2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

Safe residual reinforcement learning for helicopter aerial refueling

Damsara Jayarathne[1] Santiago Paternain[2] and Sandipan Mishra[1]

Abstract— **Autonomous helicopter aerial refueling is a challenging problem because of the complex aerodynamic interactions between the helicopter, the tanker and the refueling hose-drogue system. Methodologies solely relying on model-based control approaches are unable to directly address the aerodynamic interactions, whereas pure data-driven methods such as reinforcement learning (RL) often do not provide safety guarantees. Therefore, in this paper, we propose a novel residual RL control methodology that works in conjunction with a model-based outer-loop position controller. Further, we incorporate a *safe* RL algorithm that assures probabilistic safety guarantees by imposing appropriate constraints. This algorithm leverages the primal-dual formulation of a constrained optimal control problem to solve a sequence of RL problems that ultimately guarantees a probabilistic safety assurance requirement. The RL agent is trained in a simulation platform that consists of a reduced-order helicopter model and a state-dependent control mixer that appropriately delegates the control authority between the outer-loop controller and the RL controller. Once trained, the RL agent is deployed on a physics-based high-fidelity helicopter model without additional parameter tuning. These high-fidelity simulations reveal that the application of the proposed methodology yields a mean 2-norm error of 0.25m at the time of docking, which outperforms a purely model-based controller by 24%.**

I. Introduction

Helicopter Air-to-Air Refueling (HAAR) is the process of refueling a helicopter in-air using a fixed-wing tanker. HAAR is considered a particularly challenging flight maneuver because of (1) the limited time to dock with the drogue, (2) strict safety constraints, (3) complex interactions between the tanker-air wake-helicopter during docking maneuver and (4) the unpredictable nature of the drogue motion. Since HAAR requires a substantial pilot workload, there is a need to develop pilot-assisted or autonomous control strategies that improve performance and safety during refueling.

The standard fully autonomous control architecture for helicopters consists of an inner-loop controller for regulating attitude and altitude, with an outer-loop control for regulating the so-called zero dynamics of lateral/ longitudinal position [1]. The trajectories for these are generated either using heuristic methods (e.g. tau guidance [2]) or optimization-based trajectory generation [3]. Expanding on these ideas, complex maneuvers such as formation-flying [4] and cooperative slung-load carrying [5] have been demonstrated with the aid of advanced control schemes. More recently, control methodologies have been developed for *contact-based* maneuvers such as landing [1], [6]. In spite of these developments in control strategies toward enabling autonomous missions for helicopters, to the best of our knowledge, autonomous HAAR remains an unexplored problem.

Realizing autonomy for complex safety-critical maneuvers (such as HAAR, landing, etc.) requires careful trajectory planning, high-quality sensing and estimation, along with sophisticated feedback control design strategies. Consequently, there has been increasing interest in applying model predictive control (MPC) for helicopter control problems [6], [7], which necessitates accurate control-oriented models of the physical phenomena. For the multi-body aerodynamic interactions between the tanker, receiver (helicopter) and refueling system, generating appropriate control-oriented models is challenging. On the other hand, recently there have been advances in *simulation models* for capturing the aerodynamic interactions during aerial refueling [8]. These models, while not suitable for model-based controller design, can be used to design data-driven control strategies such as reinforcement learning (RL) for HAAR. Evaluating this potential is the key motivation for this study.

RL-based controllers have been successfully used in prior literature for controlling unmanned aerial vehicles such as quad-rotors and helicopters. For example, a learning algorithm that minimizes computational time during training was proposed for quad-rotors [9]. Building on this work, [10] has shown that RL algorithms can be trained to fly quad-rotors in unpredictable environments. The applicability of RL algorithms for autonomous helicopters has been explored in [11]. However, one of the key drawbacks in standard RL control is that safety guarantees cannot be enforced once the RL agent is incorporated into the loop [12].

However, in contact-critical systems that use RL, it is essential to guarantee safety irrespective of the operating conditions and the training methodology. One method of encouraging safe behavior of the RL agent is by incorporating explicit safety constraints, which require that certain safety criteria are met with a minimum guaranteed probability. Such constraints can be in the form of lower bounds on the value functions or additional safety-related value functions [13]. A safe RL strategy that exploits the well-known primal-dual algorithm is developed in [14], and proven to be effective in systems with multiple safety constraints. Finally, combining RL with model-based control methods can often guarantee a minimum baseline performance [15].

Motivated by the need for combining model-based control approaches and meeting safety requirements in HAAR, this paper presents a novel residual reinforcement learning methodology that combines the safe-RL scheme in [14] with

[1]Damsara Jayarathne and Sandipan Mishra are with the department of Mechanical, Aerospace and Nuclear Engineering of Rensselaer Polytechnic Institute, Troy, NY. [2]Santiago Paternain is with the Department of Electrical, Computer, and Systems Engineering of Rensselaer Polytechnic Institute, Troy, NY. {jayarj2,paters,mishs2}@rpi.edu

978-1-6654-7633-1/23/$31.00 ©2023 IEEE

a model-based baseline controller [1]. The nominal docking trajectory is planned prior to the execution of the maneuver based on the predicted drogue motion. The RL agent is then designed to adjust this nominally planned trajectory through corrections based on the current measurements of the drogue and the helicopter states. Next, we propose a safe-RL strategy that guarantees safety while executing path following/docking maneuvers. To reduce training time, the RL agent is trained on a reduced-order model. Although training the agent on the reduced-order model and deploying it on a full-scale helicopter simulation results in some degradation in performance, the training effort is reduced significantly.

II. Helicopter Air-to-Air Refueling

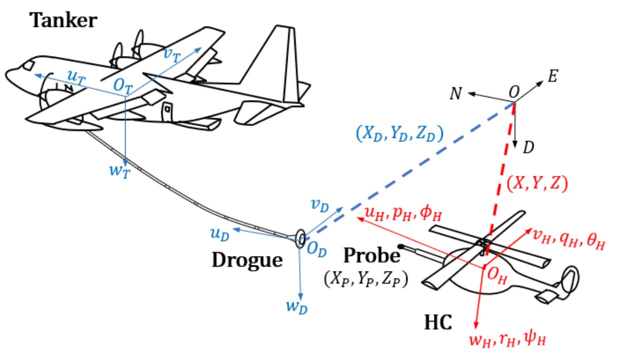


Fig. 1: Schematic of HAAR. O, O_H, O_D correspond to the North-East-Down, helicopter and drogue coordinate frames, respectively. The velocity components, the rates of rotation and the Euler angles in the helicopter coordinate frame are (u_H,v_H,w_H), (p_H,q_H,r_H) and (ϕ,θ,ψ), respectively. The velocity of the drogue is u_D,v_D,w_D. (X,Y,Z), (X_P,Y_P,Z_P) and (X_D,Y_D,Z_D) are the positions of the helicopter, the probe and the drogue, respectively.

A schematic illustrating the HAAR problem is shown in Fig. 1. Let t_0 (nominally set to 0) be the time at the initiation of the maneuver, when the helicopter, the probe and the drogue are at states $\mathbf{x}_{f0}$, $\mathbf{x}_{p0}$ and $\mathbf{x}_{D0}$. The goal is to guide the helicopter autonomously such that at the time of docking T, the helicopter probe docks with the drogue. The (time-varying) drogue state is described by $\mathbf{x}_D(t)$. Furthermore, to guarantee safety, the helicopter states must remain within a safe set S during the maneuver, i.e.

$$\begin{array}{rl} \mathbf{x}_f(t_0) = \mathbf{x}_{f0} & \text{Initial helicopter state} \\ \mathbf{x}_p(t) = \mathbf{x}_f(t) + G(\mathbf{x}_f(t)) & \text{Probe state trajectory} \\ ||\mathbf{x}_D(T) - \mathbf{x}_p(T)|| \leq \varepsilon & \text{Docking criterion} \\ \mathbf{x}_f(t) \in S \quad \forall t \in [t_0, T] & \text{Safety constraint} \end{array} \tag{1}$$

where t, $\mathbf{x}_f$, $\mathbf{x}_D$ and $\mathbf{x}_p$ are time, fuselage state, drogue state and probe state, respectively. G maps the relative state of the probe with respect to the fuselage state.

The helicopter dynamics are given generically by $\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t),\mathbf{u}(t),\mathbf{d}(t))$, where $\mathbf{d}(t)$ is the disturbance caused by the aerodynamic interactions and $\mathbf{x}_f(t)$ is a sub-state of $\mathbf{x}(t)$. The drogue is nominally moving at the speed of the tanker but is influenced by the local aerodynamics of the tanker, receiver, and atmosphere, i.e., $\mathbf{x}_D(t) = \mathbf{x}_D(t_0) + V_T \cdot (t - t_0) + \delta_a$ where V_T and δ_a correspond to the velocity vector of the tanker and the variation due to local aerodynamics, respectively.

Next, we present a brief description of the helicopter dynamics, the drogue kinematics, the trajectory generator and the helicopter control architecture.

III. Mathematical Modeling and Baseline Control Design

A. Helicopter dynamic model

In this paper, we use a UH-60A Black Hawk model developed by [16]. The dynamics, in general, are given by $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u})$ and $\mathbf{y} = \mathbf{g}(\mathbf{x},\mathbf{u})$ where $\mathbf{y}$ is the measurement used in the controller (a subset of states of rigid body dynamics). The state vector $\mathbf{x}$ is defined by $\mathbf{x} = \left[\mathbf{x}_f^T, \mathbf{x}_r^T, \mathbf{x}_t^T, \mathbf{x}_e^T\right]^T$ where $\mathbf{x}_f = \left[u,v,w,p,q,r,\phi,\theta,\psi,X,Y,Z\right]^T$, $\mathbf{x}_r = \left[\beta_0,\beta_{1s},\beta_{1c},\beta_d,\dot{\beta}_0,\dot{\beta}_{1s},\dot{\beta}_{1c},\dot{\beta}_d,\lambda_0,\lambda_{1s},\lambda_{1c}\right]^T$, $\mathbf{x}_t = \lambda_{0TR}$ and $\mathbf{x}_e = \left[\Omega,\chi_f,Q_e\right]^T$. $\mathbf{x}_f$ denotes 12 fuselage rigid body states, $\mathbf{x}_r$ denotes 8 blade flapping states and 3 inflow states of the main rotor, $\mathbf{x}_t$ denotes the tail rotor inflow state and $\mathbf{x}_e$ denotes 3 engine states. The control input is given by $\mathbf{u} = \left[u_{lat},u_{long},u_{col},u_{ped},u_{tht}\right]^T$, which consists of lateral, longitudinal, and collective joystick input to the main rotor, pedal input to the tail rotor and throttle input to the engine. The fuselage input $\mathbf{u}_f = \left[u_{lat},u_{long},u_{col},u_{ped}\right]^T$ is comprised of the input channels governing the fuselage motion.

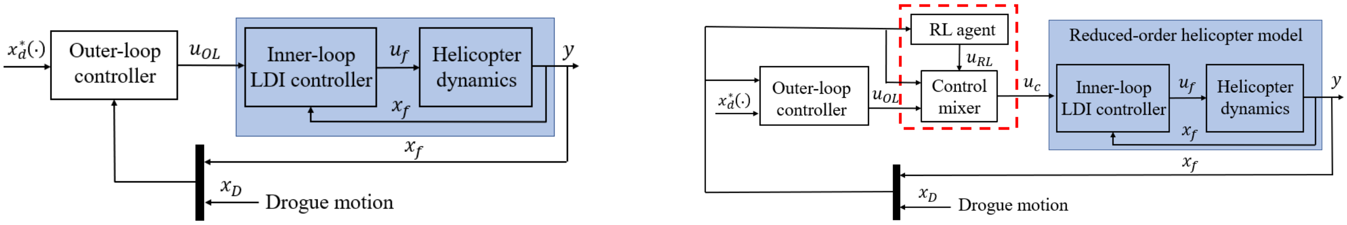
B. Drogue kinematic model

Physics-based drogue dynamical models have been developed to capture the motion of the drogue when subjected to steady wind and turbulence [17]. Furthermore, several models have been proposed in prior literature to capture the so-called *bow-wave* effect during refueling [8]. However, these models capture bow-wave effects from *fixed-wing aircraft*, which may not accurately model helicopter-drogue interaction. Therefore, as an exemplar study here we use a simplified kinematic model that mimics the bow-wave effect on the dynamics of the drogue.

For the purposes of this paper, the velocity component of the drogue in the North direction is the tanker velocity u_{T0} (110knots or 56.6m/s). The relative position between the probe and the drogue in the North direction $X_e(t)$ is defined as $X_e(t) = X_D(t) - X_p(t)$. Starting at (X_{D0},Y_{D0},Z_{D0}), the North-East-Down coordinates of the drogue are given by

$$\begin{array}{c} X_D(t) = X_{D0} + u_{T0} \cdot t \\ Y_D(t) = Y_{D0} + k_{y1} + k_{y2} \cdot tanh\left(k_{y3} \cdot X_e(t) + k_{y4}\right) \\ Z_D(t) = Z_{D0} + c_{z1} \cdot sin(c_{z2} \cdot t + c_{z3}) \end{array} \tag{2}$$

where t is time, Z_{D0} is the mean downward coordinate of the drogue. The coefficients c_{z1}, c_{z2}, c_{z3}, k_{y1}, k_{y2} and k_{y3} are chosen appropriately to capture the magnitude of the coupling and region of influence of the bow-wave. The motion of the drogue in the East direction mimics the bow-wave effect, while the motion in the Down direction mimics the effect of the tanker air wake.



(a) Baseline control architecture for full-scale helicopter [1].

(b) Proposed control architecture with the RL control scheme.

Fig. 2: Control architectures for autonomous HAAR.

C. Trajectory generation

Before the execution of the maneuver, a time-optimal path planner solves a time-optimal optimization problem given by

$$\begin{aligned}\{\mathbf{x}_d^*(\cdot),\mathbf{u}_f^*(\cdot),T^*\} = \quad & \underset{T,\mathbf{x}_f,u_f}{\arg\min}\int_{t_0}^{T} 1\; dt \\ s.t \quad & \dot{\mathbf{x}}_f(t) = F(\mathbf{x}_f(t),\mathbf{u}_f(t)), \\ & \mathbf{x}_p(t) = \mathbf{x}_f(t) + G(\mathbf{x}_f(t)), \\ & ||\hat{\mathbf{x}}_D(T) - \mathbf{x}_P(T)|| \leq \varepsilon, \\ & u_{f,min} \leq \mathbf{u}_f(t) \leq u_{f,max}, \\ & x_{f,min} \leq \mathbf{x}_f(t) \leq x_{f,max}\end{aligned} \tag{3}$$

where $\hat{\mathbf{x}}_D$, t and T correspond to the estimated drogue motion, time and final time, respectively. This generated docking trajectory $\mathbf{x}_d^*(\cdot)$ provides the nominal reference commands for the outer-loop controller (Fig. 2a).

D. Helicopter control architecture

Fig. 2a shows the baseline control architecture of the helicopter, consisting of an outer-loop and an inner-loop controller. We refer the reader to [1] for details of this control architecture. The inner-loop consists of the well-known (scheduled) linear dynamic inversion (LDI) controller, that takes command inputs $u_{OL} = [\phi_c, \theta_c, \dot{Z}_c, \dot{\psi}_c]^T$ from the outer-loop controller. ϕ_c, θ_c, $\dot{Z}_c$, and $\dot{\psi}_c$ denote the commanded roll angle and pitch angle, the vertical velocity and the rate of yaw angle, respectively. The outer-loop commands u_{OL} are generated by the outer-loop controller, which is based on dynamic inversion of the longitudinal and lateral position dynamics (assuming sufficiently fast inner loop stabilization) with the reference trajectory $\mathbf{x}_d^*(\cdot)$. The outer-loop controller is designed based only on the dynamics related to the horizontal motion given by

$$\begin{aligned}\ddot{X} &= -g\left(\tan(\theta-\theta_{trim})\cos\psi + \frac{\tan(\phi-\phi_{trim})}{\cos(\theta-\theta_{trim})}\sin\psi\right) \\ \ddot{Y} &= -g\left(\tan(\theta-\theta_{trim})\sin\psi - \frac{\tan(\phi-\phi_{trim})}{\cos(\theta-\theta_{trim})}\cos\psi\right)\end{aligned} \tag{4}$$

where θ, ϕ and ψ correspond to the Euler angles (pitch, roll and yaw) of the helicopter. θ_{trim} and ϕ_{trim} refer to the trim attitudes. Note that the control commands to velocity in the Down direction $\dot{Z}_c$ and yaw rate $\dot{\psi}_c$ are supplied from the optimal trajectory described previously (Section III-C).

IV. Reinforcement Learning Methodology

The motion of the drogue is unpredictable because of the complex aerodynamic interactions between the drogue, the helicopter and the tanker. Furthermore, the drogue and aerodynamic interaction models developed in prior literature [8], though useful for simulation, are unsuitable for model-based control design. Therefore, in this work, we propose an RL-based scheme to enable the controller to learn to dock the probe on the drogue.

One approach for designing the RL agent is to grant complete control to the agent, without the need for any model-based control augmentation. However, this may result in a long training time, and further pure RL controllers do not provide stability guarantees or baseline performance. Therefore, we propose a residual RL framework that combines the model-based controller (Section IV-A) with an RL agent as illustrated in Fig. 2b. The RL agent modifies the control command generated by the outer-loop controller u_{OL} by adding corrections in the form of u_{RL}. The control mixer governs the level of control authority granted to each control law (RL and baseline) by following a state-dependent control mixing strategy. The resulting control command u_c that is obtained after mixing is sent to the inner-loop controller for tracking. Since general RL algorithms do not guarantee safe operation, they are not suitable for this application. Therefore, we employ a safe-RL framework to assure safety.

Further, considering the time to train, the complexity of the model and the fact that we cannot avoid crashing the full-scale helicopter model during training, we employ a reduced-order helicopter model (as depicted in the blue box in Fig. 2b) to train the RL agent. We note here that the closed-loop stability of the proposed control architecture is guaranteed by limiting the maximum allowed corrections generated by the RL agent, i.e, $||u_{RL,i}|| < M_{RL}^i$, where i is the control channel and M_{RL}^i is the corresponding bound. The inner and outer loop controllers are designed with guaranteed stability nominally, i.e., without the RL agent [1].

A. Residual reinforcement learning

RL is a data-driven framework for the Markov decision process (MDP) defined by the tuple $(\mathbb{S},\mathbb{A},\mathbb{P}(s_{t+1}|s_t,u_{RL,t}),r(s_t,u_{RL,t}))$ where $\mathbb{S}$ and $\mathbb{A}$ correspond to the set of all possible states and actions, respectively. $\mathbb{P}(s_{t+1}|s_t,u_{RL,t})$ is the transition probability of state at time $t+1$ given current state $s_t \in \mathbb{S}$ and action $u_{RL,t} \in \mathbb{A}$. The RL agent draws an action $u_{RL} \in \mathbb{A}$ from a conditional distribution (called the policy $\pi_\mu(u_{RL}|s)$) when in s_t. This results in a transition to state $s_{t+1} \in \mathbb{S}$ and the agent receives a reward $r(s_t, u_{RL,t})$. The policy is parameterized by a vector $\mu \in \mathbb{R}^d$, where d is the number of parameters. We can then define a stochastic optimal control problem

$$\mu^* = \arg\max_{\mu \in \mathbb{R}^d} V(\mu) = \arg\max_{\mu \in \mathbb{R}^d} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{\,t} r(s_t, u_{RL,t})\right] \quad (5)$$

where $\gamma \in [0,1)$ is the discount factor and $V(\mu)$ is the value function.

In standard RL, the actions of the agent directly correspond to the control input of the plant, whereas in *residual* RL, the control input to the plant is the superposition of the action of the agent and an existing baseline control law [18]. This proposed control architecture is illustrated in Fig. 2b. The RL controller shown in the red dashed box is combined with the baseline controller described in Section III-D. Given the outer-loop control signal u_{OL} and the action of the RL agent u_{RL}, the control mixer generates the control signal u_c that is sent to the inner-loop LDI controller.

As illustrated in Fig. 2b, the observations of the system s from the perspective of the RL agent are given by $s = [X_e, \dot{X}_e, Y_e, \dot{Y}_e, Z_e, \dot{Z}_e, \psi_e, \dot{\psi}_e]^T$ where X_e, Y_e, Z_e correspond to the relative position in North, East, and Down directions and ψ_e corresponds to the relative yaw angle between the drogue and the probe in the North-East-Down coordinate frame. The actions executed by the RL agent are defined by $u_{RL} = [\Delta\phi, \Delta\theta, \Delta\dot{Z}, \Delta\psi]^T$. The reward function is given by

$$r = -(u_{RL}^T A u_{RL})^{1/2} - (e_s^T B e_s)^{1/2} + M_1 \mathbb{I}\Big((e_s^T B e_s)^{1/2} \leq \sigma_1\Big) + M_2 \mathbb{I}\Big((e_s^T B e_s)^{1/2} \leq \sigma_2\Big) + M_3 \mathbb{I}\Big((e_s^T \; B e_s)^{1/2} \leq \sigma_3\Big) \quad (6)$$

where $e_s = [X_e, Y_e, Z_e, \psi_e]^T$, A and B are positive definite diagonal matrices. This reward function is designed to minimize the error between the probe and the drogue with minimal control effort while motivating the agent to improve further based on the criterion defined by $\sigma_{(\cdot)}$. M_1, M_2, M_3 create a cascaded set of rewards that provide positive rewards when the squared relative distance between the probe and the drogue is within bounds defined by σ_i, where $\sigma_1 < \sigma_2 < \sigma_3$.

B. Safe reinforcement learning

Because of the unconstrained nature of the MDP, RL-based methods are often not well-suited for risky tasks [19]. This issue can be alleviated by defining a constrained optimal control problem

$$\mu^\star = \arg\max_{\mu \in \mathbb{R}^d} \quad \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{\,t} r(s_t, u_{RL,t}) \mid \pi_\mu\right], \quad (7)$$
$$\text{s.t.} \quad U_i(\mu) \geq c_i, \text{ for all } i = 1, \ldots, m$$

where $U_i(\mu) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^{\,t} \mathbf{1}(s_t \in S_i \mid \pi_\mu)\right]$. S is the safety set described in (1) and m is the number of constraints. The constants c_i are slack variables defined as $c_i = (1 - \delta_i \gamma^{T_i}(1-\gamma))/(1-\gamma)$ where T_i corresponds to the time window expressed in time steps, for which the safety criterion is required. δ_i is a probability described by

$$\mathbb{P}(\cap_{t=0}^{T_i}\{s_t \in S_i\} \mid \pi_\mu) \geq 1 - \delta_i, \quad (8)$$

which means that the trajectories generated by the policy π_μ will remain in the safe set S_i given in (1) with high probability $1 - \delta_i$ [14]. In (7), we wish to find a policy parameter μ^* such that the discounted cumulative probability of state s_t being in the safety set S is greater than a specified value c_i. Choosing c_i in this way guarantees that the inequality constraint given in (8) is satisfied.

A general approach for solving constrained problems (e.g., of the form described in (7)) is by formulating a dual relaxation [20]. We first construct the Lagrangian associated with problem (7)

$$\mathbb{L}(\mu, \lambda) = V(\mu) + \sum_{i=1}^{m} \lambda_i (U_i(\mu) - c_i) \quad (9)$$

where $\lambda_i \in \mathbb{R}_+^m$ is the multiplier associated with the i^{th} constraint. Then, the dual function is given by

$$d(\lambda) = \max_{\mu \in \mathbb{R}^d} \mathbb{L}(\mu, \lambda) \quad (10)$$

that leads to the dual problem

$$D^\star = \arg\min_{\lambda \in \mathbb{R}_+^m} d(\lambda). \quad (11)$$

The maximization in (10) is equivalent to solving the unconstrained RL problem (5) with a reward defined by

$$r_\lambda(s_t, u_{RL,t}) = r(s_t, u_{RL,t}) + \sum_{i=1}^{m} \lambda_i (1(s_t \in S) - c_i(1-\gamma)). \quad (12)$$

If we consider the expectation of the discounted cumulative reward of (12), we observe that it is equivalent to the Lagrangian given in (9). Therefore, instead of maximizing the Lagrangian, we can solve the RL problem that maximizes the expectation of the cumulative reward defined in (12) to find the policy parameter μ^*.

Considering the ease of implementation (fewer hyperparameters to tune) and ability to incorporate continuous actions, in this study, we employ the deep deterministic policy gradient (DDPG) algorithm [21], which updates the primal variable μ to solve the unconstrained RL problem in (10). Furthermore, as (11) is a convex function, one can use gradient descent to update the dual variable λ using $\lambda^{k+1} = \lambda^k - \eta_\lambda(\hat{U}(\mu_k) - c)$ where k is the current step, η_λ is the gradient descent step size and c is the slack variable. $\hat{U}(\mu_k)$ for finite time is given by $\hat{U}_i(\mu_k) = \sum_{t=0}^{T_f} 1(s_t \in S_i)$ where T_f is the final time. As shown in [14], the duality gap between (7) and (11) is almost zero, which guarantees the possibility of finding an optimal solution for problem (7) by solving the problem (10) in the dual domain. Algorithm 1 outlines the primal-dual algorithm that is used to train safe-RL policies [14] where η_μ, μ^0, λ^0, N correspond to the gradient ascent step size, the initial policy parameter, the initial dual multiplier value and the maximum number of iterations, respectively.

Algorithm 1 Stochastic primal-dual for safe policies

```
Require: $\mu^0, \lambda^0, T_i, \eta_\mu, \eta_\lambda, \gamma, \delta_i$
while $k \leq N$ do
    Simulate a trajectory with the policy $\pi_{\mu_k}(s)$
    Calculate the reward as in (12)
    Calculate the dual gradient $\hat{U}(\mu_k) - c$
    Update primal variable $\mu^{k+1}$ using DDPG
    Update dual variable $\lambda^{k+1} = \lambda^k - \eta_\lambda(\hat{U}(\mu_k) - c)$
end while
```

Control mixer: Since the proposed control architecture blends control signals from the RL framework and the model-based control strategy, it is necessary to combine them appropriately in order to extract the best performance. Therefore, we employ a state-dependent control mixing strategy $u_c = \beta u_{RL} + (1-\beta)u_{OL}$ and $\beta = \alpha_1(\alpha_2 + \alpha_3 \tanh(\alpha_4 X_e(t) + \alpha_5) + \alpha_6$. β satisfies the inequality, $0.1 \leq \beta \leq 0.9$ and the constants, α_i for $i = 1, ..., 6$ are appropriately chosen. The control mixer assigns a lower β value in the initial stage of the maneuver, granting more control authority to the outer-loop controller. However, the effect of bow-wave intensifies when the probe gets closer to the drogue. As a result, in the final stage of the operation, β is gradually increased, granting more control authority to the RL controller compared to the outer-loop controller.

C. Training in the reduced-order model

In order to train the RL agent, instead of using the full-scale model, we propose employing a reduced-order helicopter model to expedite training and allow some violation of the safety constraints during training. Note that this reduced order model is an approximation of dynamics indicated by the blue box in Fig. 2b, by assuming instantaneous attitude and altitude dynamics (which is reasonable given the time-scale separation between the outer-loop dynamics of the helicopter and the inner loop stabilization). The input to this reduced-order model is $u_{OL} = [\phi_c, \theta_c, \dot{Z}_c, \dot{\psi}_c]^T$ while the state is described by $x_s = [X, \dot{X}, Y, \dot{Y}, Z, \psi]^T$. With the above assumptions, the reduced-order model is described by

$$\begin{aligned} \ddot{X} &= -g\left(\tan(\theta_c - \theta_{trim})\cos\psi + \frac{\tan(\phi_c - \phi_{trim})}{\cos(\theta_c - \theta_{trim})}\sin\psi\right) \\ \ddot{Y} &= -g\left(\tan(\theta_c - \theta_{trim})\sin\psi - \frac{\tan(\phi_c - \phi_{trim})}{\cos(\theta_c - \theta_{trim})}\cos\psi\right) \\ \dot{Z} &= \dot{Z}_c \qquad \dot{\psi} = \dot{\psi}_c \\ \phi_{trim} &= g_\phi(\sqrt{\dot{X}^2 + \dot{Y}^2}) \qquad \theta_{trim} = g_\theta(\sqrt{\dot{X}^2 + \dot{Y}^2}) \end{aligned} \tag{13}$$

In this study, the training was performed on a single computer with specifications: Intel core i7 2.3GHz with 16GB RAM, employing the DDPG algorithm. The details of the parameters used in the simulation are shown in Table I. The sample time is 0.1s (typical for such control loops). The parameters of the agent, the actor and the critic network are determined heuristically. *Tanh* is the activation layer for both actor and critic networks. The diagonal elements of matrix A in the reward function (6) are chosen so that the minimization of the input is treated equally. The diagonal elements of the B matrix in the reward function (6) are chosen so that elements in e_s are appropriately scaled. The values of $\sigma_{(.)}$ generate a cascaded reward scheme with $M_{(.)}$ being scaling values. The parameters of the control mixer guarantee a smooth transition between the initial and final stages of the refueling process. η_λ, η_μ and λ_0 for the *Safe-RL* are chosen heuristically. T_i and T_f are chosen based on the simulation length of the maneuver, whereas x_{safe} is selected based on the fact that the probe is not allowed to overshoot the drogue. δ_i is set to 0.01 so that safety is maintained with a probability of 0.99.

TABLE I: RL training details

Group	Parameter	Value
Agent	Actor, critic learning rate Discount factor	10^{-3} 0.98
Actor network	# of hidden layers # of neurons in each layer	3 128
Critic network	# of hidden layers # of neurons in each layer	2 64
Reward	$(A_{11}, A_{22}, A_{33}, A_{44})$ $(B_{11}, B_{22}, B_{33}, B_{44})$ $(\sigma_1, \sigma_2, \sigma_3)$, (M_1, M_2, M_3)	(10,10,10,10) (5,5,10,10^4) (0.1,0.25,0.5) (5,5,5)
Control mixer	$(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6)$	(0.8,0.5,0.5,-15,-40,0.1)
Safe-RL	(T_f, T_i) $(\eta_\lambda, \eta_\mu, \lambda^0, \delta_i)$	(7(s),70 steps) $(10^{-5}, 10^{-3}, 0, 0.01)$
Drogue motion	$(k_{y1}, k_{y2}, k_{y3}, k_{y4})$	(0.25,-0.25,-2.2,-5)
Control bounds	$(M_{RL}^1, M_{RL}^2, M_{RL}^3, M_{RL}^4)$	(0.25,0.09,0.025,0.1)

V. Simulation results

In this section, we demonstrate the application of the proposed control architecture in an aerial refueling operation. The study is carried out in two stages. In stage 1, we illustrate the performance of the residual RL methodology in which the agent is trained in the reduced-order model (Section IV-C) and directly deployed in the full-scale model (Section III-A). In stage 2, we show the results obtained from the safe-RL methodology in which the RL agent is trained in the reduced-order model and deployed in the full-scale model. A statistical analysis of the performance of the proposed methodology from 100 docking simulations is given in Table II. At the start of the maneuver, the initial positions of the probe and the drogue are given by $(0+\Delta, 0+\Delta, -1000+\Delta)m$ and (5, 0, -1000)m, respectively in the **North-East-Down coordinate frame** where $\Delta \in [-0.25, 0.25]m$ is randomly generated. Note that the probe position is offset by (3.96, 1.11, 1.75)m from the center of gravity (CG) of the helicopter in the **helicopter coordinate frame**.

A. Performance evaluation of residual RL

1) Reduced-order model simulations: The RL agent is trained in the reduced-order model in which the average reward approximately converged to 40 after 4000 iterations. Then, the obtained policy is implemented in the reduced-order model to evaluate the performance. The resulting relative position between the drogue and the probe for a single *sample trajectory* is given in Fig. 3. The mean 2-norm docking error for 100 simulation runs is found to be 0.015m with a standard deviation of 0.008m and a maximum error of 0.032m as shown in Table II. Next, the policy learned in

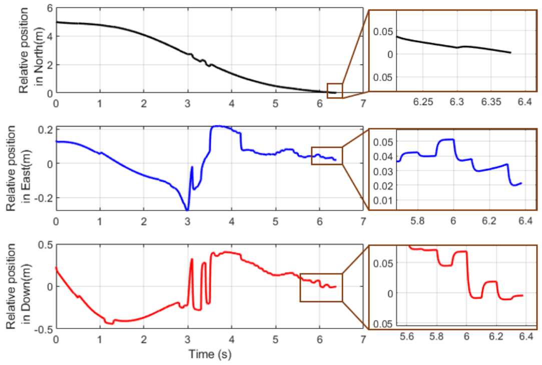


Fig. 3: Relative position between the drogue and the probe for a typical docking simulation run on the reduced-order helicopter model. Note that the scales of the graphs are different for each direction. The zoomed-in plots depict the docking error at the time of contact. (error 2-norm $0.021m$)

the reduced-order helicopter model is ported to the full-scale helicopter model to evaluate the performance.

TABLE II: The mean, the standard deviation and the maximum values of 2-norm error at the time of docking obtained through 100 simulations for each case described in Section V-A, V-B and model-based control in [1].

Case	Mean error(m)	Std. Dev.(m)	Max error(m)
RL on reduced-order model (Sec.V-A.1)	0.015	0.008	0.032
RL on full-scale model (Sec.V-A.2)	0.246	0.214	1.334
Model-based control [1]	0.324	0.117	0.551
Safe-RL on reduced-order model (Sec.V-B.1)	0.071	0.005	0.080
Safe-RL on full-scale model (Sec.V-B.2)	0.307	0.279	1.140

2) Full-scale model simulations: The policy learned in the reduced-order model is implemented in the full-scale model. The resulting relative position between the drogue and the probe for a single *sample trajectory* is given in Fig. 4. In comparison, the relative positions in the North directions share similar trends despite the discrepancies between the Down and the East direction as shown in Fig. 3 and 4. The key contributing factor to this discrepancy is the mismatch between the reduced-order model used for training and the full-scale helicopter model. Note that the time-to-contact is different in the two cases due to the behavioral difference between the two models. The mean 2-norm docking error for 100 simulation runs is found to be $0.246m$ with a standard deviation of $0.214m$ and maximum error of $1.334m$ as shown in Table II. It is evident that the proposed methodology outperforms a purely model-based controller by 24% in 2-norm mean error. Furthermore, there is a possibility that the drogue overshoots the probe since the motion of the probe in the North direction is not constrained. One method of preventing this is to impose a safety constraint on the position of the helicopter's center of gravity (CG).

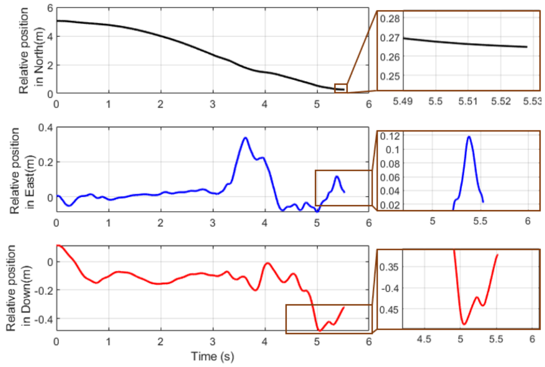


Fig. 4: Relative position between the drogue and the probe for a typical docking simulation run on the full-scale helicopter model. Note that the scales of the graphs are different for each direction. The zoomed-in plots show the docking error at the time of contact (error 2-norm $0.41m$).

B. Stage 2: Performance evaluation of safe residual RL

1) Reduced-order model simulations: In stage 2, we investigate the ability of the control algorithm to accommodate safety constraints. The position of the probe is dependent on the attitudes (ϕ, θ, ψ) of the helicopter since it is a static function of the helicopter state (1). One way to ensure a safe docking maneuver is by enforcing safety constraints on the coordinates of the CG of the helicopter. Therefore, we present a case in which we guide the CG of the helicopter towards a point in space that has similar motion characteristics as the drogue while imposing safety bounds on the CG of the helicopter. In order to enforce the safety constraint, we define the constraint $X_e(t) \geq x_{safe} \quad \forall t \leq T_f$ where x_{safe} is set to 0 and T_f is the maneuver time. The RL agent was trained in the reduced-order model employing the safe-RL algorithm described in Algorithm 1 where the average reward converged to 105 after 10000 iterations. Furthermore, λ_x converged to 0.135 which is used when simulating both the reduced-order and the full-scale helicopter models.

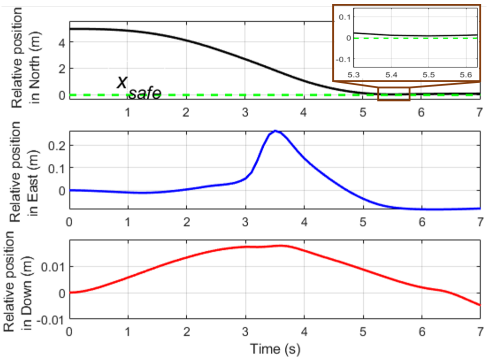


Fig. 5: Relative positional information of the trajectories obtained by training RL agent on the reduced-order model with safety constraint $X_e(t) \geq x_{safe}$ and deploying on the reduced-order model. The green dashed line represents the x_{safe}. The zoomed-in plot confirms that the safety constraint is not violated throughout the run.

In Fig. 5, we present the relative positional information of the reduced-order helicopter simulation in North, East and Down directions for a single *sample trajectory*. Clearly, the helicopter motion is contained within the safe region. Although the time taken to dock is less than 7 seconds, we plot the trajectory to emphasize that the safety constraint is met throughout the run. The mean docking error for 100 simulation runs is found to be $0.071m$ with a standard deviation of $0.05m$ as shown in Table II.

2) Full-scale model simulations: The RL agent trained in the reduced-order model is deployed on the full-scale helicopter model. As illustrated by Fig. 6, the full-scale helicopter model overshoots x_{safe} in the relative position in the North direction which violates the safety constraint. This is mainly due to the nonlinearities present in the full-scale helicopter model. Since the residual RL controller never experiences these nonlinearities during training, it fails to perform the safe maneuver with high precision.

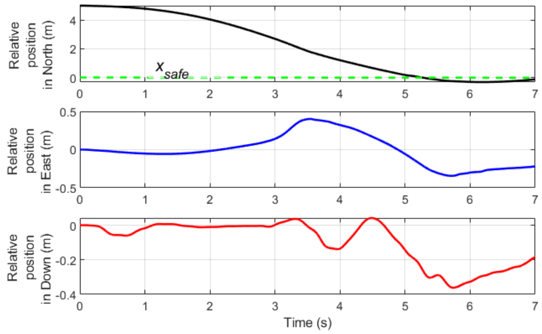


Fig. 6: Relative positional information of the trajectories obtained by training RL agent in the reduced-order model with safety constraint $X_e(t) \geq x_{safe}$ and deploying on the full-scale helicopter.

Discussion: Based on the studies above, we observe that the proposed methodology outperforms the model-based control method in 2-norm mean error by 24% in the residual RL method (Section V-A.2) and 5% in the safe-residual RL method (Section V-B.2) when deployed on the full-scale helicopter model. Furthermore, the residual RL controller meets the docking criterion in the reduced-order model where the 2-norm error at the time of docking is less than $0.1m$ (Table II). Moreover, we notice that the safety constraint is not violated when the proposed controller is executed in the reduced-order model which yields an accuracy of 100% in meeting the safety criterion. On the other hand, we observe that the proposed method *does not meet the safety criterion* or the *tight docking criterion* when deployed on the full-scale helicopter model, despite the improved performance over the model-based controller. This is primarily due to the fact that the reduced-order model does not accurately represent the dynamics of the full-scale model. As a result, during training on the reduced order model, the RL agent does not experience or learn to respond to the complex nonlinearities of the full-scale model. This issue may be mitigated by employing techniques such as *domain randomization* [22] and retraining on the full-scale model.

VI. Conclusions and Future Work

In this paper, we presented a novel residual RL control method that works in conjunction with an outer-loop position controller. By incorporating safety constraints in the residual RL controller, we demonstrated that it is capable of guaranteeing safety, which is essential in autonomous HAAR. The conducted simulations highlight that the proposed method outperforms sole model-based controllers. Since the RL agent does not experience nonlinearities in the full-scale helicopter during training, techniques such as *domain randomization* and retraining on the full-scale model can be utilized to improve the performance.

References

[1] D. Zhao, S. Mishra, and F. Gandhi, "A differential-flatness-based approach for autonomous helicopter shipboard landing," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 3, pp. 1557–1569, 2021.

[2] M. Voskuijl, G. Padfield, D. Walker, B. Manimala, and A. Gubbels, "Simulation of automatic helicopter deck landings using nature inspired flight control," *The Aeronautical Journal*, vol. 38, no. 12, pp. 25–34, Aug 2010.

[3] B. Hu and S. Mishra, "Time optimal trajectory generation for a quadrotor," in *American Control Conference, 2017.*, 2017, p. submitted.

[4] A. Karimoddini, H. Lin, B. M. Chen, and T. H. Lee, "Hybrid three-dimensional formation control for unmanned helicopters," *Automatica*, vol. 49, no. 2, pp. 424–433, 2013.

[5] J. Geng and J. W. Langelaan, "Cooperative transport of a slung load using load-leading control," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 7, pp. 1313–1331, 2020.

[6] T. D. Ngo and C. Sultan, "Model predictive control for helicopter shipboard operations in the ship airwakes," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 3, pp. 574–589, 2016.

[7] H. Chung and S. Sastry, "Autonomous helicopter formation using model predictive control," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2006, p. 6066.

[8] X. Dai, Z. Wei, and Q. Quan, "Modeling and simulation of bow wave effect in probe and drogue aerial refueling," *Chinese Journal of Aeronautics*, vol. 29, no. 2, pp. 448–461, 2016.

[9] J. Hwangbo, I. Sa, R. Siegwart, and M. Hutter, "Control of a quadrotor with reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2096–2103, 2017.

[10] C.-H. Pi, K.-C. Hu, S. Cheng, and I.-C. Wu, "Low-level autonomous control and tracking of quadrotor using reinforcement learning," *Control Engineering Practice*, vol. 95, p. 104222, 2020.

[11] H. Kim, M. Jordan, S. Sastry, and A. Ng, "Autonomous helicopter flight via reinforcement learning," *Advances in neural information processing systems*, vol. 16, 2003.

[12] C. Greatwood and A. G. Richards, "Reinforcement learning and model predictive control for robust embedded quadrotor guidance and control," *Autonomous Robots*, vol. 43, no. 7, pp. 1681–1693, 2019.

[13] C. Tessler, D. J. Mankowitz, and S. Mannor, "Reward constrained policy optimization," *arXiv preprint arXiv:1805.11074*, 2018.

[14] S. Paternain, M. Calvo-Fullana, L. F. Chamon, and A. Ribeiro, "Safe policies for reinforcement learning via primal-dual methods," *IEEE Transactions on Automatic Control*, 2022.

[15] T. Staessens, T. Lefebvre, and G. Crevecoeur, "Adaptive control of a mechatronic system using constrained residual reinforcement learning," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 10, pp. 10 447–10 456, 2022.

[16] J. Krishnamurthi and F. Gandhi, "Flight simulation and control of a helicopter undergoing rotor span morphing," *Journal of the American Helicopter Society*, vol. In Press, Aug 2017.

[17] K. Ro and J. W. Kamman, "Modeling and simulation of hose-paradrogue aerial refueling systems," *Journal of guidance, control, and dynamics*, vol. 33, no. 1, pp. 53–63, 2010.

[18] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, "Residual reinforcement learning for robot control," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6023–6029.

[19] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[20] D. Ding, K. Zhang, T. Basar, and M. Jovanovic, "Natural policy gradient primal-dual method for constrained markov decision processes," *Advances in Neural Information Processing Systems*, vol. 33, pp. 8378–8390, 2020.

[21] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[22] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE symposium series on computational intelligence (SSCI)*, pp. 737–744.