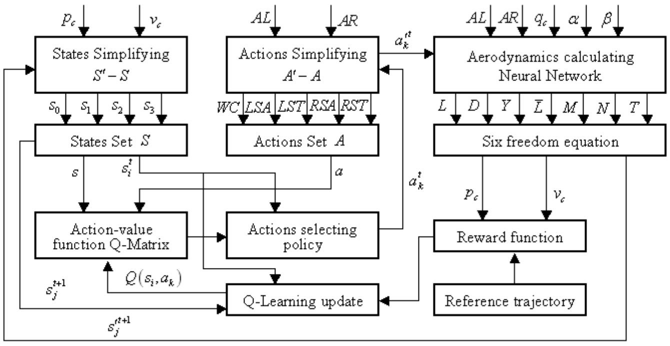

Fig.8. The RL Flight Control Architecture

The Reinforcement Learning Flight Control Architecture is composed of two sub-systems: Dynamic Behavior (DB) and Reinforcement Learning (RL).

The DB system including Aerodynamics Calculating Neural Network and Six freedom equation is used to work out the linear velocities p_c and angular velocities v_c in terms of the morphing control vectors: AL and AR , and p_c, v_c will make the RL system's current state s_i^t transfer to the next middle state $s_j'^{t+1}$.

The RL system is used to implement the iterative process of optimizing shape changes policy: s_1, s_2, s_3, and s_4 derived from p_c and v_c by the States Simplifying module make up of the States Set S ; WC, LSA, LST, RSA, and RST derived from AL and AR by the Actions Simplifying module make up of the Actions Set A ; The Action-value function Q-Matrix is formed based on the experiences of the probability that a particular state-action pair is the best choice, and is the most important module of the RL system; The Actions selecting policy module initially commands an arbitrary action a_k^t from the Actions Set A in term of the current state s_i^t and Q-Matrix, and a_k^t is transformed into $a_k'^t$ which represents the morphing control vectors for driving the DB system running; The effect of tracking trajectory is evaluated with the reward function; The action-value function updates itself in terms of the Q-Matrix, the current state s_i^t, the next state

s_j^{t+1}, the current action a_k^t , and the reward. The two sub-systems interact significantly during both the episodic learning stage, when the optimal shape change policy is learned, and the operational stage, when the plant morphs and tracks a trajectory.

4.3 Trajectory Tracking

The numerical example has been carried out based on both the RL flight control module implemented in the section 2 and section 3, and the RL flight control architecture in Fig.8 .

The results of simulating calculation are presented as follows: The effect of tracking a reference trajectory is shown in Fig.9a, and Fig.9b, 9c, 9d show the top view, side view, and front view, respectively. The gray broken line, red real line, and green real line represent the reference trajectory, the initial flight trajectory without learning, and the final flight trajectory after learning, respectively. The most bad error of tracking trajectory is greater than 100 meters in initial phase and may be limited to 50 meters when the reward of the RL module has reached stability.

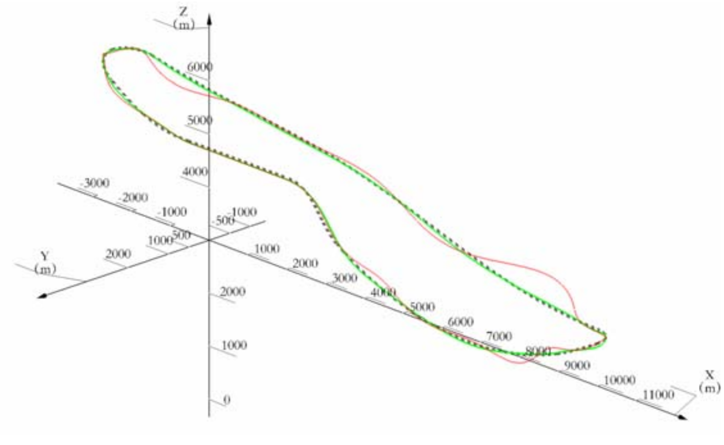

Fig.9a The Simulation Result of Tracking Trajectory

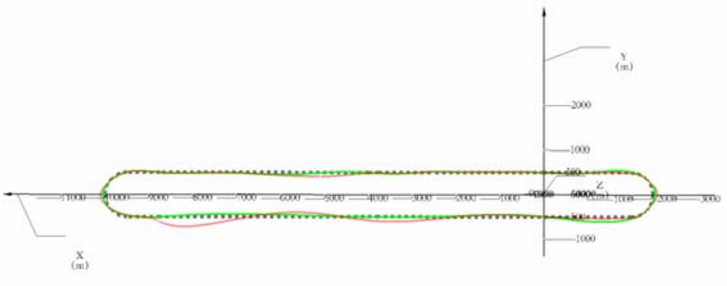

Fig.9b The Top View of the Simulation Results

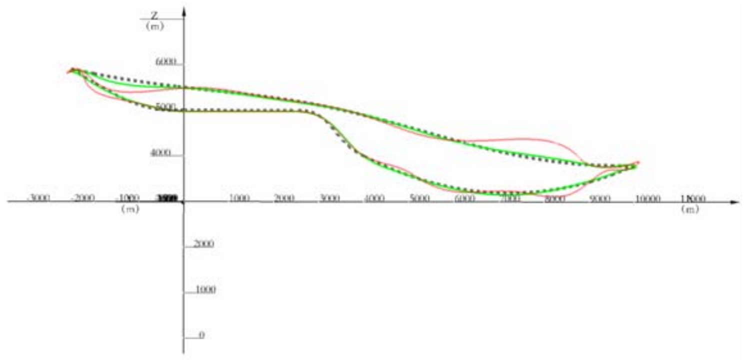

Fig.9c The Side View of the Simulation Results

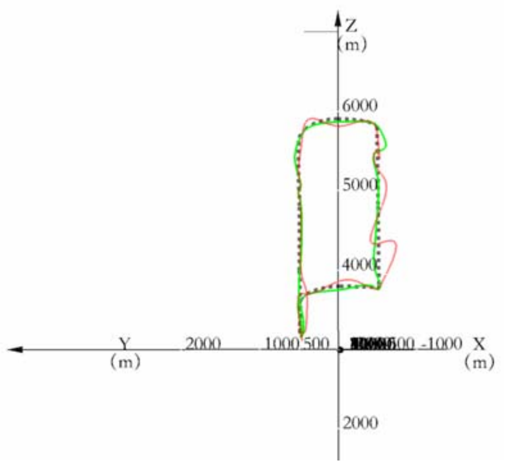

Fig.9d The Front View of the Simulation Results

In the numerical example, any experiences are actually induced to shorten the time of spending on the learning process of tracking the reference trajectory and it does not be absolutely unaware of the relationship between the flight conditions, rewards and the optimal shapes. A typical scenario is designed as:

The hypothetical morphing UCAV that can morph in both wing configuration and shape corresponding to flight condition along the trajectory experiences an attacking flight with four phases. The UCAV flights by employing straight wing in the cruise phase, by employing small-forward-bend wing in the glide and climb phase, and by employing large-forward-bend wing in the dive-bomb phase. In each flight phase, the UCAV controls Euler angles by driving those smart joints in two wings and tail-wings. By utilizing the 3D-MAX software, the flight simulation of the morphing UCAV has been made and shown in Fig.10.

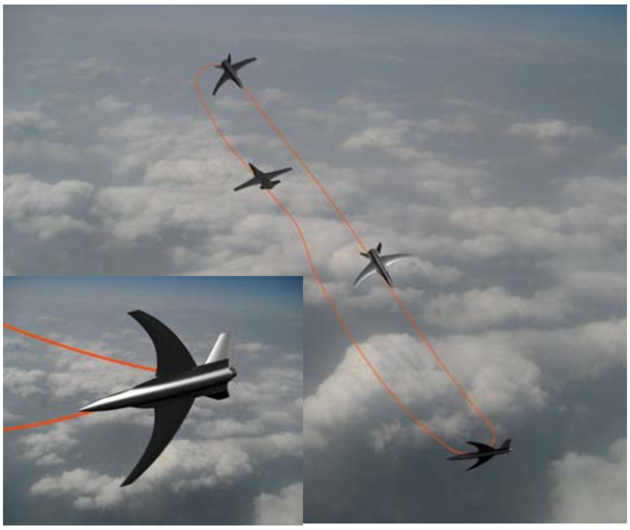

Fig.10. The Flight Simulation of the Morphing UCAV

5 Conclusions

This paper developed a Reinforcement Learning Control methodology for morphing aircraft, combining morph policy learning and dynamic behavior controlling.

An innovate morphing UCAV concept has been created to achieve the three Configuration changes (straight, small-forward-bend and large-forward-bend wing) and three Shape changes (bend, warp and twist) by controlling those smart joints in two wings.

The morphing control vectors that have be defined as the rotary angle displacements of the joint turning respectively round the three body axis and deflection of the tail-wing are used to describe the shape–change driving states of the morphing UCAV. Each shape–change driving state induces actually one only aerodynamic force and moment state of the morphing aircraft.

For establishing the complicated and nonlinear mapping from the morphing control vectors to those aerodynamic force and moment parameters, a three-layer neural network has been created. The shape–change driving states and the position states of the morphing UCAV are related by the calculating process from the neural network to the six degree of freedom equations.

For achieving the flight control to track a reference trajectory, the Reinforcement Learning Module of the Morphing UCAV has been implemented based on the Q-Learning method. For simplifying the research in this paper, the states and actions of the RL module have been designed by introducing a few transition parameters. The Action-value function Q-Matrix is formed based on the experiences of the probability that a particular state-action pair is the best choice.

The numerical example of simulating an attacking flight process has been carried out based on the RL system. The results of simulating calculation are shown that the error of tracking trajectory may be limited to a better level. However, the above result is still primary and can not be compared with traditional aircraft because it has not been considered in this paper that many factors such as the inertia and mass changes as the aircraft morphs into different shapes, the primary aerodynamic forces and moments excepted.

References

[1] Sharon L. Padula, James L. Rogers and David L. Raneyn. *Multidisciplinary Techniques and Novel Aircraft Control Systems*. AIAA 2000-4848.

[2] James Doebbler, Monish D. Tandale, John Valasek, Andrew J. Meade. *Improved Adaptive-Reinforcement Learning Control for Morphing Unmanned Air Vehicles*. AIAA 2005-7159.

[3] Holly A. Feldman. *Space-Based Antenna Morphing using Reinforcement Learning*. AIAA 2007-164.

[4] Brian C. Dutoi, Nathan D. Richards, Neha Gandhi and David G. Ward. *Hybrid Robust Control and Reinforcement Learning for Optimal Upset Recovery*. AIAA 2008-6502.

[5] Kenton Kirkpatrick and John Valasek. *Reinforcement Learning for Active Length Control of Shape Memory Alloy*. AIAA 2008-7280.

Copyright Statement