Two-Stage Reinforcement Learning-Based Upset Recovery Strategy for Aircraft

Huanhui Cao
School of Mechanical Engineering and Automation
Harbin Institute of Technology
Shenzhen, China

Given Name Surname
School of Mechanical Engineering and Automation
Harbin Institute of Technology
Shenzhen, China

Given Name Surname
School of Mechanical Engineering and Automation
Harbin Institute of Technology
Shenzhen, China

Given Name Surname
School of Mechanical Engineering and Automation
Harbin Institute of Technology
Shenzhen, China

Given Name Surname
School of Mechanical Engineering and Automation
Harbin Institute of Technology
Shenzhen, China

Hao Xiong*
School of Mechanical Engineering and Automation
Harbin Institute of Technology
Shenzhen, China
xionghao@hit.edu.cn

***Abstract*—something anasdd**

***Index Terms*—aircraft upset recovery, reinforcement learning, Twin Delayed Deep Deterministic Policy Gradient**

I. Introduction

Aircraft upset incidents are the highest risk to civil aviation for decades ago to now [1]. To this end, several researchers have devoted to address the aircraft upset issue [?]. Yildiz et al. [2] describes a novel finite-state conditional switching structure that enables autonomous recovery for a large envelope of loss-of-control conditions. Cunis et al. [3] proposed a loss of altitude minimizing economic model predictive control strategy for deep-stall recovery. The problem of aircraft spin recovery is addressed by solving a trajectory optimization problem via direct multiple shooting method in [4]. Although the above-mentioned approaches are adequate to deal with nonlinear dynamics of aircraft, they may fail to address the high complexity of the upset situation [5].

Reinforcement Learning (RL) is an approach that can deal with high complexity problems without an explicit model of the problem [6], leading to recovery strategies for aircraft suffering complex upset situation [?]. Dutoi et al. combined robust control and RL to address spin recovery problem [?]. The spin recovery performance of the achieved strategy outperforms the strategy obtained based on skilled pilots in some cases. Nonetheless, this study compressed the action space to reduce the computation load, leading to a limitation on agile spin recovery. Kim et al. [5] developed RL-based recovery strategy including 27 actions for angular rate arrest and nine actions for unusual attitude recovery for stable flat spin. The performance of the proposed RL-based strategy was compared with an optimal solution. However, the training of the RL-based strategy was not detailed in [5]. Zhu et al. [7] applied deep Q-network (DQN) and deep deterministic policy gradients (DDPG) to achieve spin recovery strategies and propose an exploring mechanism that dynamically selects between deterministic and stochastic exploration for DDPG. Whereas this study did not consider the uncertainties of the aircraft and the environment.

In this paper, we proposed a method based on RL to recover the aircraft back to steady level flight swiftly. When the aircraft is in an upset state, control the aircraft through RL and adjust the aircraft attitude so that the aircraft can resume smooth flight within a limited time.

This paper has the following major contributions.

- A pretrained-fine tuned RL method is proposed for random upset state recovery. Using this method, the agent can be trained successfully and quickly.
- A novel reward function is proposed to improve the performance of the training efficiency.
- We provide a comparison between PID and our proposed method for LOA(loss of altitude)-minimal recovery, and proved our method is quite more excellent than PID.

The rest of this paper is organized as follows. Section II introduces the preliminaries of this paper, including typical upset situations of aircraft and Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm. In section III, a TD3-based upset recovery strategy is proposed. Section IV presents the simulations to illustrate the proposed strategy. Finally, section V summarizes this paper.

*Corresponding author.

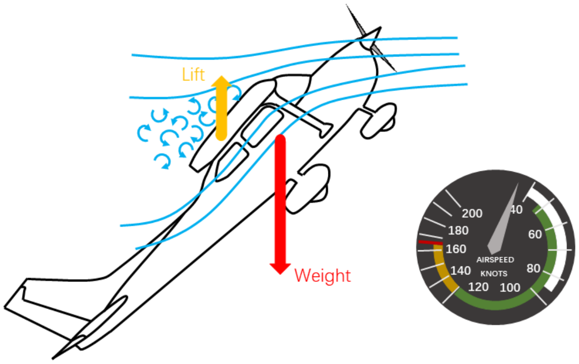

Fig. 1. The stall phenomenon of the aircraft

II. Preliminaries

A. Upset Situations of Aircraft

An upset situation of aircraft refers to an abnormal mode of the nonlinear dynamics that shows significantly altered steady-state responses and usually immediately precedes wing stall [3], such as stall and spin.

Upset states of the aircraft includes stall and spin [8], and spin is a nonlinear post-stall phenomena in which an aircraft develops a high rotational-rate and descends almost vertically in a helical trajectory.

a) Stall: [?] A stall is a condition in aerodynamics and aviation such that if the angle of attack increases beyond a certain point, then lift begins to decrease. Stalls in fixed-wing flight are often experienced as a sudden reduction in lift as the pilot increases the wing's angle of attack and exceeds its critical angle of attack. A stall does not mean that the engine have stopped working, or that the aircraft has stopped moving. The stall phenomenon is shown in Fig. 1.

b) Spin: Spin (as shown in Fig. 2) is one of the most complex aircraft maneuver and has been a subject of numerous research projects, tests and investigations since the early years of aviation. A spin is defined as an aggravated stall, which results in autorotation of an aircraft while descending in a helical pattern about the vertical spin axis. In an aggravated stall, one wing is stalled more than the other. The more stalled wing experiences less lift and more drag as compared to other and this imbalance of forces initiates autorotation and subsequent rapid decent of the aircraft [**?**]

B. Loss of Altitude

Loss of altitude (LOA) [3] is an important performance metric for upset recovery maneuvers and it can be exploited to enlarge the operational envelope during and after the maneuver, particularly at low altitudes. In this study, we use LOA to measure the performance of our method. In theory, the better our method performs, the less LOA is.

LOA is the difference from the initial altitude h_{ini} to the altitude h_{fin} when the aircraft is flying steady. It can be defined as:

$$LOA = |h_{ini} - h_{fin}| \tag{1}$$

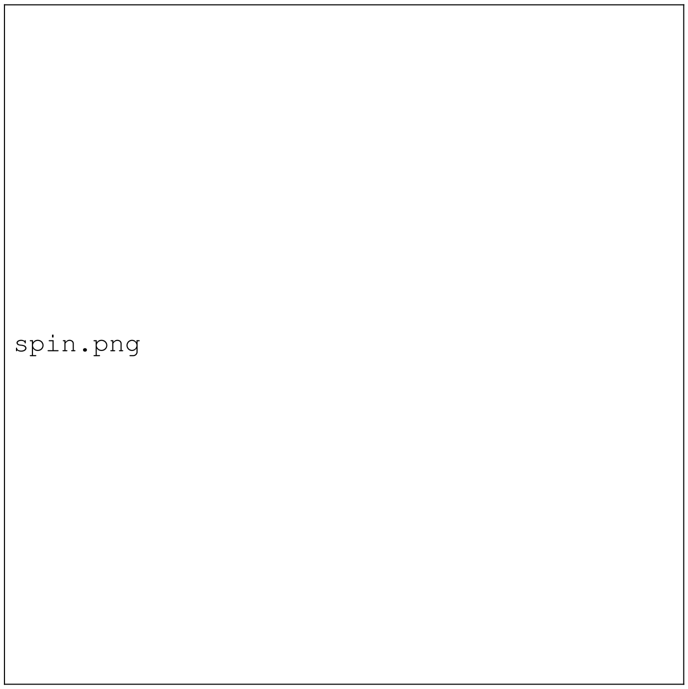

Fig. 2. The spin phenomenon of the aircraft

C. Reinforcement Learning and Twin Delayed Deep Deterministic Policy Gradient Algorithm

Reinforcement learning studies the paradigm of an agent interacting with the environment aiming to learn behaviors that maximize accumulated rewards. At time step t, the agent selects an action $a \in \mathcal{A}$ based on the current state $s \in \mathcal{S}$ with respect to its policy $\pi : \mathcal{S} \mapsto \mathcal{A}$. The agent receives a reward r and the state transfers to a new state s'. The agent aims to maximize the accumulated rewards $R_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i)$, where γ is a discount factor.

Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm [9] is a RL algorithm proposed for agents with continuous states and actions. TD3 is based on an actor-critic architecture taking the interplay between function approximation error in both policy and value updates into account. TD3 applies three novel approaches to address the overestimation issue of the critic network.

Apply a pair of critic networks. TD3 learns two Q-functions instead of one, Q_{ϕ_1}, Q_{ϕ_2}, and uses the smaller of the two Q-values to form the targets in the Bellman error loss functions.

$$y(r, s', d) = r + \gamma(1-d) \min_{i=1,2} Q_{\phi_{i,targ}}(s', a') \tag{2}$$

where $d = 0 \ or \ 1$. Then the parameters of both Q-value function ϕ_1 and ϕ_1 are updated by one step of gradient descent using:

$$\nabla_{\phi_i} \frac{1}{\mathcal{B}} \sum_{(s,a,s',r,d) \in \mathcal{B}} (Q_{\phi_i}(s,a) - y(r,s',d))^2 \tag{3}$$

where $i = 1, 2$ and $\mathcal{B}$ is a mini-batch sampled from the replay buffer D. Using the smaller Q-value for the target, and regressing towards that, helps decrease overestimation in the Q-function.

Delay policy updates. TD3 updates the policy and target networks less frequently than the Q-function, and we define the delayed frequency as p_d. The parameter of the policy network π_θ is updated by one step of gradient ascent to maximize the Q-value using:

$$\nabla_\theta \frac{1}{\mathcal{B}} \sum_{s \in \mathcal{B}} Q_{\phi_1}(s, \pi_\theta(s)) \tag{4}$$

Smooth target policy. In order to reduce the variance caused by over-fitting, TD3 uses a regularisation technique known as target policy smoothing. Ideally there would be no variance between target values, with similar actions receiving similar values. TD3 reduces this variance by adding a small amount of random noise to the target and averaging over mini batches. The range of noise is clipped in order to keep the target value close to the original action. The target actions are thus:

$$a'(s') = clip(\pi_{\theta_{targ}}(s') + clip(\epsilon, -c, c), a_{Low}, a_{High}) \tag{5}$$

where $\pi_{\theta_{targ}}$ is the target policy, action a satisfy $a_{Low} \leq a \leq a_{High}$, $\epsilon \in \mathcal{N}(0, \sigma)$.

III. Two-Stage Reinforcement Learning-Based Upset Recovery Strategy

In this section, a two-stage reinforcement learning-based upset recovery strategy is proposed for aircraft.

A. Problem Formulation

In the context of aviation, the term upset can be used to describe a variety of abnormal situations. As mentioned in Section I, if the classical methods usually provide simple and robust ways to control a system, most of the time they require a good knowledge of its dynamics and perform poorly when the level of uncertainty raises. Since the dynamics of the aircraft can get quite complex and difficult to model, a controller based on RL will be designed for its capacity to capture the complex behavior of physical system without the need to specify the laws of motion explicitly.

In order to control the aircraft using RL, we need to define the state space and action space of the agent.

The flight status can be obtained from sensors in the aircraft. Although we cannot directly obtain the flight status from sensors, we can obtain some related data from the master pilot's perspective. We noticed that our goal had no dependence on its location, so latitude and longitude can be ignored. Under comprehensive consideration, we define our state space S as a set of states $S_1, S_2, ..., S_n \in S$ as a collection of values corresponding to the following:$\{\omega, \kappa, \xi, p, q, r, h, v\}$:

$$S = \{s | s = [\omega, \kappa, \xi, p, q, r, h, v]\} \tag{6}$$

All of them are described in Table I:

Table I. The state data fed into the reinforcement learning model

Field	Description
ω	pitch
κ	roll
ξ	heading/yaw
p	rotational velocity of pitch
q	rotational velocity of roll
r	rotational velocity of heading
h	altitude
v	indicated airspeed of the aircraft

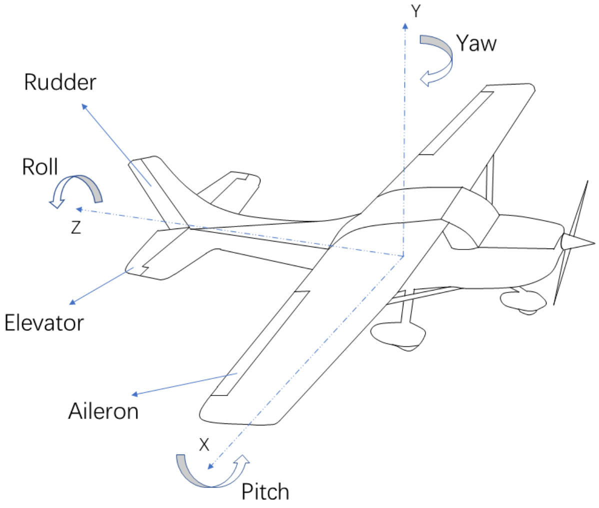

Fig. 3. Aircraft control surfaces

As for action space, we define it as a set of actions in 4 dimensions based on the high level controls available to a pilot: elevator, aileron, rudder and throttle. Among them, elevator, aileron, rudder are in the continuous range $[-1, 1]$, and the engine throttle is in $[0, 1]$. Thus, the action space can be defined as:

$$A = \{a | a = \delta_e, \delta_a, \delta_r, \delta_t\} \in [-1, 1]^3 \times [0, 1] \tag{7}$$

where $\delta_e, \delta_a, \delta_r, \delta_t$ denote elevator, aileron, rudder, throttle, respectively. Fig. 3 shows the main control surfaces.

B. Reward Function

In RL, reward is an environmental feedback that is given to the agent for every action it takes. The design of reward function is crucial as it governs not only the convergence time but also the quality of the convergent point of the learning model. In this work, the reward function is designed such that the agent earns highest reward as soon as possible for suggesting a maneuverable resolution that successfully recover the aircraft back to steady level flight under a upset initial condition.

Since our end goal is to recover the aircraft back to steady level flight which is a sparse reward problem, leading to the slow convergence rate of training. With the idea of rewarding shaping [10], we define the reward function as follows:

$$R(s, a, s') = T(s') + F(s, a, s') \tag{8}$$

where $T(s')$ is the termination reward and $F(s, a, s')$ is a bounded function.

The termination reward is the reward obtained when the next state is crushed or is successful to recover. It is defined as:

$$T(s') = \begin{cases} T_1, & \text{if the aircraft is crushed} \\ T_2, & \text{if the aircraft is successful to recover} \\ 0, & else \end{cases} \quad (9)$$

where T_1 is a very small negative value while T_2 is a large positive value. When the steady state of aircraft hold for more than n time steps, the aircraft is successful to recover.

$F(s, a, s')$ has the form:

$$F(s, a, s') = \Gamma\Phi(s') - \Phi(s) \quad (10)$$

where $\Phi(s)$ is a function over states, Γ is a constant. $\Phi(s)$ is defined as:

$$\Phi(s) = C(s) + P(s) \quad (11)$$

where $C(s)$ is the action reward which is related to the elevator and aileron and it is a variable integer. It links the current action and expected action with the current attitude change. When the current action is consistent with the expected action, $C(s)$ will increase, otherwise it will decrease.

$P(s)$ is the aircraft attitude reward, it is given by (12):

$$P(s) = \sum_{i=1}^{8} p_i |s_{norm}[i] - s_{targ,norm}[i]| + p_0 |LOA_{norm}| \quad (12)$$

where $p_i (i = 0, ..., 8)$ are non-positive weight coefficients, and p_0 is much smaller than $p_i (i = 1, ..., 8)$. s_{norm}, $s_{targ,norm}$, LOA_{norm} are the normalization of state s, target state s_{targ}, LOA of aircraft, respectively.

C. Two-Stage RL-Based Upset Recovery Strategy

For complex scenarios, successfully training an agent is very challenging. What's more, even if the training is successful, the convergence speed may be very slow. Therefore, we develop a two-stage RL-based strategy.

The whole process has two phases: model pre-training and model tuning. We apply RL algorithm to pre-train a guidance agent and then obtain a guidance policy in the pre-trained phase. The purpose of this is to pre-train a guidance agent to prepare for subsequent tuning. Based on the guidance agent, it can greatly speed up the training process and save a lot of time.

After the pre-trained process, we enter the tuning phase. Based on the guidance agent, an efficient exploration strategy is set up to train the agent. We shape the reward function using the method as described in Section III-B. The overall framework of training algorithm is shown in Fig. 4.

D. Pre-trained-fine tuned TD3-based training framework

With the idea of Section III-C, firstly, we apply TD3 algorithm to pre-train a guidance agent. Specifically, we added the uncertainty of the aircraft model and strengthened the uncertainty of the environment: we randomized the total mass of the aircraft and the harshness of the environment within the allowable range. Both critic network and policy network are initialized in the pre-trained phase. Then in the tuning phase, the uncertainty of the aircraft model is cancelled. Based on the guidance agent, We continue to use TD3 algorithm to train the agent. This method is more efficient than directly using TD3 in our scenarios. The pseudocode is shown in Algorithm 1.

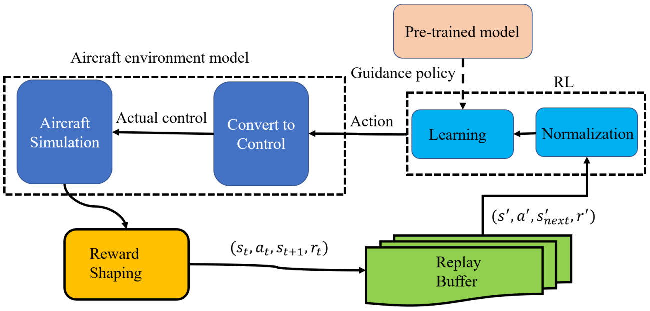

Fig. 4. The overall framework of training process.

IV. Experiments

A. Environment Set Up

We tested our well-trained agent in X-Plane which is a flight simulation software. As a professional flight simulation software, X-Plane includes different kinds of aircraft including commercial aircraft and military aircraft. Moreover, the software is very extensible, allowing developers to expand functions arbitrarily, such as adding their own designed airplanes or landscapes. As X-Plane can provide sophisticated dynamic models of various types of aircraft with a blade element approach along with a realistic flying environment, it can emulate the flight conditions relatively reliably and is thus chosen as the flight simulator for the algorithm test in this study.

The integrated simulation system, utilizing X-Plane 11.0 and Tensorflow-gpu 2.4, runs under 64-bit Windows 10 on a PC with 32G RAM, 3.4GHz Frequency, and a RTX 3070 GPU for training of neural networks.

The test aircraft is N172SP, as shown in Fig. 5. Since our goal is to test recovery performance of our RL method, N172SP needs to be deliberately upset. We mainly test two classical upset scenarios as described in Section II: stall and spin.

a) Stall: The stall state is set up by initializing the aircraft state as: $s = [\omega, \kappa, \xi, p, q, r, h, v] = [90, 90, 90, 90, 90, 90, 2500, 0]$.

b) Spin: We initialize the aircraft state as: $s = [\omega, \kappa, \xi, p, q, r, h, v] = [90, 90, 90, 90, 90, 90, 2500, 0]$, followed by a continuous excitation $a = [\delta_e, \delta_a, \delta_r, \delta_t] = [1, 1, 1, 0]$.

To study the effects of wind on upset recovery, simulations are also carried out with windy conditions. In X-Plane, we can easily set the windy condition via the settings panel and UDP(User Datagram Protocol).

For real simulation, all data of states are not precise data, they are obtained from panel data that the pilot can see. This

Algorithm 1: Pre-trained-fine tuned TD3-based algorithm

```
Input: initial policy parameters $\theta$, Q-function
       parameters $\phi_1$, $\phi_2$, empty replay buffer $\mathcal{D}$
Set target parameters equal to main parameters
  $\theta_{targ} \leftarrow \theta$, $\phi_{targ,1} \leftarrow \phi_1$, $\phi_{targ,2} \leftarrow \phi_2$;
Set Pre-trained aircraft model parameters and
  environment parameters;
Pre-train a guidance agent using original TD3;
Set the guidance agent as the baseline agent;
Reset aircraft model parameters;
for episode=1 to M do
    Receive initial observation state $s_1$ and normalize
      it;
    for t=1 to T do
        Select action $a$ according to the guidance
          policy;
        Convert the action into the aircraft control;
        Execute action $a$ in aircraft environment, and
          obtain reward $r$ and new state $s'$;
        Normalize the state $s'$;
        Get $r^{rs}$ using $r^{rs} = T(s') + F(s, a, s')$;
        Store transition $(s, a, r^{rs}, s', d)$ in $\mathcal{D}$;
        Sample a mini-batch of transitions:
          $\mathcal{B} = \{(s, a, r^{rs}, s', d)\}$ from $\mathcal{D}$;
        Compute target actions $a'(s') =$
          $clip(\pi_{\theta_{targ}}(s') + clip(\epsilon, -c, c), a_{Low}, a_{High})$;
        Compute targets $y(r^{rs}, s', d) =$
          $r^{rs} + \gamma(1-d)\min_{i=1,2} Q_{\phi_{i,targ}}(s', a')$;
        Update Q-functions using
          $\nabla_{\phi_i} \frac{1}{\mathcal{B}} \sum_{(s,a,s',r^{rs},d)\in\mathcal{B}} (Q_{\phi_i}(s,a) - y(r^{rs}, s', d))^2$;

        if t mod $p_d$=0 then
            Update policy using
              $\nabla_\theta \frac{1}{\mathcal{B}} \sum_{s\in\mathcal{B}} Q_{\phi_1}(s, \pi_\theta(s))$;
            Update the target network:
              $\phi_{targ,i} \leftarrow \tau\phi_{targ,i} + (1-\tau)\phi_i, \quad i = 1, 2$
              $\theta_{targ} \leftarrow \tau\theta_{targ} + (1-\tau)\theta$
```

means that the data obtained is inaccurate, but this is in line with reality.

B. Algorithm Set Up

As for the algorithm set up, the learning rate is 1.0×10^{-4}. The actor network has three hidden layers with 64, 64, and 32 units, respectively. The critic network has two hidden layers with 64, 64 units, respectively. The time step T is 500 and the minibatch size $\mathcal{B}$ is 64. The number of episodes M is 400 or 800 under non-wind or windy conditions.

The values range of the state space are : $\omega, \kappa, \in [-180°, 180°]$; $\xi \in [0°, 360°]$; $p, q, r \in [-90°/s, 90°/s]$; $h \in [0\ m, 5000\ m]$; $v \in [0\ m/s, 160\ m/s]$.

The coefficients of the reward function discussed in Section III-B are set: $T_1 = -1000$, $T_2 = 1000$, $n = 60$, $\Gamma = 0.99$, $p_0 = -1$, $p_1, p_2, p_3, p_6 = -0.1$, $p_4, p_5, p_7, p_8 = -0.5$, initial value of $C(s)$ is 1.

Fig. 5. The test aircraft: N172SP

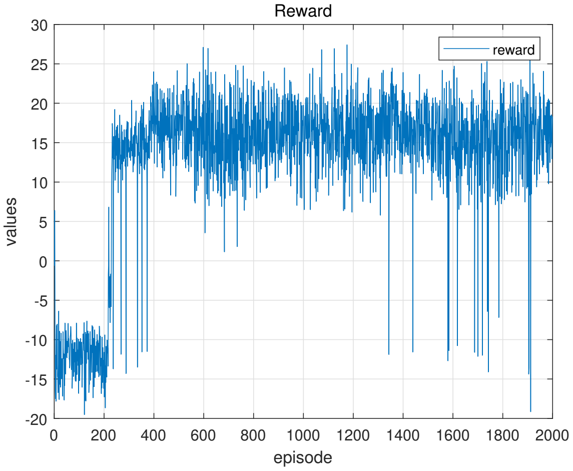

Fig. 6. The average reward during pre-trained process

C. Tests

Firstly, we save reward data for each iteration pre-trained and training, and perform simulation analysis on these data. These results are shown in Fig. 6 and Fig. 7.

In the pre-trained phase, about 200 episodes begin to converge, the average reward range of the simulation results is approximately 2000. It can be seen from the simulation results that the pre-trained model has obtained a sub-optimal policy since there are still sometimes unstable flight situations. This is also in line with the results we expected. it is almost impossible to obtain the optimal policy in the pre-trained phase, but the sub-optimal policy obtained assist the subsequent tuning.

In the tuning phase, based on the pre-trained model, We trained our agent in windy and non-wind conditions. As shown in Fig. 7, only less than 10 iterations are needed to make the training successful in both windy and non-wind conditions, which greatly improves the training efficiency. Besides, the

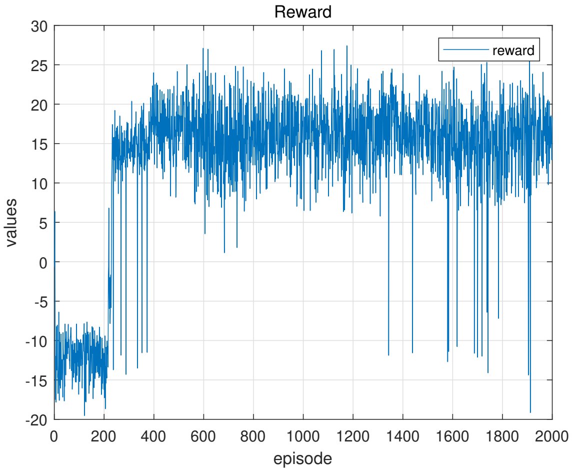

Fig. 7. The average reward during training process

training speed of non-wind is faster than the training speed of windy. Each of them has obtained the optimal policy.

Next, the performance of our method is compared with a traditional PID controller with windy and non-wind interference under stall and spin states.

(1) Stall recovery

The well-trained agent and the traditional PID controller are tested for 10 simulations under the stall state. The corresponding attitude changes and airspeed changes are shown in Fig. 8 - Fig. 10. All of angles are regulated to the target region which stands for steady state of the aircraft and the airspeed has returned to the normal controllable range. But the performance of PID controller is worse than the well-trained agent especially in the windy condition. For the loss of altitude(LOA) shown in Fig. 11, the performance of the well-trained agent is better than PID and the recovery speed of the well-trained agent is also faster than PID.

(2) Spin recovery

The same as the stall state, the well-trained agent and the traditional PID controller are tested for 10 simulations under the spin state. Fig. 12 - Fig. 14 showed the angles and airspeed changes. We can easily know all of angles and airspeed are regulated to the target region, but relatively speaking, the well-trained agent is more robust than PID controller. As expected, for the LOA shown in Fig. 15, the performance of the well-trained agent is also better than PID.

The simulation results show that our proposed method performs more efficiently and excellently than PID. It is very useful to return the aircraft to a steady state from any upset state.

V. Conclusion

References

[1] Christine M Belcastro, John V Foster, Gautam H Shah, Irene M Gregory, David E Cox, Dennis A Crider, Loren Groff, Richard L Newman, and David H Klyde. Aircraft Loss of Control Problem Analysis and Research Toward a Holistic Solution. *Journal of Guidance, Control, and Dynamics*, 40(4):733–775, 4 2017.

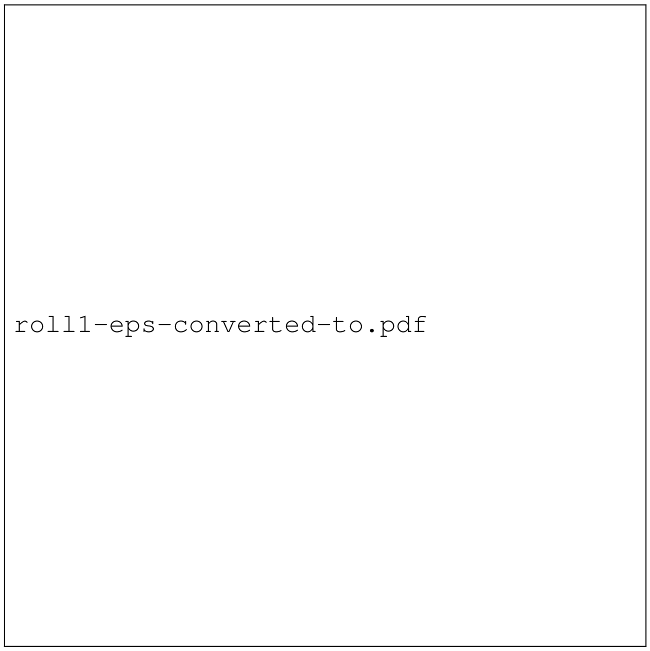

Fig. 8. The roll angle changes

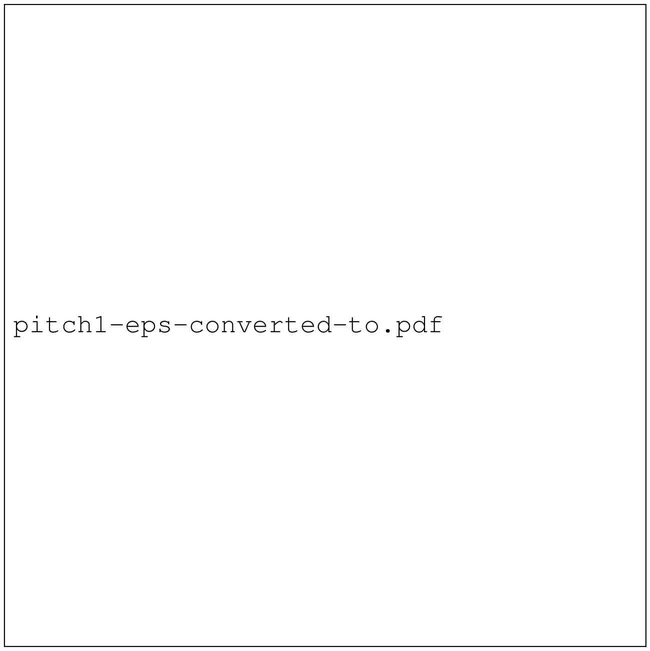

Fig. 9. The pitch angles changes

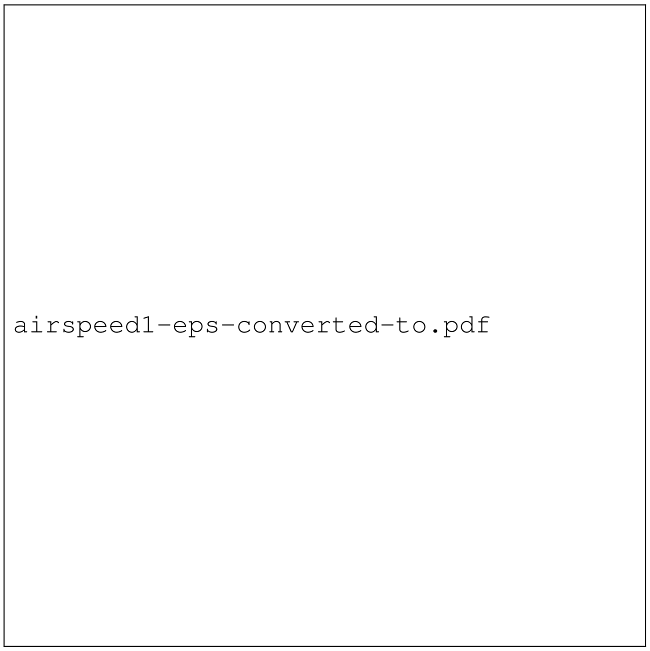

Fig. 10. airspeed

Fig. 11. The altitude angle changes

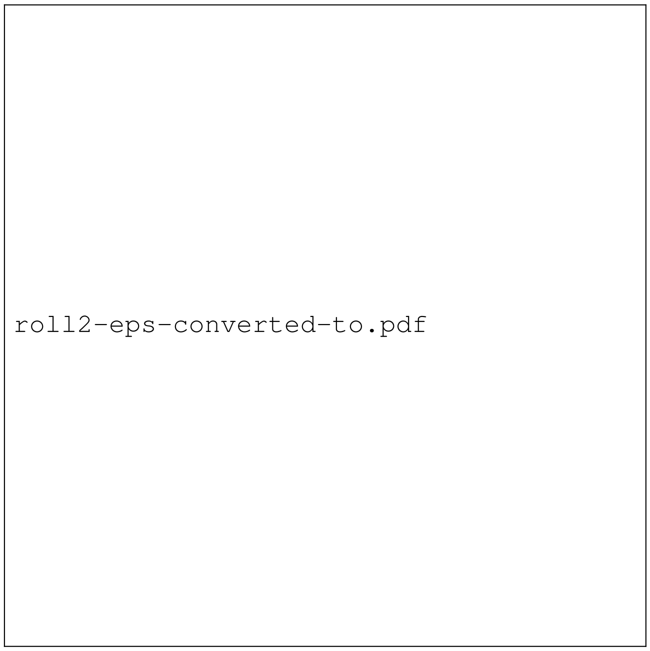

Fig. 12. The roll angle changes

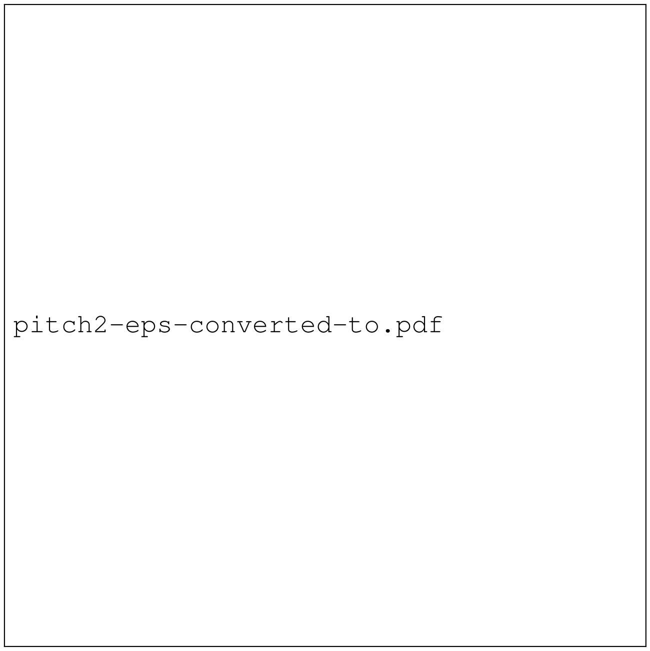

Fig. 13. The pitch angles changes

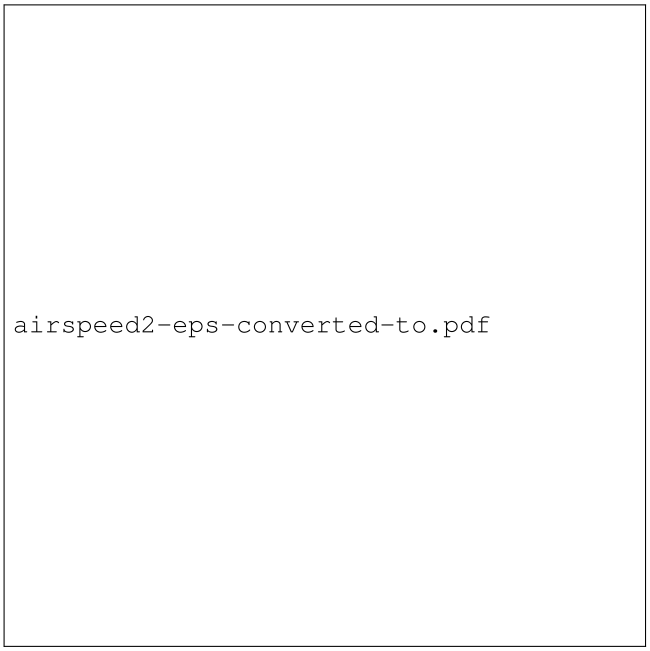

Fig. 14. airspeed

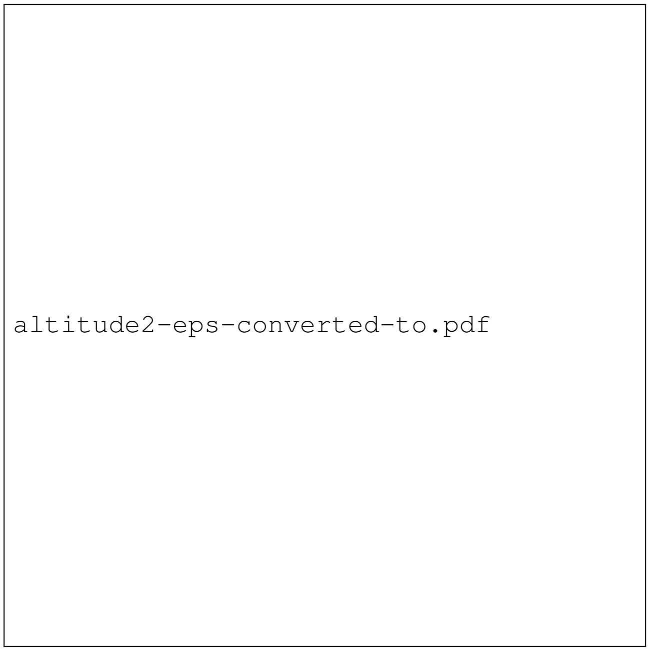

Fig. 15. The altitude angle changes

[2] Anil Yildiz, M Ugur Akcal, Batuhan Hostas, and N Kemal Ure. Switching Control Architecture with Parametric Optimization for Aircraft Upset Recovery. *Journal of Guidance, Control, and Dynamics*, 42(9):2055–2068, 4 2019.

[3] T Cunis, D Liao-McPherson, J Condomines, L Burlion, and I Kolmanovsky. Economic Model-Predictive Control Strategies for Aircraft Deep-stall Recovery with Stability Guarantees. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 157–162, 2019.

[4] D.M.K.K. Venkateswara Rao and Tiauw H Go. Optimization of aircraft spin recovery maneuvers. *Aerospace Science and Technology*, 90:222–232, 2019.

[5] Donghae Kim, Gyeongtaek Oh, Yongjun Seo, and Youdan Kim. Reinforcement Learning-Based Optimal Flat Spin Recovery for Unmanned Aerial Vehicle. *Journal of Guidance, Control, and Dynamics*, 40(4):1076–1084, 6 2016.

[6] Zhuangdi Zhu, Kaixiang Lin, and Jiayu Zhou. Transfer learning in Deep Reinforcement Learning: A survey. *arXiv*, pages 1–22, 2020.

[7] Y Zhu, H Liu, B Ren, H Duan, X She, and Z Wu. A Model-free Flat Spin Recovery Scheme for Miniature Fixed-wing Unmanned Aerial Vehicle. In *2019 IEEE International Conference on Unmanned Systems (ICUS)*, pages 623–630, 2019.

[8] Bilal Malik, Jehanzeb Masud, and Suhail Akhtar. A review and historical development of analytical techniques to predict aircraft spin and recovery characteristics. *Aircraft Engineering and Aerospace Technology*, 92(8):1195–1206, 1 2020.

[9] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing Function Approximation Error in Actor-Critic Methods. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1587–1596. PMLR, 2018.

[10] Andrew Y Ng, Daishi Harada, and Stuart J Russell. Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, page 278–287, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.