

Distributed Estimation and Control for Jamming an Aerial Target with Multiple Agents

Savvas Papaioannou, Panayiotis Kolios and Georgios Ellinas

Abstract—This work proposes a distributed estimation and control approach in which a team of aerial agents equipped with radio jamming devices collaborate in order to intercept and concurrently track-and-jam a malicious target, while at the same time minimizing the induced jamming interference amongst the team. Specifically, it is assumed that the malicious target maneuvers in 3D space, avoiding collisions with obstacles and other 3D structures in its way, according to a stochastic dynamical model. Based on this, a track-and-jam control approach is proposed which allows a team of distributed aerial agents to decide their control actions online, over a finite planning horizon, to achieve uninterrupted radio-jamming and tracking of the malicious target, in the presence of jamming interference constraints. The proposed approach is formulated as a distributed model predictive control (MPC) problem and is solved using mixed integer quadratic programming (MIQP). Extensive evaluation of the system's performance validates the applicability of the proposed approach in challenging scenarios with uncertain target dynamics, noisy measurements, and in the presence of obstacles.

Index Terms—Autonomous systems, cooperative agents, tracking, control, jamming.



NOMENCLATURE

x_t	Target state vector at time t .
y_t	Target measurement vector at time t .
u_t	Target control vector at time t .
s_t	Agent state vector at time t .
v_t	Agent control vector at time t .
ω_t	Target dynamical model uncertainty.
w_t	Measurement noise.
Φ, Γ	Target/Agent dynamical model parameters.
Δt	Sampling time interval.
$m^{\text{agent}}, m^{\text{target}}$	Agent and target mass.
η	Air resistance coefficient.
$x_{t' t}$	Predicted target state at time t' which is computed at time-step t .
$\hat{x}_{t' t}$	Estimated target state at time t' which is computed at time-step t .
$p_{t t-1}(x_t y_{1:t-1})$	Predictive distribution of the target state at time t .
$p_t(x_t y_{1:t})$	Posterior distribution of the target state at time t .
$f_{t t-1}(x_t x_{t-1})$	Target transitional density.
$g_t(y_t x_t, s_t)$	Measurement likelihood function.
\mathcal{S}	Set of cooperative pursuer agents.
R	Agent sensing range.
\mathcal{A}	Agent sensing profile.
$\Xi(x_t, s_t)$	Agent detection-and-jamming function.
T	Planning horizon length.

X_t	Target trajectory inside the planning horizon starting at time t .
U_t	Target control inputs inside the planning horizon starting at time t .
\hat{X}_t	Target estimated trajectory inside the planning horizon starting at time t .
S_t	Agent trajectory inside the planning horizon starting at time t .
V_t	Agent control inputs inside the planning horizon starting at time t .
$\mathcal{J}_{\text{target}}$	Target controller objective function.
$\mathcal{J}_{\text{centralized}}$	Tracking-and-jamming centralized objective function.
$\mathcal{J}_{\text{agent}}$	Tracking-and-jamming distributed objective function.

1 INTRODUCTION

In recent years, the interest in unmanned aerial vehicle (UAV) technologies for various application areas and industry sectors has soared. UAVs or drones, which were initially utilized as an elite high-tech military technology, have nowadays become widely available and have found widespread use in various domains including emergency response [1]–[5] and precision agriculture [6], [7] just to name a few. According to the latest Drone Market Report [8], the global market size for drones is predicted to grow to 42.8B by 2025. The proliferation of consumer drones, however, has also led to the emergence of new threats, with the use of drones for malicious purposes including targeting airports and restricted airspaces [9], attacking critical infrastructures [10], and threatening public safety [11]. These new threats necessitate the research and development of highly efficient, effective, and practical counter-drone systems which can be used to safeguard public safety. In particular, an effective

- The authors are with the KIOS Research and Innovation Centre of Excellence (KIOS CoE) and the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, 1678, Cyprus. E-mail: {papaioannou.savvas, pkolios, gellinas}@ucy.ac.cy

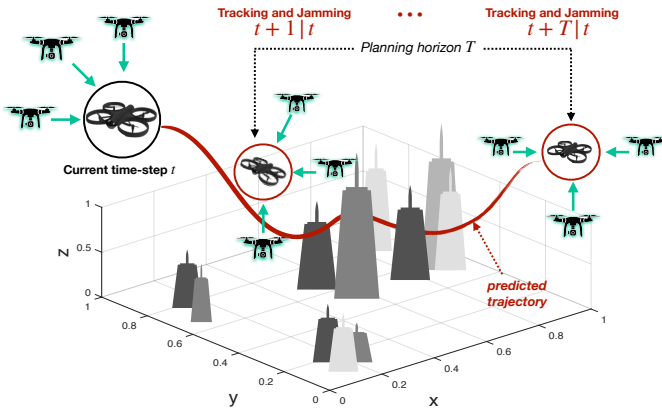


Fig. 1. A team of distributed autonomous aerial agents concurrently track and radio-jam a malicious target, while avoiding the jamming interference amongst them and collisions with obstacles in the environment.

counter-drone system [12] must be able to detect, track, and disable the operation of a malicious drone.

Over the last few years, a variety of counter-drone approaches and systems have been proposed for tackling the problem of malicious drones [13]–[15]. Anti-drone solutions, however, are still in their infancy and more work is required in order for this technology to reach the required level of maturity. As of today, no adequate solution exists for safeguarding public safety from the threat of malicious drones. Motivated by this necessity, this work proposes a distributed model predictive control (DMPC) [16], [17] approach in which a team of pursuer aerial agents equipped with radio-jamming devices, are collaborating to concurrently track-and-jam a malicious target in 3D space over an extended period of time, with the ultimate purpose of disrupting its navigation circuitry and thus forcing it to execute its fail-safe procedures [18], [19] and auto-land.

Specifically, it is assumed that the malicious target moves in 3D space according to a stochastic dynamical model and has the ability to maneuver around obstacles and other 3D structures found along its path. The pursuer agents are equipped with a 3D range-finding radar which they use for a) detecting the presence of the malicious target and b) acquiring target measurements. Specifically, the agents use their radar to receive noisy target measurements (i.e., radial distance, azimuth angle, and inclination angle), which they use to recursively estimate the target's state over time. In addition, the pursuer agents are equipped with an omnidirectional jamming system [20] which they use for emitting interference signals at specific wireless bands in an effort to jam the navigation circuitry of the malicious target i.e., disrupting the global navigation satellite system (GNSS) link of the malicious target.

Finally, the agents adapt their decisions and plans online over a finite planning horizon in order to better accommodate the collective objective of the team i.e., maximizing the collective tracking-and-jamming performance while at the same time minimizing the jamming interference amongst them.

The main contributions of this work are as follows:

- A distributed estimation and control approach is proposed for the problem of concurrently tracking

and radio-jamming a malicious target in 3D space. The problem of tracking-and-jamming is formulated as a constrained model predictive control problem. In this work a team of collaborative pursuer agents decide their control actions online over a rolling finite planning horizon, to achieve uninterrupted radio-jamming and tracking of the malicious target. At the same time the proposed control approach makes sure that the decided control actions over all pursuer agents minimize the induced jamming interference amongst the team.

- The problem of concurrent tracking-and-jamming is formulated in noisy and challenging 3D environments i.e., the agents and the malicious target must maneuver around obstacles and other 3D structures to avoid collisions, the malicious target's trajectory is uncertain and thus must be estimated from noisy measurements, and finally the pursuer agents exhibit a 3D range finding sensor with limited sensing range.
- A mixed integer quadratic mathematical program (MIQP) is derived for the problem tackled, which can be solved exactly using off-the-shelf solvers. The effectiveness of the proposed approach is demonstrated through extensive simulation experiments, addressing scenarios with uncertain target dynamics, noisy measurements, and in the presence of obstacles.

The rest of the paper is organized as follows. Section 2 reviews the state-of-the-art, Section 3 presents the problem definition and system overview, while Section 4 introduces the system model. Subsequently, Section 5 discusses target maneuvering and target state prediction and estimation, Section 6 details the distributed model predictive control approach for tracking-and-jamming, while Section 7 presents the performance results for a number of experiments. Finally, Section 8 concludes the work and discusses future research directions.

2 RELATED WORK

A comprehensive review of the technologies utilized in existing anti-drone systems can be found in [21], in which the authors also develop a ground-based anti-drone system capable of detecting, localizing, and radio-jamming a malicious drone. A ground-based radio-frequency (RF) drone detector and jammer system is also proposed in [19]. The authors design a drone detection mechanism using the RF control signal exchanged between the drone and its remote controller. Once the malicious drone is detected, a high-power jamming signal is transmitted towards the malicious drone causing it to disconnect from its controller and execute its fail-safe protocols. Furthermore, the work in [22] proposes a low-cost ground jamming system to counteract the operation of small drones.

Although the ground-based anti-drone systems can be low-cost and effective, they are also associated with the following three major limitations:

- these systems are located at fixed positions on the ground and hence they are only effective when the

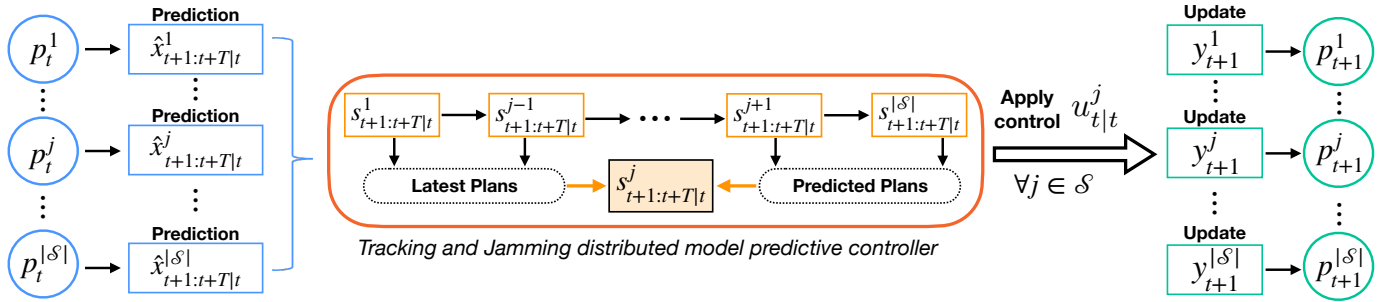


Fig. 2. Overview of the proposed tracking-and-jamming system architecture.

target resides inside their jamming range. As a consequence, an adversarial target can escape by intelligently navigating outside the system's jamming area. On the other hand in the proposed approach multiple mobile UAV agents with intelligent guidance, navigation, and tracking-and-jamming capabilities can pursue the malicious target, making its escape much harder

- as opposed to mobile anti-drone systems, ground systems appear to be less effective in environments with occlusions and obstacles, which affect their tracking and jamming signals
- necessitate the transmission of high-power jamming signals, as opposed to mobile anti-drone systems which can approach the target and use less power for jamming.

Another recent survey in [23] investigates a wide range of anti-drone technologies, including detection, identification, and neutralization, and proposes guidelines for effective drone defense operations. Various techniques for drone detection, tracking, and interdiction are also discussed in [13] and [24]. In [25] the authors propose a detection and tracking system for small and fast moving drones, whereas in [26] the authors use a consumer drone outfitted with antennas and commodity radios to autonomously localize another drone by its telemetry radio emissions. Additionally, the work in [27] proposes an anti-drone defense system for intercepting and escorting a malicious drone outside a restricted flight zone. In particular, a networked defense drone swarm forms a cluster around the malicious drone to prevent it from entering the restricted airspace. Similarly, the work in [28] proposes a decentralized formation control anti-drone system, with multiple multi-rotor aerial vehicles cooperating to establish a predefined enclosing formation around the malicious drone. Moreover, many of the existing works (e.g., [27], [28]) assume that a highly accurate drone detection and tracking system is in place, that provides the location of the malicious drone. Instead, in the proposed approach the problem of detecting, tracking, and jamming the malicious drone is investigated jointly.

More recently, the works in [29]–[32], investigate the problem of target jamming by multiple agents, and they propose centralized and distributed decision and control algorithms which can be used to tackle this problem. More specifically, in [29] the authors propose a centralized control approach where the jamming power level and mobility control actions of a team of pursuer agents are jointly decided

in such way so that the jamming power received at the malicious drone is maximized, whereas in [30] a distributed control approach is proposed for the same problem. The works in [29], [30], however, tackle the problem in obstacle-free 2D environments with simplified discrete kinematic models. In addition, compared to the proposed approach, in which the problem is solved optimally over a rolling finite planning horizon, the approaches presented in [31], [32] are myopic greedy one-step ahead planners.

In summary, compared to the existing state-of-the-art techniques, in this work a novel distributed estimation and control approach is proposed for the problem of concurrently tracking-and-jamming a malicious target with a team of distributed mobile agents. The problem is investigated in challenging conditions i.e., 3D environment with obstacles, noisy dynamic and measurement models, and limited sensing range for target detection. Distributed model predictive control (DMPC) is used as the multi-agent control method and distributed stochastic filtering as the target state estimation method. The proposed approach enables the pursuer agents to optimally decide their control actions online over a finite planning horizon, so as to maximize the team's collective objective, while at the same time taking into consideration jamming interference constraints amongst them. Finally, the proposed formulation can be solved exactly and efficiently using off-the-shelf MIQP solvers.

3 PROBLEM DEFINITION AND SYSTEM OVERVIEW

An overview of the proposed system architecture is illustrated in Fig. 2. As shown in the figure, each agent $j \in \mathcal{S}$ maintains a probability distribution $p_t^j = p_t^j(x_t | y_{1:t}^j)$ which accounts for the uncertainty in the target's state x_t (i.e., the position and velocity) given target measurements $y_{1:t}^j$ up to time t . The agents use stochastic filtering to recursively compute in time the posterior distribution of the target state using the time-prediction and measurement-update steps as shown in the figure. More specifically, each agent j uses its current belief of the target state in order to make predictions on the future target trajectory $\hat{x}_{t+\tau+1|t}^j, \tau \in [0, \dots, T-1]$ over a rolling finite planning horizon of length T time steps, where the notation $t'|t$ indicates the predicted target state at time t' which is made at time t . Subsequently, the predicted target trajectories are fused and are used in the distributed model predictive tracking-and-jamming controller as shown in the figure.

The problem of concurrently tracking-and-jamming a malicious drone with multiple pursuer agents is formulated

as a distributed model predictive control (DMPC) problem [16], [17], where at each sampling time t the agent control actions $V_t^j = \{v_{t|t}^j, \dots, v_{t+T-1|t}^j\}, \forall j$ are obtained over a rolling-horizon of length T time steps, by solving a finite horizon open-loop optimal control problem using the current state of the agent $s_{t|t}^j$ as the initial state. The first control action $v_{t|t}^j$ in the sequence is then applied to agent j and the optimization is repeated for the next sampling time. Once agent j applies its first control action in the sequence $v_{t|t}^j$, it moves to its new state where it receives the new target measurement y_{t+1}^j and computes the posterior distribution of the target state p_{t+1}^j . The local densities are then fused using covariance intersection and the final estimated target state is extracted from the fused density. The agents then use the fused target density to make predictions for the next planning horizon.

The distributed concurrent tracking-and-jamming controller shown in Fig. 2 follows a sequential processing procedure [33], where each agent j computes its own plan $s_{t+\tau+1|t}^j, \tau \in [0, \dots, T-1]$ sequentially, one after the other. To compute their plans, the agents collaborate by exchanging information in a sequential coordinated fashion where each agent j : (a) receives the latest plans from all agents $i \in [1, \dots, j-1]$ earlier in the sequence which have already computed their plans and (b) receives the predicted plans from all agents $i \in [j+1, \dots, |\mathcal{S}|]$ later in the sequence, which have not yet computed their latest plans.

Thus, with the proposed tracking-and-jamming controller the agents collaborate in computing their plans, such that the collective tracking-and-jamming objective over the planning horizon is maximized, and at the same time the jamming interference constraints are satisfied (i.e., the jamming interference amongst the team of pursuer agents is minimized). To summarize, the problem addressed in this work can be stated as follows: *Find the optimal control actions $V_t^j = \{v_{t+\tau|t}^j : \forall \tau \in [0, \dots, T-1]\}$ for all agents $j \in \mathcal{S}$, such that the collective task of concurrently tracking-and-jamming the malicious target is maximized, while at the same time the jamming interference between the agents is minimized.*

4 SYSTEM MODEL

A proof-of-concept real-world implementation of a single UAV-based pursuer counter-drone system with detection, tracking and jamming capabilities was demonstrated in our previous work [20]. This enabled us to gain key-insights in this application-domain, and propose in this work a distributed estimation and control framework for jamming an aerial target with multiple UAV agents. Based on the experiences we have gained from the implementation of a real-world counter-drone system, in this section we outline our modeling assumptions for the problem tackled in this work.

4.1 Target Dynamical Model

In this work, it is assumed that a single aerial target is maneuvering in three dimensional space according to the following stochastic, state-space [34], [35] model:

$$x_t = \Phi x_{t-1} + \Gamma u_{t-1} + \omega_{t-1} \quad (1)$$

where $x_t = [x, \dot{x}]_t^\top \in \mathbb{R}^6$ denotes the state of the target at time t which consists of position $x_t = [p_x, p_y, p_z]_t \in \mathbb{R}^3$ and velocity $\dot{x}_t = [v_x, v_y, v_z]_t \in \mathbb{R}^3$ components in 3D Cartesian coordinates. Furthermore, it is assumed that the target can change its course and speed via input force $u_t = [u_x, u_y, u_z]_t^\top \in \mathbb{R}^3$, and that ω_t is a Gaussian process noise which represents the model uncertainty [36], and which is distributed according to a zero mean multivariate Gaussian distribution with covariance matrix Q i.e., $\omega_t \sim \mathcal{N}(0, Q)$. Moreover, it is assumed that the target's initial state is distributed according to $\mathcal{N}(\mu_0, P_0)$. The matrices Φ and Γ are further given by:

$$\Phi = \begin{bmatrix} I_{3 \times 3} & \Delta t \cdot I_{3 \times 3} \\ 0_{3 \times 3} & \phi \cdot I_{3 \times 3} \end{bmatrix}, \Gamma = \begin{bmatrix} 0_{3 \times 3} \\ \gamma \cdot I_{3 \times 3} \end{bmatrix} \quad (2)$$

where Δt denotes the sampling time interval, $I_{3 \times 3}$ and $0_{3 \times 3}$ are the identity matrix and zero matrix of dimension 3×3 , respectively, and parameters ϕ and γ are given by $\phi = (1 - \eta)$ and $\gamma = \frac{\Delta t}{m^{\text{target}}}$, where η is used to model the air resistance coefficient and m^{target} denotes the mass of the target.

4.2 Agent Dynamical Model

A team $\mathcal{S} = \{s^1, \dots, s^{|\mathcal{S}|}\}$ of pursuer mobile agents operate inside the 3D surveillance region with the ultimate purpose of tracking-and-jamming the target. The total number of pursuer mobile agents is assumed to be known, fixed, and equal to $|\mathcal{S}|$, where the operator $|\cdot|$ denotes the set's cardinality. The agent dynamics are given by the following linear, discrete-time dynamical model:

$$s_t^j = \Phi s_{t-1}^j + \Gamma v_{t-1}^j, \forall j \in [1, \dots, |\mathcal{S}|] \quad (3)$$

where $s_t = [s, \dot{s}]_t^\top \in \mathbb{R}^6$ denotes the state of the agent at time t which consists of position $s_t = [p_x, p_y, p_z]_t \in \mathbb{R}^3$ and velocity $\dot{s}_t = [v_x, v_y, v_z]_t \in \mathbb{R}^3$ components in 3D Cartesian coordinates, and $v_t \in \mathbb{R}^3$ denotes the applied control input at time t . The matrices Φ and Γ are given by Eq. (2). Without loss of generality, it is assumed in this work that all pursuer agents have a mass equal to m^{agent} and, in general, the pursuer agents have equal capabilities, although this assumption is not strict and can be relaxed. With the proposed approach we aim to find the optimal mission-level agent control inputs according to the linear dynamical model of Eq. (3), essentially generating the reference trajectory to be executed by the on-board UAV controller [37], [38], which can take into account the UAV's low-level controls and aerodynamic behavior.

4.3 Agent Sensing and Jamming Model

The pursuer agents are equipped with an onboard omnidirectional active-sensing 3D radar and jamming system [20] which they use in order to detect the aerial target, acquire target measurements, and transmit power [19] to the target, thus radio-jamming its navigation circuitry. In this work, it is assumed that the task of uninterrupted radio jamming will force the target to execute its fail-safe protocols [18] and thus terminate its mission. This aforementioned fail-safe functionality assumption is motivated by the fact that the majority of non-military grade drones i.e., off-the-shelf consumer drones nowadays implement a

certain set of fail-safe procedures (e.g., the auto-land safety protocol when the drone loses its navigation link or its battery is running critically low or the return-to-home fail-safe protocol which is activated when the drone loses its connection with the remote controller) due to increased concerns for public safety. The auto-land fail-safe protocol was indeed tested and verified experimentally through our previous work in [20] and is also documented in other related works [18], [19]. We should mention here that the proposed approach can be extended with a directional jamming antenna with slight modifications. The agent's active-sensing 3D radar/jamming system exhibits the following characteristics.

4.3.1 Sensing Profile

Agent j with state s_t^j exhibits a radio sensing profile \mathcal{A}^j in 3D which is modeled as a sphere with Cartesian coordinates (x_t, y_t, z_t) given by:

$$\begin{aligned} x_t^j &= p_{t,x}^j + R^j \sin(\theta) \cos(\phi) \\ y_t^j &= p_{t,y}^j + R^j \sin(\theta) \sin(\phi) \\ z_t^j &= p_{t,z}^j + R^j \cos(\theta) \end{aligned}$$

where $\theta \in [0, \pi]$, $\phi \in [0, 2\pi]$, and $s_t^j = Hs_t^j = [p_x^j, p_y^j, p_z^j]_t^\top$ is the agent's position at time t , with H being a matrix which extracts the position coordinates from the agent's state vector s_t^j . Finally, R^j is the radius of the sensing profile and denotes the agent's radar and jamming range. Thus, a target with position coordinates Hx_t resides inside agent's j sensing range (i.e., notation $Hx_t \in \mathcal{A}^j$ is used) when $\|Hx_t - Hs_t^j\|_2 \leq R^j$.

4.3.2 Measurement Acquisition Model

Each pursuer mobile agent j uses its active-sensing radar system to acquire relative 3D target measurements [39] at every time step, composed of radial distance ρ , azimuth angle θ , and inclination angle ϕ , i.e., $y_t = [\rho, \theta, \phi]_t^\top \in \mathbb{R}^3$, according to the measurement model:

$$y_t^j = h(x_t, s_t^j) + w_t^j \quad (4)$$

with $h(x_t, s_t^j)$ given by:

$$\left[\left\| Hx_t - Hs_t^j \right\|_2, \tan^{-1}\left(\frac{\Delta_y}{\Delta_x}\right), \tan^{-1}\left(\frac{\sqrt{\Delta_x^2 + \Delta_y^2}}{\Delta_z}\right) \right]^\top$$

where Δ_x , Δ_y , and Δ_z are the differences between the target's and the agent's x, y , and z position coordinates respectively, and $w_t^j \sim \mathcal{N}(0, \Sigma)$ denotes the zero mean Gaussian measurement noise with covariance matrix Σ , which models the radar's measurement acquisition process imperfections [36], [39]. It should be noted here that target measurements are received only when the target is detected inside the agent's sensing range.

4.3.3 Target Detection and Jamming Model

The pursuer agents use their onboard active-sensing 3D radar and jamming system to detect-and-jam the target. More specifically, it is assumed that a target detection can occur only when the target resides inside the agent's radar

sensing range R . When this happens, the agent acquires target measurements as previously discussed. At the same time, when the target resides inside the agent's sensing range R , the target can also be jammed. It should be noted here that the processes of target detection and jamming do not affect each other i.e., they operate on different channels. Hence, it is important to emphasize that the target detection and jamming occurs concurrently when the target resides inside the agent's sensing profile \mathcal{A} . More specifically, a target detection-and-jamming event $\Xi_t^j = \Xi_t^j(x_t, s_t^j)$, by agent j with state s_t^j for a target with state x_t occurs when:

$$\Xi_t^j(x_t, s_t^j) = \begin{cases} 1 & \text{when } Hx_t \in \mathcal{A}^j \\ 0 & \text{o.w} \end{cases}$$

4.4 Obstacle Model

The obstacles inside the surveillance area are represented in this work as rectangular cuboids of various sizes. A rectangular cuboid is a convex hexahedron in the three-dimensional space which exhibits six rectangular faces, where each pair of adjacent faces meets in a right angle. A 3D point $\mathbf{x} = [x, y, z]^\top \in \mathbb{R}^3$ that belongs to the cuboid \mathcal{C} satisfies the following system of linear inequalities:

$$\begin{aligned} a_{11}x + a_{12}y + a_{13}z &\leq b_1 \\ a_{21}x + a_{22}y + a_{23}z &\leq b_2 \\ &\vdots \\ a_{n1}x + a_{n2}y + a_{n3}z &\leq b_n \end{aligned}$$

where $n = 6$ is the total number of faces which compose the cuboid \mathcal{C} , $\alpha_i = [a_{i1}, a_{i2}, a_{i3}]$ is the outward normal vector of the i^{th} face of the cuboid, and b_i is a constant resulting from the dot product between α_i and a known point on the plane that contains the i^{th} face. In more compact form, the expression above can be written as $Ax \leq B$, where A is a $n \times 3$ matrix, \mathbf{x} is a 3×1 column vector, and B is a $n \times 1$ column vector. That said, an obstacle in the environment is avoided by an agent j (or the target) with state s_t^j when the following condition is true:

$$\exists i \in [1, \dots, n] : \text{dot}(\alpha_i, Hs_t^j) > b_i \quad (5)$$

where $\text{dot}(\alpha, \beta)$ denotes the dot product between the vectors α and β . It should be mentioned here that the aforementioned obstacle avoidance technique is not limited only to cuboid-like obstacles and can be used for any kind of convex polyhedra. On the other hand, one way to handle a non-convex obstacle, is by embedding it into the minimum bounding convex set. Although this approach would generate conservative collision avoidance trajectories, it fits within the proposed MIQP framework which has the potential to be solved to optimality with existing optimization tools. Finally, as discussed in Sec. 6, since the agent trajectories are computed at each time-step in a receding horizon fashion, dynamic obstacle detection functionality can be easily integrated into the proposed controller. Therefore, information regarding the obstacles do not need to be known a-priori; and can be obtained in an online fashion and processed within each successive planning horizon.

5 TARGET TRAJECTORY

The target's planned trajectory over a finite planning horizon of length T is denoted as $X_t = \{x_{t+\tau+1|t} : \tau \in [0, \dots, T-1]\}$, where the notation $x_{t'|t}$ is used to denote the predicted target state at time t' generated when the target is actually at time t . The target trajectory X_t is the solution to the MIQP problem (P1), as discussed next. Essentially, it is assumed in this work that the target's plan is to move as quickly as possible to a designated goal region, while avoiding the obstacles in its path. This is formulated as a rolling-horizon model predictive control problem with linear and binary constraints and is solved using mixed integer quadratic programming (MIQP) by the target's controller.

Problem (P1) finds the target control inputs $U_t = \{u_{t+\tau|t} : \tau \in [0, \dots, T-1]\}$ which guide the target towards a known designated goal region \mathcal{G} , which is represented by a cuboid $\mathcal{C}_{\mathcal{G}}$ with centroid $\mathcal{G}_o \in \mathbb{R}^3$. To achieve this, the target's controller minimizes the objective function:

$$\begin{aligned} \min_{U_t} \mathcal{J}_{\text{target}}(X_t, U_t) = & \|Hx_{t+T|t} - \mathcal{G}_o\|_2^2 \\ & + \lambda \sum_{\tau=1}^{T-1} \|u_{t+\tau|t} - u_{t+\tau-1|t}\|_2^2 \end{aligned} \quad (6)$$

where the first term drives the predicted target position at time-step $t+T|t$ towards the goal region and the second term in the objective function, i.e., $\sum_{\tau=1}^{T-1} \|u_{t+\tau|t} - u_{t+\tau-1|t}\|_2^2$, is used to minimize the deviations between consecutive control inputs, thus leading to smooth trajectories which can be followed by the drone's controller. The term λ is a tuning weight which controls the emphasis given to trajectory smoothing. The value of λ can be fitted to data or can be tuned to match the desired trajectory behavior.

Problem (P1): Target Maneuvering

$$\min_{U_t} \mathcal{J}_{\text{target}}(X_t, U_t) \quad (7a)$$

subject to $\tau \in [0, \dots, T-1]$:

$$x_{t+\tau+1|t} = \Phi x_{t+\tau|t} + \Gamma u_{t+\tau|t} \quad \forall \tau \quad (7b)$$

$$A_{\psi,l} H x_{t+\tau+1|t} > B_{\psi,l} - M z_{\tau,\psi,l} \quad \forall \tau, \psi, l \quad (7c)$$

$$\sum_{l=1}^L z_{\tau,\psi,l} \leq L-1 \quad \forall \tau, \psi \quad (7d)$$

$$z_{\tau,\psi,l} \in \{0, 1\} \quad \forall \tau, \psi, l \quad (7e)$$

$$x_{t|t} = x_{t|t-1} \quad (7f)$$

$$|\dot{x}_{t+\tau+1|t}| \leq v_{\max}^{\text{target}} \quad \forall \tau \quad (7g)$$

$$|u_{t+\tau+1|t}| \leq u_{\max}^{\text{target}} \quad \forall \tau \quad (7h)$$

As shown in (P1), Eq. (7b) reflects the target dynamical model and the constraints in Eqs. (7c)-(7d) define the collision avoidance constraints with all obstacles $\psi \in \Psi$ in the environment. More specifically, the target avoids a collision with an obstacle $\psi \in \Psi$ when:

$$Hx_{t+\tau+1|t} \notin \mathcal{C}_{\psi}, \quad \forall \psi \in \Psi, \forall \tau \in [0, \dots, T-1] \quad (8)$$

where \mathcal{C}_{ψ} denotes the cuboid representation of obstacle ψ . Suppose that all obstacles to be avoided contain $L = 6$ faces, then at time τ a collision with obstacle ψ is avoided if

$\exists l \in [1, \dots, L] : A_{\psi,l} H x_{t+\tau+1|t} > B_{\psi,l}$, where $A_{\psi,l}$ is the outward normal vector to the l th face of the cuboid which represents the obstacle ψ and $B_{\psi,l}$ is a constant. In essence $A_{\psi,l}$ and $B_{\psi,l}$ define the equation of the plane which contains the l th face of the cuboid (i.e., see Sec. 4.4). Therefore, the constraint in Eq. (7c) uses $T \times |\Psi| \times L$ binary variables $z_{\tau,\psi,l}$ to check if the constraint $A_{\psi,l} H x_{t+\tau+1|t} > B_{\psi,l}$ is violated, where M is a big positive constant. Then, constraint (7d) counts the number of times $z_{\tau,\psi,l}$ is activated and ensures that this number is less or equal than $L-1$. Equivalently, the target control inputs are chosen so that the linear system of inequalities $A_{\psi} H x_{t+\tau+1|t} < B_{\psi}, \forall \psi$ is not satisfied, thus achieving obstacle avoidance. Finally, constraints (7g) and (7h) put an upper bound on the target speed and applied force, respectively. In summary, problem (P1) is used by the target to navigate from its position to the goal location while avoiding collisions with the obstacle in its path.

5.1 Target State Prediction

In this work, it is assumed that the agents have the capability of predicting the target's trajectory in the near future (i.e., over a finite planning horizon) with some associated uncertainty. Specifically, from an agent's perspective, the target trajectory $X_t = \{x_{t+\tau+1|t}\}$ for $\tau \in [0, \dots, T-1]$ is a stochastic process, governed by Eq. (1), where each future state $x_{t+\tau+1|t}$ is distributed according to $x_{t+\tau+1|t} \sim \mathcal{N}(\mu_{t+\tau+1|t}, P_{t+\tau+1|t})$ with $\mu_{t+\tau+1|t}$ and $P_{t+\tau+1|t}$ given by:

$$\mu_{t+\tau+1|t} = \Phi^{\tau+1} \mu_t + \sum_{i=0}^{\tau} \Phi^i \Gamma u_{t-i} \quad (9a)$$

$$P_{t+\tau+1|t} = \sum_{i=0}^{\tau} \Phi^i Q (\Phi^{\tau})^i + \Phi^{\tau+1} P_t (\Phi^{\tau})^{\tau+1} \quad (9b)$$

As discussed next in Sec. 5.2, when the agents detect the target inside their sensing range, they receive target measurements which they use to compute the posterior distribution of the target state i.e., $\mathcal{N}(\mu_t, P_t)$ for the current time-step t , using stochastic filtering. We should also note that the agents use Eq. (9) to precompute the predicted future distribution of the target's trajectory over the planning horizon i.e., $\mathcal{N}(\mu_{t+\tau+1|t}, P_{t+\tau+1|t}), \forall \tau$ and subsequently use the estimated expected target state $\hat{X}_t = \{\hat{x}_{t+\tau+1|t}\} = \{\mu_{t+\tau+1|t}\}, \tau \in [0, \dots, T-1]$ to plan their future states.

5.2 Target State Estimation

When the pursuer agents detect the target inside their sensing range, they receive target measurements as discussed in Sec. 4.3, which they use to improve their estimation of the target state. To achieve this, each mobile agent $j \in \mathcal{S}$ maintains a local Bayes filter [40] which is used to recursively estimate and propagate in time the posterior probability distribution $p_t(x_t | y_{1:t})$ of the target's state x_t given all target measurements $y_{1:t} = [y_1, \dots, y_t]$ up to the current time-step t . As previously mentioned, the target's state x_t follows a first order Markov process, i.e., Eq. (1), with transition density $f_{t|t-1}(x_t | x_{t-1}) = \mathcal{N}(\Phi x_{t-1} + \Gamma u_{t-1}, Q)$, which determines the probability density that the target with state

x_{t-1} at time $t-1$ will move to state x_t at the next time step. The Bayes filter recursion is given by:

$$p_{t|t-1}(x_t|y_{1:t-1}) = \int f_{t|t-1}(x_t|x)p_{t-1}(x|y_{1:t-1})d_x \quad (10a)$$

$$p_t(x_t|y_{1:t}) = \frac{g_t(y_t|x_t, s_t)p_{t|t-1}(x_t|y_{1:t-1})}{\int g_t(y_t|x_t, s_t)p_{t|t-1}(x_t|y_{1:t-1})d_x} \quad (10b)$$

where the function $g_t(y_t|x_t, s_t)$ is the measurement likelihood function according to Sec. 4.3 and determines the probability density that the agent with state s_t will receive measurement y_t from the target with state x_t , if the target is detected. Therefore $g_t(y_t|x_t, s_t)$ is given by:

$$g_t(y_t|x_t, s_t) = [1 - \Xi_t(x_t, s_t)] + \Xi_t(x_t, s_t)\mathcal{N}(h(x_t, s_t), \Sigma). \quad (11)$$

With that said, Eq. (10a), which is referred to as the time-prediction step, uses the transitional density and the prior belief on the target's state to compute the target's predictive density $p_{t|t-1}(x_t|y_{1:t-1})$ before receiving the latest measurement y_t . Subsequently, Eq. (10b) i.e., the measurement-update step, incorporates the latest target measurement y_t and computes the posterior distribution of the target state, from which a point estimate of the target state can be extracted using the expected a posteriori (EAP) or maximum a posteriori (MAP) estimators as discussed in more detail in Sec. 6.2. The reader should note that in the preceding discussion the agent index j was omitted for notational clarity.

Each agent j uses the filtering procedure described above to estimate at each time step the posterior distribution of the target state, i.e., $p_t^j(x_t|y_{1:t}) = \mathcal{N}^j(\mu_t, P_t)$, which is used as the initial belief for predicting the target trajectory over the planning horizon as discussed in Sec. 5.1, where μ_t^j is the EAP point estimate of the target state and P_t^j its covariance matrix.

Subsequently, the collection $\{(\mu_t^j, P_t^j)\}, j \in \mathcal{S}$ of local target state estimates is combined at each time step using covariance intersection [41] to produce the fused posterior distribution of the target's state. More specifically, in this step the agents cooperate in a sequential fashion by exchanging their local estimates and computing recursively the fused target state as follows: Agent 1 first transmits its local estimate (μ_t^1, P_t^1) to agent 2 which in turn combines the received data with its local data, i.e., (μ_t^2, P_t^2) , to compute the fused target distribution $\mathcal{N}(\tilde{\mu}_t^1, \tilde{P}_t^1)$ between agents 1 and 2 as follows:

$$(\tilde{P}_t^1)^{-1} = \omega(P_t^1)^{-1} + (1 - \omega)(P_t^2)^{-1} \quad (12a)$$

$$(\tilde{P}_t^1)^{-1}\tilde{\mu}_t^1 = \omega(P_t^1)^{-1}\mu_t^1 + (1 - \omega)(P_t^2)^{-1}\mu_t^2 \quad (12b)$$

where $\omega \in [0, 1]$ is a weighting coefficient [42] and is found by solving the optimization problem below:

$$\arg \min_{\omega \in [0, 1]} \text{tr} \left\{ [\omega(P_t^1)^{-1} + (1 - \omega)(P_t^2)^{-1}]^{-1} \right\} \quad (13)$$

where $\text{tr}\{\cdot\}$ denotes the trace of a matrix. Next, agent 2 transmits the fused estimate $(\tilde{\mu}_t^1, \tilde{P}_t^1)$ to agent 3 and the covariance intersection procedure shown in Eqs. (12)-(13) is repeated to produce the fused estimate $(\tilde{\mu}_t^2, \tilde{P}_t^2)$ by combining agent's 3 data, i.e., (μ_t^3, P_t^3) . The procedure is repeated until the final fused target distribution $\mathcal{N}(\tilde{\mu}_t^{|\mathcal{S}|}, \tilde{P}_t^{|\mathcal{S}|})$ is

computed after incorporating the data from the last agent, i.e., $(\mu_t^{|\mathcal{S}|}, P_t^{|\mathcal{S}|})$. The parameters of the fused posterior target distribution $(\tilde{\mu}_t^{|\mathcal{S}|}, \tilde{P}_t^{|\mathcal{S}|})$ are then broadcasted to all agents, which they use as a prior belief for the next time step. We should mention here that in this work we assume that any pre-existing wireless communication technology (e.g., WiFi, ZigBee, Bluetooth) can be used to handle the exchange of information and communication between the agents. Subsequently, low-level communication aspects such as multi-path fading and shadowing effects are assumed to be handled by the wireless communication module.

6 DISTRIBUTED MPC FOR TRACKING AND JAMMING

This section develops a DMPC algorithm for the cooperative guidance of the team of pursuer UAV agents for the purpose of concurrently tracking-and-jamming the malicious target, while at the same time minimizing the jamming interference among the pursuer agents. Instead of solving a large centralized optimization problem, where all the necessary information is transmitted to a central station which in turn decides the control actions for each agent, this optimization problem is decomposed into smaller sub-problems that can be solved autonomously within each agent. The proposed distributed tracking-and-jamming approach ensures that the control actions taken by one agent are consistent with the control actions of all other agents in the team and that the coupled jamming interference constrains between the pursuer agents are taken into account during the decision making process.

6.1 Centralized Control

The centralized version of the problem, as shown in (P2) below, is initially described in a high-level form, followed by a detailed discussion of the proposed DMPC tracking-and-jamming approach.

Problem (P2): Centralized MPC

$$\max_{V_t^1, \dots, V_t^{|\mathcal{S}|}} \mathcal{J}_{\text{centralized}}(\hat{X}_t, S_t^1, \dots, S_t^{|\mathcal{S}|}, V_t^1, \dots, V_t^{|\mathcal{S}|}) \quad (14a)$$

subject to $\forall j \in \{1, \dots, |\mathcal{S}|\}, \tau \in [0, \dots, T-1]$:

$$s_{t+\tau+1|t}^j = \Phi s_{t+\tau|t}^j + \Gamma v_{t+\tau|t}^j \quad \forall \tau \quad (14b)$$

$$s_{t|t}^j = s_{t|t-1}^j \quad (14c)$$

$$s_{t+\tau+1|t}^j \notin \mathcal{C}_\psi \quad \forall \psi \quad (14d)$$

$$|s_{t+\tau+1|t}^j| \leq v_{\max}^{\text{agent}} \quad \forall \tau \quad (14e)$$

$$|v_{t+\tau+1|t}^j| \leq v_{\max}^{\text{agent}} \quad \forall \tau \quad (14f)$$

$$H s_{t+\tau+1|t}^i \notin \mathcal{A}^j \quad \forall i \neq j \in \mathcal{S} \quad (14g)$$

6.1.1 Objective Function

Problem (P2) optimizes a system-wide objective function i.e., Eq. (14a), over the planning horizon of length T with $\tau \in [0, \dots, T-1]$ for the joint control inputs over all $|\mathcal{S}|$ agents. In (P2), agent's j planned trajectory is denoted

as $S_t^j = \{s_{t+\tau+1|t}^j : \forall \tau\}$ and future control inputs as $V_t^j = \{v_{t+\tau|t}^j : \forall \tau\}$. The objective in Eq. (14a) is also a function of the predicted target trajectory $\hat{X}_t = \{\hat{x}_{t+\tau+1|t} : \forall \tau\}$, since the focus is on finding the control inputs over all pursuer agents that will enable them to track-and-jam the target. The predicted target trajectory \hat{X}_t over the planning horizon has been generated by fusing the local predictions over all agents as discussed in Sec. 5.2.

In order to track-and-jam the target, the agents must be able to detect the target inside their sensing range as explained in Sec. 4.3. This will allow the agents to: (a) emit power to the target in order to jam it and (b) acquire target measurements which they can use to estimate more accurately the target's state, thus improving their tracking performance. Let us assume that the system-wide objective function shown in Eq. (14a) returns the total number of detection-and-jamming events over the planning horizon for all agents. Thus, whenever the condition $H\hat{x}_{t+\tau+1|t} \in \mathcal{A}_{t+\tau+1|t}^j, \forall \tau, \forall j$ is true, the value of the objective function is increased. Therefore, in order to maximize the number of detection-and-jamming events over the planning horizon and for all agents it suffices to maximize Eq. (14a). From the discussion above, it is clear that the overall system has decoupled agent dynamics and that the objective function in Eq. (14a) is separable and can be written as:

$$\sum_{j=1}^{|S|} \mathcal{J}_{\text{agent}}^j(\hat{X}_t, S_t^j, V_t^j) = \sum_{j=1}^{|S|} \sum_{\tau=0}^{T-1} \Xi_t^j(\hat{x}_{t+\tau+1|t}, s_{t+\tau+1|t}^j) \quad (15)$$

where function $\Xi_t^j(\cdot)$ was defined in Sec. 4.3. With this decomposition, each agent decides only its own control inputs when optimizing its local objective. Thus, the agents can work in a distributed fashion optimizing their local objective, while at the same time the system-wide objective is also optimized. Then, Eqs. (14b)-(14c) are constraints imposed by the agent dynamical model assuming a known initial state, Eq. (14d) defines obstacle avoidance constraints, and constraints (14e) and (14f) keep the agent within the allowable speed limit and control input values, respectively.

6.1.2 Jamming Interference Constraints

Constraint (14g) ensures that there is no jamming interference between the team of pursuer agents. Agent j is jamming some other agent i at time t when $Hs_t^i \in \mathcal{A}_t^j$, i.e., when agent i resides inside the sensing range of agent j at time t . Thus, to avoid the interference between the agents it is required that $Hs_{t+\tau+1|t}^i \notin \mathcal{A}^j, \forall i \neq j, \forall j$ or equivalently:

$$\left\| Hs_{t+\tau+1|t}^i - Hs_{t+\tau+1|t}^j \right\|_2 > R^j, \forall \tau, \forall i \neq j, \forall j. \quad (16)$$

The constraints in Eq. (16), however, are very challenging to be applied directly, since they create an overall non-convex feasibility region which renders their use impractical, even for specialized solvers. To overcome this problem, the jamming interference constraints of Eq. (16) are convexified and approximated by a set of linear constraints which can be handled efficiently by off-the-shelf MQIP solvers. More specifically, the sensing profile of agent j , i.e., a sphere with radius R^j , is approximated in this work by a regular dodecahedron D^j with circumradius R^j . That is, the circumscribed sphere with radius R^j of the dodecahedron is

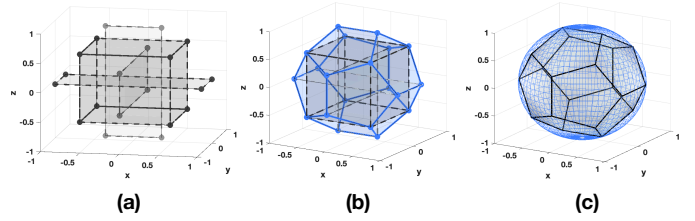


Fig. 3. (a)-(b) The 20 vertices of a regular dodecahedron (centered at the origin with $R = 1$) formed from the vertices of a cube and 3 planes according to Eq. (17), (c) The agent's sensing area i.e., a dodecahedron with circumradius R .

a sphere that contains the dodecahedron and touches each of the dodecahedron's vertices. A regular dodecahedron centered at the origin exhibits 12 regular pentagonal faces and 20 vertices. These 20 vertices are formed as the union of the vertices of a cube and the vertices of 3 rectangles on the yz , xz , and xy planes, with Cartesian coordinates given by (shown in Fig. 3):

$$\text{Cube} : \left(\pm \frac{R}{\sqrt{3}}, \pm \frac{R}{\sqrt{3}}, \pm \frac{R}{\sqrt{3}} \right) \quad (17a)$$

$$\text{Rectangle on } yz : \left(0, \pm \frac{\varphi R}{\sqrt{3}}, \pm \frac{R}{\varphi\sqrt{3}} \right) \quad (17b)$$

$$\text{Rectangle on } xz : \left(\pm \frac{R}{\varphi\sqrt{3}}, 0, \pm \frac{\varphi R}{\sqrt{3}} \right) \quad (17c)$$

$$\text{Rectangle on } xy : \left(\pm \frac{\varphi R}{\sqrt{3}}, \pm \frac{R}{\varphi\sqrt{3}}, 0 \right) \quad (17d)$$

where constant $\varphi = \frac{1+\sqrt{5}}{2}$ is the golden ratio. Similarly to the discussion of Sec. 4.4, a point $\mathbf{x} = [x, y, z]^T \in \mathbb{R}^3$ resides inside the dodecahedron if it satisfies the linear system of 12 inequalities $D\mathbf{x} \leq C$, where D is a 12×3 matrix (each row $D_l, l \in [1, \dots, 12]$ is the normal to the l^{th} face of the dodecahedron) and C is a 12×1 column vector. With the dodecahedron approximation the interference constraint in Eq. (16) can be approximated by a system of 12 linear inequalities as:

$$Hs_{t+\tau+1|t}^i \notin \mathcal{A}^j \implies \exists l \in [1, \dots, 12] : D_l(Hs_{t+\tau+1|t}^i - Hs_{t+\tau+1|t}^j) > C_l.$$

6.2 Distributed Control

Subsequently, we transform the centralized problem shown in (P2) to the distributed tracking-and-jamming controller shown in (P3), which can be implemented as an MIQP and solved using off-the-shelf solvers. The problem shown in (P3) is executed at each time step locally within each agent, in a sequential fashion [33], where agent j receives the latest plans $s_{t+\tau+1|t}^i$ from all agents $i < j$ earlier in the sequence and also receives the predicted plans $\tilde{s}_{t+\tau+1|t}^i = s_{t+\tau|t-1}^i$ from all agents $i > j$ later in the sequence, which have not yet computed their latest plans. That said, each agent j

optimizes the objective function below:

$$\min_{V_t^j} \mathcal{J}_{\text{agent}}^j(\hat{X}_t, S_t^1, \dots, S_t^{|S|}, V_t^j) = -w_1 \sum_{\tau=0}^{T-2} \tilde{b}_\tau^j + w_2 \sum_{\tau=0}^{T-2} \sum_{i \neq j=1}^{|S|} \tilde{\kappa}_{\tau,i}^j + w_3 \|s_{t+T}^j - \mathcal{G}_o\|^2 \quad (18)$$

where $\hat{X}_t = \{\hat{x}_{t+\tau+1|t} : \forall \tau\}$ is the expected target trajectory over the planning horizon $t+1|t, \dots, t+T|t$, $S_t^{i \neq j}$ are the received plans from all other agents, which contain the latest or the predicted plans depending on the position of agent j in the sequence, and $V_t^j = \{v_{t+\tau}^j : \forall \tau\}$ are the future control inputs of agent j which are to be optimized. The tracking-and-jamming objective function shown in Eq. (18) consists of three terms. The first term is responsible for the detection and jamming events and thus maximizes (i.e., minimizes the negative) the binary variable \tilde{b}_τ^j over the planning horizon (i.e., the binary variable which indicates whether the target with state $\hat{x}_{t+\tau+1|t}$ resides inside agent's j sensing range). In other words, this term implements the function Ξ_t^j as discussed in Sec. 4.3. Subsequently the second term minimizes the jamming interference events inside the planning horizon, by making use of the binary variable $\tilde{\kappa}_{\tau,i}^j$ which indicates if agent i is being jammed by agent j at time-step $t+\tau+1|t$. Finally, the last term is used to drive the agent to the goal region. Parameters w_1, w_2 , and w_3 are tuning weights which determine the preference given to each term.

Problem (P3): Distributed Control

$$\min_{V_t^j} \mathcal{J}_{\text{agent}}^j(\hat{X}_t, S_t^1, \dots, S_t^{|S|}, V_t^j) \quad (19a)$$

subject to $\tau \in [0, \dots, T-1], i \neq j \in [1..|S|]$:

$$s_{t+\tau+1|t}^j = \Phi s_{t+\tau|t}^j + \Gamma v_{t+\tau|t}^j \quad \forall \tau \quad (19b)$$

$$D_l(H\hat{x}_{t+\tau+1|t} - Hs_{t+\tau+1|t}^j) + (M - C_l)b_{\tau,l}^j < M \quad \forall \tau, l \quad (19c)$$

$$12\tilde{b}_\tau^j - \sum_{l=1}^{12} b_{\tau,l}^j \leq 0 \quad \forall \tau \quad (19d)$$

$$D_l(Hs_{t+\tau+1|t}^i - Hs_{t+\tau+1|t}^j) > C_l - M\kappa_{\tau,i,l}^j \quad \forall \tau, l, i < j \quad (19e)$$

$$D_l(H\tilde{s}_{t+\tau+1|t}^i - Hs_{t+\tau+1|t}^j) > C_l - M\kappa_{\tau,i,l}^j \quad \forall \tau, l, i > j \quad (19f)$$

$$\sum_{l=1}^{12} \kappa_{\tau,i,l}^j - 11 \leq \tilde{\kappa}_{\tau,i}^j \quad \forall \tau, i \neq j \quad (19g)$$

$$A_{\psi,l} Hs_{t+\tau+1|t}^j > B_{\psi,l} - Mz_{\tau,\psi,l} \quad \forall \tau, \psi, l \quad (19h)$$

$$\sum_{l=1}^L z_{\tau,\psi,l} \leq L - 1 \quad \forall \tau, \psi \quad (19i)$$

$$b_{\tau,l}^j, \tilde{b}_\tau^j, \kappa_{\tau,i,l}^j, \tilde{\kappa}_{\tau,i}^j, z_{\tau,\psi,l} \in \{0, 1\} \quad \forall \tau, \psi, l \quad (19j)$$

$$s_{t|t}^j = s_{t|t-1}^j \quad (19k)$$

$$|\dot{s}_{t+\tau+1|t}| \leq v_{\max}^{\text{agent}} \quad \forall \tau \quad (19l)$$

$$|v_{t+\tau+1|t}^j| \leq v_{\max} \quad \forall \tau \quad (19m)$$

In formulation (P3) of the distributed tracking-and-jamming controller, constraint (19b) is imposed by the

agent dynamical model discussed in Sec. 4.2, where constraints (19c)-(19d) are responsible for the target detection and jamming events. More specifically, the inequality $D_l(H\hat{x}_{t+\tau+1|t} - Hs_{t+\tau+1|t}^j) < C_l$ becomes true when the target's location resides inside the negative half-space created by the plane containing the l^{th} face of the dodecahedron that approximates the agent's spherical sensing range. When this happens, the corresponding binary variable $b_{\tau,l}^j$ becomes 1 to satisfy the constraint. Otherwise, $b_{\tau,l}^j = 0$ and constraint (19c) is valid with the addition of a large positive constant M (i.e., big- M). Subsequently, the binary variable \tilde{b}_τ^j in Eq. (19d) is activated when the system of linear inequalities is satisfied for all 12 faces of the dodecahedron, i.e., $D_l(H\hat{x}_{t+\tau+1|t} - Hs_{t+\tau+1|t}^j) < C_l, \forall l$ which indicates that the target's location $H\hat{x}_{t+\tau+1|t}$ resides inside agent's j sensing range at time $t+\tau+1|t$. Next, Eqs. (19e)-(19g) define the radio-jamming interference constraints amongst the team of agents. In this case, the binary variable $\kappa_{\tau,i,l}^j$ is activated when agent i resides inside the negative half-space created by the plane containing the l^{th} face of the dodecahedron which approximates agent's j sensing range. When $\sum_{l=1}^{12} \kappa_{\tau,i,l}^j$ equals 12, then agent i is being jammed by agent j at time-step $t+\tau+1|t$ (i.e., agent i is inside the dodecahedron), a scenario that must be avoided. For this reason, the binary variable $\tilde{\kappa}_{\tau,i}^j$ shown in Eq. (19g) is minimized, so that $\sum_{l=1}^{12} \kappa_{\tau,i,l}^j < 12$ and the jamming interference is avoided i.e., agent i is outside agent's j sensing range. Finally, constraints (19h)-(19i) ensure obstacle avoidance similarly to (P1) in Sec. 5, and the rest of the constraints define the decision variables and the limits on the speed and input controls of the agent.

It should also be noted that due to the non-linear measurement model, i.e., Eq. (4), the Bayes filter recursion described in Eq. (10) is implemented in this work as a particle filter [43], [44] and thus the posterior density at time $t-1$ denoted by $p_{t-1}(x_{t-1}|y_{1:t-1})$ is represented by a set of weighted particles $\{w_{t-1}^{(i)}, x_{t-1}^{(i)}\}, i \in [1, \dots, N]$:

$$p_{t-1}(x_{t-1}|y_{1:t-1}) \approx \sum_{i=1}^N w_{t-1}^{(i)} \delta(x_{t-1} - x_{t-1}^{(i)}) \quad (20)$$

where $\delta(x)$ denotes the Dirac impulse function and $\sum_{i=1}^N w_{t-1}^{(i)} = 1$. Given the proposal density $q(\cdot|x_{t-1}^{(i)}, y_{1:t})$ the posterior distribution of the subsequent time-step t is then approximated by a new set of particles $\{w_t^{(i)}, x_t^{(i)}\}$ as:

$$p_t(x_t|y_{1:t}) \approx \sum_{i=1}^N w_t^{(i)} \delta(x_t - x_t^{(i)}) \quad (21)$$

where

$$x_t^{(i)} \sim q(x_t^{(i)}|x_{t-1}^{(i)}, y_{1:t}) \quad (22a)$$

$$w_t^{(i)} = \tilde{w}_t^{(i)} \left(\sum_i \tilde{w}_t^{(i)} \right)^{-1} \quad (22b)$$

$$\tilde{w}_t^{(i)} = w_{t-1}^{(i)} \frac{g_t(y_t|x_t^{(i)}, s_t) f_{t|t-1}(x_t^{(i)}|x_{t-1}^{(i)})}{q(x_t^{(i)}|x_{t-1}^{(i)}, y_{1:t})}. \quad (22c)$$

In this work, for the proposal distribution the transition density is used, i.e., $q(x_t^{(i)}|x_{t-1}^{(i)}, y_{1:t}) = f_{t|t-1}(x_t^{(i)}|x_{t-1}^{(i)})$.

Subsequently, each agent computes the expected posteriori (EAP) target state point estimate and its associated covariance matrix as:

$$\mu_t = \sum_{i=1}^N w_t^{(i)} x_t^{(i)} \quad (23a)$$

$$P_t = \sum_{i=1}^N w_t^{(i)} (x_t^{(i)} - \mu_t)(x_t^{(i)} - \mu_t)^\top. \quad (23b)$$

To conclude, each agent implements the Bayes recursion in Eq. (10) as a particle filter, which is used to estimate the posterior distribution of the target state at each time step as discussed above. The agents cooperate by exchanging their local estimates to produce the final target distribution using covariance intersection fusion as discussed in Sec. 5, and use the distributed tracking-and-jamming controller discussed in Sec. 6 to plan their trajectories such that the target is optimally tracked-and-jammed over the rolling planning horizon with minimum jamming interference.

7 PERFORMANCE EVALUATION

7.1 Simulation Setup

The simulation setup used to evaluate the performance of the proposed approach is based on field tests conducted with the DJI M200 series drone [20]. In particular we assume that the agents and the target maneuver inside a bounded 3D surveillance area of size $300\text{m} \times 300\text{m} \times 100\text{m}$. The dynamics of the target are given by Eq. (1) with $Q = \text{diag}([1, 1, 1, 0.5, 0.5, 0.5])$, $\eta = 0.2$, $m^{\text{target}} = 3.25\text{kg}$, and $\Delta t = 1\text{s}$. The maximum target speed and control input are $v_{\text{max}}^{\text{target}} = 15\text{m/s}$ and $u_{\text{max}}^{\text{target}} = 10\text{N}$, respectively. Finally, parameter λ in Eq. (6) is set at $\lambda = 0.01$. Regarding the agents, their dynamics are given by Eq. (3), with $\eta = 0.2$ and $m^{\text{agent}} = 3.25\text{kg}$, for all agents. The covariance Σ of the measurement noise is set at $\Sigma = \text{diag}([0.5, \pi/50, \pi/50])$ and the maximum agent speed and control input are $v_{\text{max}}^{\text{agent}} = 20\text{m/s}$ and $u_{\text{max}}^{\text{agent}} = 25\text{N}$, respectively. The agent radius is $R = 20\text{m}$ for all agents and parameters w_1, w_2 , and w_3 of the objective function in Eq. (18) are $w_1 = 1, w_2 = 1$, and $w_3 = 1$. The obstacles in the environment are modeled as rectangular cuboids according to Sec. 4.4, and the target state estimation was implemented as a particle filter [43], [44], with 1500 particles. Simulations were conducted on a 2GHz dual core CPU computer running Matlab and the Gurobi solver.

7.2 Results

The first experiment aims to demonstrate the proposed approach with a tracking-and-jamming scenario taking place in an obstacle-free environment with 3 agents and 1 target. More specifically, in this setup, three agents colored blue, turquoise, and green are spawned at locations $[110, 70, 30]$, $[80, 70, 30]$, and $[80, 100, 30]$, respectively, as shown in Fig. 4(a). The target is initially distributed according to $\mathcal{N}(\mu_0, P_0)$, where $\mu = [95, 95, 30, 3, 3, 0]$ (shown with a red star) and $P_0 = \text{diag}([2, 2, 2, 0.8, 0.8, 0.8])$. The target's guidance controller executes the program shown in (P1), which is responsible for guiding the target to the goal region \mathcal{G} as shown in the figure. Figures 4(b)-(e) show the overall system state at time-step $t = 4$ with a planning

horizon $T = 5$ time steps. More specifically, Fig.4(b) shows the trajectories of the agents and the target where clear diamonds are used to denote the agent predicted states, and clear circles to denote the target's predicted states over the planning horizon. The executed trajectories are marked with solid shapes. Initially, the target does not reside inside any of the agents' sensing ranges, thus for the initial time steps the target cannot be jammed by any agent as shown in Fig. 4(c). During this time, the agents do not receive any target measurements, since the target is not detected inside their sensing ranges.

Specifically, Fig. 4(c) shows the target detection and jamming events per agent up to the current time-step $t = 4$ and over the planning horizon $t + \tau + 1 | t, \tau \in [0, T - 1]$. Similarly, Fig. 4(d) shows the jamming interference events amongst the agents, and Fig. 4(e) shows the system configuration at the current time step in a top-down view, where the distribution of the target state over the planning horizon (as shown) is approximated with a set of particles. The final target and agent trajectories of this experiment are shown in Fig. 4(f). As can be observed, the agents move in such a formation that allows them to surround the target, while at the same time avoiding the jamming interference amongst them. In this figure, the target ground truth trajectory is marked with yellow circles and the estimated trajectory is marked with orange squares. Figure 4(g) shows more clearly the agent positions at time-steps $t = [1, 3, 6, 9, 12]$, along with their sensing area which is approximated by regular dodecahedra. As can be observed, the agents position themselves in such a way so that the target is included inside their sensing area, but without causing interference to each other. The jamming interference is minimized, as shown in Fig. 4(i), with only two jamming interference events occurring by agent 2 towards agent 1 during time-steps 10 and 11. Finally, Fig. 4(j) shows the tracking error, which is defined here as the Euclidean distance between the estimated and ground truth target positions at each time step. The agents start with an initial target distribution which they use to predict the target state over the planning horizon. Because during the first few time steps the target is not detected, the agents are not receiving any measurements from the target, which leads to increased uncertainty over the target state and as a result the tracking error initially increases (as shown). However, as the objective of the proposed tracking-and-jamming controller is to maximize the tracking-and-jamming performance, the tracking error subsequently decreases once the agents intercept the target at $t = 4$, as shown in Figs. 4(h)-4(j).

The next experiment illustrates the behavior of the proposed system with the jamming interference constraints disabled. In the previous experiment, the proposed tracking-and-jamming controller was executed with a weight configuration of $(w_1 = 1, w_2 = 1)$ i.e., the objective function in Eq. (18) considers both the target tracking-and-jamming events and the agent jamming interference events through weights w_1 and w_2 , respectively. In this experiment, the jamming interference constraints was disabled by setting $w_2 = 0$ in Eq. (18) and the previous scenario was repeated (w_3 remained unchanged). The results of these simulations are depicted in Fig. 5. As it can be observed, from Fig. 5(a) the target detection-and-jamming events appear to be slightly

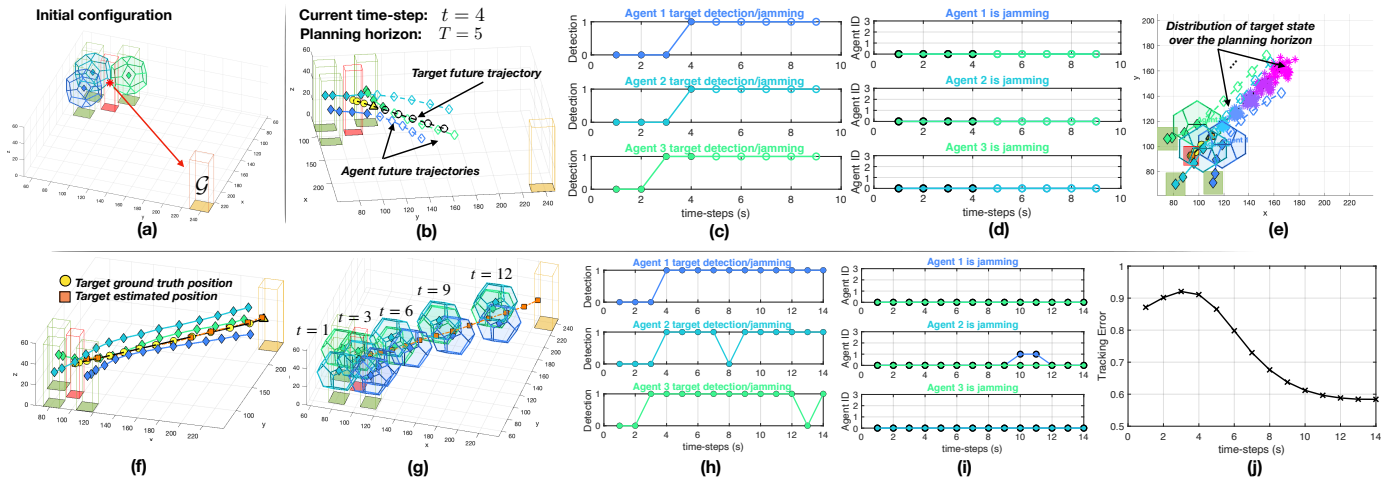


Fig. 4. Concurrent tracking-and-jamming scenario with 3 agents and 1 target (agents are marked with \blacklozenge , with agent 1, 2, and 3 colored blue, turquoise, and green respectively). The ground truth target trajectory (marked with \bullet) is colored yellow and the estimated trajectory (marked with \blacksquare) is colored orange. The clear shapes \diamond and \circ indicate the agents and target predicted states, respectively).

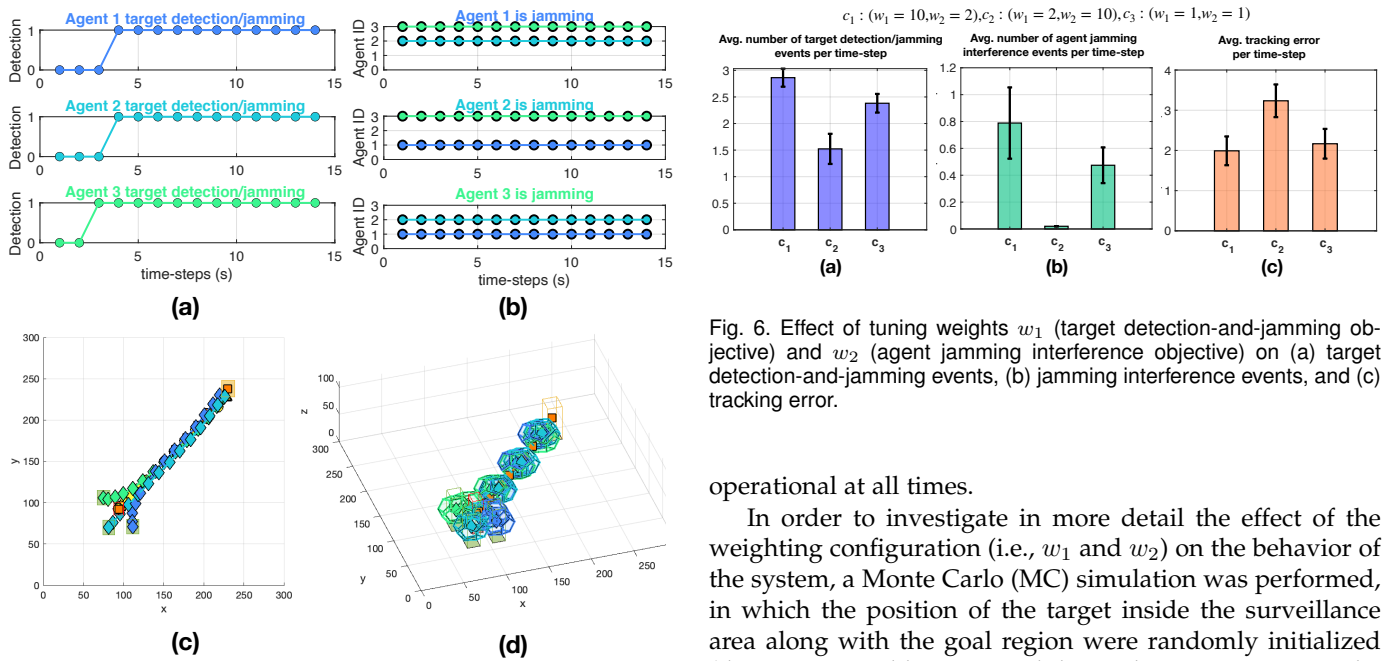


Fig. 5. Concurrent tracking-and-jamming scenario with 3 agents and 1 target with disabled jamming interference constraints. While the target detection-and-jamming objective is maximized as shown in (a), the agents come critically close to each other, causing destructive jamming interference as shown in (b)-(d).

improved compared to the previous scenario, however the jamming interference constraints are violated at every time step as shown in Fig. 5(b). In fact, Figs. 5(c)-5(d) show that the agents coalesce in their effort to maximize the target detection-and-jamming events, which causes destructive jamming interference amongst the team. Importantly, when jamming interference is accounted for (as described in the previous experiment), the proposed controller manages to achieve similar tracking-and-jamming performance as compared to the case when jamming interference is not accounted for (this experiment), while maintaining minimum jamming interference and thus keeping the system safe and

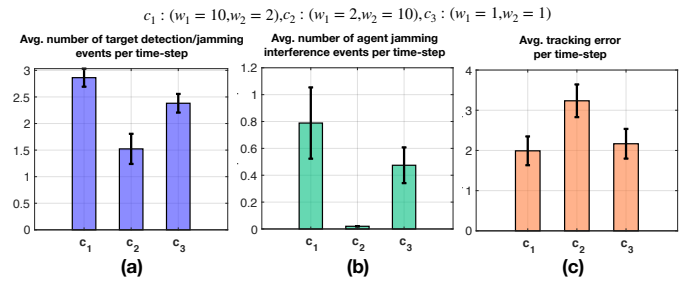


Fig. 6. Effect of tuning weights w_1 (target detection-and-jamming objective) and w_2 (agent jamming interference objective) on (a) target detection-and-jamming events, (b) jamming interference events, and (c) tracking error.

operational at all times.

In order to investigate in more detail the effect of the weighting configuration (i.e., w_1 and w_2) on the behavior of the system, a Monte Carlo (MC) simulation was performed, in which the position of the target inside the surveillance area along with the goal region were randomly initialized (the target initial location and the goal region were set to be at least 100m apart), and then the positions of 3 agents were uniformly sampled from a sphere with radius 25m centered at the target location. Fifty (50) trials were conducted with the system parameters set as discussed previously. In this experiment we compute a) the average number of target detection-and-jamming events per time step, b) the average number of agent jamming interference events per time step, and finally c) the average tracking error per time step, for 3 different weight configurations $c_1 = (w_1 = 10, w_2 = 2)$, $c_2 = (w_1 = 2, w_2 = 10)$, and $c_3 = (w_1 = 1, w_2 = 1)$. The results are shown in Fig. 6.

As can be observed, when the emphasis is on the target detection-and-jamming objective (i.e., c_1), the average number of target detection-and-jamming events is maximized as shown in Fig. 6(a); however, as Fig. 6(b) shows, the jamming interference events are also increased, since avoiding the jamming interference is not the main goal in this weight configuration. The opposite effect is obtained when the

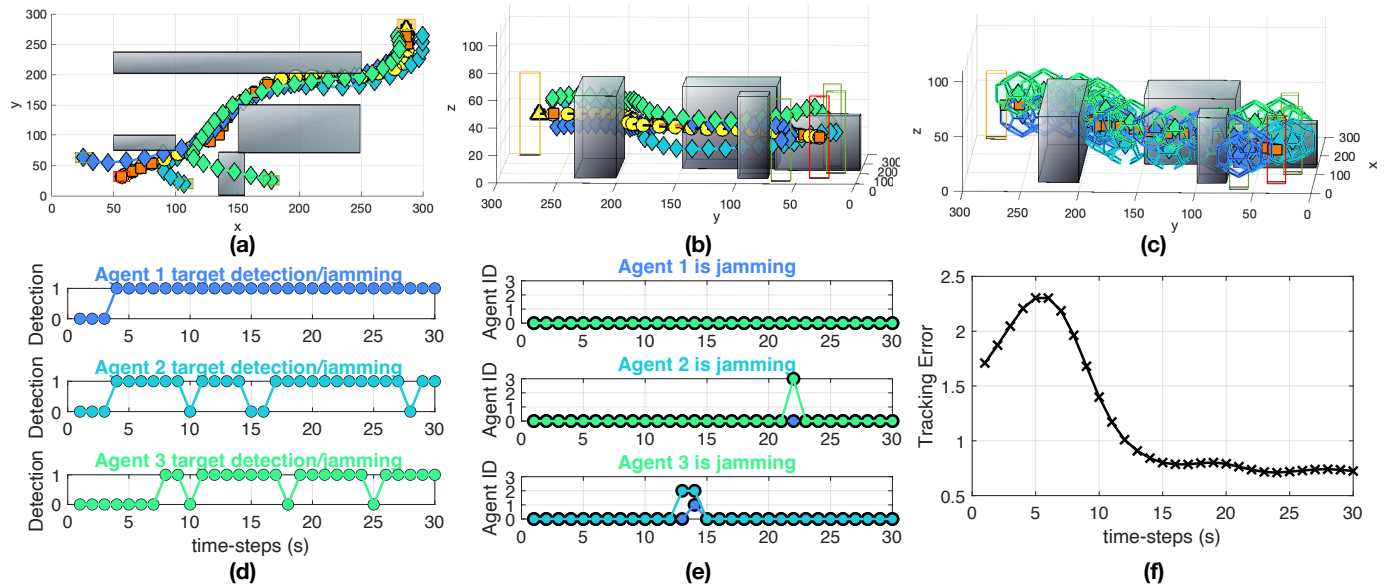


Fig. 7. A tracking-and-jamming scenario with 3 agents and 1 target in a 3D environment with obstacles. (a)-(c) Agent and target trajectories for the duration of this experiment, (d) target detection-and-jamming events per agent, (e) jamming interference events, and (f) tracking error.

emphasis is on the jamming interference objective (i.e., c_2). In this scenario, jamming interference is minimized (Fig. 6(b)); however, as in this configuration tracking the target is not the main goal, detection-and-jamming events are decreased. In contrast, configuration c_3 results in a more balanced behavior. Finally, as expected, the tracking error, shown in Fig. 6(c) for the 3 configurations, closely follows the inverse trend of the target detection-and-jamming events shown in Fig. 6(a), i.e., the more times the target is detected, the lower the tracking error.

Next, Fig. 7 illustrates a simulated cooperative tracking-and-jamming scenario with 3 agents and 1 target, taking place in a bounded 3D environment with obstacles. In this scenario, agent 1 (blue), 2 (turquoise), and 3 (green) are initialized at locations $[25, 63, 30]$, $[107, 20, 30]$, and $[117, 25, 30]$, respectively. The target is distributed according to $\mathcal{N}(\mu_0, P_0)$, where $\mu = [55, 30, 30, 3, 3, 1]$ (shown in yellow) and $P_0 = \text{diag}([2, 2, 2, 0.8, 0.8, 0.8])$, and the rest of the system parameters are set as discussed in Sec. 7.1. The goal region, marked with a clear yellow cuboid, has centroid $[286, 282, 30]$. As can be observed from Figs. 7(a)-7(b), the agents successfully track-and-jam the target, while avoiding the obstacles in their path. More specifically, as it is shown, the agents maintain a safe distance between them to avoid jamming interference, while at the same time keeping the target inside their sensing area (also shown by the target detection-and-jamming events in Fig. 7(d)). The latter allows the agents to detect the target and acquire target measurements and thus improve the target tracking accuracy. Figure 7(c) shows the agent positions and their sensing area over the duration of this experiment. As can be seen, the agents position themselves as close as possible to their peers, but without causing jamming interference, as depicted more clearly in Fig. 7(e) by the sporadic jamming interference events. Finally, Fig. 7(f) shows the tracking error for the duration of the experiment. Initially, during the time that the agents try to intercept the target, no target measure-

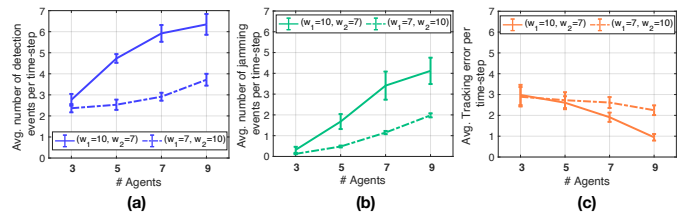


Fig. 8. System performance for different number of agents and weight (i.e., w_1 and w_2) configurations: (a) target detection-and-jamming events, (b) jamming interference events, and (c) tracking error.

ments are received, which results in increased uncertainty regarding the target state and thus increased tracking error. Once the agents detect the target (as shown in Fig. 7(d)) and start receiving measurements, the tracking error decreases and the system's performance stabilizes, as illustrated in Fig. 7(f).

The next experiment aims to demonstrate how the proposed system scales with the number of agents. For this experiment, Monte Carlo simulations were conducted for 3, 5, 7, and 9 agents, randomly initialized inside the surveillance region, along with the target. The agents are spawned at locations such that their distance with the target is at maximum 30m. Fifty (50) trials were conducted for each configuration of agents and for two different weight settings i.e., $(w_1 = 10, w_2 = 7)$ and $(w_1 = 7, w_2 = 10)$, logging the target detection-and-jamming events, the agent interference events, and the tracking error, as shown in Figs. 8(a), 8(b), and 8(c), respectively. When the emphasis is on the target detection-and-jamming objective i.e., $(w_1 = 10, w_2 = 7)$, it can be observed that as the number of agents increases the average number of detection-and-jamming events also increases and subsequently the tracking error decreases (Fig. 8(c)).

However, as shown in Fig. 8(b), as the number of agents increases, the jamming interference events also increase.

This is due to the fact that, when the number of agents increases, the number of interference-free configurations that the agents can take are reduced, i.e., there is no longer free space around the target to be occupied by the agents which results both in target detection and zero jamming interference. This issue, which is left for future work, can be alleviated in a degree by considering directional antennas and heterogeneous agents which exhibit different characteristics regarding their sensing range, sensing shape, etc. When the emphasis is on jamming interference, i.e., ($w_1 = 7, w_2 = 10$), it can be observed from Fig. 8(b) that the jamming interference constraints are much better addressed, with an average of just 2 jamming interference events per time step for the case of 9 agents. However, this occurs at the expense of target tracking-and-jamming accuracy, as shown in Figs. 8(a)-8(c). Clearly, the desired system behavior (e.g., sensitivity to jamming interference, importance of tracking accuracy, etc.) can be customized and fine-tuned to match the application scenario requirements by properly selecting the weighting scheme as shown by the results of this experiment.

7.3 Computational Complexity

Finally, the main factor that drives the computational complexity of the proposed approach is discussed. In general, the branch-and-bound/branch-and-cut [45], [46] is the main optimization method which is used to solve mixed integer quadratic programs (MIQP), and its performance is usually determined by the number of integral variables that are being utilized in the MIQP. In branch-and-bound, a search tree is constructed, which enumerates in a systematic way candidate solutions for the MIQP problem. In this tree, each node includes the original MIQP problem constraints plus additional constraints on the bounds of the integer variables (i.e., in order to enforce those variables to take integer values). The algorithm explores the nodes of this tree by optimally solving a linear programming relaxation problem (where the node's integrality constraints have been dropped). In the scenario where the optimal solution to the linear program consists of an integer constrained variable with a fractional value (i.e., $x = f$), the algorithm generates a new branch for this variable consisting of two subproblems (i.e., nodes) where new integrality constraints are imposed (i.e., $x \leq \lfloor f \rfloor$ and $x \geq \lceil f \rceil$). As the exploration proceeds, the algorithm computes and maintains lower and upper bounds on the objective function, along with the incumbent integer feasible solution, which allows it to identify and efficiently prune branches of the tree which lead to sub-optimal solutions.

The number of integral variables (i.e., only binary variables in this work) involved in the proposed distributed controller (P3), is equal to $\mathcal{B} = b_{\tau,l} + \tilde{b}_{\tau} + \kappa_{\tau,i,l} + \tilde{\kappa}_{\tau,i} + z_{\tau,\psi,\ell}$ where $\tau = [0, \dots, T-1]$, $l = [1, \dots, 12]$, $i = [1, \dots, |\mathcal{S}| - 1]$, $\psi = [1, \dots, |\Psi|]$ and $\ell = [1, \dots, L]$. Consequently, it can be concluded that the computational complexity of the proposed controller will be affected by the length of planning horizon T , the number of agents $|\mathcal{S}|$, and the number of obstacles to be avoided $|\Psi|$. That said, as T , $|\mathcal{S}|$, and $|\Psi|$ increase, $\kappa_{\tau,i,l}$, $\tilde{\kappa}_{\tau,i}$, and $z_{\tau,\psi,\ell}$ are the decision variables which grow the fastest and ultimately determine the computational complexity of the controller.

Avg. Execution Time (sec)			
# Agents	Horizon Length	Centralized	Distributed
3	3	0.0540	0.0382
5	3	0.0989	0.0419
7	3	1.2092	0.0471
3	5	0.0869	0.0710
5	5	1.8932	0.0752
7	5	8.3932	0.0792

TABLE 1
Centralized and Distributed Average Execution Time for Different (Number of agents, Horizon Length) Combinations

To better understand the computational complexity of the proposed controller, a Monte-Carlo simulation was conducted, in an obstacle-free environment, where the number of agents and the length of planning horizon were varied. More specifically, for this experiment, the centralized and distributed formulation of the proposed tracking-and-jamming controller were run for 3, 5, and 7 agents, with planning horizon lengths of 3 and 5 time-steps. For each (Number of Agents, Horizon Length) combination 20 trials were conducted, with the agents and the target being randomly initialized inside the surveillance area. The rest of the parameters were set according to the simulation setup discussed in Sec. 7.1. Table 1 summarizes the results of this experiment in terms of the execution time (i.e., the time required by the solver to find the optimal solution). In particular, Table 1 shows the average time (taken over the 20 trials) for each combination of the parameters for the centralized and distributed controllers. As can be observed from Table 1, although both the centralized and the distributed controllers seem to perform reasonably well for some values of the parameters, in terms of execution time, the distributed controller scales much better with respect to the number of agents and the length of the planning horizon. More specifically, the results show that the increasing number of agents greatly affect the computational complexity of the centralized controller as opposed to the proposed distributed approach. Observe that the problem in (P3) is a mixed integer quadratic program which contains only linear constraints with 0 – 1 integral variables. These properties allow optimal solutions to be obtained on the order of milliseconds, especially with the use of the proposed distributed controller as illustrated in Table 1. It is also worth noting that additional computational savings can be obtained through various heuristics [47]–[49] and approximations [50], [51] which can provide adequate near-optimal MIP solutions.

7.4 Communication Overhead

We close the evaluation section of the proposed approach with a brief discussion on the communication requirements (i.e., amount of information exchange) of the proposed approach. In the proposed distributed estimation and control framework the agents exchange information in order to a) accurately estimate the target's trajectory over time, and b) coordinate their control decisions thus generating cooperative plans. The information exchange which takes place during the target state estimation phase is described in detail in Sec. 5.2. In essence, during this phase, at each time

step t the agents exchange their posterior belief on the target's state in a sequential fashion in an effort to compute the fused posterior distribution of the target's state. This process requires that the parameters of the target state distribution i.e., (μ, P) are transmitted in total $|S|$ times (where $|S|$ is the number of agents) amongst the agents. Since μ is a 6-by-1 column vector, and P is a 6-by-6 covariance matrix, the total amount of numerical values transmitted for this purpose in each time step is equal to $42|S|$. Finally, the agents exchange their plans via a sequential processing procedure as discussed in Sec. 6.2. The total amount of information exchanged in this phase over a planning horizon of length T time-steps is equal to $T|S|(|S| - 1)s$, where s is a 6-by-1 column vector representing the agent's state, resulting in the transmission of $6T|S|(|S| - 1)$ numerical values.

8 CONCLUSION AND FUTURE DIRECTIONS

This work investigates the problem of concurrently tracking-and-jamming a malicious target in 3D environments with a team of autonomous distributed aerial agents. A distributed model predictive control approach is proposed which allows the team of aerial agents to continuously track-and-jam the malicious drone, and at the same time minimize the jamming interference amongst them. The problem is studied in challenging scenarios with uncertain target dynamics, noisy measurements, and in the presence of obstacles that need to be avoided. The proposed approach is implemented as a mixed integer quadratic mathematical program (MIQP), which can be solved using off-the-shelf solvers. The effectiveness of the proposed approach is demonstrated through extensive simulation experiments.

Future work will utilize our existing proof-of-concept single agent counter-drone system [20] as a basis for implementing the proposed distributed multi-agent jamming controller in a real-world setting, and solving possible issues which might arise from a practical implementation. For instance, the use of an omnidirectional jamming antenna might in practice interfere with the navigation circuitry of the jamming agent. A possible solution to this problem is to revert to a backup positioning system while the agent is jamming the target. Alternatively, a directional jamming antenna can be utilized. Due to battery constraints the amount of time the agents can be in the air and at the same time jam the target might be insufficient for the success of the mission; this is another issue that we plan to investigate through the practical implementation of the proposed approach.

The extension of the proposed approach to multiple malicious targets will be investigated in a future step. Specifically, we will investigate the possibility of designing a pre-planning step which incorporates an instance of the assignment problem [52], [53] to the proposed approach, in order to dynamically assign in an optimal way the pursuer agents to the multiple targets over a rolling time horizon. Other future directions include the investigation of the optimal number of pursuer agents that must be deployed in order to track-and-jam multiple targets within a specified bounded surveillance area and the optimization of the total time required for multiple targets to be jammed and eventually be forced to land, by multiple pursuer agents.

ACKNOWLEDGMENTS

This work has been supported by the European Union's H2020 research and innovation programme under grant agreement No 739551 (KIOS CoE - TEAMING) and from the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

REFERENCES

- [1] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "Towards automated 3D search planning for emergency response missions," *Journal of Intelligent & Robotic Systems*, vol. 103, no. 1, pp. 1–19, 2021.
- [2] J. Moon, S. Papaioannou, C. Laoudias, P. Kolios, and S. Kim, "Deep reinforcement learning multi-UAV trajectory control for target tracking," *IEEE Internet of Things Journal*, vol. 8, no. 20, pp. 15 441–15 455, 2021.
- [3] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "3D trajectory planning for UAV-based search missions: An integrated assessment and search planning approach," in *Proc. IEEE Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 517–526.
- [4] —, "Decentralized search and track with multiple autonomous agents," in *Proc. IEEE 58th Conf. on Decision and Control (CDC)*, 2019, pp. 909–915.
- [5] —, "A cooperative multi-agent probabilistic framework for search and track missions," *IEEE Trans. on Control of Network Systems*, vol. 8, no. 2, pp. 847–858, 2021.
- [6] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, "A review on UAV-based applications for precision agriculture," *Information*, vol. 10, no. 11, p. 349, 2019.
- [7] P. Radoglou-Grammatikis, P. Sarigiannidis, T. Lagkas, and I. Moscholios, "A compilation of UAV applications for precision agriculture," *Computer Networks*, vol. 172, p. 107148, 2020.
- [8] DRONEII. (2020) Drone market report 2020. [Online]. Available: <https://droneii.com/the-drone-market-size-2020-2025-5-key-takeaways/>
- [9] D. Schneider, "Regulators seek ways to down rogue drones: Growing anti-drone industry offers radar, remote ID, and other tools," *IEEE Spectrum*, vol. 56, no. 4, pp. 10–11, 2019.
- [10] S. Crino and C. Dreby, "Drone attacks against critical infrastructure: A real and present threat," Atlantic Council, Tech. Rep., 2020. [Online]. Available: <http://www.jstor.org/stable/resrep24632>
- [11] T. Humphreys, Oversight and Management Efficiency Subcommittee, Homeland Security Committee, US House of Representatives, Washington, DC, Tech. Rep.
- [12] K. Wesson and T. Humphreys, "Hacking drones," *Scientific American*, vol. 309, no. 5, pp. 54–59, 2013.
- [13] I. Guvenc, F. Koohifar, S. Singh, M. L. Sichert, and D. Matolak, "Detection, tracking, and interdiction for amateur drones," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 75–81, 2018.
- [14] J. Loeb, "Exclusive: Anti-drone technology to be tested in UK amid terror fears," *Engineering & Technology*, vol. 12, no. 3, pp. 9–9, 2017.
- [15] G. Lykou, D. Moustakas, and D. Gritzalis, "Defending airports from UAS: A survey on cyber-attacks and counter-drone sensing technologies," *Sensors*, vol. 20, no. 12, p. 3537, 2020.
- [16] E. Camponogara, D. Jia, B. H. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Mag.*, vol. 22, no. 1, pp. 44–52, 2002.
- [17] R. Scattolini, "Architectures for distributed and hierarchical model predictive control—a review," *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.
- [18] M. B. Revill, "UAV swarm behavior modeling for early exposure of failure modes," *Naval Postgraduate School*, Tech. Rep., 2016.
- [19] A. H. Abunada, A. Y. Osman, A. Khandakar, M. E. H. Chowdhury, T. Khattab, and F. Touati, "Design and implementation of a RF based anti-drone system," in *Proc. IEEE Int. Conf. on Informatics, IoT, and Enabling Technologies (ICIOT)*, 2020, pp. 35–42.
- [20] N. Souli, R. Makrigiorgis, A. Anastasiou, A. Zacharia, P. Petrides, A. Lazanas, P. Valianti, P. Kolios, and G. Ellinas, "Horizonblock: Implementation of an autonomous counter-drone system," in *Proc. Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 398–404.
- [21] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi, and J. Chen, "Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 68–74, 2018.

- [22] T. Multerer, A. Ganis, U. Pecht, E. Miralles, A. Meusing, J. Mietzner, M. Vossiek, M. Loghi, and V. Ziegler, "Low-cost jamming system against small drones using a 3D MIMO radar based tracking," in *Proc. European Radar Conf. (EURAD)*, 2017, pp. 299–302.
- [23] S. Park, H. T. Kim, S. Lee, H. Joo, and H. Kim, "Survey on anti-drone systems: Components, designs, and challenges," *IEEE Access*, vol. 9, pp. 42 635–42 659, 2021.
- [24] S. R. Ganti and Y. Kim, "Implementation of detection and tracking mechanism for small UAS," in *Proc. Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2016, pp. 1254–1260.
- [25] S. Srigrarom and K. H. Chew, "Hybrid motion-based object detection for detecting and tracking of small and fast moving drones," in *Proc. Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2020.
- [26] L. Dressel and M. J. Kochenderfer, "Hunting drones with other drones: Tracking a moving radar target," in *Proc. Int. Conf. on Robotics and Automation (ICRA)*, 2019, pp. 1905–1912.
- [27] M. R. Brust and G. Danoy and P. Bouvry and D. Gashi and H. Pathak and M. P. Goncalves, "Defending against intrusion of malicious UAVs with networked UAV defense swarms," in *Proc. IEEE Conf. on Local Computer Networks Workshops*, 2017, pp. 103–111.
- [28] G. C. Karras, C. P. Bechlioulis, G. K. Fournas, and K. J. Kyriakopoulos, "Formation control and target interception for multiple multi-rotor aerial vehicles," in *Proc. Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 85–92.
- [29] P. Valianti, S. Papaioannou, P. Kolios, and G. Ellinas, "Multi-agent coordinated interception of multiple rogue drones," in *Proc. IEEE Global Communications Conf. (GLOBECOM)*, 2020, pp. 1–6.
- [30] —, "Multi-agent coordinated close-in jamming for disabling a rogue drone," *IEEE Trans. on Mobile Computing*, pp. 1–1, 2021.
- [31] S. Papaioannou, P. Kolios, C. G. Panayiotou, and M. M. Polycarpou, "Cooperative simultaneous tracking and jamming for disabling a rogue drone," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2020, pp. 7919–7926.
- [32] S. Papaioannou, P. Kolios, and G. Ellinas, "Downing a rogue drone with a team of aerial radio signal jammers," in *2021 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 2555–2562.
- [33] A. Richards and J. P. How, "Robust distributed model predictive control," *International Journal of control*, vol. 80, no. 9, pp. 1517–1531, 2007.
- [34] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking, part i. dynamic models," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1333–1364, 2003.
- [35] C. E. Luis, M. Vukosavljev, and A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 604–611, 2020.
- [36] S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*, Norwood, MA, 1999.
- [37] I. D. Cowling, O. A. Yakimenko, J. F. Whidborne, and A. K. Cooke, "A prototype of an autonomous controller for a quadrotor UAV," in *Proc. IEEE European Control Conf. (ECC)*, 2007, pp. 4001–4008.
- [38] A. Fradkov and B. Andrievsky, "Combined adaptive controller for UAV guidance," *Eur. J. Control*, vol. 11, no. 1, pp. 71–79, 2005.
- [39] S.-T. Park and J. G. Lee, "Improved kalman filter design for three-dimensional radar tracking," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 37, no. 2, pp. 727–739, 2001.
- [40] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, vol. 10, no. 3, pp. 197–208, 2000.
- [41] S. Julier and J. K. Uhlmann, "General decentralized data fusion with covariance intersection," in *Handbook of Multisensor Data Fusion*. CRC Press, 2017, pp. 339–364.
- [42] Z. Deng, P. Zhang, W. Qi, J. Liu, and Y. Gao, "Sequential covariance intersection fusion Kalman filter," *Information Sciences*, vol. 189, pp. 293–309, 2012.
- [43] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *IEEE Trans. on Signal Processing*, vol. 50, no. 2, pp. 174–188, 2002.
- [44] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electr. Syst. Mag.*, vol. 25, no. 7, pp. 53–82, 2010.
- [45] J. E. Mitchell, "Branch-and-cut algorithms for combinatorial optimization problems," *Handbook of Applied Optimization*, vol. 1, pp. 65–77, 2002.
- [46] J. Eckstein, "Parallel branch-and-bound methods for mixed integer programming," in *Applications on Advanced Architecture Computers*. SIAM, 1996, pp. 141–153.
- [47] V. V. Naik and A. Bemporad, "Exact and heuristic methods with warm-start for embedded mixed-integer quadratic programming based on accelerated dual gradient projection," *arXiv preprint arXiv:2101.09264*, 2021.
- [48] G. Hendel, "Adaptive large neighborhood search for mixed integer programming," *Mathematical Programming Computation*, pp. 1–37, 2021.
- [49] N. Sonnerat, P. Wang, I. Ktena, S. Bartunov, and V. Nair, "Learning a large neighborhood search algorithm for mixed integer programs," *arXiv:2107.10201*, 2021.
- [50] E. Klotz and A. M. Newman, "Practical guidelines for solving difficult mixed integer linear programs," *Surveys in Operations Research and Management Science*, vol. 18, no. 1, pp. 18–32, 2013.
- [51] D. Bertsimas and B. Stellato, "Online mixed-integer optimization in milliseconds," *arXiv:1907.02206*, 2021.
- [52] D. W. Pentico, "Assignment problems: A golden anniversary survey," *Eur. J. Oper. Res.*, vol. 176, no. 2, pp. 774–793, 2007.
- [53] S. Deb, K. R. Pattipati, Y. Bar-Shalom, and H. Tsaknakis, "A new algorithm for the generalized multidimensional assignment problem," in *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, 1992, pp. 249–254.



Savvas Papaioannou received the B.S. degree in electronic and computer engineering in 2011 from the Technical University of Crete, Greece, the M.S. degree in electrical engineering in 2013 from Yale University, USA, and the Ph.D. degree in computer science in 2017 from the University of Oxford, U.K. He is currently a Research Associate with the KIOS Research and Innovation Center of Excellence at the University of Cyprus. His research interests include multi-agent and autonomous systems, state estimation and control, multi target tracking, probabilistic inference, Bayesian reasoning, and intelligent unmanned-aircraft vehicle (UAV) systems and applications.



Panayiotis Kolios is a Research Assistant Professor at the KIOS Research and Innovation Center of Excellence at the University of Cyprus. He received his B.Eng and Ph.D degrees in Telecommunications Engineering from King's College London in 2008 and 2011, respectively. His interests focus on both basic and applied research on networked intelligent systems. Some examples of such systems include intelligent transportation systems, autonomous unmanned aerial systems, and the plethora of

cyber-physical systems that arise within the Internet of Things. Particular emphasis is given to emergency response aspects in which faults and attacks could cause disruptions that need to be effectively handled. He is an active member of IEEE, contributing to a number of technical and professional activities within the Association.



Georgios Ellinas holds a B.S., M.Sc., M.Phil., and a Ph.D. in Electrical Engineering from Columbia University. He is a Professor and the past Chair (2014-2020) of the Department of Electrical and Computer Engineering and a founding member of the KIOS Research and Innovation Center of Excellence at the University of Cyprus. Previously, he served as Associate Professor of Electrical Engineering at City College of New York, Senior Network Architect at Tellium Inc., and Research Scientist/Senior Research Scientist at Bell Communications Research (Bellcore). Prof. Ellinas is a Fellow of the IET (2019), a Senior Member of IEEE, OSA, and ACM, and a Member of the Marie Curie Fellows Association (MCFA). His research interests are in optical networks, transportation networks, IoT, and unmanned aerial systems.