2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Identifying Optimal Launch Sites of High-Altitude Latex-Balloons using Bayesian Optimisation for the Task of Station-Keeping

Jack Saunders[1], Sajad Saeedi[2], Adam Hartshorne[1], Binbin Xu[3], Özgür Şimşek[1], Alan Hunter[1], and Wenbin Li[1]

***Abstract*— Station-keeping tasks for high-altitude balloons show promise in areas such as ecological surveys, atmospheric analysis, and communication relays. However, identifying the optimal time and position to launch a latex high-altitude balloon is still a challenging and multifaceted problem. For example, tasks such as forest fire tracking place geometric constraints on the launch location of the balloon. Furthermore, identifying the most optimal location also heavily depends on atmospheric conditions. We first illustrate how reinforcement learning-based controllers, frequently used for station-keeping tasks, can exploit the environment. This exploitation can degrade performance on unseen weather patterns and affect station-keeping performance when identifying an optimal launch configuration. Valuing all states equally in the region, the agent exploits the region's geometry by flying near the edge, leading to risky behaviours. We propose a modification which compensates for this exploitation and finds this leads to, on average, higher steps within the target region on unseen data. Then, we illustrate how Bayesian Optimisation (BO) can identify the optimal launch location to perform station-keeping tasks, maximising the return from a given rollout. We show BO can find this launch location in fewer steps compared to other optimisation methods. Results indicate that, surprisingly, the most optimal location to launch from is not commonly within the target region. Please find further information about our project at https://sites.google.com/view/bo-lauch-balloon/.**

I. Introduction

Recently, high-altitude balloons have shown promise in applications such as environmental [1] and wildlife [2], [3] surveillance, communication relay [4], [5], and atmospheric analysis [6]. Furthermore, high-altitude balloons can sustain flights for many days [7], [4]. Whereas in comparison, conventional unmanned aerial vehicles succumb to power and weight constraints, leading to reduced flight time [8]. Latex-balloon alternatives are currently being explored as a low-cost alternative to super-pressure balloons [7], [9]. Where super-pressure balloons depend on a high-strength plastic envelope to prevent expansion of the lifting gas, which can be cost-prohibitive for research purposes. These passively actuated balloons are under-actuated, with direct control limited to the ascent rate. Hence, their flight path is dictated by the direction of the wind and atmospheric conditions.

This work is supported by the UKRI Centre for Doctoral Training in Accountable, Responsible & Transparent AI (ART-AI), under UKRI grant number EP/S023437/1.

[1]University of Bath, {js3442,ath35,os435,A.J.Hunter,w.li}@bath.ac.uk
[2]Toronto Metropolitan University, s.saeedi@torontomu.ca
[3]University of Toronto, binbin.xu@utoronto.ca

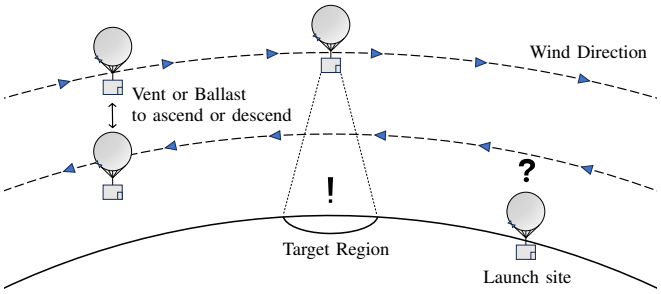

Fig. 1: Here, we illustrate our objective to identify a launch configuration that maximises the duration within the target region. Despite the challenges: of under-actuated navigation, the influence of wind and atmospheric conditions, and environment exploitation.

Reinforcement learning (RL) based controllers for station-keeping have increased in popularity, particularly due to the wind-forecast uncertainties and the non-linear relationship between actions and optimal states as a result of atmospheric conditions [4]. This relationship leads to difficulties in applying search-based and model-based approaches for station-keeping tasks. However, RL-based controllers are known to lead to unintended behaviour that can emerge from under-defined reward functions [10]. We show that agents trained for station-keeping can game the reward function by exploiting the geometries of the region, leading to poorer performance on out-of-distribution data. Additionally, this exploitation can degrade the performance of the identified optimal launch configuration. Some previous reward functions [11], [4] gave equal value to states within the region. In some instances, this incentivised agents to fly along the boundary of the region, which allowed the agent to obtain more reward at the expense of risky behaviours. Instead, we propose a modification which increases the reward closer to the target when inside the region using a Tanh function. We find this modified reward function leads to, on average, higher steps within the 50 km region on unseen data.

Researchers have previously established the temporal factors, such as diurnal cycles and seasonal variations, on station-keeping performance [4]. Despite this, the effects of varying spatial positions on balloon performance have not been thoroughly investigated, and optimisation strategies accounting for spatial variability remain unexplored. As a result, we illustrate how Bayesian Optimisation (BO) can locate launch configurations and identify latitude, longitude, and time to launch. Furthermore, we also show that BO is capable of finding the launch configuration in almost half as

many iterations compared to other optimisation approaches. For our study, we focus on latex-based balloons, however the use of BO to find the launch configuration can be used for any balloon type.

Our contributions are as follows: **(1)** To our knowledge, we are the first to propose an optimization strategy for high-altitude balloon station-keeping. Considering both spatial and temporal variability to determine the optimal launch configuration. **(2)** We propose a modified reward function that mitigates reward hacking for station-keeping tasks, and which increases maximum station-keeping performance. **(3)** We then show that Bayesian Optimisation can identify the optimal launch location in fewer steps compared to other optimisation approaches.

II. Literature Review

Researchers have proposed controllers to maintain a high-altitude balloon's position within a target region, also known as station-keeping. These controllers exploit the varying wind directions at different altitudes to optimise the balloon's time spent in a designated area.

Du *et al.* introduces a geometric method to calculate the optimal altitude to reach in order to maximise the cosine similarity between the orientation of the balloon [12]. The method, however, assumes stable wind fields, which is unpredictable. Liu *et al.* extends this work by proposing their Time-Varying Range of Achievable Altitude Algorithm (TR3A). This algorithm incorporates the bursting and over-pressure altitudes, which more accurately represent the range of altitudes the balloon can safely reach [13]. The authors illustrated improved station-keeping performance by taking advantage of an increased range of altitudes.

Bellemare *et al.* argues that standard model-predictive control algorithms face challenges performing station-keeping due to the complex non-linear relationship between control decisions and the target objective [4]. As a result, the authors use Quantile Regression-based Deep Q-learning (Qr-DQN) [14], parameterised using a neural network. The agent's actions represent the intake from a pump or exhaust of air within the balloon envelope. Furthermore, the authors model the thermal expansion of the balloon envelope and solar energy generated for a renewable resource for the pump. Effectively learning the effect of the diurnal cycle. Furthermore, a Gaussian Process (GP) was used to model the uncertainty between the wind forecast errors and the true wind speed. Bellemare's work illustrated the large computation required to train a reasonable policy. Xu *et al.* showed how beneficial a prioritised experience replay [15] based on high-value samples can improve the training stability for station-keeping tasks [16] whilst using a Deep Q-Network [17].

Saunders *et al.* illustrated that reinforcement learning could also be used for latex balloons, a low-cost alternative to super-pressure balloons [11]. The use of sand ballast and venting helium enables latex balloons to reach much larger ascent rates compared to super-pressure balloons. However, venting and ballasting come at a greater cost of resources, which is non-renewable. The authors investigate resource-constrained station-keeping while incorporating feasible limits on the resources exhausted.

Alternative balloon systems have been proposed to improve station-keeping performance. Jiang *et al.* proposes a double-balloon system which uses a winch to control the altitude of an assistant balloon [18]. This assistant balloon reaches desired altitudes using the winch to change the flight direction, using less energy than air ballasts. Furthermore, Wynsberghe *et al.* proposes electro-hydrodynamic thrusters, which deliver power wirelessly from a ground-based transmitter [19]. However, the disadvantage of this approach is the requirement to be in the line of sight of the transmitter.

The uncertainty in modelling the wind and atmospheric effects is still an open research question [20], [21], which can lead to a large sim-to-reality gap. Fields *et al.* attempts to overcome this gap using online wind data [21] to more accurately predict the landing location. Wind data is collected on the ascent phase and used to correct the flight parameters on the descent. Alternatively, Sóbester *et al.* produces a balloon flight simulation model that considers an empirically derived stochastic drag model along with uncertainties in the wind profile and the balloon envelope [20]. Then, Monte-Carlo ensembles to predict a trajectory along with the landing site with location error estimates.

Within the context of capacity optimisation, researchers have used similar techniques to optimise for cell tower coverage [22] and wind turbine placement [23]. To our knowledge, we are the first to optimise the launch location of a latex high-altitude balloon in order to enhance the station-keeping performance.

III. Background

Here we discuss the force balance of the latex balloon and the formulation for the reinforcement learning policy used for station-keeping. Then we outline the background of the Gaussian Process and the Bayesian Optimisation employed to search for the optimal launch configuration.

A. Equations of Motion

For our study, we use the dynamic model of a latex balloon, where the equations of motion are explained below. For further details, we direct the reader's attention to our previous work [11]. The forces acting on the balloon include the buoyancy F_b, weight F_w, and drag F_d force, such that $\sum F = F_b - F_d - F_w$ [24]. The buoyancy force is a result of the displaced air caused by the balloon, $F_b = \rho_a V g$, where both the density of air ρ_a and the volume of the balloon envelope V vary with altitude. The drag force acts opposite to the relative motion of the balloon with respect to the wind $F_d = \frac{1}{2}\rho c_d A|\mathbf{v_r}|\mathbf{v_r}$ [25]. Given that $\mathbf{v_r}$ is the motion of the balloon $\mathbf{v_b}$ relative to the wind $\mathbf{v_w}$, such that $\mathbf{v_r} = \mathbf{v_b} - \mathbf{v_w}$. A is the cross-sectional area, and c_d is the drag coefficient. Finally, the gravitational force is the combined weight of the inert mass m_i, including the payload and balloon envelope, helium mass m_h, and sand mass m_s. Where the combined mass is $m = (m_p + m_h + m_s)$, and $F_w = mg$. We represent the

basis vector for which the weight and buoyancy forces act along as $\mathbf{e_z} = [0,0,1]^T$:

$$m\mathbf{a} = \rho_a V g \mathbf{e_z} - \frac{1}{2}\rho c_d A |\mathbf{v_r}| \mathbf{v_r} - (m_p + m_h + m_s) g \mathbf{e_z}. \quad (1)$$

Assuming helium acts as an ideal gas and the latex material acts perfectly elastic [7], the balloon envelope volume can be calculated using $V = \frac{nRT}{P}$, where P is the ambient pressure, R is the universal gas constant, n is the number of mols of helium, and T is the ambient temperature. The atmospheric variables are all modelled using the US Standard Atmosphere Model 1976. Using atmospheric lapse rates L, the ambient temperature T, given a reference temperature T_0, can be calculated by $T = T_0 + (h \times L)$. We assume the internal temperature is equivalent to the ambient temperature. Then, we can calculate the drag area as $A = \pi \left(\frac{3V}{4\pi}\right)^{\frac{2}{3}}$.

B. Reinforcement Learning

We formulate the task of station-keeping as a Markov Decision Process (MDP), characterized by the tuple $(\mathcal{S}, \mathcal{A}, P, R)$. For each decision step t, the agent in state $s_t \in \mathcal{S}$ transitions to a new state s_{t+1} after taking an action $a_t \in \mathcal{A}$, guided by the probability distribution $P(s_{t+1}|s_t, a_t)$. As a result of reaching s_{t+1}, the agent receives a reward r_{t+1} according to the reward distribution $R(s_t, a_t)$. The agent's objective during training is to maximise the expected future discounted cumulative reward $\mathbb{E}[G_t|s_t]$, where $G_t = \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k$ and $\gamma \in [0,1]$ is the discount factor.

We make use of Soft-Actor Critic (SAC) [26], which is an off-policy deep RL algorithm within the maximum entropy framework, to control the balloon. The policy aims to maximise both the expected reward and policy entropy, and simultaneously learns a policy π_θ and a Q-function Q_ϕ, both parameterised using neural networks. The Q-function parameters are optimised using the mean squared loss function $J_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\phi(s_t,a_t) - \hat{Q}(s_t,a_t)\right)^2\right]$, which measures the discrepancy between the target Q_ϕ and predicted $\hat{Q}$ action-value, where samples are drawn from a replay buffer $\mathcal{B}$. The predicted action-value is defined as $\hat{Q}(s_t,a_t) = r(s_t,a_t) + \gamma(1-d)\mathbb{E}_{a\sim\pi}[Q_\phi(s_{t+1}, \pi_\theta(a_{t+1}|s_{t+1})) - \alpha \log \pi_\theta(a_{t+1}|s_{t+1})]$ and the entropy temperature α is adjusted by taking the gradient of $J_\alpha = \mathbb{E}[-\alpha \log \pi_\theta(a_t|s_t;a) - \alpha\bar{H}]$ [27] towards achieving a desired minimum, $\bar{H} = 0$.

The policy is optimised by minimising the simplified Kullback-Leibler divergence $J_\pi(\phi) = \mathbb{E}_{s_t\sim\mathcal{B},\varepsilon_t\sim\mathcal{N}}[\log \pi_\phi(\tilde{a}(s_t,\varepsilon_t)|s_t) - Q_\theta(s_t, \tilde{a}(s_t,\varepsilon_t))]$. Here, samples from π_θ are generated through a squashed Gaussian policy using the reparameterisation trick $\tilde{a}_{t\theta}(s,\varepsilon) = \tanh(\mu_\theta(s) + \sigma_\theta(s) \odot \varepsilon)$, with epsilon drawn from a standard normal distribution, $\varepsilon \sim \mathcal{N}(0, I)$.

C. Spatial-Temporal Gaussian Process Modelling

For this study, a GP is used to model the spatial-temporal station-keeping performance of the balloon from trails collected from running the policy through a simulated episode.

A GP is a versatile Bayesian non-parametric method that can learn unknown non-linear functions by placing a prior distribution over the space of functions [28]. Noise observed target values y are modelled as $y = f(\mathbf{x}) + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, \sigma^2\mathbf{I})$ is noise associated to each independent observation. The GP is determined by a mean function $m(\mathbf{x})$ and a positive semi-definite covariance function $k(\mathbf{x}, \mathbf{x}')$,

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), K(\mathbf{x}, \mathbf{x}')) \quad (2)$$

For this paper, we assume a zero mean function, $m(\mathbf{x}) = 0$. There are several covariance functions within the literature, each with different characteristics. For our study, we make use of the $\nu = \frac{5}{2}$ Matérn kernel due to its robustness against non-smooth functions

$$K_{\text{Matérn}}(\mathbf{x}, \mathbf{x}') = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\sqrt{2\nu}d\right)^{\nu} K_\nu\left(\sqrt{2\nu}d\right). \quad (3)$$

Where d is the distance between x and x' scaled by the lengthscale parameter θ, ν is the smoothness parameter, Γ is the gamma function, and K_ν is the modified Bessel function of the second kind.

Finding the optimal hyper-parameters θ^*_{gp} can be achieved by maximising the log marginal likelihood of the data:

$$\theta^*_{\text{gp}} = \max_\theta \log\left(-\frac{1}{2}\log|\mathbf{K}| - \frac{1}{2}\mathbf{y}^T\mathbf{K}^{-1}\mathbf{y} - \frac{n}{2}\log 2\pi\right). \quad (4)$$

D. Bayesian Optimisation

To build the GP model, we use a Bayesian Optimisation (BO) method to estimate the next sample point. To achieve this, BO utilises an acquisition function h, which guides the search over the GP. The process involves finding the parameters $\mathbf{x}$ that maximises the acquisition function at each iteration: $\mathbf{x} = \arg\max_{\mathbf{x}} h(\mathbf{x})$.

Algorithm 1 Generic Bayesian Optimization Algorithm

Require: $f, h, \mathcal{D}$
Ensure: $\mathbf{x}^*, f(\mathbf{x}^*)$
for $j = 1, 2, 3, \ldots$ **do**
  Find $\mathbf{x}_i = \arg\max_{\mathbf{x}} h(\mathbf{x})$ ▷ Max Acquisition function
  $y_i \leftarrow f(\mathbf{x}_i)$ ▷ Obtain sample from f
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{x}_i, y_i)\}$ ▷ Update training set
  $\theta^*_{\text{gp}} = \max_\theta \log p(\mathbf{y}|\mathbf{x}, \theta_{\text{gp}})$ ▷ Update GP
  if $y_j > \mu(\mathbf{x}^*)$ **then**
    $\mathbf{x}^* \leftarrow \mathbf{x}_i$ ▷ Update location of optimum
  end if
end for

For our study, we use the expected improvement acquisition function $h(\mathbf{x}) = \mathbb{E}[I(\mathbf{x})]$, given that improvement is $I(\mathbf{x}) = \max(f(\mathbf{x}) - f(\mathbf{x}^*), 0)$, where $f(\mathbf{x})$ denotes our current sampled value and $f(\mathbf{x}^*)$ is our current best-sampled value. The closed form can be expressed as the following [29]

$$h(x) = (\mu - f(\mathbf{x}*))\Theta(\frac{\mu - f(\mathbf{x}*)}{\sigma}) + \sigma\psi(\frac{\mu - f(\mathbf{x}*)}{\sigma}) \quad (5)$$

where we emit the exploration tuning parameter, $\xi = 0$. The generic algorithm for BO is illustrated in Algorithm

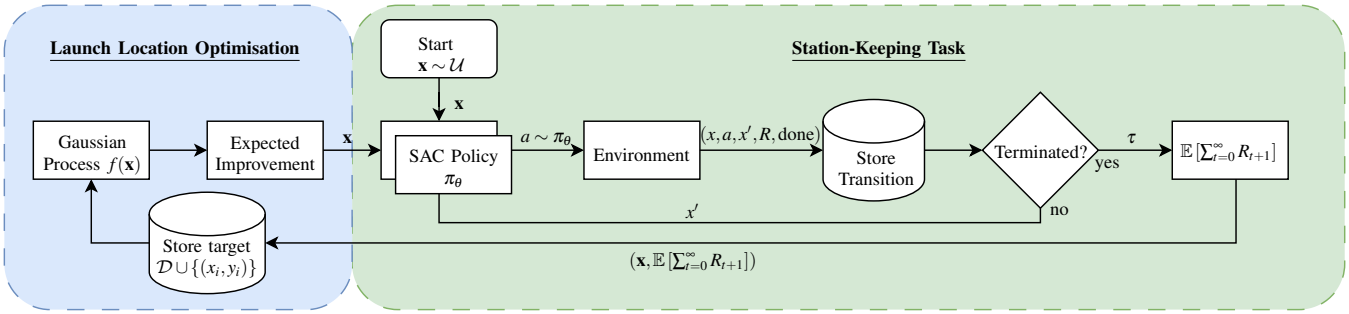

Fig. 2: Block diagram of our method to identify the optimal launch configuration. Initially, we randomly sample a launch configuration which forms part of the initial state x of the RL policy. After the episode has terminated, the expected undiscounted return is calculated and stored along with the launch configuration $\mathbf{x}$. Then, a GP models the performance of the policy by optimising the log marginal likelihood. The next launch configuration is chosen by maximising the expected improvement.

1. Although BO can be efficient in optimising expensive functions, it can suffer from computational overheads due to the GP covariance inversion of $O(n^3)$, which is updated each epoch. Furthermore, the covariance matrix requires storage of $O(n^2)$. For our study, we limit the number of sample points.

IV. Problem Statement

This paper aims to identify the launch configuration $\mathbf{x} \in \{x_0, y_0, \Delta t\}$ such that the undiscounted return for a given policy π is maximised. The launch configuration consists of the initial longitude x_0, latitude y_0, and time offset $\Delta t \in \mathbb{R}^+$ from a specific start date. Here, $\mathbf{x}$ forms part of the initial state, $s_0 \in \{..., ||(x_0, y_0)||, ...\}$ and the wind vector is sampled from $v_w = W(x, y, z, t + \Delta t)$.

For our study, we limit the spatial distance of the search space to $x_{\max} = y_{\max} = 400$ km. Furthermore, the time offset is limited to $\Delta t_{\max} = 24$ hours, as longer offsets require further study into the variance of the wind forecast [30]. The problem is formally defined as:

$$\begin{aligned} &\underset{x_0, y_0, \Delta t}{\arg\max} \quad \left[\sum_{t=0}^{T} R_{t+1} | S_t = s_0 \right] \\ &\text{s.t.} \quad |x_0| \leq x_{\max}, \quad |y_0| \leq y_{\max}, \quad \Delta t_0 \leq t_{\max} \end{aligned} \tag{6}$$

V. Method

In this section, we formulate the Soft Actor-Critic policy used for station-keeping, detailing the state and action space. We then describe the data utilised for training and testing. Then, we propose a reward function designed to improve the generalisation of unseen data and reduce reward hacking. Finally, we outline how Bayesian optimisation maximises the expected improvement to identify the optimal launch configuration. We outline the components of our method in Figure 2.

A. Soft Actor-Critic Controller

The Soft Actor-Critic policy, which controls the balloon, is parameterised by a fully connected neural network as illustrated in our previous work [11]. Both actor and critic networks have two hidden layers of 256 neurons each. The actor network is parameterised by a Gaussian policy, where actions are represented as the hyperbolic tangent Tanh applied to z values sampled from the mean and covariance given by the network. Meanwhile, the critic is modelled as a soft Q-function.

The MDP state space ($\mathcal{S}$) consists of a collection of wind and ambient features. Wind features, consisting of magnitude $|v|$ and bearing error θ, are sampled at 25 equally-spaced points, between the vertical pressure limits $[5000, 14000]$ Pa of the forecast. The ambient features consist of onboard measurements, which include the altitude h_t, ascent rate $\dot{h}_t$, balloon envelope drag area A and volume V, helium mols n_h, and total system m_T and sand m_s mass. Furthermore, the wind velocity $|v_h|$ and bearing error at the current altitude θ_h, and distance d and heading to the target $[\sin(\theta_x), \cos(\theta_x)]$ also form part of the ambient features. $d = ||(x, y)||$ is the distance at the current decision step to the target, which has a radius of r. The past three altitudes $[h_{t-1}, h_{t-2}, h_{t-3}]$, ascent rates, $[\dot{h}_{t-1}, \dot{h}_{t-2}, \dot{h}_{t-3}]$, and float actions $[a_{2,t-1}, a_{2,t-2}, a_{2,t-3}]$ are included to incorporate agent memory. Both wind and ambient features are concatenated into a single vector of length 77.

The MDP action space ($\mathcal{A}$) consists of three actions $a \in [a_0, a_1, a_2]$. Consisting of desired altitude $a_0 \in [14, 21]$ Km, time-factor $a_1 \in [1, 5]$, and finally, if to float $a_2 \in [-1, 1]$. The desired ascent rate $\dot{h}_d$ is calculated as:

$$\dot{h}_d = \mathbb{1}_{a_2 \in [-1, 0]} \left(\frac{a_0 - h_t}{a_1 \times T} \right), \tag{7}$$

where $\mathbb{1}$ represents the indicator function. Then, given the desired ascent rate, the desired sand ballast can be calculated if the desired ascent rate is larger than the current ascent rate, $\dot{h}_d > \dot{h}_t$.

$$m_{\text{s,calc}} = \rho V - \frac{1}{2g} C_d A \left| \dot{h}_d \right| \dot{h}_d - m_p - m_h \tag{8}$$

Or, the desired vented helium can be calculated if the desired ascent rate is less than the current ascent rate, $\dot{h}_d < \dot{h}_t$. Where we solve n_{calc} in,

$$\begin{aligned} \rho g \left(\frac{RT}{P} - M \right) n_{\text{calc}} - \frac{1}{2} \rho \left| \dot{h}_d \right| \dot{h}_d C_d \pi \left(\frac{3RT}{4\pi P} \right)^{\frac{2}{3}} n_{\text{calc}}^{\frac{2}{3}} \\ - (m_p + m_s) g = 0 \end{aligned} \tag{9}$$

B. Wind Data

We make use of the ECMWF's ERA5 global reanalysis dataset [31]. The wind vectors are located over a grid of

points in a parameter space of $\mathcal{X} \times \mathcal{Y} \times \mathcal{P} \times \mathcal{T} \times \mathcal{V}$. The longitude $\mathcal{X}$ and and latitudinal $\mathcal{Y}$ positions are sampled with a resolution of 0.4° at pressure points $\mathcal{P}$ ranging from 2000 Pa to 17500 Pa. Finally, the wind fields have a time separation of 6 hours and are collected between 1st November 2022 and 28th February 2023 at longitude -113° latitude 1°. The data is split into train and test, such that the training set consists of dates between 1st November 2022 to 31st January 2023 and the test set contains dates between 1st February 2023 to 28th February 2023. Furthermore, simplex noise [32] is used to augment the wind forecast to emulate forecasting errors [33].

C. Reward Function Evaluation

The effectiveness of the policy generalising to unseen data can be influenced by the reward function. A trade-off exists between the risk of reward hacking with ambiguous reward functions and the potential over-constraint policy behaviour with overly defined functions. Contextually, the agent could overfit to specific idiosyncrasies of the environment, such as weather patterns, and, if not entirely constrained, could also disregard theoretical venter or ballast limits [11].

Previous studies employed reward functions which provided no distinction of state values within the region. We find, for latex balloons, this approach led the agent to navigate close to the circumference of the region, frequently causing it to approach the region by flying around the perimeter. Therefore, we augment the previously used Step reward function R_{Step} [4], [11], by incorporating a Tanh function R_{Tanh}. This augmentation provides further incentives to fly closer to the centre radius.

The Step function, used first by Bellemare *et al.* illustrates a cliff edge, with a cliff edge constant of $c = 0.4$ to provide an immediate distinction between inside and outside the region. Additionally, a decay function is used to give higher rewards to states closer to the target. Given that $d - \rho$ is the distance from the target and τ is the half-life.

$$R_{\text{Step}} = \begin{cases} 1.0, & \text{if } d < r \\ c \times 2^{-(d-\rho)/\tau}, & \text{otherwise} \end{cases} \tag{10}$$

We make a slight adjustment to the Step reward function to encourage the agent to fly closer to the centre. As mentioned previously, states on the border of the region are inadvertently risky. Therefore, Tanh is used instead of the singular $+1.0$. We adjust the Tanh function's scaling to ensure the maximum reward at the centre is $+1.0$, which then progressively decreases to 0.72 at the boundary, maintaining a clear distinction between the two areas.

$$R_{\text{Tanh}} = \begin{cases} -(\tanh((d/20) - 3) - 1)/2, & \text{if } d < r \\ c \times 2^{-(d-\rho)/\tau}, & \text{otherwise} \end{cases} \tag{11}$$

Two separate policies are trained using the two reward functions R_{Step} and R_{Tanh}. With both policies trained using the same seeded environment and the same MDP formulation. Launch locations during training are randomly sampled uniformly across a 400 km radius $r \sim \mathcal{U}(-400, 400)$ and $\theta \sim \mathcal{U}(0, 2\pi)$.

To evaluate how well the policy generalises to unseen data, we evaluate both policies on the test dataset. We calculate the average steps within region over all trajectories as $\frac{1}{N \times T} \sum_{n=0}^{N} \sum_{t=0}^{T} \mathbb{1}_{d_t < r}$. Where N represents the total number of trajectories, and T represents the total number of decision steps. We also calculate the ratio of trajectories passing through region $\frac{1}{N} \sum_{n}^{N} \mathbb{1}_{d_t < r}$. The average time within region illustrates the average performance of the policy, whereas the number of trajectories passing through the region indicates the spread of viable launch locations.

D. Launch Location Optimisation

BO is used to build a GP model representing the spatial-temporal performance of each trained policy given the two reward functions: R_{Step} and R_{Tanh}.

An initial sample location, as previously done, is randomly sampled from a uniform distribution $(x_0, y_0) \sim \mathcal{U}(-400, 400)$, $\Delta t \sim \mathcal{U}(0, 24)$. For each episode, actions are selected by taking the mean of the GP $\mu_\theta(s)$ where each transition (s, a, s', r) is saved until a terminal condition is reached. The expected undiscounted return received throughout the trajectory is calculated, $\mathbb{E}[\sum_{t=0}^{T} R_{t+1}]$, which acts as the target, along with the initial launch parameter $\mathbf{x} \in (x_0, y_0, \Delta t)$. Both the target and sample are appended to the GP training set $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\mathbf{x}, \mathbb{E}[\sum_{t=0}^{T} R_{t+1}])\}$. The GP is then updated with this new sample point, by maximising the log marginal likelihood $\max_\theta \log p(\mathbf{y}|\mathbf{x}, \theta_{\text{gp}})$. The next sample point is chosen which maximises the expected improvement $\mathbb{E}[I(\mathbf{x})]$. This next sample point is used as the initial state of the balloon.

This procedure is performed for every day within the test set, 1st to 28th February, to evaluate the best location and time-offset to launch the balloon. To evaluate the performance of BO, we compare the average converged score and convergence time for all days within the test set against Particle Swarm Optimisation (PSO) and by uniformly sampling over the space. Given that the average convergence score is defined as $\frac{1}{N} \sum_{n=0}^{N} \mathbf{x}^*$ where in this instance n represents the date in the test set. Furthermore, the average converged index is defined as $\frac{1}{N} \sum_{n=0}^{N} \arg\max_{\mathbf{x}^*}$. We set a maximum number of steps to 1500 for each day within the test set. We compare BO against PSO and uniform sampling. To give PSO a fair chance, we linearly decay the cognitive c_1, social c_2, and inertia w such that $w = 0.4 \frac{N_i - n_i}{N_i^2} + 0.4$, $c_1 = -3 \frac{n_i}{N_i} + 3.5$, $c_2 = 3 \frac{n_i}{N_i} + 0.5$ [34]. Where n_i and N_i are the current index and maximum index respectively. The initial values for both the cognitive and social parameters are set at one, $c_1 = c_2 = 1$, and the inertial weight is $w = 0.8$.

VI. Results

Our findings reveal that the Tanh reward function generalises to the unseen test data better than the Step function. Furthermore, we illustrate that BO is able to locate an optimal launch configuration in less time compared to PSO and uniform sampling.

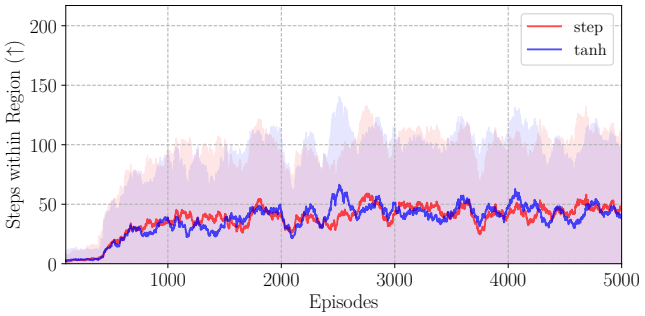

Fig. 3: Training curve of the two policies, indicating performance converging to approximately 45 steps within region. Illustrating similar performance to the training data.

A. Reward Function Evaluation

Both policies during training converge, on average, to the same time within region, as illustrated in Figure 3. Previous studies have pointed out the diminishing returns after longer training times, with some studies training a policy for weeks [4]. Hence, we stop training after 5000 episodes, given that both policies have converged.

The average time within the 50 km radius and the ratio of trajectories reaching the region in unseen test data is illustrated in Table I. The policy trained using a Tanh reward function achieved a higher average score on unseen data, showing that this policy was able to generalise better to unseen data. Furthermore, the Tanh reward function illustrated a higher proportion of trajectories reaching the region. To illustrate this further, we plot a kernel density estimate which visualises the distribution of locations reached by the balloon in Figure 4.

Expectedly, the plot illustrates a decrease in the distribution for both policies on the unseen test dataset relative to the training dataset. Notably, the policy trained with the Step function shows a greater reduction in density. To clarify the

Reward Function	Average Steps Within Region (↑)	Ratio of Trajectories Reaching the Region (↑)
Step	14.11	0.29
Tanh	**23.20**	**0.49**

TABLE I: Test data results of average time within the 50 km region and the ratio of trajectories reaching the region for both policies.

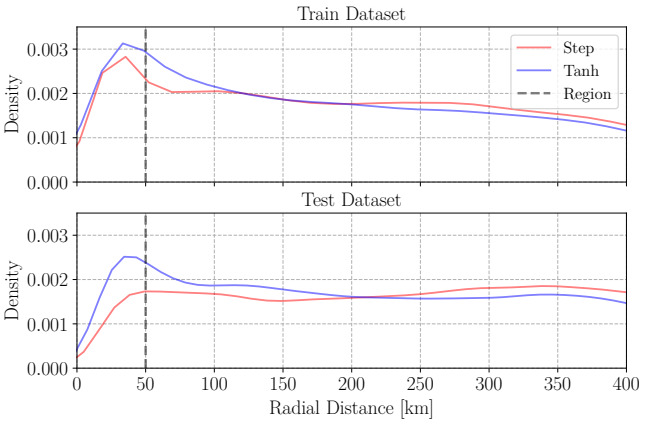

Fig. 4: Kernel density estimate visualising the distribution of positions for both policies between 0 and 400 k, indicating the worse generalisation for the Step function to unseen data.

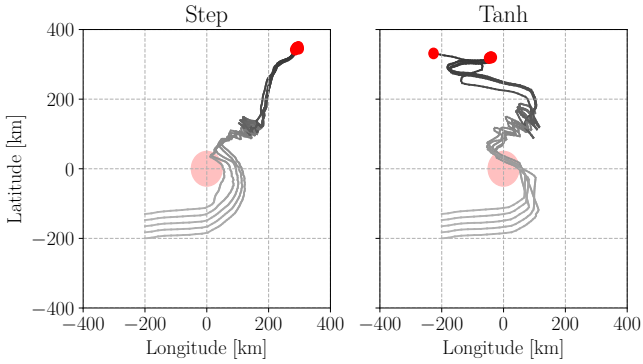

Fig. 5: Projected view of both policies trajectories given the same environment state with wind fields chosen from the test dataset. Where initial positions are chosen at the same longitude initial position with varying latitudes. The trajectories indicate the Tanh reward function incentivises the agent to fly closer to the target region. The policy trained on the Step function attempts to traverse the circumference of the region which leads to fewer steps within the region.

observed behaviour, we plot the trajectories of both policies in Figure 5.

The policy trained with the Step function inadvertently maximises its reward by exploiting the environment. More steps within the region, and hence more reward, were achieved by circumnavigating the region's circumference rather than traversing the diameter. Alternatively, the Tanh reward function incentivised flying closer to the target location, thus creating a higher separation between the bounds of the region and the balloon. This leads to a higher proportion of trajectories reaching the region and a, on average, higher time within region score.

B. Launch Location Optimisation

Average time within region, as illustrated by Figure 7, is not a good indicator for the maximum performance of the latex balloon. The effect of seasons and diurnal cycles on performance has already been illustrated in previous literature [4]. There exists a clear gap in the analysis of spatial performance for station-keeping. Take the dates in Figure 7, the most optimal time to launch given the average time within region would be between $2^{nd} - 6^{th}$ November. However, given the maximum performance, the best time would instead be between the $17^{th} - 20^{th}$ November. Clearly, the wind field pattern has a significant effect on this performance as illustrated in Figure 6.

The convergence time and the maximum converged index for BO, PSO, and uniformly sampled for the test dataset is shown in Table II. Furthermore, the table is split into evaluation metrics for both reward functions. All methods consistently converge to a similar maximum on average. However, BO converges in half the iterations required by other methods. Additionally, the Tanh method consistently attains a higher average TW50 score.

By analysing the optimal positions across two different wind field dates, as depicted in Figure 6, we observe that both PSO and BO identify subtly different launch configurations in both spatial (longitude and latitude) and temporal (launch

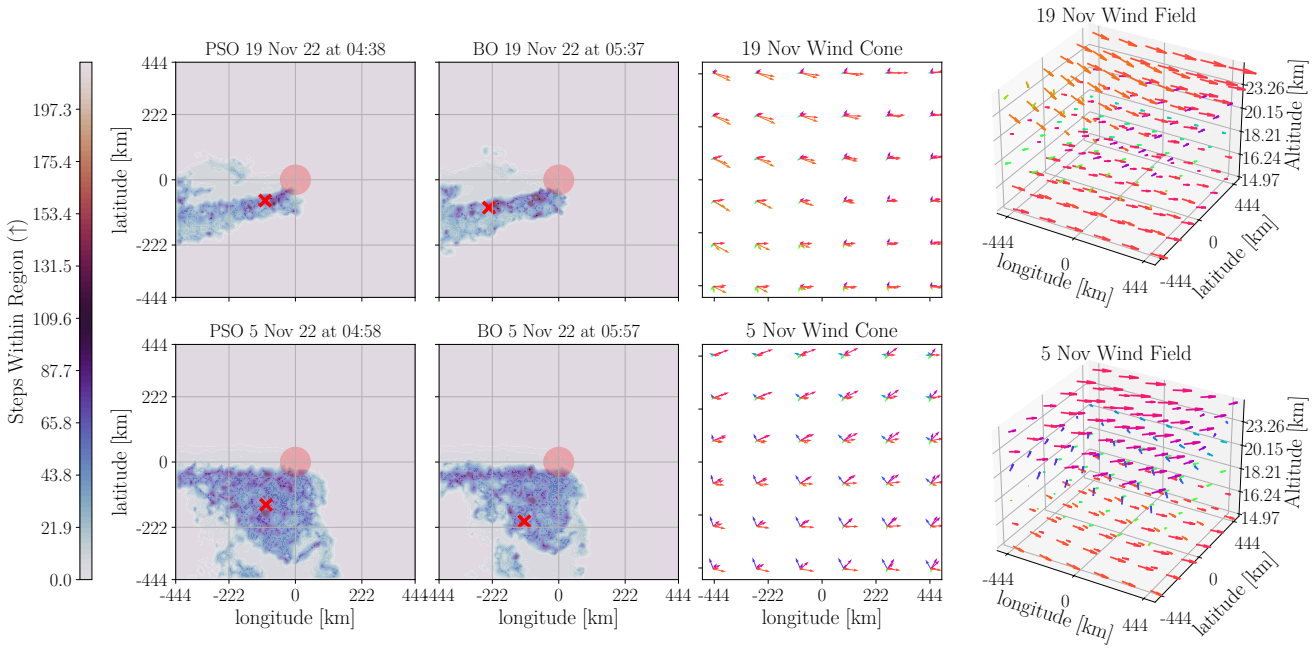

Fig. 6: Illustrating optimal positions, indicated by the red cross, across two different wind field dates. It can be observed that both PSO and BO identify subtly different launch configurations, spatially and temporally. Furthermore, it is obvious that the wind field has a significant effect on the performance, as shown by the wind cone. Where the wind cone illustrates the diversity of the wind vectors at that position.

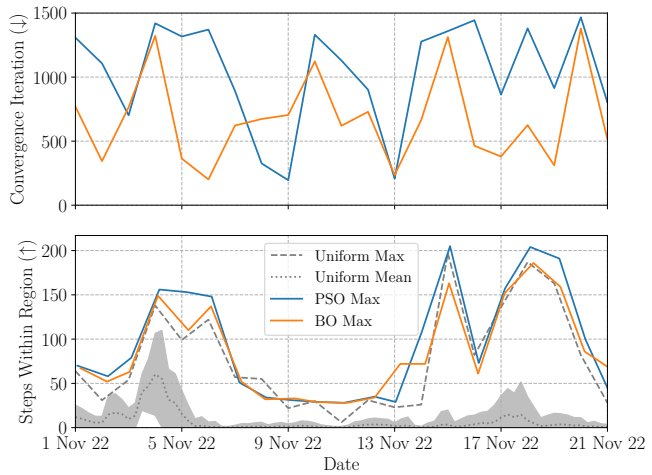

Fig. 7: Illustrates the converged index and station-keeping performance within the train dataset. Showing that BO converges to the optimal performance in fewer steps than PSO. All optimisation methods identify, on average, similar optimal performances. The range of values achievable is indicated by the standard deviation obtained by uniformly sampling.

Method	Average Converged Max (↑)	Average Converge Index (↓)
BO + Step	68.42	**573.36**
PSO + Step	69.97	669.87
Uniform + Step	67.78	805.11
BO + Tanh	86.82	**459.32**
PSO + Tanh	89.71	976.03
Uniform + Tanh	83.68	787.61

TABLE II: Evaluation metrics for all optimisation methods over the test dataset. BO reaches the maximum converged score in fewer steps compared to PSO and Uniform. Furthermore, Tanh reaches a higher average maximum time within region.

offset) aspects. The contour plots illustrate the dependence on the wind field and can be easily visualised with the wind cone. The wind cone is obtained by projecting onto the latitude-longitude plane. The wind cone is a good indicator of the diversity of wind vectors in the altitude layers. Large wind cones indicate diverse wind vectors, creating better conditions for station-keeping. Contrary to expectations, the chosen launch location is often outside the target region.

VII. Conclusion

High-altitude balloons have shown promise for their potential in atmospheric analysis and communication relay tasks, particularly through station-keeping. Yet, the efficacy of station-keeping is significantly influenced by atmospheric conditions, and determining the optimal launch location remains an open research question. This challenge is particularly pronounced for latex balloons, which have a shorter flight time compared to their super-pressure counterparts. To our knowledge, we are the first to address the problem of identifying the optimal launch location. This problem involves identifying the most optimal time and location to launch the balloon in order to optimise the total steps within the target region. This problem is multifaceted and relies on underacted balloon dynamics driven by atmospheric conditions. As done in previous works, we use a Soft Actor-Critic policy to control the balloon for station-keeping, and illustrate the problem, of misaligned behaviour at distances much further away from the target, which can limit the search space. This misaligned behaviour leads to less maximum reward and poorer generalisation of unseen wind fields. We propose optimising the agent by incentivising the agent to

fly closer to the center, achieved using a Tanh function, and through experiments we find this new reward function achieves more steps within 50 km in unseen data. We illustrate how Bayesian Optimisation can identify optimal launch locations in fewer steps compared to other optimisation methods. Furthermore, we show that the proposed reward function finds launch configurations which achieve, on average, higher steps within region. Future research will focus on examining the impact of spatial constraints to emulate restricted regions and applying our methodology to a variety of other balloon models, such as super-pressure balloons.

REFERENCES

[1] M. Tironi and M. Valderrama, "The militarization of the urban sky in Santiago de Chile: the vision multiple of a video-surveillance system of aerostatic balloons," *Urban Geography*, vol. 42, pp. 161–180, Feb. 2021.

[2] K. R. Adams, L. Gibbs, N. A. Knott, A. Broad, M. Hing, M. D. Taylor, and A. R. Davis, "Coexisting with sharks: a novel, socially acceptable and non-lethal shark mitigation approach," *Scientific Reports*, vol. 10, p. 17497, Oct. 2020. Publisher: Nature Publishing Group.

[3] Z. Wang, M. Huang, W. Han, B. Zhao, G. Zhang, L. Qian, G. Wang, and B. Li, "Optical sensing in Tibet Plateau wildlife observation based on tethered balloon," *Optik*, vol. 243, p. 167425, Oct. 2021.

[4] M. G. Bellemare, S. Candido, P. S. Castro, J. Gong, M. C. Machado, S. Moitra, S. S. Ponda, and Z. Wang, "Autonomous navigation of stratospheric balloons using reinforcement learning," *Nature*, vol. 588, pp. 77–82, Dec. 2020.

[5] S. H. Alsamhi, M. S. Ansari, L. Zhao, S. N. Van, S. K. Gupta, A. A. Alammari, A. H. Saber, M. Y. A. M. Hebah, M. A. A. Alasali, H. M. Aljabali, M. Najim, and A. Srivastava, "Tethered Balloon Technology for Green Communication in Smart Cities and Healthy Environment," Dec. 2019. arXiv:1912.11251 [eess].

[6] S. S. Alam, A. J. Islam, M. Mahmudul Hasan, and M. Mehedi Farhad, "Design and Implementation of an Embedded System to Observe the Atmospheric Condition using a Helium Balloon," in *2018 International Conference on Innovations in Science, Engineering and Technology (ICISET)*, pp. 242–246, Oct. 2018.

[7] A. Sushko, A. Tedjarati, J. Creus-Costa, S. Maldonado, K. Marshland, and M. Pavone, "Low cost, high endurance, altitude-controlled latex balloon for near-space research (ValBal)," in *2017 IEEE Aerospace Conference*, (Big Sky, MT, USA), pp. 1–9, IEEE, Mar. 2017.

[8] J. Saunders, S. Saeedi, and W. Li, "Autonomous aerial robotics for package delivery: A technical review," *Journal of Field Robotics*, vol. 41, no. 1, pp. 3–49, 2024.

[9] S. L. Jeger, N. Lawrance, F. Achermann, O. Pang, M. Kovac, and R. Y. Siegwart, "Reinforcement Learning for Outdoor Balloon Navigation: A Successful Controller for an Autonomous Balloon," *IEEE Robotics & Automation Magazine*, vol. 31, pp. 26–38, June 2024. Conference Name: IEEE Robotics & Automation Magazine.

[10] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, "Concrete Problems in AI Safety," July 2016. arXiv:1606.06565 [cs].

[11] J. Saunders, L. Prenevost, O. Şimşek, A. Hunter, and W. Li, "Resource-Constrained Station-Keeping for Latex Balloons Using Reinforcement Learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1102–1109, Oct. 2023. ISSN: 2153-0866.

[12] H. Du, M. Lv, J. Li, W. Zhu, L. Zhang, and Y. Wu, "Station-keeping performance analysis for high altitude balloon with altitude control system," *Aerospace Science and Technology*, vol. 92, pp. 644–652, Sept. 2019.

[13] Y. Liu, Z. Xu, H. Du, and M. Lv, "Increased utilization of real wind fields to improve station-keeping performance of stratospheric balloon," *Aerospace Science and Technology*, vol. 122, p. 107399, Mar. 2022.

[14] W. Dabney, M. Rowland, M. Bellemare, and R. Munos, "Distributional Reinforcement Learning With Quantile Regression," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, Apr. 2018.

[15] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," Feb. 2016. arXiv:1511.05952 [cs].

[16] Z. Xu, Y. Liu, H. Du, and M. Lv, "Station-keeping for high-altitude balloon with reinforcement learning," *Advances in Space Research*, vol. 70, pp. 733–751, Aug. 2022.

[17] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2015. Publisher: Nature Publishing Group.

[18] Y. Jiang, M. Lv, and J. Li, "Station-keeping control design of double balloon system based on horizontal region constraints," *Aerospace Science and Technology*, vol. 100, p. 105792, May 2020.

[19] E. van Wynsberghe and A. Turak, "Station-keeping of a high-altitude balloon with electric propulsion and wireless power transmission: A concept study," *Acta Astronautica*, vol. 128, pp. 616–627, Nov. 2016.

[20] A. Sóbester, H. Czerski, N. Zapponi, and I. Castro, "High-Altitude Gas Balloon Trajectory Prediction: A Monte Carlo Model," *AIAA Journal*, vol. 52, pp. 832–842, Apr. 2014.

[21] T. Fields, M. Heninger, J. LaCombe, and E. Wang, "In-flight Landing Location Predictions using Ascent Wind Data for High Altitude Balloons," in *AIAA Balloon Systems (BAL) Conference*, (Daytona Beach, Florida), American Institute of Aeronautics and Astronautics, Mar. 2013.

[22] R. M. Dreifuerst, S. Daulton, Y. Qian, P. Varkey, M. Balandat, S. Kasturia, A. Tomar, A. Yazdan, V. Ponnampalam, and R. W. Heath, "Optimizing Coverage and Capacity in Cellular Networks using Machine Learning," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (Toronto, ON, Canada), pp. 8138–8142, IEEE, June 2021.

[23] P. Asaah, L. Hao, and J. Ji, "Optimal Placement of Wind Turbines in Wind Farm Layout Using Particle Swarm Optimization," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 2, pp. 367–375, 2021.

[24] R. Farley, "BalloonAscent: 3-D Simulation Tool for the Ascent and Float of High-Altitude Balloons," in *AIAA 5th ATIO and16th Lighter-Than-Air Sys Tech. and Balloon Systems Conferences*, (Arlington, Virginia), American Institute of Aeronautics and Astronautics, Sept. 2005.

[25] J. R. Taylor, *Classical mechanics*. Sausalito, California: University Science Books, 2005.

[26] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," 2018.

[27] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft Actor-Critic Algorithms and Applications," Jan. 2019. arXiv:1812.05905 [cs, stat].

[28] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, vol. 2. Cambridge, MA: MIT press, 2006.

[29] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, vol. 13, pp. 455–492, Dec. 1998.

[30] L. S. Friedrich, A. J. McDonald, G. E. Bodeker, K. E. Cooper, J. Lewis, and A. J. Paterson, "A comparison of Loon balloon observations and stratospheric reanalysis products," *Atmospheric Chemistry and Physics*, vol. 17, pp. 855–866, Jan. 2017. Publisher: Copernicus GmbH.

[31] H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horányi, J. Muñoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers, A. Simmons, C. Soci, S. Abdalla, X. Abellan, G. Balsamo, P. Bechtold, G. Biavati, J. Bidlot, M. Bonavita, G. Chiara, P. Dahlgren, D. Dee, M. Diamantakis, R. Dragani, J. Flemming, R. Forbes, M. Fuentes, A. Geer, L. Haimberger, S. Healy, R. J. Hogan, E. Hólm, M. Janisková, S. Keeley, P. Laloyaux, P. Lopez, C. Lupu, G. Radnoti, P. Rosnay, I. Rozum, F. Vamborg, S. Villaume, and J. Thépaut, "The ERA5 global reanalysis," *Quarterly Journal of the Royal Meteorological Society*, vol. 146, pp. 1999–2049, July 2020.

[32] K. Perlin, "An image synthesizer," vol. 19, no. 3, 1985.

[33] L. Coy, M. R. Schoeberl, S. Pawson, S. Candido, and R. W. Carver, "Global Assimilation of Loon Stratospheric Balloon Observations," *Journal of Geophysical Research: Atmospheres*, vol. 124, pp. 3005–3019, Mar. 2019.

[34] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, (Anchorage, AK, USA), pp. 69–73, IEEE, 1998.