IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 December 2022, accepted 26 February 2023, date of publication 6 March 2023, date of current version 31 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3253481

RESEARCH ARTICLE

A Soft-Kill Reinforcement Learning Counter Unmanned Aerial System (C-UAS) With Accelerated Training

DOUGLAS LOPES DA SILVA, (Member, IEEE), RENATO MACHADO, (Senior Member, IEEE), OLYMPIO L. COUTINHO, (Member, IEEE), AND FELIX ANTREICH, (Senior Member, IEEE)
Electronic Warfare Laboratory (LAB-GE), Aeronautics Institute of Technology (ITA), São José dos Campos 12228-900, Brazil

Corresponding author: Douglas Lopes da Silva (douglas@ita.br)

This work was supported in part by the Brazilian Agencies Coordination for the Improvement of Higher Education Personnel (CAPES); in part by the Research Development Foundation (FUNDEP); in part by the São Paulo Research Foundation (FAPESP), Brazilian Institute of Data Science (BI0S), under Grant 2020/09838-0; and in part by the National Council for Scientific and Technological Development (CNPq), under Grant 312394/2021-7 PQ-2.

ABSTRACT In recent years, unmanned aerial vehicles (UAVs) have gained significant popularity and are used for many applications, from entertainment to surveillance and the modern battlefield. As regulation demands arose worldwide, controling and reacting to unauthorized flights of UAVs became a pressing issue. In this work, we present an algorithm to accelerate the training of a reinforcement learning drone agent for a counter unmanned aerial system (C-UAS). The main objective of this C-UAS is to guide an invading drone to a safe-killing zone (SZ) using a hunter quadrotor drone. The hunter quadrotor launches a spoofing, or meaconing, attack on the GNSS receiver of the invading drone. The proposed algorithms employ an abstraction of the C-UAS problem to accelerate the training step and enable training during the mission. Results for different SZ radii are discussed using a software-in-the-loop simulation for ground truth based on a detailed model of the UAV embedded system and flight dynamics, including error metrics and action time. We show that a 99% probability of successful target steering to the SZ can be achieved considering a SZ radius of 75 meters and a Q-table trained with the proposed accelerated training model.

INDEX TERMS C-UAS, drones, Q-learning, reinforcement learning, software-in-the-loop, UAV.

I. INTRODUCTION

In the last decade, unmanned aerial vehicles (UAV) received increasingly academic and commercial attention [1], [2]. While autonomous and self-driving cars are being developed and starting to appear in the market slowly, aerial vehicles have quickly dominated several areas of interest, from commercial to military purposes [3], [4], [5]. These vehicles can be used for obscure purposes and terrorism, i.e., as a vector of explosives and even for biological/chemical payloads [4], [5]. The reason for such advances can be traced to the increased battery capacity and weight reduction, powerful embedded controllers widely available to the public, and easy access to lighter brushless motors. Therefore, today a heavy payload such as a camera or a sensor can be carried at a low-cost [1].

Nowadays, large areas can be covered and remotely sensed. A task that a few years ago had to be performed by conventional fixed-wing aircraft and helicopters resulted in prohibitive costs and a long operation time [6]. These applications today are more and more performed by UAVs, such as inspection of power lines, search and rescue tasks, filming, railway and wind turbine inspections, agriculture, security and surveillance, and uses on the battlefield in modern warfare [7] to the delivery of products [8].

Furthermore, due to the easy access to UAVs by the general public, their low cost, and their easy operation, the use of UAVs in prohibited areas has been increasing. Consequently, the need to protect such areas against unauthorized access is growing. Areas that need to be protected or where UAVs

The associate editor coordinating the review of this manuscript and approving it for publication was Cong Pu.

are not authorized, for example, are: protest areas, stadiums, airports, water supply plants, atomic plants, and military zones [9]. The high cost of conventional methods to destroy such aircraft motivates research on protecting restricted areas from UAVs by other means.

In general, a counter unmanned aerial system (C-UAS) is composed of two steps: detection and mitigation [10]. The detection of drones is a very challenging problem, as due to their size and structure they are difficult to spot and locate compared to conventional aircraft. After the detection, the mitigation process usually requires a low reaction time and can be composed of: radio frequency (RF) and global navigation satellite systems (GNSS) jamming as well as spoofing, lasers, kinetic, or a combination of these elements [11]. Also, C-UAS platforms are flexible and more expeditious than static or mobile ground platforms [12].

Our proposed C-UAS differs from the current state-of-the-art solutions as it employs a ''drone countering drone'' approach, along with the so-called attacker hunter drone (AHD) being able to carry a GNSS spoofing payload. The proposed C-UAS reduces the reaction time as the AHD can fly closer to the so-called target invader drone (TID). A non-destructive or soft-kill method can be employed to catch the TID preserving vital forensic information on the target system. Most importantly, we present a C-UAS being able to adapt to different target behavior and missions using a reinforcement learning (RL) agent. The decision of how the AHD moves to automatically guide the TID to a pre-defined safe zone (SZ) is provided by an RL technique, specifically Q-learning. Due to their various capabilities and robustness, RL algorithms have received much attention in recent years [13]. RL algorithms for guidance and navigation became possible based on recent advances in theoretical fields of intelligent agents to provide guidance, as presented in [14], [15], and [16]. Problems of moving vehicles in space in a scenario where its dynamics are not fully known or are not implemented directly in the controller are addressed in [17], [18], and [19].

The C-UAS design presented in [20] was tested using a software-in-the-loop (SITL) implementation based on the Ardupilot platform. The SITL simulations are performed to emulate the complex operational scenarios and are used to train the Q-values in the Q-tables. The computational cost of the SITL simulation is high as we simulate the movement and sensors of two drones. Therefore, the time the training takes is in the order of minutes, resulting in only a few missions/scenarios that could be presented to the system in the training phase. Consequently, the high computational load of the SITL method should be bypassed using an alternative to derive the Q-values in the Q-tables of the system in the training.

This work presents an alternative method of training the C-UAS using an abstraction of the problem to reduce the training time and prepare the C-UAS for different mission scenarios. This proposed new training uses an error-tracking approach (ETA) to emulate the UAVs' movements and guidance. We call this model the simplified UAV movement model (SUMM). We use the SUMM to obtain the Q-tables faster, which is later loaded to the SITL simulation, which we use to simulate the complex mission scenarios of the two UAVs. Hence, in [20], the SITL simulation was used while training the C-UAS and evaluating its performance. In this work, we use the SUMM to accelerate the training and the SITL as ground truth for the performance evaluation of the SUMM.

The dynamic model of the SITL simulation used is the standard model for rotary-wing aircraft (Copter module) on the Ardupilot platform. The platform uses the SITL simulation as a test tool for the developed control algorithms, and in this work, it is used as a standard with which we compare the results of the proposed algorithm.

In [20], the proposed C-UAS also uses spoofing, and a much simpler RL, e.g., a reward function with fixed values for movement, and the initial Q-table was chosen to represent the heading of the SZ based on the TID's position. Also, the training and evaluation were achieved only using the SITL simulation. No acceleration or pretraining steps were included. In this work, we introduce different initial positions for both UAVs in contrast to [20] where the initial positions were in the origin of the coordinate system.

The main contribution of this work is a methodology to drastically accelerate training for a C-UAS which can soft-kill a UAV. This contribution is made through a simpler model describing the movement information of two drones. The essential information for the attacker's training is maintained. The attitude information of both aircraft is not used in the proposed algorithm and is, therefore, discarded. Furthermore, we have further developed the Q-learning algorithm and the dimension of the Q-table with respect to the algorithm presented in [20]. Hence, this new approach has the following benefits:

- Lower computational load while training, and therefore cheaper training;
- Different strategies can be tested in scenarios where the C-UAS could not guide the TID to the SZ so far;
- As the training time is reduced to the order of seconds, we can retrain the system while hunting, using the onboard CPU resources of the AHD;

This article is organized as follows. In Section II related work and essential information related to the presented topic are discussed. In Section III we discuss GNSS spoofing and RL to guide the AHD and the SITL simulation, while Section IV presents the SUMM simulation details of the proposed ETA method. In Section V we present the training process along with its parameters. In Section VI we present and discuss the results of the evaluation step of the SITL simulations of SUMM trained Q-tables together with a Monte Carlo (MC) simulation for three scenarios of different radius of the SZ. Finally, in Section VII we discuss our conclusions and in Section VIII we present future work improvements.

II. RELATED WORK

As the UAV system complexity increases with the capabilities of such vehicles, they become more dependent on their onboard instruments, such as the inertial navigation system (INS) and its sensors as accelerometers, gyroscope, and barometer [21], [22]. Other sensors such as altimeters, light detection and ranging (LiDAR) sensors, power measurement sensors of the RF link, and GNSS receivers [23] are also used. GNSS receivers are vital to correct the drift error of the INS using fusion algorithms and to derive robust and highly reliable position, navigation, and timing (PNT).

The systems and subsystems of UAVs are susceptible to several attacks and malicious actions, such as electronic warfare measures, i.e., jamming and spoofing [24], [25], [26], and cyberattacks [24], [27]. The embedded systems of a UAV rely on the information provided by the sensors attached to the main control board. This control board is responsible for capturing the sensor data, process, and forwarding the control signals to the actuators. Therefore, to infiltrate the system, one of its sensors has to be attacked to send false information to the embedded hardware. The inertial measurement units (IMUs) responsible for measuring the angular velocities and the linear acceleration in the three-axis require a high level of access to the embedded platform to change the reading of the INS. Generally, to alter the measured quantities by the IMU, a cyberattack on the embedded system is required [28], [29]. The RF - control and telemetry - links are the subsystems that are predominantly attacked [25], [27], [30] with simple jamming techniques. In contrast, most commercial systems use known protocols and network tools similar to the ones found on modern computers. The operational system of these embedded systems usually is a modified Linux system.

As an alternative to launching cyberattacks on the telemetry links, the GNSS receiver, present in almost any commercial UAV that can cover a large area, are susceptible to RF link attacks using electronic warfare measures, such as jamming and spoofing [25], [30], [31]. These attacks are powerful since the control loop is affected, calculating the wrong position, and even for remotely controlled flights, a pilot will have great difficulties to control the aircraft even when the telemetry and control links are not directly affected.

Detecting jamming and spoofing does not mean an autonomous drone, or a pilot controlling the drone, will quickly regain control of the aircraft. The controls may not respond correctly to the inputs, and onboard cameras sensing the environment may not be sufficient to correct the injected errors. Several techniques have already been presented to fuse terrain data to estimate aircraft position [29], [32] or to use visual cameras with optical-flow estimation [33], but such techniques require a very high computational load or additional sensors. The use of point clouds [34] and 3D target detection [35] along with position estimation from terrain data can automate flight steps [21], [22], [36], [37], but the navigation while under a GNSS repeater attacker is still a big challenge for UAVs and even for conventional aircraft.

III. SPOOFING MODULE AND REINFORCEMENT LEARNING (RL)

In this work, we consider two kinds of the positioning of a UAV: based on external observation and GNSS. External observation using a visual camera with object detection and optical flow tracking or a LiDAR sensor is considered for the AHD to estimate the position of the TID. We assume that both the AHD and the TID use GNSS to estimate their position internally. To guide the TID to the SZ, the AHD launches a spoofing repeater attack [38] on the TID's GNSS receiver. Such an attack is also called meaconing [38], [39]. The main purpose of this meaconing system is to superimpose the relayed signal to the signals from the GNSS satellites received by the TID. Thus, the GNSS satellite signals are received by the meaconing system on board the AHD, amplified, and then relayed (with or without modification) to the TID.

As the AHD can get closer to the TID at the beginning of the attack, the additional time delay that the spoofing signals experience when propagating from the AHD to the TID with respect to the direct GNSS satellite signals can be in the order of nanoseconds to zero. The situation when the AHD is flying over the TID is called the zero-delay meaconing attack, *i.e.*, the GNSS satellite and the meaconing signals are aligned [38]. When aligned, the propagation time delay difference between the AHD and the TID with respect to the direct GNSS satellite signals is close to zero, and the tracking loops of the TID's receiver can accumulate drift over time, accumulating small clock offsets. If then, additionally, the power of the meaconing signals is a bit stronger than the power of the direct satellite signals, the tracking loops of the TID will keep tracking the meaconing signals while the AHD can slowly move farther away from the TID's original true position. If the attack was successful, then the TID will calculate its position using the meaconing signals from the AHD and thus will calculate the same position as the AHD, but with a slight receiver clock offset, [20]. However, in case the zero-delay meaconing attack is not successful, the AHD also should be able to perform a jamming attack to deny tracking of the direct GNSS satellite signals by the TID's GNSS receiver and force it into re-acquisition. During the jamming the AHD can initiate the meaconing attack and thus ensure the TID's receiver tracking loops to lock on the meaconing signals.

Assuming a successful spoofing attack, in our simulations, the output coordinates of the GNSS receiver in the TID embedded system will be the coordinates of the AHD embedded system with added errors due to the hardware implementation of the repeater.

So far, autonomous target drones could be guided to the desired SZ by a pilot controlling the AHD but this task imposes a high cognitive load on the AHD pilot as the TID's mission is unknown. Therefore, we automate this task by using an RL Q-learning agent to guide the TID to the SZ in the shortest time possible, as it requires little to no knowledge of the system dynamics [4] described by the SITL model or the SUMM.

Considering the TID's position vector $w_T[n] = [w_{T_x}[n], w_{T_y}[n]]^{\mathrm{T}}$ and the SZ position vector $w_Z[n] = [w_{Z_x}[n], w_{Z_y}[n]]^{\mathrm{T}}$ at time instant n, we have the displacement vector

$$w[n] = \begin{bmatrix} w_x[n] \\ w_y[n] \end{bmatrix} = w_T[n] - w_Z[n]. \tag{1}$$

The observed states $s[n]$ are defined as

$$s[n] = \left\lfloor \frac{\arctan 2(w_y[n], w_x[n])}{\pi/4} \right\rfloor. \tag{2}$$

As we can observe in (2), the states depend only upon the SZ's and TID's position. The TID's position (relative) is assumed to be estimated by the AHD embedded sensors, such as visual cameras and a LiDAR system, while the TID's mission is not known to the AHD. In the Q-learning algorithm, the AHD agent performs an action $a[n]$ at time n, chosen from a set of possible actions $\mathcal{A}$, and the environment returns a state $s[n]$ from the set of possible states $\mathcal{S}$ and

$$a[n] = \left\lfloor \frac{\arctan 2(d_y[n], d_x[n])}{\pi/4} \right\rfloor, \tag{3}$$

where

$$d[n] = \begin{bmatrix} d_x[n] \\ d_y[n] \end{bmatrix} = w_A[n+1] - w_A[n] \tag{4}$$

is the displacement vector of the AHD at time instant n and $w_A[n] = [w_{Ax}[n], w_{Ay}[n]]^{\mathrm{T}}$ is the position vector of the AHD. The norm $\|d[n]\|_2$ is called the action size. Thus, the AHD moves in cardinal and intercardinal directions by its norm $\|d[n]\|_2$ units. The reward is a result of an action taken by the agent, *i.e.*, the reward $R(s[n], a[n])$ is obtained considering the state $s[n] \in \mathcal{S}$ when the action $a[n] \in \mathcal{A}$ was performed. We define the reward function $R(s[n], a[n])$ as

$$R(s[n], a[n]) = \begin{cases} \kappa, & \text{if } \|\mathbf{w}[n]\|_2 <= r_s \\ -\kappa, & \text{if } \|\mathbf{w}[n]\|_2 > \sigma \\ & \text{or } \|\mathbf{t}[n]\|_2 > \rho \\ -0.0125\,\|\mathbf{w}[n]\|_2\,, & \text{otherwise, called} \\ & \text{movement penalty,} \end{cases} \tag{5}$$

where r_s is the SZ radius, ρ is the maximum allowed distance or also called antenna distance, *i.e.*, $\|t[n]\|_2$ with

$$t[n] = \begin{bmatrix} t_x[n] \\ t_y[n] \end{bmatrix} = w_A[n] - w_T[n]. \tag{6}$$

κ is the kill reward the agent receives when a successful kill is made. Also, $-\kappa$ is the evasion reward when the TID evades the flight area with radius σ or the maximum antenna distance between the TID and the AHD is reached. Each action of the AHD results in a a new state $s[n+1]$ and reward $R(s[n], a[n])$. This iterative process is called action-sense-learn cycle [40].

The agent selects its next action based on the maximum Q-value policy $\pi[n]$ [41], *i.e.*,

$$\pi[n] = \underset{a \in \mathcal{A}}{\arg\max}\, Q(s[n], a), \tag{7}$$

where $Q(s[n], a)$ is called the Q-value, obtained from the Q-table, a list that comprises the possible states (observation) and Q-values based on which the agent will decide which action is best. During the operation, the values in the Q-table are updated following Bellman's equation [19],

$$\begin{aligned} Q(s[n], a[n]) \leftarrow\ & Q(s[n], a[n]) + \alpha[R(s[n], a[n]) \\ & + \gamma \underset{a \in \mathcal{A}}{\arg\max}\, Q(s[n+1], a) - Q(s[n], a[n])], \end{aligned} \tag{8}$$

where α is the learning rate, $Q(s[n+1], a)$ is the Q-value from state $s[n+1]$ after the action a was taken, and γ is the discount factor responsible for balancing the short-term and long-term reward, *i.e.*, acting as a memory measure in the system. We use the ϵ-greedy action selection policy to introduce randomness to the system balancing between exploration and exploitation.

In contrast to [20], the velocity ratio of AHD and TID were reduced in the present work. Also, as the acceleration step is introduced, the Q-values for a given configuration were improved as the randomness ϵ during training is higher.

The SITL simulation which is applied to simulate operational behavior of UAVs in various scenarios is based on the Ardupilot [42] embedded firmware and was presented in details in [20]. As shown in [20], this approach enables simulating the behavior of a real embedded firmware, but the SITL simulations performed take the same time as the real flights. This means at least three minutes per flight as the test bed also has to initiate and then perform the complete episode of the training process. The number of flights needed to populate the respective Q-table grows as we introduce more parameters to the guidance and control problem. Thus, we propose a generalized model of the UAV movement in the next section called SUMM as a mean to obtain the Q-table faster, which is loaded in the SITL simulation [20], and the performance of the SUMM-trained Q-table in the SITL is compared. The SITL test bed is used in this work to validate the SUMM approach in reducing the training time and thus is considered to emulate realistic UAV behavior in the considered scenarios.

IV. SUMM SIMULATION

Proportional–integral–derivative controllers (PIDs) are commonly used to control attitude and movement of UAVs, several combinations of linear and nonlinear PIDs are discussed in [43], [44], and [45]. The ETA derived in this work uses a PID-like approach to represent the general behavior of a UAV. We consider only the UAV's position and velocity in a 2D space, flying towards the desired waypoint. The behavior of the UAV can be characterized by a simple closed-loop control system where the process variable is the position in space, and the linear velocity is limited to match the real-life maximum velocities of commercially available UAVs. For this approach, the UAV is considered a point mass in space, and only the cinematic movement is considered. Thus, we neglect rigid body quantities, as stated by [46], which are assumed to be symmetrical and with constant mass. Also, the mass distribution over the vehicle is minimal, resulting in a negligible inertia matrix.

Furthermore, as presented in [45], it is common to separate the vehicle controller into several stages, such as position controller, attitude controller, and adaptive PIDs, to mix multiple quantities to control a UAV with unknown dynamics. In contrast to the classical implementation of error-tracking controllers to control the UAV motion in a given space, for example, shown in [47], in our case, we emulate a closed-loop problem. In contrast, the position error for the desired waypoint is driven to zero for increasing iterations. We also use a pair of controllers responsible for the x and y directions in a 2D Cartesian coordinate system since we assume an independent multivariate positioning sensor model. Thus, the output of the ETA in the SUMM simulation for the x and y coordinate directions are modeled each as a PID controller with a feedback loop [48], [49]

$$u_{x,y}(t) = K_{p_{x,y}} e_{x,y}(t) + K_{i_{x,y}} \int_0^t e_{x,y}(\tau) d\tau + K_{d_{x,y}} \frac{de_{x,y}(t)}{dt} \tag{9}$$

or, in the discrete form,

$$u_{x,y}[n] = K_{p_{x,y}} e_{x,y}[n] + K_{i_{x,y}} \sum_{k=0}^{n} e_{x,y}[k] + K_{d_{x,y}} \left[e_{x,y}[n] - e_{x,y}[n-1]\right] \tag{10}$$

where $K_{p_{x,y}} \in \mathbb{R}$, $K_{i_{x,y}} \in \mathbb{R}$, $K_{d_{x,y}} \in \mathbb{R}$ are the proportional, integral, and derivative constants of the controllers, $e_{x,y}(t) \in \mathbb{R}$ is the error in time and $e_{x,y}[n] \in \mathbb{R}$ is the representation of the discrete error, defined as the difference between the reference input and the output (from the feedback loop [48], [49])

$$e_{x,y}[n] = r_{x,y}[n] - u_{x,y}[n] + \epsilon_{x,y}[n] \tag{11}$$

where $r_{x,y}[n] \in \mathbb{R}$ is the discrete input reference signal, the desired waypoint in our case, $u_{x,y}[n] \in \mathbb{R}$ is the output of the PID, and $\epsilon_{x,y}[n] \in \mathbb{R}$ is an error from external sources, e.g. the error from the target detection system and the error from the GNSS receiver of the UAV. It is essential to notice that the x and y subscripts refer to the equation in each coordinate system axis.

Even though the presented algorithm is considering a 2D space for guidance, the flight levels of the AHD and TID are different, as required by the meaconing. Therefore, there is no chance of collision. Also, the flight phases in modern flight controllers are separated, e.g., the take-off and landing phases have an altitude controller. In contrast, the cruise flight phase has a controller providing navigation and guidance. Thus, small changes in altitude from the initial position and the TW can be neglected. Thus, the presented 2D algorithm can be used in the cruise phase of the flight. As a requirement, the detection and tracking systems of the AHD must maintain a minimum vertical distance of the TID, e.g., maintaining the AHD at a flight level above the TID.

For the SUMM, we define the desired waypoint coordinate tuple in the x and y coordinates as $(r_x[n], r_y[n])$. The constants $K_{p_{x,y}}$, $K_{i_{x,y}}$, $K_{d_{x,y}}$ are the same for the x and y directions but are chosen differently for the AHD and TID. The controller output tuple is defined as $(u_x[n], u_y[n])$ and is updating the vehicle position, whereas $K_{p_{x,y}}$, $K_{i_{x,y}}$, $K_{d_{x,y}}$ are kept rather small. In our proposed SUMM simulation, we are not interested in controlling a vehicle, but, instead we are interested in finding a simple descriptive model that represents the overall behavior of a quadrotor while a waypoint is set to its embedded firmware and the movement it performs towards the desired direction. The SUMM simulation procedure is outlined in Algorithm 1. In the following, the same structure is applied to emulate the behavior of the AHD and the TID.

Algorithm 1 SUMM UAV Simulation Procedures

```
1: Create the drone and movement related classes.
2: Load initial parameters $K_{p_{x,y}}$, $K_{i_{x,y}}$, $K_{d_{x,y}}$ for the given vehicle.
3: Sets the initial waypoint pair $(r_x[0], r_y[0])$.
4: procedure Simulation Loop
5:     Update error $e_{x,y}[n]$.
6:     Calculate output by $u_{x,y}[n] = K_{p_{x,y}} e_{x,y}[n] + K_{i_{x,y}} \sum_{k=0}^{n} e_{x,y}[k] + K_{d_{x,y}} \left[e_{x,y}[n] - e_{x,y}[n-1]\right]$.
7:     Update UAV position.
8:     Logs simulation parameters.
9: end procedure
10: End of the simulation.
```

Finally, we present the final TID UAV model for x and y axis positions as

$$\begin{aligned} u_x[n] &= 0.03\, e_x[n] + 0.0001 \sum_{k=0}^{n} e_x[k] \\ &\quad + 0.001\, (e_x[n] - e_x[n-1]) \\ u_y[n] &= 0.03\, e_y[n] + 0.0001 \sum_{k=0}^{n} e_y[k] \\ &\quad + 0.001\, \left(e_y[n] - e_y[n-1]\right) \end{aligned} \tag{12}$$

along with the final AHD UAV model for x and y axis positions as

$$\begin{aligned} u_x[n] &= 0.8\, e_x[n] + 0.01 \sum_{k=0}^{n} e_x[k] \\ &\quad - 0.3\, (e_x[n] - e_x[n-1]) \\ u_y[n] &= 0.8\, e_y[n] + 0.01 \sum_{k=0}^{n} e_y[k] \\ &\quad - 0.3\, \left(e_y[n] - e_y[n-1]\right) \end{aligned} \tag{13}$$

for the n-ith iteration.

TABLE 1. Training information of the q-learning agent.

Parameter	SITL [20]	SUMM
MC Episodes	1000	10000
learning rate α	0.1	0.1
ϵ-greedy policy randomness ϵ	0.01	0.5 (min 0.02)
discound factor γ	0.95	0.95
AHD Action Size $\|\mathbf{d}\|_2$	15 m	15 m
Actions per Episode	400	1000
Movement Penalty	1	$R(s[n], a[n])$
Kill Reward κ	400	400
Evasion Reward $-\kappa$	-400	-400
Initial TID Position (r_x, r_y)	(-300,-200)	(-300,-200)
Initial AHD Position (r_x, r_y)	(300,-200)	(300,-200)
Target Waypoint TW	(150,-150)	(150,-150)
Safe-zone Position SZ	(-300,100)	(-300,100)
TID $K_{p_{x,y}}$	-	0.03
TID $K_{i_{x,y}}$	-	0.0001
TID $K_{d_{x,y}}$	-	0.001
AHD $K_{p_{x,y}}$	-	0.8
AHD $K_{i_{x,y}}$	-	0.01
AHD $K_{d_{x,y}}$	-	-0.3
TID Max. Linear Velocity	1 m/s	8 m/s
AHD Max. Linear Velocity	10 m/s	52 m/s
AHD Flight Level	40m	40m
TID Flight Level	30m	30m

V. TRAINING PROCESS

The training process aims to derive Q-values of the Q-table so that the agent learns the best way to interact with the environment. In our problem at hand, this means learning the best possible trajectory for the AHD to guide the TID to the SZ. Then, during the actual flight, the Q-table resulting from the final training episode can be used to resolve the guidance and control problem. The training can also be continued during the flight to adapt to changing situations. Thus, in any case, the training process needs to be of low computational complexity.

In Table 1 the training parameters used for the SITL and SUMM training cases are listed as well as the location of the TW and the SZ. The SUMM parameters were chosen to match the behavior of the SITL simulation represented by the resulting Q-table with its Q-values. Generally, the Q-table can only be generated for a finite number of discrete actions and states. Additionally, Q-learning, in its basic form, starts to fail for a larger number of actions and states, as the likelihood of the agent observing a particular state and performing a particular action is increasingly small. Therefore, the Q-table with its Q-values can only provide rough information about the environment and the involved systems but also sufficient information to solve complex control problems. Consequently, when designing dynamic system models for training, the model depth can be traded-off against its complexity, as the resulting Q-table based on different models can provide a similar solution to the problem at hand. Therefore, the training hyperparameters were not included in the Q-table as new state-action pairs, keeping the RL algorithm decoupled from the ETA.

To compare the simulation time for both training cases that we discussed earlier, we consider 400 episodes being simulated with the SITL and the SUMM. The results are shown in Fig. 1. In general, we can observe that the SUMM simulation achieves a much shorter simulation time for each episode. The peaks at multiples of 100 for the SUMM simulations are caused by the flight log insertion to the overall training database. We begin by observing Figure 1 where the comparison of both methods yields the main reason for proposing an approach: a low simulation time per flight against keeping the same overall mechanics between the AHD and the TID.

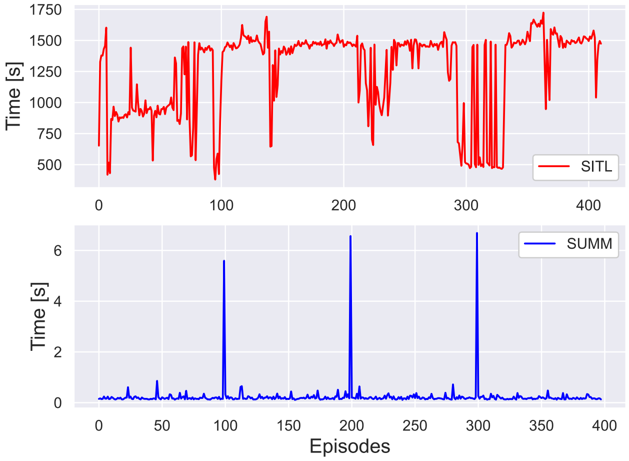

FIGURE 1. Episode time comparison between SITL and SUMM simulations.The spikes in the blue line happens when the data is saved to the database.

Additionally, we define the maximum-kills stop criteria (MKSC), where we early-end the training process in the SUMM when the number of overall kills is achieved. Also, we define the successive-kills stop criteria (SKSC), where the simulation is stopped when a given number of successive kills is achieved. In the following, MKSC500 means we stop the training process after 500 accumulated kills. The MKSC and SKSC can be combined in the SUMM simulations. These two parameters help to decide when the training is sufficient to start the mission of the AHD based on the derived Q-table.

VI. RESULTS AND DISCUSSION

To evaluate our proposed C-UAS method, we choose representative flight scenarios and we generate the trajectories of the AHD and TID during a hunting mission using the described SITL simulation. We also perform Monte Carlo (MC) SITL simulations of flights using pre-trained SUMM Q-tables to derive and analyze the performance of our proposed method.

A. SUMM LOADED SITL FLIGHTS

In Fig. 2, we show nine flights simulated with the SITL approach loaded with a pre-trained Q-table using the SUMM training. The red dashed line represents the flight trajectory of the AHD, and the solid blue line shows the TID's flight trajectory. The SZ is illustrated as a yellow circle while the red *X* denotes the next TW as planned in the TID's mission. It is important to note that for these examples, no previous SITL flights were performed to train the used Q-table. Based on these example flights, we will discuss the behavior of our proposed Q-learning approach for specific cases. Afterward, we will analyze the performance of our proposed approach based on MC simulations, especially the kill probability.

In general, we can observe that, as the AHD moves in space, the TID is guided towards the SZ. Thereby, guiding the TID through the movements of the AHD would impose a high cognitive load on a human pilot steering the AHD. Thus, the need for our Q-learning agent arises, which accomplishes this complex task without directly involving a human pilot.

In Fig. 2 (a), Fig. 2 (f) and Fig. 2 (h) scenarios are shown in which the AHD is not capable of steering the TID towards the SZ, but the TID also does not reach its next TW. In some cases the TID is approaching its next TW, but not reaching it as shown in Fig. 2 (a) and Fig. 2 (h). In other cases the TID simply tangents the next TW and is then guided away by the AHD, as shown in Fig. 2 (f). The TID is driven away from the TW even in the case the TID is not approaching the SZ due to the TW's location and the implemented reward function, as it penalizes the movements closer to the safe zone less. In this manner, the TID first is guided towards the SZ and then driven away from the TW. Afterwards, as the TID is still seeking to get to the TW, its position drifts until it leaves the defined flight zone.

The SZ kill radius r_s together with the action size $\|\mathbf{d}\|_2$ influence the kill rate of the TID. In the case $\|\mathbf{d}\|$ and r_s are chosen quite small the TID might pass through or by the SZ without a successful kill. Such cases can be seen in Fig. 2 (b) and Fig. 2 (i), where the TID is driven to the SZ, but the $r_s = 25$ m criteria is not met and the AHD decides to simply drive the TID away from its next TW. The flight example shown in Fig. 2 (b) addresses another important aspect. It demonstrates that it is vital that the AHD's velocity is higher than the TID's velocity, as the AHD has to fly ahead of the TID to recover the steering closer to the SZ.

In the example flights shown in Fig. 2 (c) to Fig. 2 (e) and in Fig. 2 (g) the AHD successfully guides the TID to the SZ, the kill criteria is met, and we can consider the TID successfully killed. Also, we can observe that the AHD and TID trajectories are quite similar but not equal, as in general convergence can be obtained from the SUMM training. However, the act-sense-learn cycle is still active during the flight and thus, the AHD adapts while flying and consequently is behaving a bit different for the depicted scenarios.

Even though the Q-table only has a discrete set of states, the system can adapt to changing conditions as the state-action pairs comprise a set of possible AHD and TID positions in space quantized by $\pi/4$ angles. If the discretization is not high enough to guide the TID to the SZ, the AHD can retrain in flight. As can be seen in Fig. 2 (b) and (i), the AHD velocity must be higher than the TID's velocity, so the AHD can move in space in the opposing direction the TID is heading. The closer the AHD's velocity is to the TID, the broader the steering area has to be accepted and more iterations are needed to guide the TID to the SZ.

Finally, an important element of the overall problem that needs to be discussed is the overall size of the space needed for the maneuvers of the AHD. In practical terms, the larger the needed area, the higher the probability that the AHD will lose its capability to continue spoofing the TID due to the increased antenna distance. On the other hand we cannot make the maneuver box smaller due to the AHD's and TID's velocity, as the AHD needs sufficient space to guide the TID to the SZ.

B. SUMM LOADED SITL MONTE CARLO (MC) SIMULATION

In order to derive the kill probability p_k we performed MC simulations of a SUMM trained scenario. p_k is the probability that the TID will be guided into the SZ with a radius of r_s meters. We can calculate the p_k by assuming the number of trials m in our experiment is large enough, i.e., $m \rightarrow \infty$, we can approximate p_k by the relative frequency as

$$p_k = \lim_{m \to \infty} \frac{m_k}{m} = \frac{m_k}{m}, \tag{14}$$

where m_k is the number of kills in m tries. The SITL MC simulations are performed with the parameters as described in the previous section and $m = 1000$. We analyze different SZ radii r_s for the case of a MKSC of 800. The convergence of the relative frequency to p_k for different r_s is depicted in Fig. 3 and the summarized results of the MC simulations are listed in Table 2. To calculate the standard deviation we used a binomial approximation, with 95% confidence interval.

TABLE 2. SITL MC results.

r_s	p_k
	MKSC800
25	0.424 ± 0.0306
50	0.559 ± 0.0308
75	0.991 ± 0.0059

The radius r_s plays a key-role in the problem at hand and p_k increases significantly if r_s is increased. The results show that the proposed C-UAS system is able to guide the TID to the SZ in 42% of the cases for $r_s = 25$ m which can be considered a feasible scenario in urban areas. However, even if the drone is not captured in the SZ, it is steered away from its original TW. Thus, the AHD in all cases can confound the mission of the THD.

To further analyze the results, we show the final position spatial distribution for the SITL MC simulations in Fig. 4. The higher the r_s, the more the AHD's final positions, represented by the red dots, are clustered in a small final region, indicating that, besides the non-deterministic portion of the presented problem, the AHD is adapting while guiding the TID towards the SZ (act-sense-learn cycle). The TID's final positions are concentrated either well inside the SZ or the TID is guided away from the next TW. It is essential to note the arrangement of the points in Fig. 4 when we compare the three scenarios: in the first and second scenarios presented in Fig. 4(a) and (b), with a radius of $r_s = 25m$ and $r_s = 50m$, the distribution

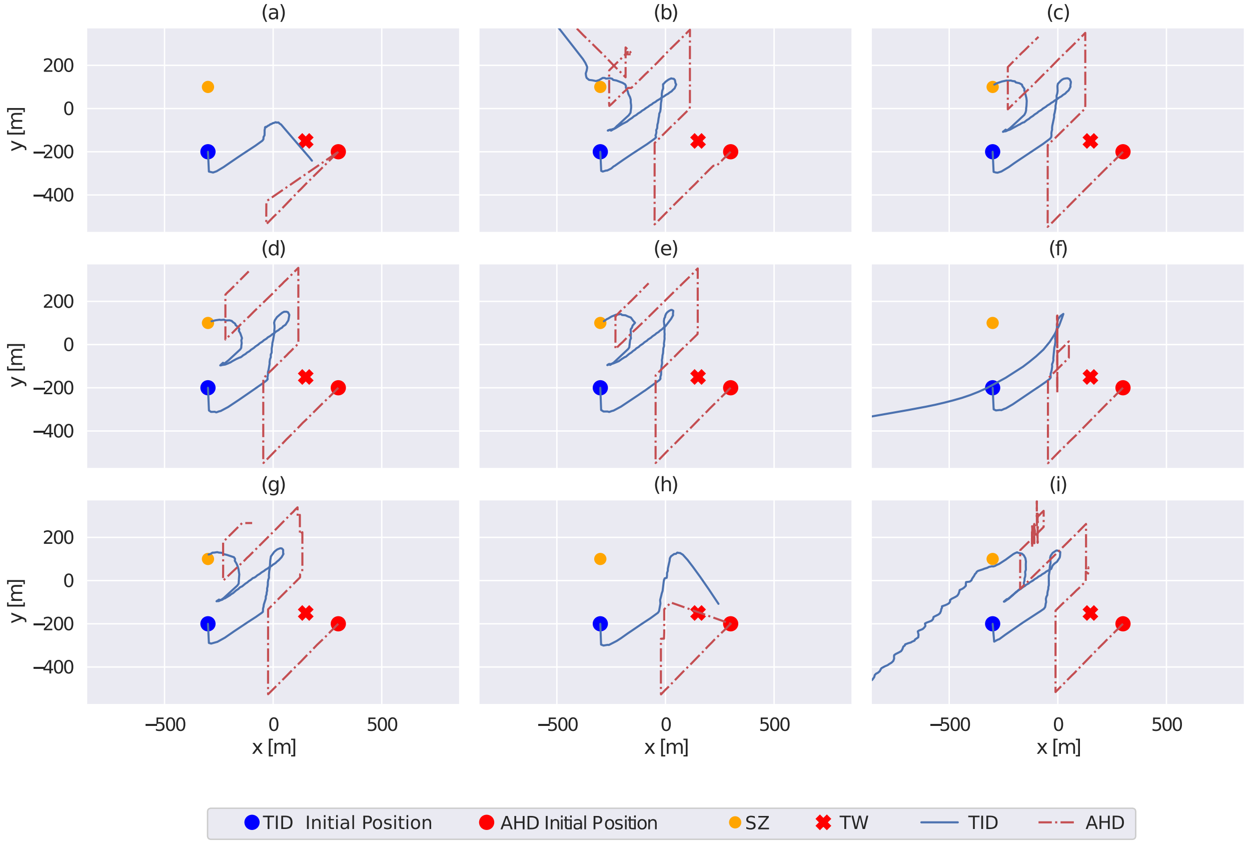

FIGURE 2. SITL flight trajectories loaded with previously pre-trained Q-tables from the SUMM training algorithm.

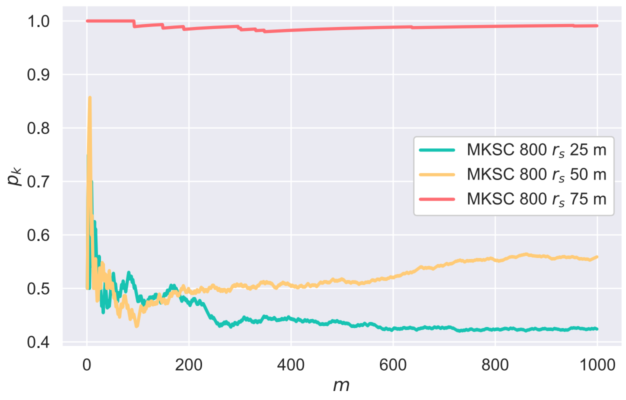

FIGURE 3. Probability of Kill p_k obtained in SITL MC simulations for multiple MKSC and r_s.

of the final position of both drones is by a larger region of the map when compared to the scenario represented in Fig. 4(c) with greater r_s.

Also, in Fig. 4(c), the final target positions, represented in blue inside the yellow region, are concentrated in an arc of the safe zone perimeter. This result represents the pattern from the UAV dynamics, along with the agent adopting an equivalent strategy to guide the TID to the SZ using similar maneuvers.

For the AHD and TID, the distances of their final positions to the SZ r_{sz} and to the TW r_{tw} are calculated. The results are shown in Fig. 5 for MKSC800 and $r_s = 25$ m. We can state that $r_s = 25$ m is the hardest case as it has the smallest safe zone while maintaining the same velocity ratio between the AHD and the TID. On the other hand, in the cases the TID's velocity is higher than that of the AHD, the latter would not be able to steer the TID to the safe zone. Therefore, in such a case, the AHD is still capable of driving the TID away from its TW.

Furthermore, the action size defines the movement of the TID from time instant to instant. A larger action size $\|d\|$ would lead to the TID flying over the SZ at cruise velocity and a soft-kill would not be successful.

In Fig. 5, the diagonal graphs are the histograms of the respective variables. There are three categories of final positions of the AHD and the TID

- the TID was killed in the SZ,
- the AHD or TID flew out of the defined area (out of box),
- the AHD reached the maximum movements and could not further guide the TID.

The r_{sz} and r_{tw} for the ADH or the TID have a linear dependency, as can be expected for fixed SZ and TW. If the AHD or the TID is out of the defined flight area or box, r_{sz} and r_{tw}

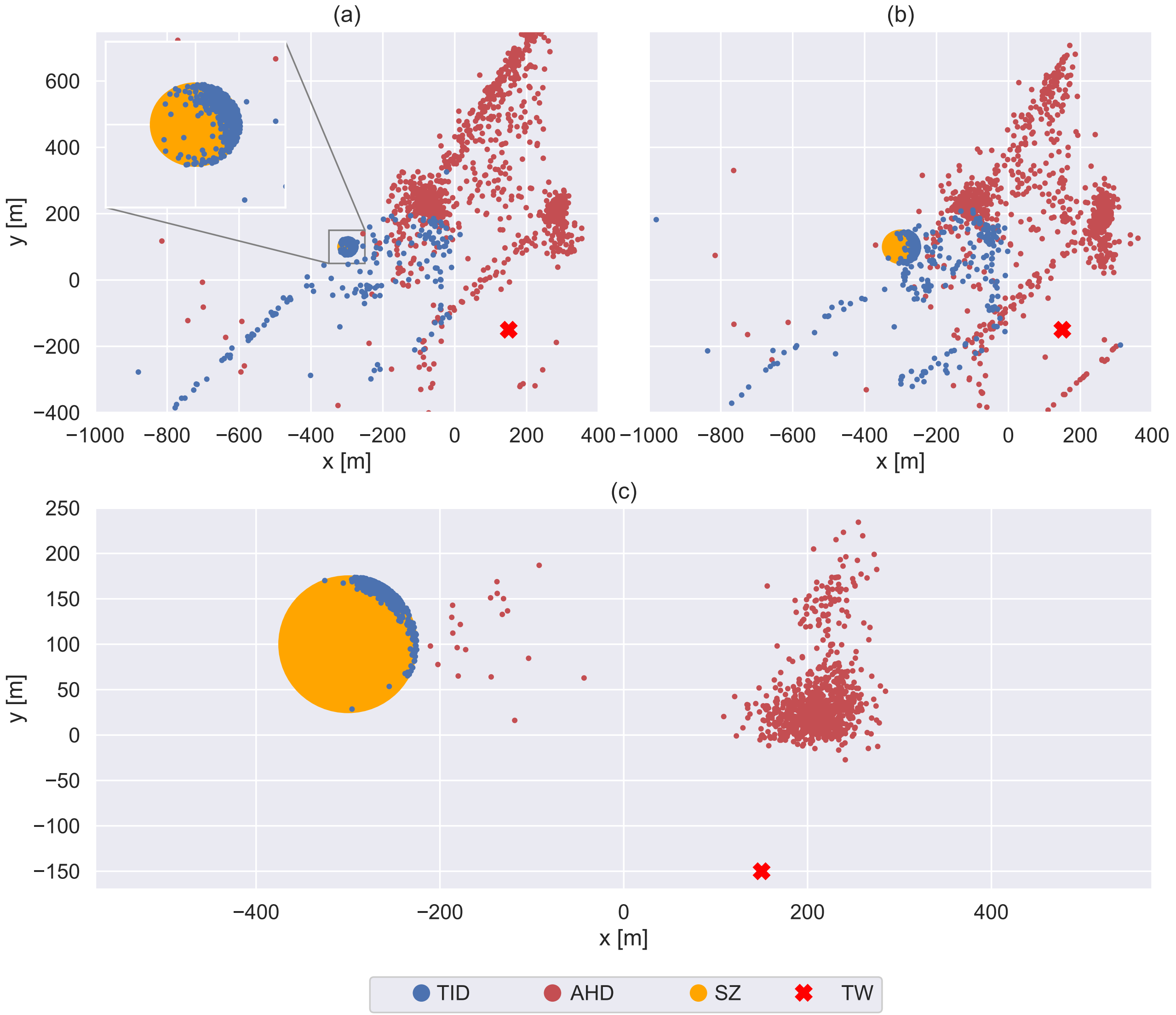

FIGURE 4. Scatter plot of the final TID (blue) and AHD (red) positions obtained in the SITL MC simulations, using MKSC 800 and different r_S. (a) MKSC $r_S = 25$m. (b) MKSC $r_S = 50$m. (c) MKSC $r_S = 75$m.

have rather high values. The same is true when either drone is far from the region of interest. The distribution of the r_{tw} higher values describes our objective to drive the TID away from the TW. The narrow peaks of the killed states represent the capability of our system to steer the TID to a bounded SZ region. The closer to zero and narrower the peaks, the smaller the bounding box required to execute the soft-kill process. If the final status is maximum movements, we note the wide range over which these points are distributed. In each case, the flight trajectory is still contained in the bounding box but the AHD were not able to drive the TID to the SZ effectively - as is represented in Fig. 2 (a), 2 (f), and 2 (h).

Finally, it is essential to note that the target is successfully steered towards the SZ using a small region no more than a kilometer wide in most flights. These results show that the proposed method is promising for dense urban regions with an increased p_k. The SUMM pre-training and SITL evaluation combination are complementary when evaluating C-UAS systems for a wide range of mission scenarios.

VII. CONCLUSION

As UAVs became available to the general public, the need for C-UAS arises in order to protect a given area from malicious or non-authorized UAVs. The high cost of conventional methods to destroy UAVs from airports to stadiums motivates our research and development of a cheaper and more general approach to capture and soft-kill these vehicles effectively and safely.

This work proposed a soft-kill approach to C-UAS by automatically guiding a UAV to an SZ with a Q-learning algorithm. The proposed approach achieved high success rates and additionally it was designed so that even if the TID

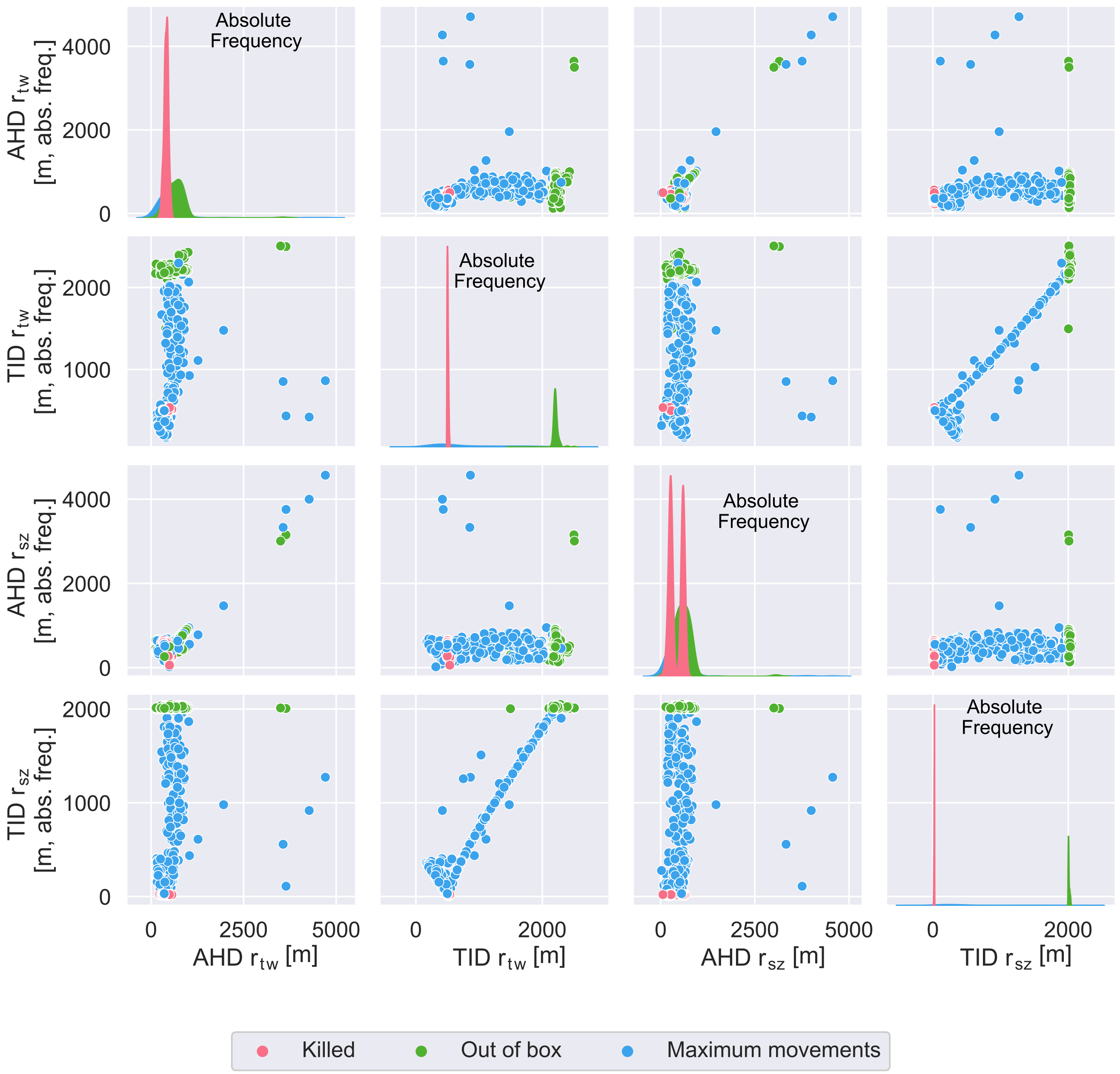

FIGURE 5. Final AHD and TID distances to SZ (r_{sz}) and distances to TW (r_{tw}) distributions obtained by SITL MC simulation, for MKSC 800 and $r_s = 25m$. The diagonal plots show the histogram of either TID r_{sz}, AHD r_{sz}, TID r_{tw}, and AHD r_{tw}, respectively.

does not reach the SZ, it cannot fly to the next TW. Thus, the Q-learning algorithm in the AHD could always identify and fail the mission of the TID.

The SUMM method was introduced as an acceleration tool compared to the SITL simulations for the training to achieve a better trade-off between high fidelity and low computational cost. The SITL simulation was designed to simulate the true behavior of UAVs in dedicated mission scenarios, while the SUMM method only implements a coarse model of the behavior of UAVs, but captures its main features to accelerate training drastically.

Finally, this work showed a new way to implement expanded Q-tables for complex Q-learning algorithms, perform the training in reasonable time, and perform extensive evaluation by a more straightforward training method.

VIII. FUTURE WORK

The proposed algorithm allows more complex scenarios to be considered, and a large Q-table can be used in future developments and system improvements. Also, as more scenarios can be considered for different strategies for the attacker, the C-UAS can be prepared for different mission requirements.

The presented techniques and system designs can also simulate strategies against drone swarms since the SITL alone would require prohibitive resources for training such scenarios. The strategy to be used against drone swarms is to spoof the centroid of the swarm in the same presented way a meaconer payload would with a single drone.

Furthermore, the presented 2D algorithm can easily be extended to 3D to additionally consider different flight levels and changes in the target's altitude during flight. The Q-table also can be expanded to allow a quantized azimuth vector pointing from the AHD to the TID as states.

REFERENCES

[1] J. M. Sullivan, "Evolution or revolution? The rise of UAVs," *IEEE Technol. Soc. Mag.*, vol. 25, no. 3, pp. 43–49, Fall 2006.

[2] G. Sachs. (2019). *Drones: Reporting for Work*. [Online]. Available: http://www.goldmansachs.com/insights/technology-driving-innovation/drones/

[3] J. Lockett. (Sep. 2019). *Saudi Oil Attack—Drone Strikes are Impossible to Stop and NO Country Can Defend Against Them, Military Expert Warns*. [Online]. Available: https://www.thesun.co.uk/news/9971225/saudi-oil-attack-drone-strikes-no-defence/

[4] M. Min, L. Xiao, D. Xu, L. Huang, and M. Peng, "Learning-based defense against malicious unmanned aerial vehicles," in *Proc. IEEE VTC-Spring*, Jun. 2018, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8417685/

[5] J. Warrick. (2019). *Use of Weaponized Drones by Isis Spurs Terrorism Fears—The Washington Post*. [Online]. Available: https://www.washingtonpost.com/world/national-security/use-of-weaponized-drones-by-isis-spurs-terrorism-fears/2017/02/21/9d83d51e-f382-11e6-8d72-263470bf0401_story.html

[6] G. A. Venkatesh, P. Sumanth, and K. R. Jansi, "Fully autonomous UAV," in *Proc. Int. Conf. Tech. Advancements Comput. Commun. (ICTACC)*, Melmaurvathur, India, Apr. 2017, pp. 41–44.

[7] U. A. Drones, "Unmanned aircract system airspace integration plan," version 2.0, Tech. Rep., Mar. 2011.

[8] (Jun. 2022). *Amazon Prime Air Prepares for Drone Deliveries*. [Online]. Available: https://www.aboutamazon.com/news/transportation/amazon-prime-air-prepares-for-drone-deliveries

[9] M. Kratky and V. Minarik, "The non-destructive methods of fight against UAVs," in *Proc. Int. Conf. Mil. Technol. (ICMT)*, Brno, Czech Republic, May 2017, pp. 690–694. [Online]. Available: http://ieeexplore.ieee.org/document/7988845/

[10] J. Wang, Y. Liu, and H. Song, "Counter-unmanned aircraft systems (C-UAS): State of the art, challenges, and future trends," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 36, no. 3, pp. 4–29, Mar. 2021.

[11] (Feb. 2018). *Report: Counter-Drone Systems*. [Online]. Available: https://dronecenter.bard.edu/counter-drone-systems/

[12] H. Kang, J. Joung, J. Kim, J. Kang, and Y. S. Cho, "Protect your sky: A survey of counter unmanned aerial vehicle systems," *IEEE Access*, vol. 8, pp. 168671–168710, 2020.

[13] M. Saito, K. Masuda, and K. Kurihara, "A flexible Q-relearning method to accelerate learning under the change of environments by reusing a portion of useful policies," in *Proc. SICE Annu. Conf.*, 2012, pp. 1223–1227.

[14] T. Sugimoto and M. Gouko, "Acquisition of hovering by actual UAV using reinforcement learning," in *Proc. 3rd Int. Conf. Inf. Sci. Control Eng.*, Beijing, China, Jul. 2016, pp. 148–152. [Online]. Available: http://ieeexplore.ieee.org/document/7726139/

[15] S. C. Stubberud and K. A. Kramer, "Navigation error experiment using dynamic system matching," in *Proc. 6th Int. Symp. Commun., Control Signal Process. (ISCCSP)*, Athens, Greece, May 2014, pp. 372–375. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6877891

[16] W. Qiang and Z. Zhongli, "Reinforcement learning model, algorithms and its application," in *Proc. Int. Conf. Mechatronic Sci., Electr. Eng. Comput. (MEC)*, Aug. 2011, pp. 1143–1146.

[17] S. Ravichandiran, *Hands-On Reinforcement Learning With Python: Master Reinforcement and Deep Reinforcement Learning Using OpenAI Gym and TensorFlow*. Birmingham, U.K.: Packt Publishing, 2018.

[18] M. Lapan, *Deep Reinforcement Learning Hands-On: Apply Modern RL Methods, With Deep Q-Networks, Value Iteration, Policy Gradients, TRPO, AlphaGo Zero and More*. Birmingham, U.K.: Packt, 2018.

[19] B. S. Ciftler, A. Tuncer, and I. Guvenc, "Indoor UAV navigation to a Rayleigh fading source using Q-learning," 2017, *arXiv:1705.10375*.

[20] D. L. Da Silva, F. Antreich, O. L. Coutinho, and R. Machado, "Q-learning applied to soft-kill countermeasures for unmanned aerial vehicles (UAVs)," in *Proc. IEEE/ION Position, Location Navigat. Symp. (PLANS)*, Apr. 2020, pp. 91–99.

[21] C. Gronwall, J. Rydell, M. Tulldahl, E. Zhang, F. Bissmarck, and E. Bilock, "Two imaging systems for positioning and navigation," in *Proc. Workshop Res., Educ. Develop. Unmanned Aerial Syst. (RED-UAS)*, Linkoping, Sweden, Oct. 2017, pp. 120–125. [Online]. Available: http://ieeexplore.ieee.org/document/8101654/

[22] P. Guo, X. Li, Y. Gui, X. Zhou, H. Zhang, and X. Zhang, "Airborne vision-aided landing navigation system for fixed-wing UAV," in *Proc. 12th Int. Conf.*, Hangzhou, China, Oct. 2014, pp. 1215–1220. [Online]. Available: http://ieeexplore.ieee.org/document/7015193/

[23] G. Miko and A. Nemeth, "Combined communication and radio navigation system for small UAVs," in *Proc. 23rd Int. Conf. Radioelektronika*, Pardubice, Czech Republic, Apr. 2013, pp. 284–287. [Online]. Available: http://ieeexplore.ieee.org/document/6530932/

[24] C. G. L. Krishna and R. R. Murphy, "A review on cybersecurity vulnerabilities for unmanned aerial vehicles," in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot. (SSRR)*, Shanghai, China, Oct. 2017, pp. 194–199. [Online]. Available: http://ieeexplore.ieee.org/document/8088163/

[25] H. Zhang, H. Zhang, X. Liu, and T. A. Gulliver, "A new electromagnetic jamming system for unmanned aerial vehicles," in *Proc. IEEE Pacific Rim Conf. Commun., Comput. Signal Process. (PACRIM)*, Victoria, BC, USA, Aug. 2017, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/8121921/

[26] D. He, Y. Qiao, S. Chan, and N. Guizani, "Flight security and safety of drones in airborne fog computing systems," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 66–71, May 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8360852/

[27] S. M. Giray, "Anatomy of unmanned aerial vehicle hijacking with signal spoofing," in *Proc. 6th Int. Conf. Recent Adv. Space Technol. (RAST)*, Istanbul, Turkey, Jun. 2013, pp. 795–800. [Online]. Available: http://ieeexplore.ieee.org/document/6581320/

[28] S. Bao, J. Lai, Z. Chen, P. Lyu, and W. Chen, "Aerodynamic model/INS/GPS failure-tolerant navigation method for multirotor UAVs based on federated Kalman filter," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 1121–1125. [Online]. Available: http://ieeexplore.ieee.org/document/8242934/

[29] O. Eroglu and G. Yilmaz, "A novel fast and accurate algorithm for terrain referenced UAV localization," in *Proc. Int. Conf. Unmanned Aircr. Syst.*, Atlanta, GA, USA, May 2013, pp. 660–667. [Online]. Available: http://ieeexplore.ieee.org/document/6564746/

[30] D. He, S. Chan, and M. Guizani, "Communication security of unmanned aerial vehicles," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 134–139, Aug. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7792372/

[31] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "A hierarchical detection and response system to enhance security against lethal cyber-attacks in UAV networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 9, pp. 1594–1606, Sep. 2018. [Online]. Available: https://ieeexplore.ieee.org/document/7890467/

[32] M. D. Pritt and K. J. LaTourette, "Aircraft navigation by means of image registration," in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop*, Washington, DC, USA, Oct. 2013, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/6749335/

[33] M. E. Angelopoulou and C.-S. Bouganis, "Vision-based egomotion estimation on FPGA for unmanned aerial vehicle navigation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 6, pp. 1070–1083, Jun. 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6668877/

[34] L.-P. Berge, N. Aouf, T. Duval, and G. Coppin, "Generation and VR visualization of 3D point clouds for drone target validation assisted by an operator," in *Proc. 8th Comput. Sci. Electron. Eng. (CEEC)*, Colchester, U.K., Sep. 2016, pp. 66–70. [Online]. Available: http://ieeexplore.ieee.org/document/7835890/

[35] O. Kechagias-Stamatis and N. Aouf, "A new passive 3-D automatic target recognition architecture for aerial platforms," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 1, pp. 406–415, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8423710/

[36] Y. Zhigui and L. ChuanJun, "Review on vision-based pose estimation of UAV based on landmark," in *Proc. 2nd Int. Conf. Frontiers Sensors Technol. (ICFST)*, Shenzhen, China, Apr. 2017, pp. 453–457. [Online]. Available: http://ieeexplore.ieee.org/document/8210555/

[37] V. F. Vidal, L. M. Honorio, M. F. Santos, M. F. Silva, A. S. Cerqueira, and E. J. Oliveira, "UAV vision aided positioning system for location and landing," in *Proc. 18th Int. Carpathian Control Conf. (ICCC)*, Sinaia, Romania, May 2017, pp. 228–233. [Online]. Available: http://ieeexplore.ieee.org/document/7970402/

[38] P. Teunissen and O. Montenbruck, *Handbook of Global Navigation Satellite Systems*. Berlin, Germany: Springer, 2017. [Online]. Available: https://link.springer.com/book/10.1007/978-3-319-42928-1

[39] M. L. Psiaki and T. E. Humphreys, "GNSS spoofing and detection," *Proc. IEEE*, vol. 104, no. 6, pp. 1258–1270, Jun. 2016.

[40] A. E. N. Asli and A. Jannesari, "A simulation of UAV power optimization via reinforcement learning," *CoRR*, 2019.

[41] M. M. U. Chowdhury, F. Erden, and I. Guvenc, "RSS-based Q-learning for indoor UAV navigation," 2019, *arXiv:1905.13406*.

[42] Ardupilot. (2019). *ArduPilot Open Source Autopilot*. [Online]. Available: http://www.ardupilot.org/

[43] A. A. Najm and I. K. Ibraheem, "Nonlinear PID controller design for a 6-DOF UAV quadrotor system," 2018, *arXiv:1806.05153*.

[44] O. A. Somefun, K. Akingbade, and F. Dahunsi, "'Closed proportional-integral-derivative-loop model' following control," 2020, *arXiv:2006.00314*.

[45] A. Goel, A. M. Salim, A. Ansari, S. Ravela, and D. Bernstein, "Adaptive digital PID control of a quadcopter with unknown dynamics," 2020, *arXiv:2006.00416*.

[46] S. Zouaoui, E. Mohamed, and B. Kouider, "Easy tracking of UAV using PID controller," *Periodica Polytechnica Transp. Eng.*, vol. 47, no. 3, pp. 171–177, Mar. 2018. [Online]. Available: https://pp.bme.hu/tr/article/view/10838

[47] A. L. Salih, M. Moghavvemi, H. A. F. Mohamed, and K. S. Gaeid, "Flight PID controller design for a UAV quadrotor," *Sci. Res. Essays*, vol. 5, p. 9, Dec. 2010.

[48] K. Ogata, *Modern Control Engineering* (Instrumentation and Controls Series), 5th ed. Boston, MA, USA: Prentice-Hall, 2010.

[49] V. van Veldhuizen, "Autotuning PID control using actor-critic deep reinforcement learning," Bachelor's Thesis, Univ. Amsterdam, Amsterdam, The Netherlands, 2020.

DOUGLAS LOPES DA SILVA (Member, IEEE) received the bachelor's degree in electronic and telecommunication engineering from the Federal University of Uberlândia (UFU), Uberlândia, Brazil, in 2018, and the M.Sc. degree in electronic engineering and computing from the Aeronautics Institute of Technology (ITA), São José dos Campos, Brazil, in 2020, where he is currently pursuing the Ph.D. degree in electronic and telecommunications engineering, researching on counter unmanned aerial systems (C-UAS). Since 2018, he has been a member of the Electronic Warfare Laboratory (LAB-GE), ITA. His research interests include UAVs, C-UAS, artificial intelligence, GNSS, and signal processing.

RENATO MACHADO (Senior Member, IEEE) received the B.S.E.E. degree from São Paulo State University (UNESP), Ilha Solteira, Brazil, in 2001, and the M.Sc. and Ph.D. degrees in electrical engineering from the Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, in 2004 and 2008, respectively. From November 2013 to February 2015, he was a Visiting Research Fellow with the Blekinge Institute of Technology (BTH) in partnership with Saab AB. From August 2009 to December 2017, he was an Assistant (2008–2016) and an Associate (2017) Professor with the Federal University of Santa Maria, Brazil, where he lectured many courses in bachelor's and graduate programs. He assumed different positions in the institution, such as a Research Leader of the Communications and Signal Processing Research Group, a Coordinator of the Telecommunications Engineering Program, and the Director of the Aerospace Science Laboratory. Since December 2017, he has been an Associate Professor with the Aeronautics Institute of Technology (ITA), where he is currently a Research Leader of the Digital and Signal Processing Laboratory and SAR and Radar Signal Processing Laboratory. He is also a Principal Investigator of the Brazilian Institute of Data Science (BI0S), a Research Center for Applied Science and Technology in Artificial Intelligence—FAPESP/MCTI/CGI. From August 2021 to July 2022, he was the Head of the Graduate Program in Electronics and Computer Engineering, ITA. Since June 2022, he has been the Head of the Electronic Engineering Division, ITA. His research interests include SAR processing, SAR image processing, change detection, radar signal processing, and digital signal processing.

OLYMPIO L. COUTINHO (Member, IEEE) received the B.Eng. degree in electrical engineering from the Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, Brazil, in 1993, and the M.Sc. and Ph.D. degrees in electronic and computing engineering from the Technological Institute of Aeronautics, São José dos Campos, Brazil, in 2005 and 2011, respectively. He was a Postdoctoral Fellow with the Microwave Photonic Research Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON, Canada, sponsored by the Technological Institute of Aeronautics (ITA), Brazil. He is currently an Adjunct Professor and a Researcher with ITA. His research interests include microwave photonic signal processing for radar applications and fiber optic sensors.

FELIX ANTREICH (Senior Member, IEEE) received the Diploma degree in electrical engineering and the Ph.D. degree from the Technical University of Munich (TUM), Munich, Germany, in 2003 and 2011, respectively. From 2003 to 2016, he was an Associate Researcher with the Department of Navigation, Institute of Communications and Navigation of the German Aerospace Center (DLR), Wessling, Germany. From 2016 to 2018, he was a Visiting Professor with the Department of Teleinformatics Engineering (DETI), Federal University of Ceará (UFC), Fortaleza, Brazil. Since July 2018, he has been a Professor with the Division of Electronics Engineering, Department of Telecommunications, Aeronautics Institute of Technology (ITA), São José dos Campos, Brazil. His research interests include sensor array signal processing for global navigation satellite systems (GNSS), wireless communications, estimation theory, wireless sensor networks, positioning, localization, and signal design for synchronization.