AIAA SciTech Forum
January 3-7, 2022, San Diego, CA & Virtual
AIAA SCITECH 2022 Forum

10.2514/6.2022-2442

Hazard Avoidance Landing of Loaded Parafoil using Deep Reinforcement Learning

Junwoo Park*, Seongheon Lee†, Yehyun Kim‡, and Hyochoong Bang§
Korea Advanced Institute of Science and Technology, Daejeon, 34141, Korea

This study proposes a data-driven autonomous landing hazard avoidance technique (ALHAT) of a parafoil that carries a payload of a large aerospace platform and lands on a non-hazardous region. Steering law of the parafoil is realized by applying soft actor-critic(SAC) deep reinforcement learning algorithm to carefully tailored Markov decision process which implicitly encompass parafoil states, ground hazard population and their evolution in time. Model-based approaches to parafoil guidance and control are subject to both fidelity of dynamic model and accuracy of model parameters all of which are hardly precisely known. Thus, the research aims at deriving near end-to-end steering logic of parafoil brakes to avoid hazard landing in a data-driven manner. Multiple grayscale images over several time steps stacked in channel direction are utilized as observation of Markov decision process(MDP) to imply temporal evolution of hazard while each image is a projection of hazard population onto parafoil viewpoint whose intensity is adjusted according to parafoil height. The reward function is shaped to let agent value avoiding hazards using the spatially weighted mask. Results of numerical simulation show that the parafoil ALHAT is achievable without any prior knowledge about parafoil dynamics, hazard transition, and human supervision.

Nomenclature

x_C, y_C, z_C	=	position of confluence point resolved in the inertial frame
u_C, v_C, w_C	=	velocity of confluence point resolved in confluence frame
ϕ, θ, ψ	=	roll, pitch, and yaw angle of either parafoil canopy or payload frame with respect to inertial frame
p, q, r	=	body angular rate of either parafoil canopy or payload frame
r, R	=	relative position of canopy or payload frame from confluence point, and its skew-symmetric matrix
M, I	=	mass and inertia matrix
$F^{\mathrm{A}}, M^{\mathrm{A}}$	=	aerodynamic force and moment of canopy or payload
o, a, r	=	observation, action, and reward of reinforcement learning framework
G, U, V	=	gray scale image of proposed observation design, and its width and height
W	=	2-dimensional mask that spatially weights hazard distribution
L, k, ϵ	=	shaping parameters of logistic reward function.
Subscripts		
t	=	given time step
$\mathcal{I}, \mathcal{P}, \mathcal{B}, \mathcal{C}$	=	the inertial frame, parafoil canopy frame, payload body frame, confluence point frame
$\mathcal{CP}, \mathcal{CB}$	=	from confluence point to parafoil center, and from confluence point to payload center
F	=	apparent term when jointed with mass or inertia

I. Introduction

TECHNICAL development of aerospace systems including small reusable launch vehicles or personal aerial vehicles(PAV) has expanded our capability on Earth and has shown their applicability to space. However, such aerial vehicles easily turn into potential threats whenever their full controllability is lost due to systematic and/or operational failures

*Ph.D. Candidate, Department of Aerospace Engineering, KAIST, junwoopark@kaist.ac.kr, Student Member of AIAA.
†Ph.D., Department of Aerospace Engineering, KAIST, skynspace@kaist.ac.kr, Member of AIAA.
‡Ph.D. Candidate, Department of Aerospace Engineering, KAIST, yehyunkim@kaist.ac.kr, Student Member of AIAA.
§Professor, Department of Aerospace Engineering, KAIST, hcbang@kaist.ac.kr, Member of AIAA.

induced by actuator fault, communication loss, battery outage, or human error. The failure situation drives the vehicle to land on or to fall onto an unexpected site causing subsequent damages. While the fault-tolerant control technique applied to them [1, 2] seeks for the solution, it mostly trades off control of less significant axes to secure that of major axes. Moreover, its application is system dependent, and thus it cannot be a general remedy.

Parafoil, or a steerable parachute, can help a wide range of aerospace systems land safely on the ground using the littlest energy even in their emergency situations with sufficiently slow dynamics [3]. It becomes more efficient when the target platform gets larger or heavier in the context that retro-thrusting or driving actuators to rescue them from expected crash requires a huge amount of energy and/or come at a high cost. Since either a parafoil or a parachute is a well-known method of an assistant system that slows the aerial vehicles down during their entry, descent, and/or landing(EDL) phase [4], a number of aerospace systems utilized it as their primary tool of passive decelerator or lander, e.g. Skycrane that deployed NASA's Perseverance rover on Mars [5], parachute recovery system of NASA's Orion capsule [6], Crew V2 SpaceX, or Boeing's CST-100 Starliner [7]. It is worth mentioning that safety backup is always suggested for multiple redundancy and reliability of the huge aerospace system and safety backup realized by parachute prevents catastrophic events from happening as it gets decelerated by huge aerodynamic drag trading off additional substructures.

Highly practical application of large parafoil should be a safe recovery of reusable rockets, emergency landing of near-Earth flying objects, or any civil unmanned aerial vehicles even small failures of which can cause significant damages on both themselves and the surroundings. Since the landing of those platforms onto the substantially hazardous area or unprepared terrain [8], e.g. middle of the ocean, thick forest, or urban area, demands additional costly works for the retrieval, we propose a near-Earth version of autonomous landing hazard avoidance technique(ALHAT) [9] using a steerable aerodynamic decelerator [10], parafoil, as an attached module to them. Such a system becomes especially more effective in near-Earth conditions where aerodynamic forces take a significant portion of external force.

This study considers the realization of the system using reinforcement learning(RL). Classical model-based guidance/control of a parafoil is vulnerable to the accuracy of state estimation and model parameters verification of which must go with experimental and/or empirical evidence [3]. Apparent mass and inertia, which are one of the major components in modelling parafoil dynamics, mostly depend on Lissaman's work [11] that involves several assumptions and approximations. Moreover, the nonlinear dynamics of parafoil relies significantly on numerous aerodynamic coefficients which also necessitate flight experiment or wind tunnel test. Such parameters are hardly precisely known or valid under only limited situations. Therefore, it makes more sense to steer parafoil in a data-driven manner. This study details the end-to-end autonomous landing of parafoil and its payload avoiding potential threats on the ground. Without any specific prior knowledge of parafoil dynamics, hazards population, and their time evolution, the RL agent learns better steering commands in avoiding ground hazards at the moment by interacting with a simulated environment. As the environment is enveloped and the RL agent communicates with it only by means of observation, reward, and action, it's possible to learn more from the real-world data when properly post-processed. The study utilizes soft actor-critic(SAC) deep RL algorithm to train RL agent that produces hazard-avoiding steering commands.

A. Related Works

Several model- and rule-based guidance, control, and/or path planning of a parafoil were studied in application to modern airborne delivery systems [12] especially in tactical purposes. Works of Slegers et al. [3, 13] are based on model predictive control(MPC) to make parafoil track trajectory by minimizing certain error metrics between current trajectory and designated one. In order for the MPC to be practical, however, significant model reduction and linearization were made upon the dynamic model and the identification of aerodynamic coefficients is required. Bergeron et al. [14] and Ward et al. [15] use the concept of glide slope control(GSC) to make an approach to the goal based on estimated gliding slope and wind effects, yet they provide no relationship with ground hazards. While each study has its own strength, most of them require additional wind estimation module or need to exploit knowledge of wind conditions [16, 17]. From the authors' best knowledge, only [12] has addressed the entire problem similar to us that incorporates terrain, obstacle avoidance and parafoil guidance, yet it requires an explicit notion of wind, and involves many heuristics.

Meanwhile, a number of studies that makes use of reinforcement learning techniques to guide multirotor vehicles show very promising results. Lee et al. [18] and Koch et al. [19] addressed multirotor UAV's attitude control problem using RL framework. [19] has developed an inner-loop controller using various state-of-the-art RL algorithms showing that the approach can resolve the problems with fast dynamics. [18] also used RL in developing attitude control of quadrotor vehicle that works as an adaptive gain tuning module for classical PID controller. Especially in application to automatic landing of multirotor UAVs, [20], [21], and [22] have solved similar problems using slightly different problem settings. [20] proposed an end-to-end control for landing problems using a down-looking image. Rather than using a

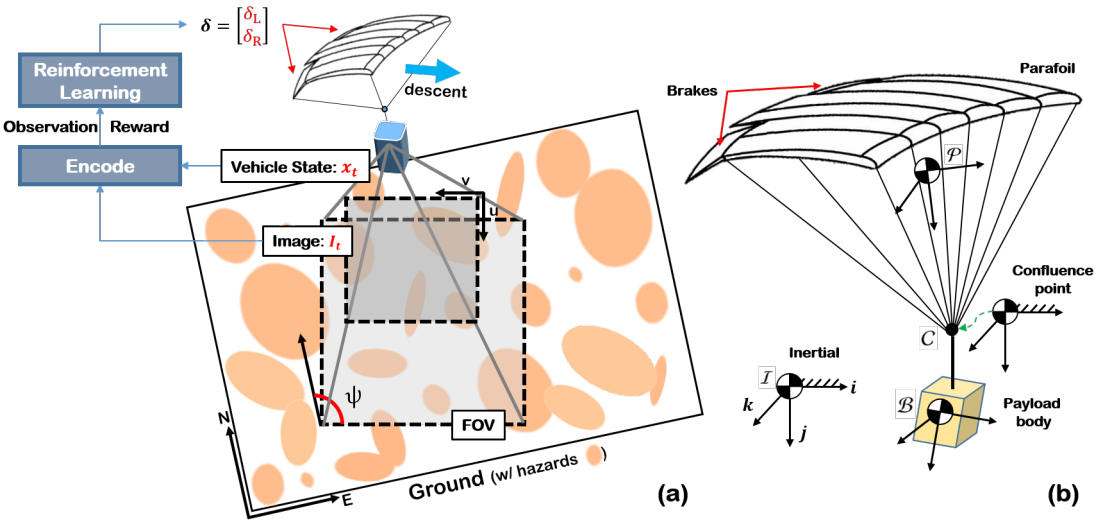

Fig. 1 (a) Schematics of parafoil ALHAT problem (b) 9-DOF parafoil system frames of reference

hand-crafted geometric features of the image, it utilizes raw image as it is. Both [21] and [22] use RL for generating high-level commands, e.g. velocity or position, assuming that inner control is available. They showed that RL agent trained in a simulated environment is robust enough to be deployed in the real-world by conducting flight experiments that examine the landing performance of multirotor UAV on a moving target.

B. Problem Statement

A parafoil is composed of three parts as shown on the right side of figure 1; parafoil canopy $\mathcal{P}$, payload body $\mathcal{B}$, and massless confluence point C with which both the canopy and payload are rigidly interconnected. The vehicle state is

$$\boldsymbol{x} = \left[\boldsymbol{x}_C^T, \boldsymbol{v}_C^T, \Psi_{\mathcal{P}}^T, \boldsymbol{\omega}_{\mathcal{P}}^T, \Psi_{\mathcal{B}}^T, \boldsymbol{\omega}_{\mathcal{B}}^T\right]^T \tag{1}$$

where $\boldsymbol{x}_C$ and $\boldsymbol{v}_C$ denote position and velocity of C in the inertial frame $\mathcal{I}$, Ψ and $\boldsymbol{\omega}$ denotes Euler angles and body angular rates of corresponding subscripts, all of whose elements are listed in (2). Note that $\boldsymbol{x}$, $\boldsymbol{x}_C$, and $\boldsymbol{v}_C$ in (1) are expressed in boldface.

$$\boldsymbol{x}_C = [x_C, y_C, z_C]^T, \boldsymbol{v}_C = [u_C, v_C, w_C]^T, \Psi = [\phi, \theta, \psi]^T, \boldsymbol{\omega} = [p, q, r]^T. \tag{2}$$

The parafoil is gliding from given initial state and its descent trajectory will vary according to the deflection of parafoil brakes $\boldsymbol{\delta} = [\delta_L, \delta_R]^T$ located at either end of the parafoil canopy. Detailed dynamics of (1) incorporating $\boldsymbol{\delta}$ will be given in section II.A. The parafoil is considered to be equipped with a down-looking camera aligned with its heading, and let's denote the image acquired from it at time step t as I_t. Moreover, it is assumed that hazards within images are fully identified with the help of detection algorithms [23, 24], and/or prescribed database as in [25] while the height of the ground hazard is neglected.

The problem this study targets is to derive meaningful steering logic of brakes, or also known as policy $\pi(\boldsymbol{\delta}|\cdot)$, out of information from (1) and I_t that will guide parafoil and its payload to descend and to avoid ground obstacles identified as potential hazards using deep reinforcement learning. This paper directly encodes (1) and I_t in a generic sense in order to derive the end-to-end steering logic of parafoil brakes from it. We propose observation as stacked post-processed grayscale images to encompass all the necessary information for hazard avoidance landing in a compact manner while detailed proposals will be given in section III.A, and section III.B. The RL agent interacts with the environment which includes parafoil, true ground hazards, and their time evolution as a whole; by exercising $\boldsymbol{\delta}$, observing partial information of down-looking images combined with vehicle state, and getting rewards. Left side of figure 1 highlights the schematic of the problem.

II. Theoretical Background

A. Dynamic Model of Parafoil

Among many representation models for parafoil dynamics including 4-degrees of freedom(DOF) [26], or 6-DOF model [27], 9-DOF model [28–30] is widely used to approximate parafoil specific dynamic characteristics. Newton-euler equations that describe translational and rotational motion of a rigid body is applied to both $\mathcal{P}$ and $\mathcal{B}$. Since translational motions of $\mathcal{P}$ and $\mathcal{B}$ are subject to C, those are expressed in terms of translational motion of C using corresponding kinematic equations. State variable $\boldsymbol{x}$ in (1) fully describes parafoil system at a given time, while time derivatives of $\boldsymbol{x}_C$ and Ψ in (1) are given as

$$\dot{\boldsymbol{x}}_C = \boldsymbol{v}_C, \dot{\Psi} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \boldsymbol{\omega} \tag{3}$$

where latter equation is valid for both $\mathcal{P}$ and $\mathcal{B}$. Dynamic equation for $\boldsymbol{v}_C$ and $\boldsymbol{\omega}$s are

$$\begin{bmatrix} (M_\mathcal{P}+M_F)T_I^\mathcal{P} & -(M_\mathcal{P}+M_F)R_{C\mathcal{P}} & 0 & -T_I^\mathcal{P} \\ 0 & I_\mathcal{P}+I_F & 0 & R_{C\mathcal{P}}T_I^\mathcal{P} \\ M_\mathcal{B}T_I^\mathcal{B} & 0 & -M_\mathcal{B}R_{C\mathcal{B}} & T_I^\mathcal{B} \\ 0 & 0 & I_\mathcal{B} & -R_{C\mathcal{B}}T_I^\mathcal{B} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{v}}_C \\ \dot{\boldsymbol{\omega}}_\mathcal{P} \\ \dot{\boldsymbol{\omega}}_\mathcal{B} \\ F_C \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \\ B_4 \end{bmatrix}. \tag{4}$$

In (4), $T_I^\mathcal{P}$ denotes coordinate transformation matrix from inertial frame $\mathcal{I}$ to $\mathcal{P}$ as represented in (5) where $C_\cdot$ and $S_\cdot$ are shorthand notation for corresponding trigonometric functions $\cos(\cdot)$ and $\sin(\cdot)$. $T_I^\mathcal{B}$ can be achieved in the same manner by replacing $\mathcal{P}$ of (5) into $\mathcal{B}$. F_C denotes internal force acting on interconnected point C, and B_i $(1 \le i \le 4)$ are given in (6). M and I denote mass and inertia matrix of corresponding subscript $\mathcal{P}$ and $\mathcal{B}$, $R_{C\mathcal{P}}$ denotes skew-symmetric matrix of relative position of $\mathcal{P}$ from C denoted as $r_{C\mathcal{P}}$, and $R_{C\mathcal{B}}$ denotes that of $r_{C\mathcal{B}}$.

$$T_I^\mathcal{P} = \begin{bmatrix} C_{\theta_\mathcal{P}}C_{\psi_\mathcal{P}} & C_{\theta_\mathcal{P}}S_{\psi_\mathcal{P}} & -S_{\theta_\mathcal{P}} \\ S_{\phi_\mathcal{P}}S_{\theta_\mathcal{P}}C_{\psi_\mathcal{P}} - C_{\phi_\mathcal{P}}S_{\psi_\mathcal{P}} & S_{\phi_\mathcal{P}}S_{\theta_\mathcal{P}}S_{\psi_\mathcal{P}} + C_{\phi_\mathcal{P}}C_{\psi_\mathcal{P}} & S_{\phi_\mathcal{P}}C_{\theta_\mathcal{P}} \\ C_{\phi_\mathcal{P}}S_{\theta_\mathcal{P}}C_{\psi_\mathcal{P}} + S_{\phi_\mathcal{P}}S_{\psi_\mathcal{P}} & C_{\phi_\mathcal{P}}S_{\theta_\mathcal{P}}S_{\psi_\mathcal{P}} - S_{\phi_\mathcal{P}}C_{\psi_\mathcal{P}} & C_{\phi_\mathcal{P}}C_{\theta_\mathcal{P}} \end{bmatrix}. \tag{5}$$

$$\begin{aligned} B_1 &= F_\mathcal{P}^\mathrm{A} + W_\mathcal{P} - \Omega_\mathcal{P}(M_\mathcal{P}+M_F)\Omega_\mathcal{P}r_{C\mathcal{P}} - \Omega_\mathcal{P}M_F\boldsymbol{v}_\mathcal{P}, \\ B_2 &= M_\mathcal{P}^\mathrm{A} - \Omega_\mathcal{P}(I_\mathcal{P}+I_F)\boldsymbol{\omega}_\mathcal{P}, \\ B_3 &= F_\mathcal{B}^\mathrm{A} + W_\mathcal{B} - \Omega_\mathcal{B}M_\mathcal{B}\Omega_\mathcal{B}r_{C\mathcal{B}}, \\ B_4 &= -\Omega_\mathcal{B}I_\mathcal{B}\boldsymbol{\omega}_\mathcal{B}. \end{aligned} \tag{6}$$

In both (4) and (6), subscript F stands for the apparent term of mass and inertia of $\mathcal{P}$, which is generated from mutual interaction between a large moving object and a fluid especially when the motion is unstable [31]. One can find a detailed explanation of apparent mass and inertia from [11, 31]. These parameters are acquired from the experimental results, or calculated based on numerous assumptions and approximation that brings about inaccuracy. In (6), F^A and M^A refers to aerodynamic force and moment acting on the corresponding subscript as described in (7) while W_p and W_b are weight of each component represented in the inertial frame. Ω denotes skew-symmetric matrix of $\boldsymbol{\omega}$.

$$\begin{aligned} F_\mathcal{B}^A &= \frac{1}{2}\rho|\boldsymbol{v}_\mathcal{B}|S_\mathcal{B}C_D^\mathcal{B}\boldsymbol{v}_\mathcal{B}, \\ F_\mathcal{P}^A &= \bar{q}_\mathcal{P}S_\mathcal{P}[C_X, C_Y, C_Z]^T, \\ M_\mathcal{P}^A &= \bar{q}_\mathcal{P}S_\mathcal{P}[bC_l, cC_m + x_{pc}C_Z, bC_n]^T. \end{aligned} \tag{7}$$

In (7), $\boldsymbol{v}_\mathcal{P}$ and $\boldsymbol{v}_\mathcal{B}$ is calculated from kinematic relationship as shown in (9), x_{pc} denotes moment arm to center of pressure. ρ, $\bar{q}$, S, and b, c stands for air density, dynamic pressure, reference area, and reference lengths. Numerous aerodynamic coefficients are expressed as $C_\bullet$ where each of them is related to parafoil brakes $\boldsymbol{\delta}$ and flight data as described in [29] and (8). Here imprecise aerodynamic coefficients also pitch in the entire model inaccuracy.

$$\begin{bmatrix} C_X \\ C_Z \end{bmatrix} = \frac{1}{|\boldsymbol{v}_{\mathcal{P}}|} \begin{bmatrix} -u_{\mathcal{P}} & w_{\mathcal{P}} \\ -w_{\mathcal{P}} & -u_{\mathcal{P}} \end{bmatrix} \begin{bmatrix} C_D^{\mathcal{P}} + C_{D_{\delta_a}} \delta_a \\ C_L^{\mathcal{P}} + C_{L_{\delta_a}} \delta_a \end{bmatrix}, \begin{bmatrix} C_Y \\ C_l \\ C_m \\ C_n \end{bmatrix} = \frac{1}{2|\boldsymbol{v}_{\mathcal{P}}|} \begin{bmatrix} bC_{Y_r} r_{\mathcal{P}} \\ b(C_{l_p} p_{\mathcal{P}} + C_{l_r} r_{\mathcal{P}}) \\ cC_{m_q} q_{\mathcal{P}} \\ b(C_{n_p} p_{\mathcal{P}} + C_{n_r} r_{\mathcal{P}}) \end{bmatrix} + \begin{bmatrix} C_{Y_\beta} \\ C_{l_\beta} \\ 0 \\ C_{n_\beta} \end{bmatrix} \beta_{\mathcal{P}} + \begin{bmatrix} C_{Y_{\delta_a}} \\ C_{l_{\delta_a}} \\ C_{n_{\delta_a}} \\ C_{n_{\delta_a}} \end{bmatrix} \delta_a. \tag{8}$$

$$\begin{aligned} \boldsymbol{v}_{\mathcal{P}} &= T_{\mathcal{I}}^{\mathcal{P}} \boldsymbol{v}_C + \Omega_{\mathcal{P}} r_{C\mathcal{P}}, \\ \boldsymbol{v}_{\mathcal{B}} &= T_{\mathcal{I}}^{\mathcal{B}} \boldsymbol{v}_C + \Omega_{\mathcal{B}} r_{C\mathcal{B}}. \end{aligned} \tag{9}$$

In (8), all of the coefficient and derivatives are self-explaining except that C_L, and C_D are functions of δ_s where $[\delta_a, \delta_s] \equiv [\delta_{\mathrm{R}} - \delta_{\mathrm{L}}, \min(\delta_{\mathrm{R}}, \delta_{\mathrm{L}})]$ is equivalent to $\boldsymbol{\delta}^T$. Inverting 12×12 matrix on the left side of (4) yields $\dot{\boldsymbol{v}}_C$, $\dot{\boldsymbol{\omega}}_{\mathcal{P}}$, and $\dot{\boldsymbol{\omega}}_{\mathcal{B}}$ integration of which from initial state leaves us full description of parafoil state at any given time. We exploit the dynamic equations (3) and (4) to simulate the dynamic motion of parafoil while the reinforcement learning agent is not aware of it.

B. Reinforcement Learning and Soft Actor Critic Algorithm

Since the major part of the learning algorithm would be a replication of the study of Haarnoja et al. [32], only relevant key concepts of RL and soft actor-critic(SAC) algorithm are introduced herein. RL formalism maximizes some notion of cumulative reward over trajectory, or frequently known as the expected return as described in (10), that an agent receives when it exhibits action a_t according to its policy $a_t \sim \pi(\cdot|o_t)$ calculated based on given observation o_t.

$$J(\pi) = \mathop{\mathbb{E}}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t \right], \tag{10}$$

where τ denotes trajectory $(o_0, a_0, o_1, a_1, \cdots)$, r_t denotes reward that depends on current state, action, and optionally on next state of trajectory $r_t = R(o_t, a_t, o_{t+1})$, $\gamma \in (0, 1]$ denotes a discount factor. Central optimization problem in RL is then to find optimal policy represented as (11). Since deep RL parametrizes the policy and/or relevant value functions that evaluate given situation using deep neural network Θ, it seeks for the optimal Θ^* whose calculation is enabled by the chain rule.

$$\pi^* = \arg\max_{\pi} J(\pi). \tag{11}$$

The key part of SAC is an entropy regularization over the policy that prevents it from being locally optimum. Therefore, a slightly different notion of cumulative reward represented as

$$J(\pi) = \mathop{\mathbb{E}}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha H \left(\pi(\cdot|s_t) \right) \right) \right] \tag{12}$$

is adopted in SAC where $H(P)$ is entropy of a random variable x using its distribution $P(x)$, i.e. $H(P) = \mathbb{E}_{x \sim P}[-\log P(x)]$, and $\alpha > 0$ is a gain for the entropy. Redefinition of $J(\pi)$ yields following value functions (13), which describe how good or bad given observation and/or observation/action pair are, with additional entropy bonuses term.

$$\begin{aligned} V^{\pi}(s) &= \mathop{\mathbb{E}}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(r_t + \alpha H \left(\pi(\cdot|s_t) \right) \right) \middle| s_0 = s \right], \\ Q^{\pi}(s, a) &= \mathop{\mathbb{E}}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r_t + \alpha \sum_{t=1}^{\infty} H \left(\pi(\cdot|s_t) \right) \middle| s_0 = s, a_0 = a \right]. \end{aligned} \tag{13}$$

Entropy term helps prevent the policy from converging to a bad local solution in a premature sense since its value gets higher when the policy is widely distributed over action space. Peaky distribution of π, i.e. being too certain of its current action, often refers to the local optimum. Therefore, adding entropy term in (12) induces an agent to explore over policy space by preventing it from being too assertive. SAC learns two Q-functions and utilizes the smaller value of the twos in calculating the Bellman error loss function [33] which leaves more stable learning. In addition to the fact that SAC is applicable to continuous action space $\delta_\cdot \in [0, 1]$, off-policy and model-free characteristics of it let us utilize past experiences leaving sample efficiency. It's also known to be robust to hyper parameters with flexible applicability over various simulated environments. For the thorough derivation of SAC, readers may consult with [32].

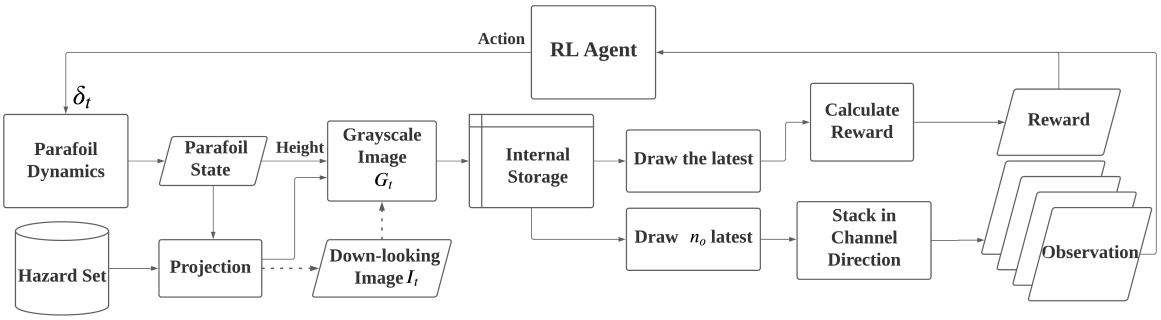

Fig. 2 Flowchart of proposing Markov decision process design and learning cycle

III. Design of Markov Decision Process and Policy Network

Figure 2 highlights flowchart of learning cycle and that of section III. As prescribed in section I.B, hazards are fully identifiable with the aid of supporting algorithms. Therefore, we simulate over true ground hazards rather than raw images (note dotted lines in figure 2).

A. Observation and Action Design

The population of ground hazards can be modelled as either a Gaussian mixture model(GMM) [34], or a set of ellipses or rectangles. Single hazard is then either a mode of a 2-dimensional Gaussian distribution, an ellipse, or a rectangle all of which can be represented as a combination of center positions (μ_x, μ_y), and some notion of its shape and orientation $\boldsymbol{s}$. $\boldsymbol{s}$ will be an upper triangle of the covariance matrix, length of semi-major/minor axis with rotation angle, or width/height of a rectangle with rotation angle respectively. In this research, we choose the ellipse model due to its intuitive and universal feature that can cover various shapes of hazard and many representations to be converted from, e.g. GMM with standard deviations as corresponding radii. The true ground hazards are simulated as

$$H = \bigcup_{i=1}^{N_\mathrm{h}} \left\{ (x + \mu_x^{[i]}, y + \mu_y^{[i]}) \,\middle|\, \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta^{[i]}) & -\sin(\theta^{[i]}) \\ \sin(\theta^{[i]}) & \cos(\theta^{[i]}) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}, \frac{x'^2}{a^{[i]2}} + \frac{y'^2}{b^{[i]2}} \le 1 \right\} \tag{14}$$

where N_h denotes the entire number of hazard, superscription $[i]$ denotes ith element, μ denotes center position, and a, b, θ denote semi major axis, semi minor axis, and orientation of each ellipse respectively. Similar models can be achieved from the other cases. In implementation phase, such parameters are randomly sampled every episode from uniform distribution related to it, i.e. $a^{[\cdot]} \sim \mathcal{U}(a_\mathrm{min}, a_\mathrm{max})$.

One of the simplest designs of observation that encodes the current situation out of I_t would be a feature vector composed of represented hazards that are captured within field of view(FOV) as

$$o_t = \left[\mu_x^{[1]}, \mu_y^{[1]}, \boldsymbol{s}^{[1]T} \quad \cdots \quad \mu_x^{[n_\mathrm{h}]}, \mu_y^{[n_\mathrm{h}]}, \boldsymbol{s}^{[n_\mathrm{h}]T} \right]^T \tag{15}$$

where n_h is the number of hazard included in FOV, and $\boldsymbol{s}$ denotes a vector that describes shape and rotation of an ellipse $\boldsymbol{s}^{[i]} = \left[a^{[i]}, b^{[i]}, \theta^{[i]} \right]^T$. The number of hazard in FOV n_h, however, varies with respect to parafoil's position, attitude, and/or population of ground hazards. Therefore, dimension of feature vector (15) also varies. Since multi-layer perceptron(MLP) type neural network, which is the most widely used function approximator for policy and value functions in deep RL, requires fixed size input, (15) is not directly applicable. One of the remedies would be to fix n_h and to reshape captured hazards with respect to it by duplicating or eliminating some portion of it, however, such approach introduces additional heuristic-based selection or trimming procedure and is thus inefficient.

Proposing design captures hazard within FOV as it is and transforms it into a grayscale image whose intensity is negatively proportional to parafoil's height at the moment. In this way one can keep the spatial distribution of hazard while augmenting one of the most critical information in landing problem from (1); altitude $|z_C|$ onto it. For the

grayscale image at tth time step G_t, its value at pixel position (u, v) is calculated as

$$G_{t,(u,v)} = \begin{cases} \lceil 255(1 - \eta \bar{h}_t) \rceil & \text{when } \left(Q_{\psi_\mathcal{P}} \frac{|z_{C,t}|}{f} \left(\begin{bmatrix} -v \\ u \end{bmatrix} + \begin{bmatrix} \frac{V}{2} \\ -\frac{U}{2} \end{bmatrix} \right) + \begin{bmatrix} x_{C,t} \\ y_{C,t} \end{bmatrix} \right) \in H \\ 1 & \text{otherwise} \end{cases} \tag{16}$$

where h_t denotes height of a parafoil at the moment ($h_t = |z_{C,t}|$), and $\bar{h}_t$ is normalized value of it so that its range becomes $[0, 1]$ taking initial height $|z_{C,0}|$ as its maximum and ground as its minimum. $\eta \in [0, 1]$ is a regulating gain that lets pixel intensity vary according to height. Note that value 1, which is the minimum among integer intensity space except 0, is utilized to express non-hazardous region in order to prevent null gradient. Also, note that ceiling operator $\lceil \cdot \rceil$ is utilized to ensure integer value for pixel intensity of gray image. Further in (16), U, V denote width and height of an image respectively, f denotes focal length, and x_C, y_C, z_C with additional subscript t denote 3-dimensional position of parafoil at given time t. $Q_{\psi_\mathcal{P}}$ is a 2-dimensional rotation matrix that produces clockwise rotation of a point with respect to the origin. Note that the order of u, v is switched since they are pointing East and North respectively when parafoil is facing North.

The problem of interest, however, is still left as partially observable MDP(POMDP) when (16) is exploited as the observation because the full dynamic state of parafoil, and evolution of captured hazard is still unknown. A trick to augment the given observations and to form a history of them, or windowed history for certain period, can effectively release the partially observable condition. Such an approach has been applied to Atari games in Deep Q-network(DQN) [35], so that an agent can infer time-related features or transition model. Our approach adopt the scheme to relax the problem to be more observable and to be more Markovian. Designing observation as (17), which is a windowed history of G_t over certain time step length n_o, lets an agent infer time-related properties of the problem, e.g. parafoil rate of descent, expected hazard transition at the moment.

$$o_t = \left(G_{t-n_o+1}, \ldots, G_t \right). \tag{17}$$

Here n_o denotes size of time horizon looking back. Especially when each grid $G_k (2 \leq k \leq n_o, k \in \mathbb{N})$ is stacked on top of former one G_{k-1} in channel direction, o_t becomes 3-dimensional tensor. It becomes typical RGB image whose width and height follows that of G_t when n_o equals 3, or RGBD data for $n_o = 4$ case. In this research we use $n_o = 4$. Channel dimension of (17) represents temporal evolution of hazards as parafoil glides, while width/height dimension of it includes spatial distribution of ground hazards seen from parafoil's point of view rotated as much as canopy's heading angle $\psi_\mathcal{P}$, and parafoil's height. For further simplicity, let bracketed indexing of o_t denotes corresponding layer, i.e., ${o_t}^{[1]} = G_{t-n_o+1}$, ${o_t}^{[n_o]} = {o_t}^{[-1]} = G_t$. We designed action as parafoil brakes themselves $\boldsymbol{\delta}$ with no modification.

B. Reward Design

Rewarding or penalizing the agent with respect to given observation and/or action necessitates the notion of riskiness at the moment. Normalization of hazard over image space, however, ambiguates positional importance of hazard, i.e., hazard at the center of FOV is regarded as equally hazardous as the other at the corner. In hazard avoiding problem, such two cases should be distinguished as the former case is more critical at the moment. In order to preserve spatial information of hazard distribution, we use a spatially weighted mask W through which we can pass the latest hazard image. The size of the mask is the same as that of the image in (16) and is weighted higher on the centric portion while peripherals are less weighted. Figure 3 highlights an example of a pyramidically weighted mask and weighting hazard image using it. One can get a normalized measure of hazard $\bar{r} \in [0, 1]$ by spatially weighting hazard image using mask W followed by normalization as

$$\bar{r}(o_t) = \frac{\sum_{i=1}^{U} \sum_{j=1}^{V} g\left({o_t}^{[-1]}{}_{(i,j)}, 1 \right) W_{(i,j)}}{\sum_{i=1}^{U} \sum_{j=1}^{V} W_{(i,j)}} \tag{18}$$

where $g(\cdot, \kappa)$ is a simple gating function using threshold κ:

$$g(x, \kappa) = \begin{cases} x & \text{if } x > \kappa \\ 0 & \text{otherwise} \end{cases}. \tag{19}$$

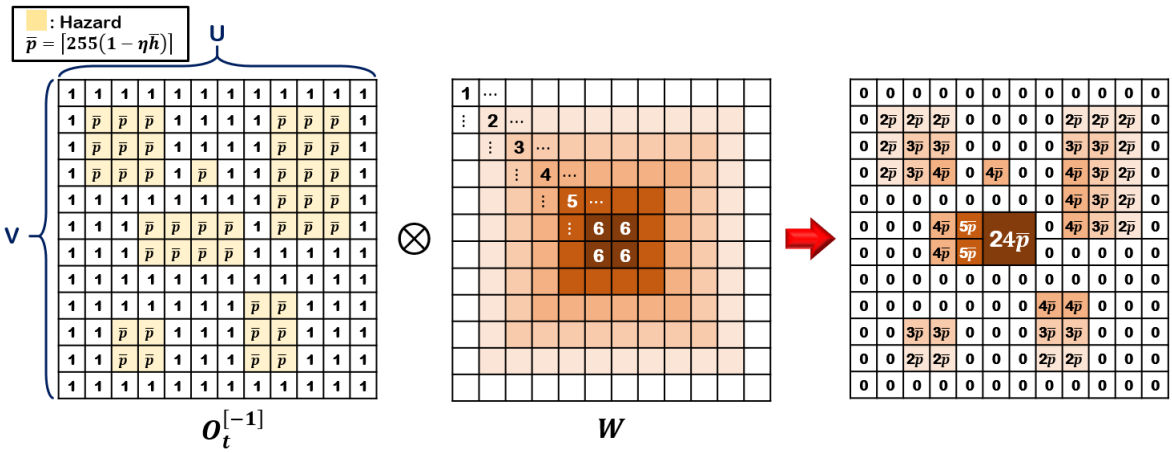

Fig. 3 Spatially weighting hazard distribution using mask

Moreover, one can imprint alternative intention by shaping the mask in a different manner. Higher weight on the upper quadrant of W, i.e., area that looks like $y > |x|$, can penalize the case when the expected trajectory is more occupied with hazard than the others. Using (18) we design reward function as

$$r(o_t) = f\left(\bar{r}(o_t), \epsilon\right) \tag{20}$$

where ϵ denotes a threshold for spatially normalized hazard so that (18) beyond this value is regarded as a potential threat at the moment. Function f is a logistic curve with L as its scaling parameter and k as slope shaping parameter as

$$f(z, \epsilon; L, k) = -L\left(\frac{1}{1 + e^{-k(z-\epsilon)}} - 0.5\right). \tag{21}$$

In such a way, the RL agent values more on keeping the hazards away from its immediate center region. The case when $\epsilon = 0.25$ in (21), for example, implies that spatially normalized hazard larger than a quarter is regarded as an undesired situation and negative reward is fed back. By designing reward function as a combination of (18) and (21) one can easily manipulate agent's behavior in an egocentric sense using explainable parameters and iterating over various values of them, i.e. ϵ provides criterion upon which assessment of hazards is made, k regulates strictness of the assessment since (21) with large k has steep curve at its ambiguous phase $|\bar{r} - \epsilon| \ll 1$. it's possible to distinctively determine whether a given hazard population is hazardous enough or not, from the steep curve of (21) and/or large value of L.

In order to keep the agent from acting too aggressively so that the parafoil gets diverged or flipped over, (20) is augmented with few more episodic negative rewards to penalize such cases as

$$r(o_t) = \begin{cases} r_{\text{Fail}}/10 & \text{flipped over} \\ r_{\text{Fail}} & \text{landed on hazard} \\ \text{same as (20)} & \text{otherwise} \end{cases}, \tag{22}$$

where r_{fail} is a large negative value that penalizes aggressive maneuver of parafoil and hazard landing. Its magnitude was derived from a rough calculation that even consecutive all positive reward accumulated along the trajectory, i.e. when $\bar{r} = 0$ in (20), eventually gets deducted resulting in negative return in non-discounted case. Episode gets terminated whenever $|\phi_{\mathcal{P}}|$, $|\theta_{\mathcal{P}}|$ becomes larger than certain threshold conveying huge negative reward to provoke negative reinforcement. Landing on hazard is treated in a similar sense but with severe penalty. With positive ϵ and moderate assumption that $\bar{r} \sim \mathcal{U}(0, 1)$ at the initial learning phase, reward is more likely to be negative value, i.e. $\mathbb{E}_{\bar{r}}[r(o_t)] < 0$. Therefore, (20) mostly provoke negative reinforcement that pressurizes the agent into not gliding over a hazardous region.

Through (16), (17), (18), and (20) we envelope the given problem in an end-to-end sense so that the agent does not need to be informed any of the followings; parafoil dynamics, evolution of hazard population with respect to parafoil's

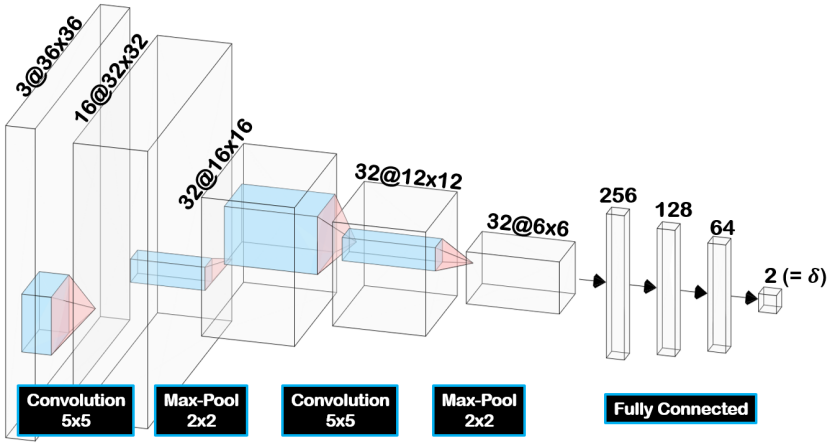

Fig. 4 CNN-MLP type policy architecture

perspective. It can learn implicit relationship among those using experience set of state, action, and reward triples by interacting with the simulated environment. The proposed scheme can be generalized further to many aerospace platforms, e.g. fixed-wing aircraft, multirotor UAV, that utilizes vision sensor.

C. Policy Architecture

Since the observation (17) is image-like data and action $\boldsymbol{\delta}$ is a 2-dimensional vector, the deep neural network that approximates policy should begin with a convolutional neural network(CNN) to take both spatial and temporal information of the stacked grayscale images, and terminate with a flat layer of two nodes. In order to prevent the policy from over-fitting or being parametrized with too many parameters, pooling layers are attached after each convolution network. Yet, since the network should be complex enough to be able to solve the given problem, we did several experiments by enlarging the architecture from the smallest possible one. We found only little improvement as the policy architecture highlighted in figure 4 gets more complex or deeper from it. Therefore, we settle down to policy architecture shown in figure 4 which is self-explaining.

IV. Training and Simulation Result

A. Training and Test Scenario

For both the training and testing of an agent, parafoil gets respawned at a random initial state, and ground hazards are randomly resampled in every episode. For the sake of stable learning, however, the stochasticity of the initial state gets regulated under certain boundary to keep episode length from varying too much. The parafoil glides according to its policy until it reaches the ground, or flips over as described in (22) and thereafter. The episode also terminates when parafoil flies beyond the predefined playground. We made use of the SAC implementation of Hill et al. [36] for stable learning and for further comparison with other problems and/or algorithms. Since the nature of SAC's policy is stochastic, We left out the sampling procedure of action and utilized deterministic action in the test cases. Hyper parameters that are used for the environment reset and learning is listed in table 1 while one can look up for parafoil-specific parameters in [11, 28, 29].

B. Resultant Trajectory and Numerical Analysis

As a result of successful training of the RL agent, we present several case-wise complete trajectories of a parafoil. Three cases are highlighted in figure 5, 6, and 7 respectively, all of which includes 3D trajectories and ground projection of them. For the 2-dimensional graphs, each grid represents 50m. The first case in figure 5 is the most frequently sampled one which shows that randomly spawned parafoil over randomly distributed hazards steers the brakes *against* the perceived ground hazard. Due to the help of spatially weighted measure of riskiness at the moment (18) and negative reward (21) that penalize the risky condition, parafoil keeps hazards away from its region of interest every moment.

Parameter	Value	Sampled Parameter	Minimum	Maximum
N_h	75	θ	$-\pi$ rad	π rad
World size	250×250 m^2	a, b	10 m	40 m
Angle of view	60 deg	$h_0 (= \lvert z_{C,0} \rvert)$	100 m	150 m
η	0.5	$x_{C,0}, y_{C,0}$	-150 m	150 m
ϵ	0.15	$\psi_0 - \tan^{-1}(\frac{y_{C,0}}{x_{C,0}}) - \pi$ (for $\mathcal{P}, \mathcal{B}$)	$-2.5\frac{\pi}{180}$ rad	$2.5\frac{\pi}{180}$ rad
L	10	$\boldsymbol{v}_{C,0}$	$[0,0,0]^T$ m/s	$[0.1,0.1,0.1]^T$ m/s
k	7.5	ϕ_0, θ_0 (for $\mathcal{P}, \mathcal{B}$)	$[-1,-1]^T$ deg	$[1,1]^T$ deg
r_{fail}	-10^3	ω_0 (for $\mathcal{P}, \mathcal{B}$)	$[0,0,0]^T$ deg/s	

Table 1 Simulation parameters

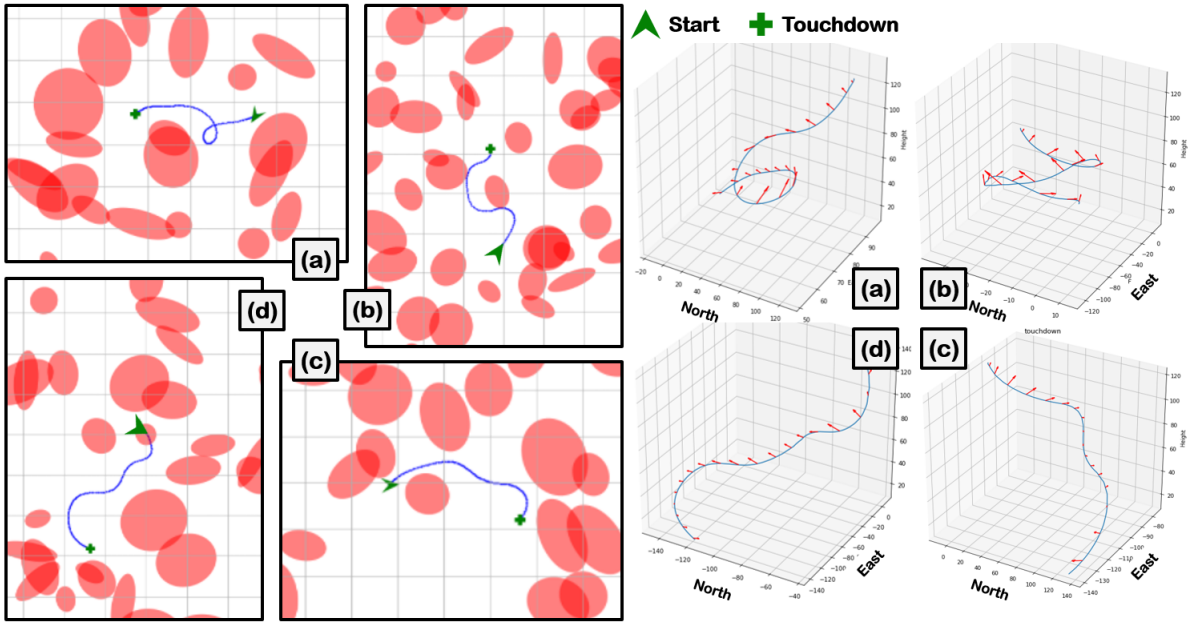

Fig. 5 2D(left side) and 3D(right side) trajectories of parafoil sampled from ALHAT trained agent. Steering against ground hazards(easy case).

Resultant trajectories show that the RL agent learned the dynamics of parafoil inherent in the environment without any prior knowledge of it, and figured out the correct steering logic that drives the vehicle against the imminent hazards.

The second case in figure 6, which comes as frequent as the first case, shows that parafoil forms a spiral at open space. Since the majority of the learning is negative reinforcement by (21), the action that elongates episode length when consecutive hazard-free observations were made is regarded as an inappropriate decision. Such behavior most likely to lower the value of (10) or (12). It also makes sense that since consecutive hazard-free observation leaves no preference on long-term directional guidance, it's best to land at the immediate hazard-free location at the moment collecting maximum rewards, i.e. (20) and (21) when $\bar{r} = 0$, along the spiral. The trained agent prefers left turn in forming a spiral. A similar pattern is observed in the following case.

The third case which is highlighted in figure 7 shows rapid maneuver of the parafoil in the middle of *dead end* or nearly dead end. It is a relatively hard case compared to the first case and the second case since hazardless space between hazards is narrow leaving the RL agent negative rewards calculated from (21). Moreover, keep exploring over hazard space for the non-hazardous regions is less likely to be successful around the dead end. This is especially true when the parafoil is close to the ground. Nevertheless, the agent effectively mitigates the risky moment by landing as quickly as possible forming a very sharp spiral in the proximity of the ground wherever such maneuver and safe landing is possible. When the parafoil faces such blockage during the initial phase, the agent again performs rapid maneuver seeking for relatively safe space.

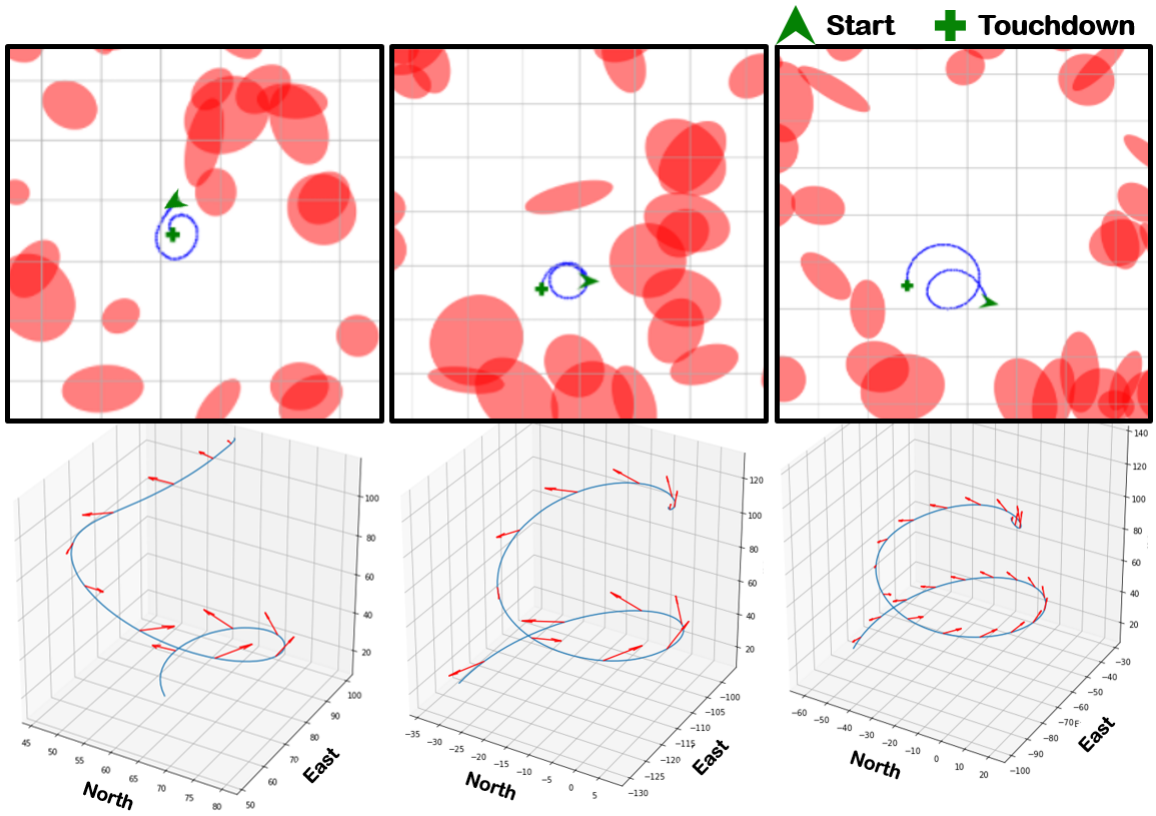

Fig. 6 2D(upper row) and 3D(lower row) trajectories of parafoil sampled from ALHAT trained agent. Spiral at the consecutive non-hazardous observation.

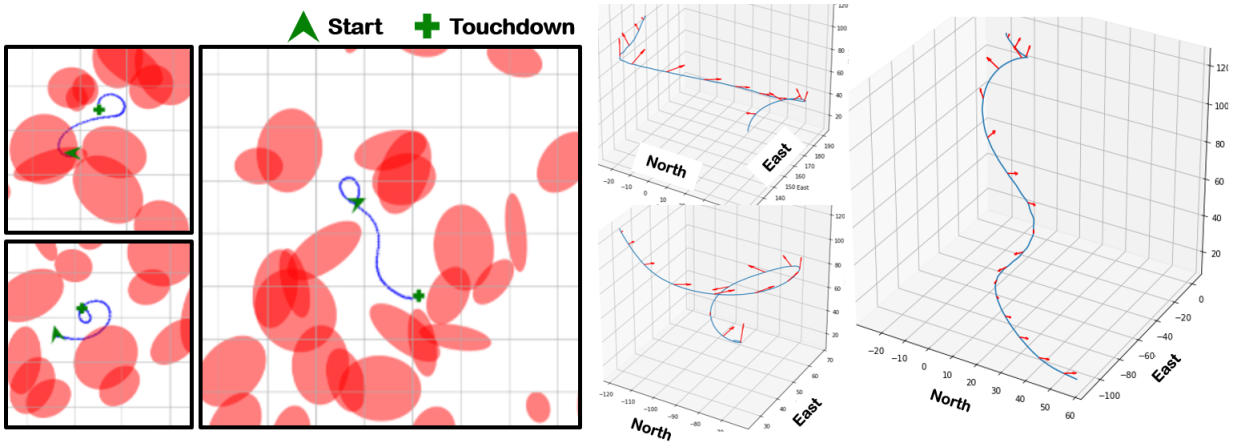

Fig. 7 2D(left side) and 3D(right side) trajectories of parafoil sampled from ALHAT trained agent. Sharp spirals around dead ends(hard case)

V. Conclusion

We presented an end-to-end autonomous hazard avoidance landing technique of a parafoil equipped with a down-looking camera by means of reinforcement learning. The simulation environment that includes 9-degrees of freedom dynamics of the parafoil, and projection of ground hazards onto the vehicle point of view was constructed in order to train reinforcement learning agent which directly exercises the brakes of a parafoil. We proposed a synthetic design of Markov decision process whose observation envelopes the parafoil dynamic state, hazard population on the ground at the moment, and transition of the hazards with respect to the parafoil's viewpoint change using the concept of stacked observation. It also released partial observability of the problem by augmenting vehicle height onto the grayscale image. The reward function of the design is featured with a spatially weighted mask that preserves the positional importance of the hazards. We solved the decision process by applying soft actor-critic reinforcement algorithm to it. The resultant trajectory of the parafoil shows that learnt steering logic of the brakes can have a parafoil and its payload touch down on non-hazardous region without any specific knowledge about dynamic relation between the brakes and parafoil states, hazard transition model, and human supervision. Learnt behavioral pattern primarily focuses on keeping spatially weighted hazards away from the center of viewpoint. When consecutive non-hazardous fields are seen, however, it tends to land on the immediate open space minimizing expected negative rewards. We showed that the end-to-end steering law for ALHAT of parafoil is achievable. We are also optimistic about refining trained policy using real experimental data. We are planning to compare the performance of steering law of parafoil brakes realized by reinforcement learning to other rule-based approaches to analyze it further in both analytic and numerical sense.

References

[1] Jung, W., and Bang, H., "Fault and Failure Tolerant Model Predictive Control of Quadrotor UAV," *International Journal of Aeronautical and Space Sciences*, 2021, pp. 1–13.

[2] Bateman, F., Noura, H., and Ouladsine, M., "Fault diagnosis and fault-tolerant control strategy for the aerosonde UAV," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 47, No. 3, 2011, pp. 2119–2137.

[3] Slegers, N., and Costello, M., "Model predictive control of a parafoil and payload system," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 4, 2005, pp. 816–821.

[4] Prakash, R., Burkhart, P. D., Chen, A., Comeaux, K. A., Guernsey, C. S., Kipp, D. M., Lorenzoni, L. V., Mendeck, G. F., Powell, R. W., Rivellini, T. P., et al., "Mars Science Laboratory entry, descent, and landing system overview," *2008 IEEE Aerospace Conference*, IEEE, 2008, pp. 1–18.

[5] Witze, A., and Kowsky, J., "NASA has launched the most ambitious Mars rover ever built: here's what happens next." *Nature*, Vol. 584, No. 7819, 2020, pp. 15–16.

[6] Taylor, A. P., Machin, R., Royall, P., and Sinclair, R., "Developing the Parachute System for NASA's Orion: An Overview at Inception," *19th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2007, p. 2577.

[7] Hsu, J., "Boeing and SpaceX test the next US ride to space: The international space station is expecting two visitors this month: Starliner and Crew Dragon-[News]," *IEEE Spectrum*, Vol. 55, No. 8, 2018, pp. 6–8.

[8] Wyllie, T., "Parachute recovery for UAV systems," *Aircraft Engineering and aerospace technology*, 2001.

[9] Epp, C. D., and Smith, T. B., "Autonomous precision landing and hazard detection and avoidance technology (ALHAT)," *2007 ieee aerospace conference*, IEEE, 2007, pp. 1–7.

[10] Matthies, L., Daftry, S., Rothrock, B., Davis, A., Hewitt, R., Sklyanskiy, E., Delaune, J., Schutte, A., Quadrelli, M., Malaska, M., et al., "Terrain Relative Navigation for Guided Descent on Titan," *2020 IEEE Aerospace Conference*, IEEE, 2020, pp. 1–16.

[11] Lissaman, P., and Brown, G., "Apparent mass effects on parafoil dynamics," *Aerospace Design Conference*, 1993, p. 1236.

[12] Luders, B. D., Sugel, I., and How, J. P., "Robust trajectory planning for autonomous parafoils under wind uncertainty," *AIAA Infotech@ Aerospace (I@ A) Conference*, 2013, p. 4584.

[13] Slegers, N., and Yakimenko, O., "Optimal control for terminal guidance of autonomous parafoils," *20th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2009, p. 2958.

[14] Bergeron, K., Tavan, S., and Fejzic, A., "Accuglide: Precision airdrop guidance and control via glide slope control," *21st AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2011, p. 2530.

[15] Ward, M., and Costello, M., "Adaptive glide slope control for parafoil and payload aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 4, 2013, pp. 1019–1034.

[16] Rademacher, B. J., Lu, P., Strahan, A. L., and Cerimele, C. J., "In-flight trajectory planning and guidance for autonomous parafoils," *Journal of guidance, control, and dynamics*, Vol. 32, No. 6, 2009, pp. 1697–1712.

[17] Chiel, B. S., and Dever, C., "Autonomous parafoil guidance in high winds," *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 5, 2015, pp. 963–969.

[18] Lee, S., and Bang, H., "Automatic Gain Tuning Method of a Quad-Rotor Geometric Attitude Controller Using A3C," *International Journal of Aeronautical and Space Sciences*, 2019, pp. 1–10.

[19] Koch, W., Mancuso, R., West, R., and Bestavros, A., "Reinforcement learning for UAV attitude control," *ACM Transactions on Cyber-Physical Systems*, Vol. 3, No. 2, 2019, pp. 1–21.

[20] Polvara, R., Patacchiola, M., Sharma, S., Wan, J., Manning, A., Sutton, R., and Cangelosi, A., "Toward end-to-end control for UAV autonomous landing via deep reinforcement learning," *2018 International conference on unmanned aircraft systems (ICUAS)*, IEEE, 2018, pp. 115–123.

[21] Rodriguez-Ramos, A., Sampedro, C., Bavle, H., De La Puente, P., and Campoy, P., "A deep reinforcement learning strategy for UAV autonomous landing on a moving platform," *Journal of Intelligent & Robotic Systems*, Vol. 93, No. 1-2, 2019, pp. 351–366.

[22] Lee, S., Shim, T., Kim, S., Park, J., Hong, K., and Bang, H., "Vision-based autonomous landing of a multi-copter unmanned aerial vehicle using reinforcement learning," *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, IEEE, 2018, pp. 108–114.

[23] Batavia, P. H., and Singh, S., "Obstacle detection using adaptive color segmentation and color stereo homography," *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, Vol. 1, IEEE, 2001, pp. 705–710.

[24] Labayrade, R., Aubert, D., and Tarel, J.-P., "Real time obstacle detection in stereovision on non flat road geometry through" v-disparity" representation," *Intelligent Vehicle Symposium, 2002. IEEE*, Vol. 2, IEEE, 2002, pp. 646–651.

[25] Jung, Y., Lee, S., and Bang, H., "Digital Terrain Map Based Safe Landing Site Selection for Planetary Landing," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 1, 2019, pp. 368–380.

[26] Jann, T., "Aerodynamic model identification and GNC design for the parafoil-load system ALEX," *16th AIAA aerodynamic decelerator systems technology conference and seminar*, 2001, p. 2015.

[27] Mortaloni, P., Yakimenko, O., Dobrokhodov, V., and Howard, R., "On the development of a six-degree-of-freedom model of a low-aspect-ratio parafoil delivery system," *17th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, 2003, p. 2105.

[28] Slegers, N., and Costello, M., "Aspects of control for a parafoil and payload system," *Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, 2003, pp. 898–905.

[29] Prakash, O., and Ananthkrishnan, N., "Modeling and simulation of 9-DOF parafoil-payload system flight dynamics," *AIAA Atmospheric Flight Mechanics Conference and Exhibit*, 2006, p. 6130.

[30] Toglia, C., and Vendittelli, M., "Modeling and motion analysis of autonomous paragliders," *Department of computer and system sciences Antonio Ruberti technical reports*, Vol. 2, No. 5, 2010.

[31] Kowaleczko, G., "Apparent masses and inertia moments of the parafoil," *Journal of Theoretical and Applied Mechanics*, Vol. 52, 2014.

[32] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S., "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," *International Conference on Machine Learning*, PMLR, 2018, pp. 1861–1870.

[33] Achiam, J., "Spinning Up in Deep Reinforcement Learning," 2018.

[34] Hong, K., Kim, S., and Bang, H., "Vision-based Navigation using Gaussian Mixture Model of Terrain Features," *AIAA Scitech 2020 Forum*, 2020, p. 1344.

[35] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M., "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[36] Raffin, A., Hill, A., Ernestus, M., Gleave, A., Kanervisto, A., and Dormann, N., "Stable Baselines3," `https://github.com/DLR-RM/stable-baselines3`, 2019.