Underactuated MIMO Airship Control Based on Online Data-Driven Reinforcement Learning*

Derek Boase[1], Wail Gueaieb[1,2] and Md Suruz Miah[3]

***Abstract*— In this work, a novel online model-free controller for an underactuated dirigible is developed based on reinforcement learning and optimal control theory. A reinforcement learning structure is used while overcoming the dependence of the value function on future values by introducing a neural network that is adapted using input-output data. The suboptimal critic neural network is structured such that optimality is guaranteed over the interval from which the data is valid. The system performance is validated using a highly realistic physics engine, Gazebo, with the robot operating system (ROS) interface and the results are compared to the performance of a model-based controller specifically designed to control the airship model. It is emphasized that the proposed formulation does not leverage any knowledge of vehicle dynamics and thus is considered a vehicle agnostic control strategy.**

I. Introduction

A large advancement in the field of unmanned aerial vehicles (UAV) control has been seen in recent years, with much of the focus being on fixed-wing and multi-rotor style vehicles. With this advancement comes novel applications in the field of surveillance, delivery, aerial cinematography and data acquisition platforms [1]. However, there remain areas where long flight times, high payload-to-weight ratio and low mechanical noise are required [1]–[3]. In such cases, lighter-than air-vehicles (LTAVs) are the best, if not the only, viable option. Some specific examples include search and rescue applications [4], long flight time tasks [5], communications relays and observation tasks, such as climatological monitoring [6]. Our work contributes to the development of a real-time model-free control scheme, formulated as an optimal control problem, for a dirigible using data collected online along its trajectory. It is worth mentioning at this point that herein, the terms LTAV, dirigible, and airship, are used interchangeably.

Historically, the limitations on the commonplace use of LTAVs in these applications include challenges in control strategies stemming from modelling complexity, lack of simulation environment and high-susceptibility to aerodynamic forces [2]. With the recent development of a highly realistic airship model and turbulent wind simulation model in [2], modern control strategies can be implemented easily in a ROS/Gazebo framework, with no risk to the physical models while ensuring a smooth transition to physical systems [2], [4], [7]. The challenge of dirigible control remains and is still heavily dampened by the modelling complexity of such vehicles. Traditionally, blimp controls rely on classical PID controllers, which struggle with nonlinear systems [8], [9], and nonlinear control, which is model-based [4], [10].

Model-free control lends itself well to nonlinear systems in the face of an unknown or uncertain environment and has already been shown to be a suitable choice for UAVs [4], [7], [11], [12]. In general, machine learning based control schemes have been found to outperform classical control strategies, especially in cases of ill-defined systems, but still lacks the stability and convergence guarantees that are afforded by the latter [13]. Reinforcement learning is a data-driven subset of machine learning in which a system gains knowledge about its environment through analysis of input-output data [14]. The learning agent can evaluate the action taken by the system based on the evolution of the states following that action.

A challenge associated with reinforcement learning control is that the value function is structured recursively and therefore requires knowledge of future values. The estimation of these values can be achieved through rigorous system modelling, which is unfeasible in the case of dirigible control as the dynamics are often time-varying and the chaotic nature of the operating environment can not precisely be modelled and tested sufficiently. Even if all the parameters are known, a mathematical model for complex nonlinear systems is often difficult to formulate exactly and designers resort to simplification techniques which themselves accompany practical limitations [9]. When system models are not known, the future values are refined iteratively using Monte Carlo techniques. Monte-Carlo methods have the advantage that the values of state-action pairs are guaranteed to converge as the number of iterations tends towards infinity. However, this becomes unrealistic from the perspective of computational expense and time, and does not guarantee a satisfactory disturbance rejection behavior in the face of uncertainties that have not been seen in the training phase [14]. The limitation of such offline Monte-Carlo techniques becomes amplified for large state and action spaces.

Optimal control is a control strategy that chooses its action, or policy, by optimizing an objective function dependent on the system's design requirements and performance measures [15]. Like reinforcement learning, the optimization of the objective function requires knowledge of the system's trajectory, making traditional techniques unfeasible when a

*This work was partially supported by NSERC Grant RGPIN-2014-06512.

[1]Derek Boase and Wail Gueaieb are with the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada K1N 6N5 {dboas065, wgueaieb}@uottawa.ca

[2]Wail Gueaieb is a visiting faculty at Mohamed bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi, UAE

[3]Md Suruz Miah is with the Department of Electrical and Computer Engineering, Bradley University, Peoria, IL, 61625, USA smiah@bradley.edu

This is the postscript version of the published paper. (doi: 10.1109/IROS55552.2023.10341752)

system model or previous experience is unavailable.

The online model-free control strategy developed and tested in this work resolves the issues of unknown future values and system dynamics by using a neural network as a value function approximator. The neural network is updated using gradient descent techniques and is structured such that the value function is guaranteed to be monotonic decreasing in time. By borrowing from linear quadratic regulation, an objective function is formulated to be quadratic in the state-error and control actions ensuring a single global minimum over the interval of the time span of the gathered data. The objective function is developed with input-output data gathered along the trajectory of the vehicle. The adaptive data-driven approach of the proposed controller demonstrates robustness to external disturbances and is not limited to a priori seen scenarios. The online nature of the algorithm lends itself well to applications without a highly realistic computer simulation environment or costly hardware for accelerated computation.

Notations: We shall denote vectors and matrices with bold lowercase and uppercase letters, respectively. Let $\mathbb{N} = \{1, 2, \ldots\}$ with $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. The set of nonnegative real numbers is denoted by $\mathbb{R}_+$, where $\mathbb{R}$ represents the set of all real numbers. Nonbold letters denote scalar quantities.

II. Related Work

The earliest work in the autonomous control of airships leverage the use of empirical offline data and generalized models for the development of the controllers. A model is presented in [16] which assumes a rigid structure and deduces the aerodynamic coefficients for the model based upon 600 hours of wind tunnel testing on a specific airship. From this model, a PID and sliding mode control strategy are developed and tested in numerical simulation using MATLAB in [1], [6]. Similar model-based works are presented using PID controllers [2], [3], [17], [18], however, PID controllers are often tuned empirically, only using the calculated values as a starting point. These controllers show strong performances in simulation. Nonetheless, with the exception of the work of Price *et al.* in [2], all the others are validated only through more or less simplistic numerical simulations, and thus the results might not reflect how the controllers would work in a realistic simulation or on a physical system. Another limitation is that the developed controllers are often based on linearization techniques making their parameters only valid in specific trimmed flight regimes [8].

More recent approaches employ adaptive fuzzy sliding mode control [19], backstepping sliding mode control [20], direct adaptive fuzzy predictive control [21], and model predictive control based on linear parameter-varying system design [22]. Machine learning techniques for the control of LTAVs have also been applied with success both in simulation and in some cases on physical testing. Reinforcement learning techniques are used alongside a Gaussian process for system identification in [10]. Gaussian processes are used to advise a Q-learning altitude controller in [11], [23]. Although this approach is online and shows promising performance, the heavy computational burden of controlling high-dimensional plants limits its applications to simpler systems. A deep reinforcement learning controller is presented in [7] where it is tested in simulation using the Gazebo physics engine. The problem formulation discretizes the action space and trains the model offline for seven days to learn the parameters of the deep neural network. This process is computationally demanding and requires an accurate and realistic simulation environment for training. It also suffers from the generally known poor extrapolation performance of artificial neural networks, such as with disturbances and state values which they have not been explicitly trained on [14]. A deep residual neural network is trained using Gazebo and validated experimentally in [4]. The model therein is again trained offline mixing the policy chosen by the deep reinforcement learning agent with the action selected by a PID controller in a hybrid scheme between classical and intelligent control techniques taking advantage of the benefits of both philosophies. A stable PID formulation is still required in this scheme and therefore a strong knowledge of the vehicle dynamics, or trial-and-error tuning is still required given the dependency of the PID controller on the action selection. The controller in [4] outperforms the reference models in the testing but does so at an expense of a 28-day training period for full control of the blimp.

The proposed model-free control scheme is developed such that it may be extended to systems with high dimensions, in contrast to the limitation seen by the Gaussian process system identification method. The measurement-driven formulation allows for flexibility across vehicle design strategies and actuation schemes. Its online learning structure eliminates the need for long training periods and high-powered costly equipment, such as modern graphics processing units. The work in this paper is meant to highlight the high-level philosophies and benefits of this algorithm with a focus on its application to dirigibles. The general problem of convergence and stability for the proposed approach is still open.

III. Preliminaries and Problem Setup

This section outlines preliminaries on a deformable airship simulation model followed by the mathematical formulation of its data-driven optimal control problem that we address in this work. Fig. 1 shows a deformable LTAV simulation model highlighting its body coordinate frame. The linear airship parameters that are of interest in this work are the altitude (z), vertical velocity ($\dot{z}$), longitudinal velocity ($\dot{x}$), and longitudinal acceleration ($\ddot{x}$), where $\dot{z}$ acts along the v_{vert} axis and $\dot{x}$ and $\ddot{x}$ act along the v_{long} axis, as shown in Fig. 1. The angular parameters of interest are the roll (ϕ), roll rate ($\dot{\phi}$), pitch (θ), and pitch rate ($\dot{\theta}$).

Without loss of generality, we consider an n-dimensional partially-observable multi-input multi-output (MIMO) state-space model of an airship described in discrete-time as

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \tag{1a}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{u}_k) \tag{1b}$$

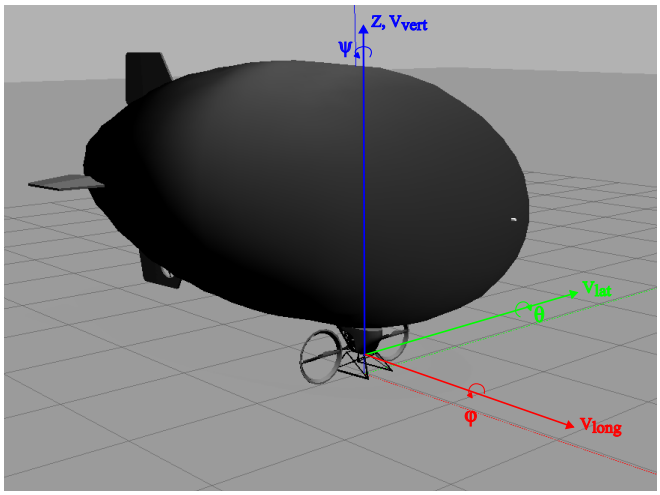


Fig. 1: Body coordinate system the LTAV model.

where $k \in \mathbb{N}_0$ denotes the discrete-time index at time $t = kT$ for some sampling period $T \in \mathbb{R}_+$, $\mathbf{x}_k = [x_{k,1}, x_{k,2}, \ldots, x_{k,n}]^T \in \mathbb{R}^n$ and $\mathbf{u}_k = [u_{k,1}, u_{k,2}, \ldots u_{k,m}]^T \in \mathbb{R}^m$ represent the airship's state and actuator input vectors, respectively. We emphasize that the state transition function $\mathbf{f} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, which contains the system drift dynamics and any disturbances, is unknown. The output of the airship, $\mathbf{y}_k \in \mathbb{R}^p$, is modelled by some unknown output function $\mathbf{g} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$ for some $p \leq n$. It is important to note that, unlike a common assumption in the literature [1], [24], the state variables in $\mathbf{x}_k$ are not assumed to be independent or decoupled.

Given the unknown dynamics of the airship model (1), the control problem is to determine the required angular velocity of the propellers, ω_k, and the propeller's pitch, γ_k, shown in Fig. 2, such that the output signals of the airship $\mathbf{y}_k = \left[z, \dot{z}_k, \dot{x}_k, \ddot{x}_k, \phi_k, \dot{\phi}_k, \theta_k, \dot{\theta}_k\right]^T$ track their corresponding references $\mathbf{y}_k^{\text{ref}} = \left[z^{\text{ref}}, \dot{z}_k^{\text{ref}}, \dot{x}_k^{\text{ref}}, \ddot{x}_k^{\text{ref}}, \phi_k^{\text{ref}}, \dot{\phi}_k^{\text{ref}}, \theta_k^{\text{ref}}, \dot{\theta}_k^{\text{ref}}\right]^T$, as $k \to \infty$. It is worth noting that the signals z_k, $\dot{x}_k$ and θ_k

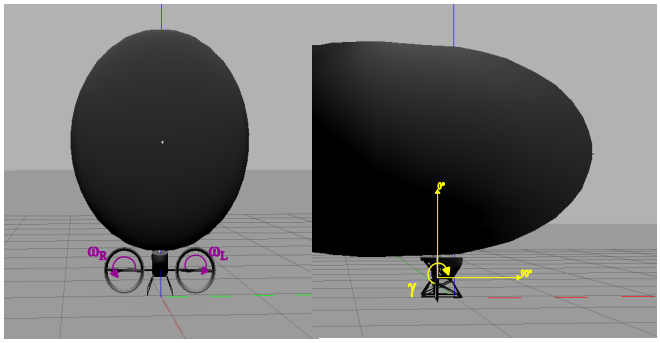


Fig. 2: Airship actuation signals.

and their derivatives are all directly coupled by the inputs while ϕ_k is not.

Note that the angular velocity applied to the left and right propellers are equal in magnitude and opposite in direction, *i.e.*, $\omega_{k,R} = -\omega_{k,L}$.

IV. Model-Free Control Design

This work employs online data-driven control technology while formulating the airship as an optimal control problem. Therefore, the proposed control scheme does not rely on the mathematical model of the airship.

Consider the airship's unknown state-space model (1). An optimal control strategy in coooperation with a (critic) neural network following a reinforcement learning structure is developed to control system outputs. The performance of the controller is measured using a cost function that allows the designer to weigh the controller's ability to eliminate steady-state error while balancing its actuation energy consumption. To formulate this behavior, the error dynamics defined by $\mathbf{e}_{k+1} = \mathbf{h}(\mathbf{e}_k, \mathbf{u}_k)$, for $k \in \mathbb{N}_0$ is introduced, where $\mathbf{e}_k = [e_{k,1}, e_{k,2}, \ldots e_{k,n}]^T \in \mathbb{R}^n$. Here $e_{k,i}$ represents the normalized error of $\tilde{e}_{k,i} = (y_{k,i} - y_{i_\infty})/y_i^{max}$ saturated in the range $[-1, 1]$ for $i = 1, 2, \ldots, p$. This is done to prevent divergence or domination of larger terms in the error vector [25]. The normalization is, $e_{k,i} = \text{sgn}(\tilde{e}_{k,i}) \min(1, |\tilde{e}_{k,i}|)$. Similar to (1), the function $\mathbf{h} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^p$, representing the error dynamics of the states to be controlled is unknown.

The quadratic scalar performance index is defined over an infinite horizon as $J = \frac{1}{2}\sum_{k=0}^{\infty} \mathbf{e}_k^T \mathbf{Q} \mathbf{e}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k$. The matrices $\mathbf{Q} \in \mathbb{R}^{n \times n}$ and $\mathbf{R} \in \mathbb{R}^{m \times m}$ are symmetric positive-definite weighting matrices.

Using a modification of the discrete-time Bellman equation for state control in linear quadratic regulators, the value function V of the system is taken as

$$V(\mathbf{e}_k, \mathbf{u}_k) = \frac{1}{2}\left[\mathbf{e}_k^T \mathbf{Q} \mathbf{e}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k\right] + V(\mathbf{e}_{k+1}, \mathbf{u}_{k+1}). \quad (2)$$

By requiring that $\mathbf{Q}$ and $\mathbf{R}$ are symmetric positive-definite, it can be confirmed that the value function is monotonically decreasing. Applying Bellman's principle of optimality to (2) yields the following optimal value function:

$$V^*(\mathbf{e}_k, \mathbf{u}_k^*) = \frac{1}{2} \min_{\mathbf{u}_k} \left\{ \left[\mathbf{e}_k^T \mathbf{Q} \mathbf{e}_k + \mathbf{u}_k^T \mathbf{R} \mathbf{u}_k\right] + V^*(\mathbf{e}_{k+1}, \mathbf{u}_{k+1}^*) \right\} \quad (3)$$

where $\mathbf{u}_k^*$ denotes the optimal action at time instant k. Minimizing the right-hand side of (3), the $\mathbf{u}_k^*$ must satisfy $\partial V^*(\mathbf{e}_k, \mathbf{u}_k)/\partial \mathbf{u}_k = 0$. This formulation presents an issue for both online and model-free control techniques as it assumes a priori knowledge of the future optimal value function, $V^*(\mathbf{e}_{k+1}, \mathbf{u}_{k+1})$, which requires exhaustive experience, accurate system modelling, and often both [8], [9], [14]. By augmenting the error and the control action, a new notation can be introduced. Let $\mathbf{w}_k \equiv [w_{k,1}, w_{k,2}, \ldots, w_{k,n+m}]^T \equiv [\mathbf{e}_k^T \ \mathbf{u}_k^T]^T \in \mathbb{R}^{n+m}$. Reformulation (2) leads to

$$V(\mathbf{w}_k) = \frac{1}{2} \mathbf{w}_k^T \bar{\mathbf{P}} \mathbf{w}_k + V(\mathbf{w}_{k+1}) \quad (4)$$

where $\bar{\mathbf{P}}$ is a block diagonal matrix whose upper-left and lower-right elements are $\mathbf{Q}$ and $\mathbf{R}$, respectively.

Motivated by Weierstrass higher-order approximation theorem, let $\hat{V}$ denote an approximation of V [15]. Then, (4)

implies

$$\hat{V}(\mathbf{w}_k) \equiv \frac{1}{2}\mathbf{w}_k^T \mathbf{P}^{[r]} \mathbf{w}_k, \quad \mathbf{P}^{[r]} = \begin{bmatrix} \mathbf{P}_{ee}^{[r]} & \mathbf{P}_{eu}^{[r]} \\ \mathbf{P}_{ue}^{[r]} & \mathbf{P}_{uu}^{[r]} \end{bmatrix} \tag{5}$$

where $\mathbf{P}_{ee}^{[r]} \in \mathbb{R}^{n\times n}$, $\mathbf{P}_{eu}^{[r]} \in \mathbb{R}^{n\times m}$, $\mathbf{P}_{ue}^{[r]} \in \mathbb{R}^{m\times n}$, and $\mathbf{P}_{uu}^{[r]} \in \mathbb{R}^{m\times m}$, are design parameters defining the symmetric positive-definite matrix $\mathbf{P}^{[r]}$. The diagonal elements, $\mathbf{P}_{ee}^{[r]}$ and $\mathbf{P}_{uu}^{[r]}$, represent the weighting terms for the error and control actions, while the off-diagonal terms $\mathbf{P}_{eu}^{[r]}$ and $\mathbf{P}_{ue}^{[r]}$, are left for the approximation of the unknown and uncertain terms encapsulated in the future values.

Applying Bellman's principle of optimality to (5) yields

$$V^*(\mathbf{w}_k) = \frac{1}{2}\min_{\mathbf{u}_k}\left\{\mathbf{w}_k^T \mathbf{P}^{[r]} \mathbf{w}_k\right\}. \tag{6}$$

Evaluating (6) and solving for $\mathbf{u}_k^*$, the optimal control action is then

$$\mathbf{u}_k^* = -{\mathbf{P}_{uu}^{[r]}}^{-1}\mathbf{P}_{ue}^{[r]}\mathbf{e}_k \tag{7}$$

The inverse of $\mathbf{P}_{uu}^{[r]}$ is guaranteed to exist since $\mathbf{P}^{[r]}$ is symmetric positive-definite. Although (7) satisfies the requirements of optimality, the matrix $\mathbf{P}^{[r]}$ still needs to be constructed such that the error, the control action, and future value dynamics, are encapsulated. To that end, we formulate a single-layer critic neural network whose weights are the stacked elements of $\mathbf{P}^{[r]}$. The neural network updates its weights using a gradient descent technique at every policy iteration $r \in \{0, 1, \ldots\}$ based on the previous input-output characteristics.

To estimate the value function, we define a vector consisting of stacked columns of $\mathbf{P}^{[r]}$ at policy index r, $\tilde{\mathbf{p}}^{[r]} = [p_{1,1}^{[r]}, p_{1,2}^{[r]}, \ldots, p_{i,j}^{[r]}]^T, \forall i, j \in \{1, 2, \ldots, n+m\}$. We can then rewrite (5) as

$$\hat{V}(\mathbf{w}_k) = \frac{1}{2}{\tilde{\mathbf{p}}^{[r]}}^T(\mathbf{w}_k \otimes \mathbf{w}_k) \tag{8}$$

where $\otimes$ represents the Kronecker product. Formulation (8) represents the structure of a single layer partially connected neural network whose output is $\hat{V}(\mathbf{w}_k)$ with input and output weights being $\mathbf{w}_k \otimes \mathbf{w}_k$ and $\tilde{\mathbf{p}}^{[r]}$, respectively. Due to the symmetry of $\mathbf{P}^{[r]}$ and the repeated terms in the Kronecker product, (8) is further simplified, removing the redundant terms and reducing its order. Let $\mathbf{\Omega}^{[r]} = [p_{1,1}^{[r]}, p_{1,2}^{[r]}, \ldots, p_{i,j}^{[r]}]^T$ and $\rho(\mathbf{w}_k) = [C_{1,1}z_{k,1}^2, C_{1,2}z_{k,1}z_{k,2}, \ldots, C_{i,j}z_{k,i}z_{k,j}]^T$, $\forall\{(i,j) \mid i \in \{1, 2, \ldots, n+m\},\ j \leq i\}$, where $C_{i,j} = \frac{1}{2}$ if $i = j$, and $C_{i,j} = 1$ if $i \neq j$. The reduced order value function approximation is then

$$\hat{V}(\mathbf{w}_k) = {\mathbf{\Omega}^{[r]}}^T\rho(\mathbf{w}_k) \tag{9}$$

Expression (9) represents the critic network and is also in the form of a single layer partially connected neural network but with fewer terms than (8).

The weights of (9) are updated using the gradient descent method applied on the squared temporal difference error between the desired value error function $\tilde{V}_{k,k+1}^{[d]} = V(\mathbf{w}_k) - V(\mathbf{w}_{k+1}) = \frac{1}{2}\mathbf{w}_k^T\bar{\mathbf{P}}\mathbf{w}_k$ and the estimated value error function $\tilde{V}_{k,k+1}^{[r]} = {\mathbf{\Omega}^{[r]}}^T(\rho(\mathbf{w}_k) - \rho(\mathbf{w}_{k+1})) = {\mathbf{\Omega}^{[r]}}^T\tilde{\rho}_{k,k+1}$. The temporal difference equation for the critic network is then defined over the previous $\eta = \frac{1}{2}(n+m)(n+m+1)$ discrete-time steps in order to ensure that the system is persistently excited [8]. The squared error of the desired value error function and estimated value error function is then

$$\delta_c = \frac{1}{2}\sum_{k=0}^{\eta}\left[\tilde{V}_{k,k+1}^{[d]} - \tilde{V}_{k,k+1}^{[r]}\right]^2 \tag{10}$$

By defining, $\mathbf{\Upsilon} = [\mathbf{\Upsilon}_0, \mathbf{\Upsilon}_1, \ldots, \mathbf{\Upsilon}_{\eta-1}]^T \in \mathbb{R}^{\eta\times\eta}$, with $\mathbf{\Upsilon}_\kappa = \tilde{\rho}_{k,k+1}^T$, $\forall\kappa \in \{0, 1, \ldots, \eta-1\}$, and $\mathbf{v}^{[d]} = \left[v_0^{[d]}, v_1^{[d]}, \ldots, v_{\eta-1}^{[d]}\right] \in \mathbb{R}^\eta$, where $v_\kappa^{[d]} = \frac{1}{2}\mathbf{w}_{k+\kappa}^T\bar{\mathbf{P}}\mathbf{w}_{k+\kappa}$, $\forall\kappa \in \{0, 1, \ldots, \eta-1\}$, the squared error modeled in (10) can be written in compact form as

$$\delta_c = \frac{1}{2}\left\|\mathbf{v}^{[d]} - \mathbf{\Upsilon}\mathbf{\Omega}^{[r]}\right\|^2$$

The critic weights are updated on every policy iteration $r \in \mathbb{N}_0$ such that $k+1$ is an integer multiple of η. The weights of the network are updated using gradient descent for some learning rate parameter $\alpha \ll 1$. The update rule is given as,

$$\mathbf{\Omega}^{[r+1]} = \mathbf{\Omega}^{[r]} - \alpha\frac{\partial\delta_c}{\partial\mathbf{\Omega}^{[r]}} = \mathbf{\Omega}^{[r]} - \alpha\mathbf{\Upsilon}^T\left(\mathbf{\Upsilon}\mathbf{\Omega}^{[r]} - \mathbf{v}^{[d]}\right) \tag{11}$$

The matrix $\mathbf{P}^{[r]}$ can now be reconstructed using the weight vector $\mathbf{\Omega}^{[r]}$ determined by the update rule (11) for some policy index $r \in \mathbf{N}_0$. The optimal control action (7) is then computed using the updated weight matrix $\mathbf{P}^{[r]}$.

V. Experiments

This section validates the proposed control strategy by simulating a partially observable dirigible in robot operating system (ROS) using the highly realistic Gazebo physics engine. The results of the data-driven controller are compared to those of a hierarchical cascade control (HCC) strategy actuated using LibrePilot ground control station, as outlined in [2].

A. Experimental Setup

The simulation environment and model parameters for the experimental validation of the proposed model-free controller are outlined for reproducibility. The proposed controller is applied to a realistic deformable airship model through a ROS/Gazebo interface presented in [2]. The airship is tested in nominal and disturbed test cases. The disturbances of the latter case are simulated using the Dryden turbulence model detailed in [26], [27] with wind speeds reaching $1.5\ [\mathrm{m\,s^{-1}}]$ in the xy-plane.

The dirigible model considered for this research takes two input signals to control eight airship state (output) variables, *i.e.*, $m = 2$, $p = 8$. The output variables are tabulated in Table I. The two input signals are the angular velocity of the propellers $\omega_k = \omega_{k,L} = -\omega_{k,R} \in [-859.4, 1718.9]^\circ\,\mathrm{s^{-1}}$ and their pitch $\gamma_k \in [-90, 90]^\circ$ with respect to the longitudinal axis. The control loop is triggered at a sampling frequency of $13.5\,\mathrm{Hz}$. The learning rate is set to $\alpha = 0.005$.

TABLE I: Controlled states, targets, and normalization values

State	Parameter	Target	Max Val, y_i^{max}
x_1	Height z	$15\,\mathrm{m}$	$6\,\mathrm{m}$
x_2	Vertical Velocity $\dot{z}$	$0\,\mathrm{m/s}$	$1\,\mathrm{m\,s^{-1}}$
x_3	Longitudinal Velocity $\dot{x}$	$2\,\mathrm{m/s}$	$1.5\,\mathrm{m\,s^{-1}}$
x_4	Longitudinal Acceleration $\ddot{x}$	$0\,\mathrm{m/s^2}$	$1\,\mathrm{m\,s^{-2}}$
x_5	Roll ϕ	$0°$	$180°$
x_6	Roll Rate $\dot{\phi}$	$0\,°/\mathrm{s}$	$180\,°\,\mathrm{s^{-1}}$
x_7	Pitch θ	$0°$	$180°$
x_8	Pitch Rate $\dot{\theta}$	$0\,°/\mathrm{s}$	$180\,°\,\mathrm{s^{-1}}$

The matrices $\mathbf{Q}$, $\mathbf{R}$, and $\mathbf{P}^{[0]}$ where chosen through trial and error with random initializations. It is noted that the response is only minimally effected by the choice of $\mathbf{Q}$ and $\mathbf{R}$ while the choice of $\mathbf{P}$ had a large impact on the convergence of the weights and systems response. The relative independence of $\mathbf{Q}$ and $\mathbf{R}$ comes from the fact that they effect the desired value error function such that $\tilde{V}_{k,k+1}^{[d]} \to 0$ as $\mathbf{w} \to 0$.

$$\mathbf{Q} = \begin{bmatrix} 2.11 & 1.55 & 1.87 & 1.54 & 1.80 & 1.94 & 1.35 & 1.18 \\ 1.55 & 2.47 & 2.12 & 2.52 & 2.71 & 2.46 & 1.76 & 2.35 \\ 1.87 & 2.12 & 2.65 & 1.98 & 2.66 & 2.39 & 1.31 & 1.58 \\ 1.54 & 2.52 & 1.98 & 2.86 & 2.77 & 2.42 & 1.78 & 2.48 \\ 1.80 & 2.71 & 2.66 & 2.77 & 3.55 & 2.74 & 1.84 & 2.28 \\ 1.94 & 2.46 & 2.39 & 2.42 & 2.74 & 2.94 & 1.64 & 2.14 \\ 1.35 & 1.76 & 1.31 & 1.78 & 1.84 & 1.64 & 1.89 & 1.76 \\ 1.18 & 2.35 & 1.58 & 2.48 & 2.28 & 2.14 & 1.76 & 2.54 \end{bmatrix}$$

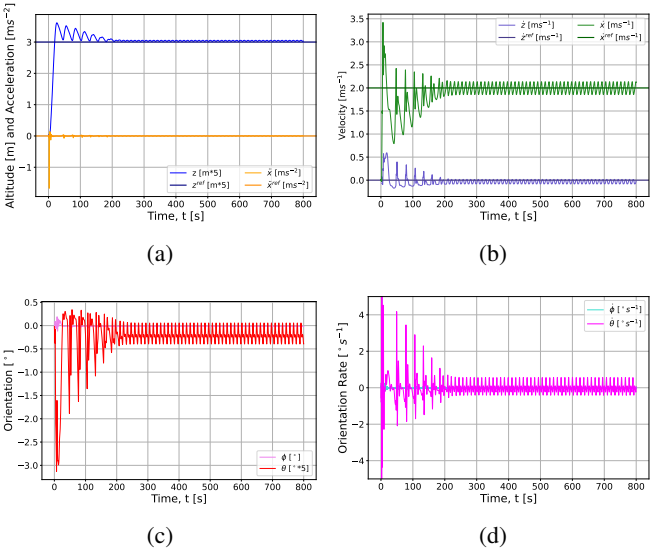
$$\mathbf{R} = \begin{bmatrix} 0.66 & 0.72 \\ 0.72 & 0.81 \end{bmatrix}$$

$$\mathbf{P}_{ee}^{[0]} = \begin{bmatrix} 3.42 & 2.16 & 1.99 & 2.35 & 2.04 & 3.16 & 2.44 & 3.39 \\ 2.16 & 2.08 & 1.71 & 1.86 & 1.32 & 2.27 & 1.83 & 2.66 \\ 1.99 & 1.71 & 2.68 & 2.33 & 1.78 & 2.31 & 2.32 & 3.03 \\ 2.35 & 1.86 & 2.33 & 2.86 & 2.16 & 2.76 & 2.45 & 3.47 \\ 2.04 & 1.32 & 1.78 & 2.16 & 2.17 & 2.25 & 2.14 & 3.00 \\ 3.16 & 2.27 & 2.31 & 2.76 & 2.25 & 3.94 & 2.75 & 3.25 \\ 2.44 & 1.83 & 2.32 & 2.45 & 2.14 & 2.75 & 2.73 & 3.35 \\ 3.39 & 2.66 & 3.03 & 3.47 & 3.00 & 3.25 & 3.35 & 5.16 \end{bmatrix}$$

$$\mathbf{P}_{eu}^{[0]} = \mathbf{P}_{ue}^{[0]^T} = \begin{bmatrix} 2.59 & 1.40 \\ 1.85 & 9.23 \\ 1.76 & 1.27 \\ 2.09 & 1.66 \\ 1.59 & 1.73 \\ 2.50 & 1.55 \\ 2.23 & 1.55 \\ 2.84 & 2.32 \end{bmatrix}, \mathbf{P}_{uu}^{[0]} = \begin{bmatrix} 2.93 & 1.22 \\ 1.22 & 1.51 \end{bmatrix}$$

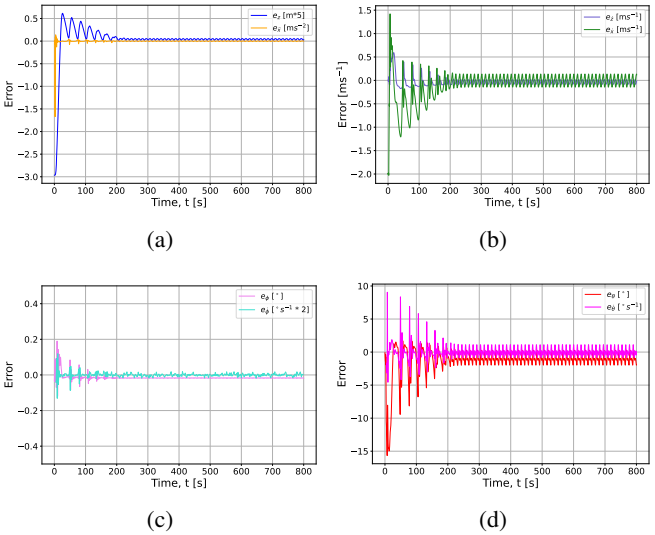
B. Results and Discussion

The trajectories from the nominal test case for the data-driven controller are shown in Fig. 3 with corresponding output errors plotted in Fig. 4. In general, the proposed control strategy is successful in reducing the steady-state error of the output states while accepting small amplitude oscillations about the reference. The notable exception is the pitch θ which oscillates around $-1°$. This behaviour is due to the thrust vector from the propellers which induces a negative moment about the pitch axis since the propellers are mounted below the center of gravity [24], [28]. Given the actuators that are used in this experiment, there are no means for the airship to influence the pitch of the vehicle without also influencing the longitudinal velocity or altitude, hence the controller balances the elimination of the steady-state



(a) (b) (c) (d)

Fig. 3: Trajectories for the nominal test case using the proposed controller. ?? height and longitudinal acceleration, ?? vertical and longitudinal velocities, ?? roll and pitch, and ?? roll and pitch rates.



(a) (b) (c) (d)

Fig. 4: Error signals for the nominal test case using the proposed controller. ?? error in height and longitudinal acceleration, ?? error in vertical and longitudinal velocities, ?? error in roll and pitch, and ?? error in roll and pitch rates.

error for each of the coupled states, inducing oscillations. In fact, in the comparison testing introduced in Section V-C, the steady-state pitch of the comparison controller, is approximately $-3.5°$ for flying at a constant altitude and longitudinal velocity.

Fig. 5 shows the control actions in $\mathbf{u}$ taken on the propellers and thruster servo, respectively. The propellers angular velocity decays in magnitude for the first $225\,\mathrm{s}$ before exhibiting a harmonic motion with a small amplitude in an attempt to regulate the competing control demands. The thrust angle also shows a small decrease in peak-to-peak

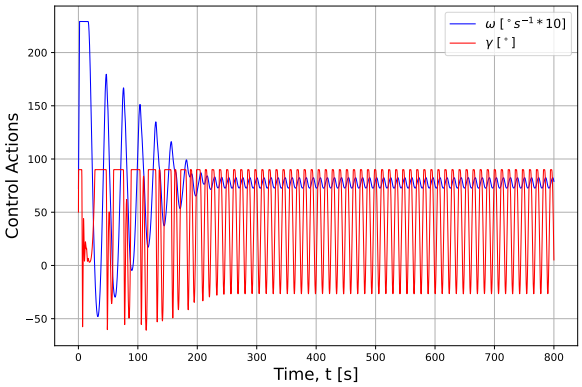
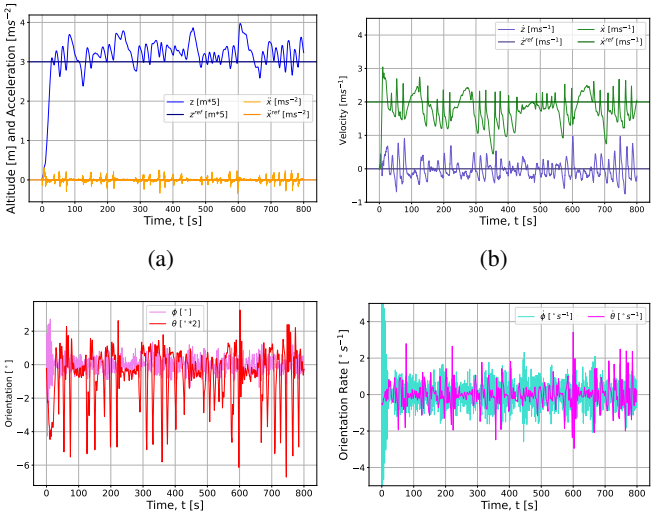


Fig. 5: Control actions for the nominal test case with the model-free controller.

values following the 225 s mark. However, the oscillations do not decay further. These oscillations in both signals are a result of the coupling of states and the competing nature of the reference output signals. To quantify the efforts of each output for comparison purposes, the integral of the squared effort is taken for the angular velocities, *i.e.*, $\omega_{\text{nominal}} = T\sum_{k=0}^{11000} \omega_k^2 \approx 359{,}257\,\text{rad}^2\,\text{s}^{-1}$ as well as for the angles, *i.e.*, $\gamma_{\text{nominal}} = T\sum_{k=0}^{11000} \gamma_k^2 \approx 957\,\text{rad}^2\,\text{s}$.

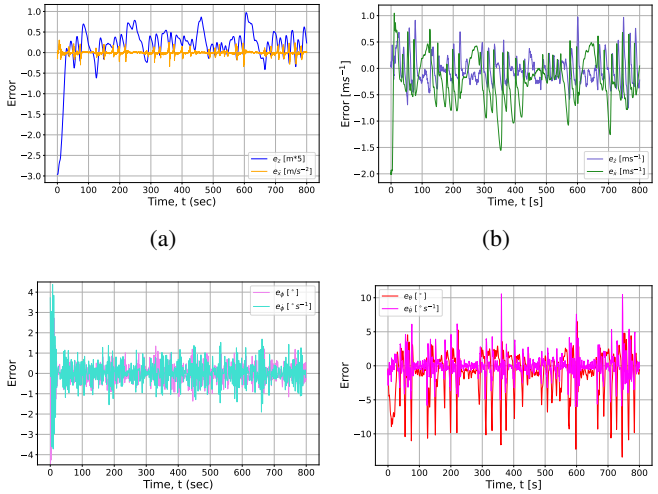
The output trajectories and the corresponding error signals from the disturbed test case for the proposed controller are shown in Fig. 6 and in Fig. 7. The states and targets of



(a) (b) (c) (d)

Fig. 6: Trajectories for the disturbed test case using the proposed controller. ?? height and longitudinal acceleration, ?? vertical and longitudinal velocities, ?? roll and pitch, and ?? roll and pitch rates.

the disturbed case are the same as in the nominal case. The proposed controller is able to oscillate about the reference signals despite the influence of turbulent wind gusts of up to 1.5 $\left[\text{m}\,\text{s}^{-1}\right]$. Higher oscillation amplitudes are observed in the trajectories of each of the states, which is an expected



(a) (b) (c) (d)

Fig. 7: Error signals for the disturbed test case using the proposed controller. ?? error in height and longitudinal acceleration, ?? error in vertical and longitudinal velocities, ?? error in roll and pitch, and ?? error roll and pitch rates.

result as the vehicle is inherently susceptible to aerodynamic forces. In all cases, the controller is able to learn the disturbances and guide the vehicle trajectories back towards their references.

Fig. 8 shows the control actions taken on the propellers and thrust vector servo for the disturbed test case. As expected, a higher oscillation in the control actions is observed with respect to the nominal case in an attempt to counteract the disturbances while the moel-free controller learns in the changing environment.

Again, numerically approximating the integral of the square of the control actions, the values are quoted as, $\omega_{\text{wind}} = T\sum_{k=0}^{11000} \omega_k \approx 865{,}444\,\text{rad}^2\,\text{s}^{-1}$ and $\gamma_{\text{wind}} = T\sum_{k=0}^{11000} \gamma_k \approx 2703\,\text{rad}^2\,\text{s}$. Predictably, these values are higher than in the nominal case as the controller requires more effort in the face of the disturbances.

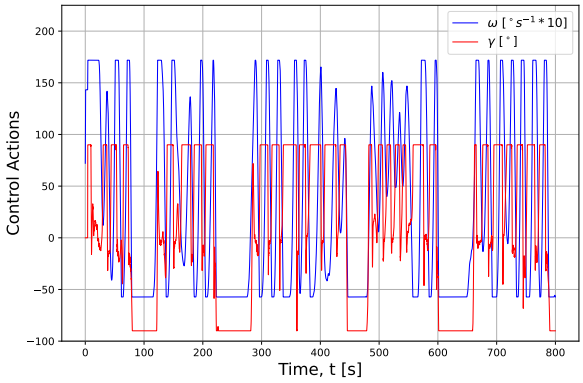


Fig. 8: Control actions for the disturbed test case with the data-driven controller.

C. Comparison

To better assess the performance of the proposed controller, it is compared to the model presented in [2]. Before presenting the results, the differences between the two controllers and their implementations are introduced. In addition to the same actuators that are used for the proposed controller, the existing control scheme presented in [2] uses four control surfaces (rudders and elevators) at the rear of the vehicle and a tail on the lower vertical stabilizer propeller. The control mechanism for the comparison model is based upon the control structures presented in [6], [29] with the use of a vehicle-agnostic control structure offered by the LibrePilot ground control station. The control structure is based on hierarchical proportional (P) and proportional-integral (PI) type controllers whose reference trajectories are found using a path-following algorithm developed in [2]. To create a test that is as similar as possible to the one used for the measurement-driven controller, two waypoints are used, the first is placed a distance of $500\,\mathrm{m}$ away from the initial position and the second is placed at a distance of $2500\,\mathrm{m}$ away both with a bearing of 0° with respect to the initial pose of the airship. The desired altitude with respect to the initial position of the airship is $15\,\mathrm{m}$ and the target velocity is $2.0\,\mathrm{m\,s^{-1}}$. It is noted for completeness that the comparison algorithm attempts to control the pitch, pitch rate, yaw rate, thrust and 3-dimensional velocity vector. However, the choice of trajectory attempts to minimize the demand for yaw control.

It is worth noting that the model-free control algorithms presented in [12] and [30] were attempted as comparison algorithms to offer a more meaningful comparison, but both algorithms were unstable when applied to the dirigible model in hand since they are not designed for general MIMO systems.

Fig. 9 shows a comparison of the altitude, longitudinal velocity and pitch for the proposed model-free controller (MFC) and the hierarchical cascade controller (HCC) in the nominal case. The mean and the peak-to-peak values of the comparison at steady state are tabulated in Table II.

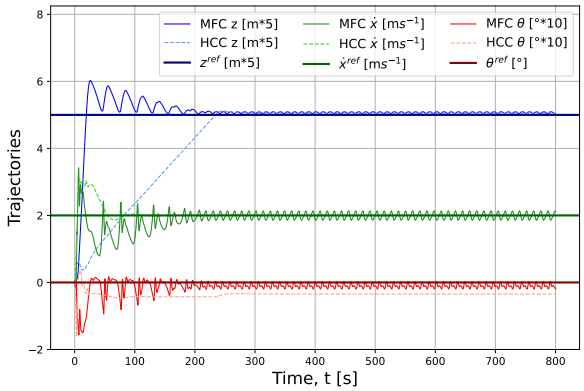


Fig. 9: Height, longitudinal velocity, and pitch, for the model-free controller and the comparison controller in the nominal test case.

It is observed that both controllers find the reference for the altitude and longitudinal velocity with small amplitude oscillations about the targets with the system response for the altitude rising towards the reference signal well before the comparison controller. The pitch for the comparison algorithm is approximately 3.5 times larger in the model-free controller for $t > 300\,\mathrm{s}$. The mean and the peak-to-peak values of the comparison at a steady state are tabulated in Table II. These values show comparable mean, with a smoother trajectory seen in the HCC. This is due to the under-actuation of the simulation of the model-free control scheme.

TABLE II: Trajectory statistics for controller comparison in the nominal test case for $t > 300\,\mathrm{s}$

	Proposed MFC		HCC	
Quantity	Mean	Peak-to-Peak	Mean	Peak-to-Peak
z [m]	15.2	0.18	15.15	0.007
$\dot{x}$ [m s^{-1}]	1.99	0.10	2.00	0.005
θ [°]	−0.98	2.27	−3.46	0.07

The trajectories for the disturbed case are shown in Fig. 10 with the results summarized in Table III. Given the consistent oscillations of the responses, the mean and the variance over the entire trajectory are calculated. It is clear that in the face of disturbances the proposed MFC outperforms the HCC in both altitude and longitudinal velocity control. The statistics for the pitch in both cases are comparable with the only notable difference being the reduced variance in the MFC controller.

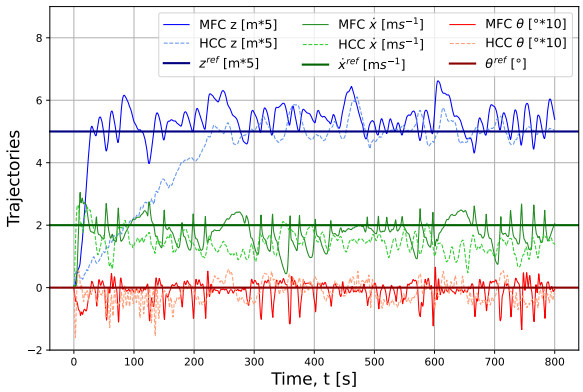


Fig. 10: Height, longitudinal velocity, and pitch, for the model-free controller and the comparison controller in the nominal test case.

TABLE III: Trajectory statistics for controller comparison in the disturbed test case

	Proposed MFC		HCC	
Quantity	Mean	Variance	Mean	Variance
z [m]	15.79	6.13	13.01	17.34
$\dot{x}$ [m s^{-1}]	1.77	0.19	1.41	0.12
θ [°]	−1.11	9.67	−1.09	10.50

The integral of the squared errors for both controllers in

the nominal and disturbed cases are summarized in Table IV. The model-free controller outperforms its counterpart in

TABLE IV: Comparison between the integral of the squared errors and control actions for both MFC and HCC controllers.

	Proposed MFC		HCC	
Quantity	Nominal	Disturbed	Nominal	Disturbed
$T\sum_{k=0}^{11000} e_{k,z}$	2501	10,797	15,594	17,037
$T\sum_{k=0}^{11000} e_{k,\dot{x}}$	69	396	36	352
$T\sum_{k=0}^{11000} e_{k,\theta}$	4604	17,455	11,095	10,501

tracking the altitude and pitch in the nominal test case, with the squared error being comparable for the longitudinal velocity. The results of both controllers in the disturbed case are closer, with the proposed MFC still outperforming the HCC in altitude regulation, but compromising the pitch control to do so. This is expected due to the higher number of degrees of freedom in the HCC due to its four additional actuators than the ones used by the MFC.

VI. Conclusion

In this paper, we present a model-free MIMO algorithm for the control of LTAVs based on reinforcement learning. The proposed technique adopts an online data-driven learning scheme overcoming the burden and flaws of other machine learning paradigms that are based on offline training. The controller has shown promising qualities as a MIMO control strategy for coupled nonlinear systems without the need for a priori knowledge of the plant or the environment. The model-free strategy has successfully contended with and, by certain metrics, outperformed a model-based controller despite being underactuated. This work has demonstrated the model-free controller's ability to balance competing control targets and has further shown itself to be more robust to disturbances.

Acknowledgment

The authors would like to thank Eric Price, Yu Tang Liu, Michael Black, and Aamir Ahmad, for their help and support throughout this work. The authors are also grateful to the Natural Sciences and Engineering Research Council of Canada (NSERC) for partially financing this research.

References

[1] E. Carneiro de Paiva, F. Benjovengo, and S. Siqueira Bueno, "Sliding mode control for the path following of an unmanned airship," *IFAC Proceedings Volumes*, vol. 40, no. 15, pp. 221–226, 2007.

[2] E. Price, Y. T. Liu, M. J. Black, and A. Ahmad, "Simulation and control of deformable autonomous airships in turbulent wind," in *Intelligent Autonomous Systems 16*, M. H. Ang Jr, H. Asama, W. Lin, and S. Foong, Eds. Cham: Springer International Publishing, 2022, pp. 608–626.

[3] W. Adamski, P. Herman, Y. Bestaoui, and K. Kozłowski, "Control of airship in case of unpredictable environment conditions," in *2010 SysTol*, 2010, pp. 843–848.

[4] Y. T. Liu, E. Price, M. J. Black, and A. Ahmad, "Deep residual reinforcement learning based autonomous blimp control," in *2022 IEEE/RSJ IROS*, 2022, pp. 12 566–12 573.

[5] S. Recoskie, "Autonomous hybrid powered long ranged airship for surveillance and guidance," Ph.D. dissertation, University of Ottawa, 2014.

[6] E. de Paiva, S. Bueno, S. Gomes, J. Ramos, and M. Bergerman, "A control system development environment for aurora's semi-autonomous robotic airship," in *Proceedings 1999 IEEE ICRA*, vol. 3, May 1999, pp. 2328–2335 vol.3.

[7] Y. T. Liu, E. Price, P. Goldschmid, M. J. Black, and A. Ahmad, "Autonomous blimp control using deep reinforcement learning," 2021. [Online]. Available: https://arxiv.org/abs/2109.10719

[8] K. J. Astrom and B. Wittenmark, *Adaptive Control*, 2nd ed., Dover, Ed. Addison-Wesley Publishing Company, 2008.

[9] N. S. Nise, *Control Systems Engineering*, J. Brady, Ed. Wiley, 2019.

[10] J. Ko, D. J. Klein, D. Fox, and D. Haehnel, "Gaussian processes and reinforcement learning for identification and control of an autonomous blimp," in *Proceedings 2007 IEEE ICRA*, 2007, pp. 742–747.

[11] A. Rottmann, C. Plagemann, P. Hilgers, and W. Burgard, "Autonomous blimp control using model-free reinforcement learning in a continuous state and action space," in *2007 IEEE/RSJ IROS*, Oct 2007, pp. 1895–1900.

[12] M. Clouâtré, M. Thitsa, and M. F. C. Join, "A robust but easily implementable remote control for quadrotors: Experimental acrobatic flight tests," *arXiv preprint arXiv:2008.00681*, 2020.

[13] S. Moe, A. M. Rustad, and K. G. Hanssen, "Machine learning in control systems: An overview of the state of the art," in *Artificial Intelligence XXXV*, M. Bramer and M. Petridis, Eds. Cham: Springer International Publishing, 2018, pp. 250–265.

[14] A. G. B. Richard S. Sutton, *Reinforcement Learning: An Introduction*, 2nd ed., F. Bach, Ed. The MIT Press, 2018.

[15] F. Lewis L., V. L. Draguna, and S. L. Vassilis, *Optimal Control*, 3rd ed., J. W. . Sons, Ed. John Wiley & Sons, 2012.

[16] S. B. V. Gomes and J. G. Ramos, "Airship dynamic modeling for autonomous operation," in *Proceedings. 1998 IEEE ICRA*, vol. 4, 1998, pp. 3462–3467 vol.4.

[17] J. Azinheira, P. Rives, J. Carvalho, G. Silveira, E. de Paiva, and S. Bueno, "Visual servo control for the hovering of all outdoor robotic airship," in *Proceedings 2002 IEEE ICRA*, vol. 3, May 2002, pp. 2787–2792 vol.3.

[18] J. Rao, Z. Gong, J. Luo, and S. Xie, "A flight control and navigation system of a small size unmanned airship," in *IEEE ICMA, 2005*, vol. 3, 2005, pp. 1491–1496 Vol. 3.

[19] Y. Yang, J. Wu, and W. ZhengWei, "Adaptive fuzzy sliding mode control for robotic airship with model uncertainty and external disturbance," *Journal of Systems Engineering and Electronics*, vol. 23, no. 2, pp. 250–255, 2012.

[20] Y. Yang, J. Wu, and W. Zheng, "Positioning control for an autonomous airship," *Journal of Aircraft*, vol. 53, pp. 1–9, 05 2016.

[21] S. Yu, G. Xu, K. Zhong, S. Ye, and W. Zhu, "Direct-adaptive fuzzy predictive control for path following of stratospheric airship," in *2017 29th CCDC*, May 2017, pp. 5658–5664.

[22] S. Liu, Y. Sang, and H. Jin, "Robust model predictive control for stratospheric airships using lpv design," *Control Engineering Practice*, vol. 81, pp. 231–243, dec 2018.

[23] A. Rottmann and W. Burgard, "Adaptive autonomous control using online value iteration with gaussian processes," in *2009 IEEE ICRA*, May 2009, pp. 2106–2111.

[24] A. Alsayed and E. Lanteigne, "Experimental pitch control of an unmanned airship with sliding ballast," in *2017 ICUAS*, 2017, pp. 1640–1646.

[25] M. Puheim and L. Madarasz, "Normalization of inputs and outputs of neural network based robotic arm controller in role of inverse kinematic model," jan 2014.

[26] J. C. Yeager, "Implementation and testing of turbulence models for the f18-harv simulation," NASA, Tech. Rep., 1998.

[27] A. R. Perry, "The flightgear flight simulator," in *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, ser. ATEC '04. USA: USENIX Association, 2004, p. 31.

[28] G. T. Navajas, "Modeling and pitch control of a re-configurable unmanned airship," mathesis, University of Ottawa, 2021.

[29] A. Elfes, S. Bueno, J. Ramos, E. Paiva, M. Bergerman, J. Carvalho, S. Maeta, L. Mirisola, B. Faria, and J. Azinheira, "Modelling, control and perception for an autonomous robotic airship," 01 2002, pp. 216–244.

[30] J. Xu, N. Lin, and R. Chi, "Improved high-order model free adaptive control," in *2021 IEEE DDCLS*, 2021, pp. 704–708.