Model Predictive Control For Multiple Castaway Tracking with an Autonomous Aerial Agent

Andreas Anastasiou[§*], Savvas Papaioannou[*], Panayiotis Kolios[*] and Christos G. Panayiotou[§*]

Abstract—**Over the past few years, a plethora of advancements in Unmanned Areal Vehicle (UAV) technology has paved the way for UAV-based Search and Rescue (SAR) operations with transformative impact to the outcome of critical life-saving missions. This paper dives into the challenging task of multiple castaway tracking using an autonomous UAV agent. Leveraging on the computing power of the modern embedded devices, we propose a Model Predictive Control (MPC) framework for tracking multiple castaways assumed to drift afloat in the aftermath of a maritime accident. We consider a stationary radar sensor that is responsible for signaling the search mission by providing noisy measurements of each castaway's initial state. The UAV agent aims at detecting and tracking the moving targets with its equipped onboard camera sensor that has limited sensing range. In this work, we also experimentally determine the probability of target detection from real-world data by training and evaluating various Convolutional Neural Networks (CNNs). Extensive qualitative and quantitative evaluations demonstrate the performance of the proposed approach.**

Index Terms—**UAS model predictive control, Multiple castaway tracking, Marine Environment, Computer Vision, Neural Networks**

I. Introduction

Tracking targets in a sea environment is a challenging task due to the vast operation environment resulting in sparse observations, leading to estimation errors in their positions. Recent advancements in Unmanned Aerial Vehicle (UAV) technology have enabled their utilization in various emergency response operations, including search-and-rescue (SAR) [1], [2] and search-and-track (SAT) missions [3], [4]. Moreover, UAVs are increasingly being used in domains such as critical infrastructure inspection [5], surveillance [6], disaster management missions [7], [8], and emergency response [9].

In this paper, we propose an autonomous UAV system for maritime disaster management, security, and safety. The number of lost ships and shipwrecks worldwide has decreased compared to a decade ago, but it is still alarmingly high [10]. Moreover, the recent migration problem has forced many people to take the risk of illegal sea travel, resulting in a high number of castaways. These effects are especially evident in the Mediterranean Sea, where arrivals through the Mediterranean Sea to European countries such as Italy, Spain, Greece, and Cyprus have exceeded one hundred thousand people, and almost two thousand people are currently considered deceased or missing according to the United Nations High Commissioner for Refugees [11].

Therefore, in this work, we propose a Model Predictive Control (MPC) framework for tracking multiple castaway targets in maritime environments using an autonomous UAV agent. The proposed MPC formulation employs a Non-linear Mixed Integer Program (NMIP) to determine the optimal control inputs of an autonomous UAV agent over a finite rolling horizon. This enables accurate multiple castaway tracking and monitoring. Our approach generates the UAV's control inputs in an online fashion so that the overall tracking error for monitoring multiple castaway targets is minimized. Specifically, we consider each castaway's predicted location uncertainty and minimize the distance between the castaway and the UAV's onboard camera sensor, thereby gathering observations with the lowest noise possible. The key contributions of this work are:

- We propose a Model Predictive Control (MPC) framework that allows an autonomous UAV agent to track multiple castaways. To achieve this, we formulate an NMIP to compute the optimal control inputs of the UAV over a finite rolling planning horizon, which minimizes the tracking error over multiple targets.
- We experimentally analyze the target detection probability using water buoys by training various Convolutional Neural Networks (CNNs) from images taken by the UAV from various altitudes. From the evaluation of the CNNs, we construct a piece-wise linear function that describes the confidence of detection probability based on the UAV's altitude.
- Finally, we create a new open-source dataset consisting of aerial images of water buoys taken with a UAV from various altitudes above sea level [12]. Different wave conditions have also been captured to create a machine learning object detection algorithm that reflects the challenges of castaway detection in marine environments.

The rest of this paper is structured as follows. The related work is discussed in Sec. II. The system model and our setup are introduced in Sec. III. The proposed approach along with the NMIP formulation and the derivation of the detection probability is given in Sec. IV-B. The evaluation of the proposed approach is discussed in Sec. V, and finally Sec. VI concludes the paper and discusses future research directions.

[*] Authors are with KIOS Research and Innovation Center of Excellence (KIOS CoE), [§] and with the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, 1678, Cyprus. E-mail:{anastasiou.antreas, papaioannou.savvas, kolios.panayiotis, christosp} @ucy.ac.cy

II. Related Work

Castaway search in post shipwreck disasters has been examined in several works in the past, assuming a number of different setups. Xinming Hu et al. [13] explored the use of Kalman Filtering (KF) [14] to predict the castaway position and a Dynamic Window Approach (DWA) [15] for the Unmanned Surface Vehicle's (USV) path planning and obstacle avoidance. Others have investigated the use of both UAV and USV for aiding in the rescue and response. Xiao Xuesu et al. [16] introduced a UAV for the use of an overhead view camera, helping in the state estimation of the USV. Thus, the system is able to autonomously navigate while First Responders (FRs) can focus on task-level needs. Anibal Matos et al. [17] took a different approach to the multi-robot solution, utilizing a USV for a first large-scale bathymetry of the interest area and then deploying an Autonomous Underwater Vehicle (AUV) for the close inspection of suspicious targeted areas. In our work, we utilize multiple instances of KF for estimating the state of each castaway. We also employ a single UAV for its maneuverability and its ability to rapidly adapt between flying and hovering.

On the other hand, Ramirez et al. [18] proposed an expert system utilizing a UAV and a USV for maritime SAR operations with the UAV acting as a remote fast moving sensor that assists the USV during the mission by providing measurements about the castaways thus, letting the USV correct their estimated position using a Particle Filter (PF) [19] for each one of them. As mentioned previously, we use KF instead, purely for its lower computational needs since using multiple PF on an embedded device is computationally intensive.

The use of Non-linear Programming has also been used widely over the years. Some have investigated an Artificial Potential Field path planning technique for producing flyable paths for multi-rotor UAVs in ground target tracking systems [20] and obstacle avoidance systems [21]. Such techniques are beneficial for ground moving targets but, their use has mainly been researched for tracking smooth paths. In addition, Dai and Cochran [22] investigated the generation of cooperative paths for UAVs that are parameterized under the Cornu-Sprials. Such paths however, are mostly suitable for fixed-wing aircrafts. Non-linear and Linear MPC have also been exhaustively researched in terms of target tracking, but most of the work has mainly been focused on ground vehicles [23].

Many efforts have also been done for solving the problem of UAV control in 3D environments. Yang and Sukkarieh [24] exploited rapidly-exploring random trees for collision avoidance in cluttered environments. Tisdale et al. [25] proposed a decentralized online path planning framework for a team of fixed wing UAVs with the purpose of search and locate missions, while Papaioannou et al. [26] exploided a novel control strategy for tracking mallicious UAVs and in [27] [28] the authors explored various MPC formulations for SAR missions in a 3D environments utilizing UAVs. In contrast, our work focuses on optimal control of a single UAV in 3D environment for multiple target tracking. By controlling the UAV in all 3 dimensions, gives the ability to improve the performance of a target tracking system, by unlocking the capability to choose between the quality of the observation and the effective sensing range.

In addition, research has also been done for calculating the Probability of Success (POS) of such search missions. Donatien et al. [29] explained in their work how using optimal search theory and analyzing the Probability of Containment and Probability of Detection at the area of interest, one can calculate the POS of the operation.

Finally, closely related to our work is the approach presented in [30]. Chang Liu and Karl Hedrick, also proposed a framework that has the objective of minimizing the covariance of the target's estimated location, by using an MPC to control the agent's movements. However, their work was limited to a two-dimensional environment and to a single ground target. Moreover, the authors did not consider any detection probability assumptions in the sensing model of the agent.

In this paper, we propose an MPC formulation for multiple target tracking missions utilizing a UAV agent for tracking multiple castaways that are assumed to be drifting afloat in the aftermath of a maritime accident. To achieve this, we take into account the target detection probability which is learned from real-world data, as well as the inherent uncertainty in the target dynamical and measurement model. Leveraging multiple KFs to estimate each castaways' drift at sea, the proposed system is able to predict the castaways' location and provide informed position inputs to the agent's control system. The system minimizes the measurement noise and simultaneously aids the throughput of Computer Vision (CV) techniques that are usually used in such missions. As a note, and to better simulate a real-life CV detector, we have experimentally analyzed the target detection probability by training various CNNs. From the evaluation of the CNNs, we constructed a piece-wise linear function that describes the confidence of detection based on the UAV's altitude. Overall, the system aims at tracking all castaways, giving the FRs actionable information about the survivors. The proposed technique considers the UAV's kinematic and sensing constraints, and optimizes the agent's control actions over a rolling finite planning horizon, which aims at minimizing the overall tracking uncertainty.

III. Preliminaries

A. System Architecture

For the proposed architecture we consider two main components, the mission initiating system (i.e. a coastal radar) and a UAV agent taking over the mission after initially being alerted by the former component. Starting from the initial sensor measurements, we initialize one KF [14] for every target with initial priori states the measurements received from the coastal radar. As long as the castaways are within the field of view (FoV) of the onboard camera sensor (further defined in Sec. III-B), the KF uses the measurement from the onboard camera sensor to correct the prediction. When a castaway is not within the FoV of the UAV's onboard camera,

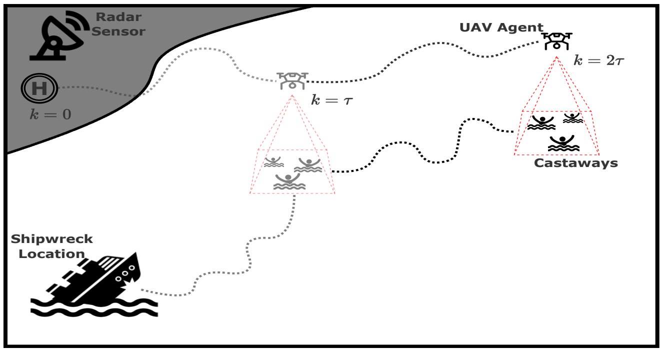

Fig. 1: Multiple castaway track scenario illustration at two time steps

then the KF is used to update the location estimates but without the correction steps. While the KF correction steps are not used, the uncertainty of their location increases. In order to minimize the uncertainty and get a better estimate of the castaways' locations, the UAV has to observe the targets with its camera's limited sensing range. However, detecting the targets is stochastic and depends on the probability of detection (described in Sec. III-C), which is inversely proportional to the agent's altitude. An illustration of the described scenario can be seen in Fig. 1.

B. Agent Dynamics and Sensing Model

In our work, the UAV agent is free to move in the 3D Cartesian space and its movement is governed by the following discrete time, linear dynamical model:

$$\chi^a_{k+1} = A\chi^a_k + Bu^a_k \tag{1}$$

where $\chi^a_k = [p^a_k, v^a_k]^T$ is the state of the agent at the k^{th} time step which is consisted from its position $p^a_k = [x^a_k, y^a_k, z^a_k]^T \in \mathbb{R}^3$ in three-dimensional Cartesian coordinates and its velocity $v^a_k = [\dot{x}^a_k, \dot{y}^a_k, \dot{z}^a_k]^T \in \mathbb{R}^3$. The term $u^a_k = [u^a_x, u^a_y, u^a_z]^T \in \mathbb{R}^3$ denotes the input control force for each dimension while the system matrices $A \in \mathbb{R}^{6\times 6}, B \in \mathbb{R}^{6\times 3}$ are given below:

$$A = \begin{bmatrix} \mathbf{I}_{3\times 3} & \delta t \mathbf{I}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \rho \mathbf{I}_{3\times 3} \end{bmatrix}, B = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \xi \mathbf{I}_{3\times 3} \end{bmatrix} \tag{2}$$

The term δt denotes the sampling interval while, $\mathbf{I}_{3\times 3}$ and $\mathbf{0}_{3\times 3}$ denote the identity and zero matrix, respectively. The parameter $\rho \in [0, 1]$ is used to model air resistance whereas parameter $\xi = \frac{\delta t}{m}$ is used to convert the input control force to acceleration, where m is the agent's mass.

As mentioned previously, the agent is carrying an onboard camera sensor. The onboard camera is capable of detecting the castaways and measuring their 2D location i.e., Cartesian (x,y)-coordinates. The effective sensing area of the onboard camera is defined by a rectangle projected on the sea surface which's size is further defined by the horizontal (θ_h) and vertical (θ_v) FoVs in degrees. The horizontal (l_h) and vertical (l_v) lengths of the rectangle are calculated by $l_{h|k} = 2z^a_k \tan\left(\frac{\theta_h}{2}\right)$ and $l_{v|k} = 2z^a_k \tan\left(\frac{\theta_v}{2}\right)$.

The measurement provided by the sensor is defined by the random set Z_k, which can be either empty with probability $1 - p_k, p_k \in [0, 1]$ or be a singleton set with probability p_k (see Sec. III-C) with its element distributed over the state space of the target $\chi^{c_i}_k$, where c_i refers to the i^{th} castaway, according to the probability density function $pr(\chi^{c_i}_k)$. Therefore, the PDF of Z_k is given by:

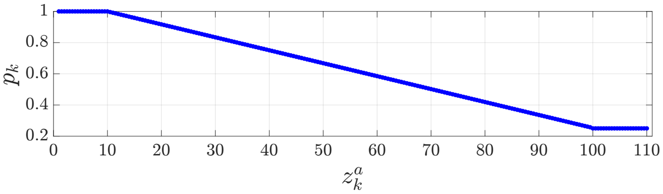

Fig. 2: Piece wise function emulating the probability of receiving a measurement from the sensor. Parameters were set to $\alpha_1 = 10, \alpha_2 = 100$ and $\beta_1 = -0.0083, \beta_2 = 1.083$

$$f(Z_k) = \begin{cases} 1 - p_k & \text{if } Z_k = \emptyset \\ p_k pr(\chi^{c_i}_k) & \text{if } Z_k = \{y^{c_i}_k\} \end{cases} \tag{3}$$

The single target measurement likelihood function is further given by $y^{c_i}_k = h(\chi^{c_i}_k) + w(p_k)$ where:

$$h(\chi^{c_i}_k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \chi^{c_i}_k, \qquad w(p_k) \sim \mathcal{N}(0, p_k^{-1}\gamma) \tag{4}$$

$w(p_k)$ is the measurement noise, which is randomly distributed by a zero mean Normal density with standard deviation $p_k^{-1}\gamma$, where γ is the scaling parameter.

C. Target detection probability

This subsection elaborates on the derivation of the probability of detection which we assume to follow a piece-wise linear function. As previously mentioned in Sec. III-B, we receive a non-empty measurement from the onboard camera sensor with probability p_k. Thus, probability p_k is assumed to obey the following piece-wise function(see Fig. 2):

$$p_k = \begin{cases} 1 & \text{if } z^a_k \leq \alpha_1 \\ p_{min} & \text{if } z^a_k \geq \alpha_2 \\ \beta_1 z^a_k + \beta_2 & \text{otherwise} \end{cases} \tag{5}$$

where, parameters α_1 and α_2 are used to tune the altitudes where the probability gets at its maximum and its minimum value. Parameters β_1 and β_2 are used to define the slope of the linear function. Note that these tuning parameters must be chosen as such as to not violate the probability rule of $p_k \in [0, 1]$. The above modeling assumptions have been verified experimentally, as discussed next in Sec. IV-B.

D. Castaway Motion Model

Each castaway is moving in 3D space and its motion is governed by a modified version of the Stokes' drift equations [31]. The equations are used to model and generate the ground truth states of the castaways $\chi^{c_i}_k = [x^{c_i}_k, y^{c_i}_k, z^{c_i}_k]^T, \forall i = [1, .., \mathcal{C}]$ given by (6). In partiqular Eq. (6a) describes the water velocity in the direction of the wave propagation that a specific castaway experiences. Thus, the castaway's location is updated as show in Eq. (6b).

$$v^{c_i}_k = \frac{\omega H}{2} e^{wd^{c_i}} \sin\left(qd^{c_i} - \omega k\right) \tag{6a}$$

$$\chi^{c_i}_{k+1} = \chi^{c_i}_k + v^{c_i}_k \begin{bmatrix} \cos(\varphi) & \sin(\varphi) & 1 \end{bmatrix}^T dt \tag{6b}$$

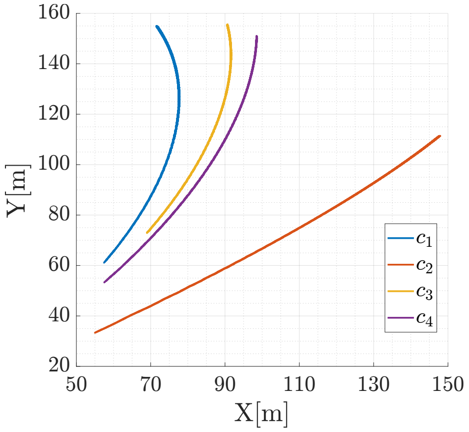

Fig. 3: Drift paths of four castaways induced by small amplitude waves for the span of an hour.

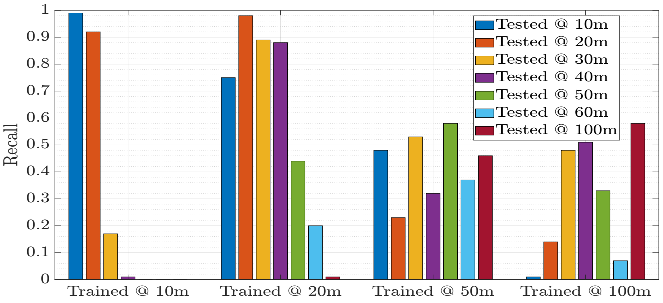

Fig. 4: Recall of each CNN at every altitude.

where $\omega = 2\pi/T$ is the wave frequency, H is the wave height, w is the wave decay-rate and d^{c_i} is the distance of the i^{th} castaway from the wave source. The wave number is given by $q = 2\pi/L$ and depends on the wavelength L. The angle between the castaway and the wave source is given by φ while the wave period is given by $T = \sqrt{2\pi L/gK}$ where, $K = \tanh(qD)$, D is the water depth and g is the gravitational acceleration. The above wave equation in (6a) assumes small amplitude waves ($qH \ll 1$), deepwater conditions ($K < 1$) and no wave reflections. The ground truth drift of multiple castaways is shown in Fig. 3. The figure shows the drift of 5 castaways induced over the period of 60 minutes.

IV. Proposed Approach

A. Determining the Target Detection Probability

To better determine a realistic target detection probability, the proposed piece-wise linear function was validated by real-life data of water objects and more specifically buoys. We collected more than 10 thousand images of two different types of buoys (white and red) in the Republic of Cyprus. The images were taken using an off-the-shelf UAV carrying a 4K camera from the altitudes of 10-100 meters in two different sea conditions. The first set of images was taken during a calm sea with zero wind while the second set, was taken during small waves and windy conditions. Each image was manually labeled for training a single class detector. Four CNNs were trained at detecting buoys from the various altitudes. The initial resolution of the images was 1920×1080 pixels and the custom dataset was split into four groups:

1^{st}: Trained with 2300 images taken from 10 meters altitude and validated with roughly 300 images and tested with roughly 300 images.

2^{nd}: Trained with 2900 images taken from 20 and 30 meters altitude and validated with roughly 370 images and tested with roughly 370 images.

3^{rd}: Trained with 3700 images taken from 50 and 60 meters altitude and validated with roughly 470 images and tested with roughly 470 images.

4^{th}: Trained with 1400 images taken from 100 meters altitude and validated with roughly 200 images and tested with roughly 200 images.

Manually labeling the images resulted to more than 14 thousand labeled buoys. The images in the custom dataset were split into 80% for training, 10% for validation and 10% for testing. For training, we utilized the AlexeyAB's version of the Darknet framework [32] by using the YOLOv4-tiny CNN mainly for its real-time capabilities. During training, all the default network parameters were chosen and the YOLOv4-tiny's default augmentation configurations were used. The training image size was set to 416x416 while 200 epochs of batch 32 for each network were done. Another functionality given by the YOLOv4 framework called random resizing was also used. Random resizing as its name suggests, randomly resizes the network input size every 10 batches (iterations) form scale 0.7 to scale 1.4 while keeping the initial aspect ratio of the image. Training of the networks was done on an NVIDIA Tesla v100 GPU, with the validation set being used during training. After training, each network was tested with all the test sets, at all altitudes. This was done to derive each networks ability of detection when given images that contain at least one buoy. During testing, the number of true and false positives as well as the amount of false negatives for each network at every altitude were counted. Fig. 4 shows the recall that each network achieved at every altitude. Recall is calculated by dividing the count of the true positives over the sum of true positives and false negatives. It can be seen that our previous assumption of decreasing confidence with increasing altitude was logical and close to truth. Thus, from Fig. 4 we constructed the confidence of detection function in Eq. (7) which we use in the proposed solution to mimic the behavior of a traditional CNN at detecting overboard castaways. A sample of three images of detected buoys can be seen in Fig. 5. The images were taken at an altitude of 30 meters. In the first image (a), a true positive detection of a white buoy can be seen during small amplitude waves. In the second image (b), a single false positive detection can be seen in the absence of buoys. Finally, in the last image (c) a false negative detection can be seen where the detector failed to recognize the white buoy.

$$f^d(z_k^a) = \begin{cases} 1 & \text{if } z_k^a \leq 10 \\ 0.25 & \text{if } z_k^a \geq 100 \\ -0.0083 z_k^a + 1.083 & \text{otherwise} \end{cases} \tag{7}$$

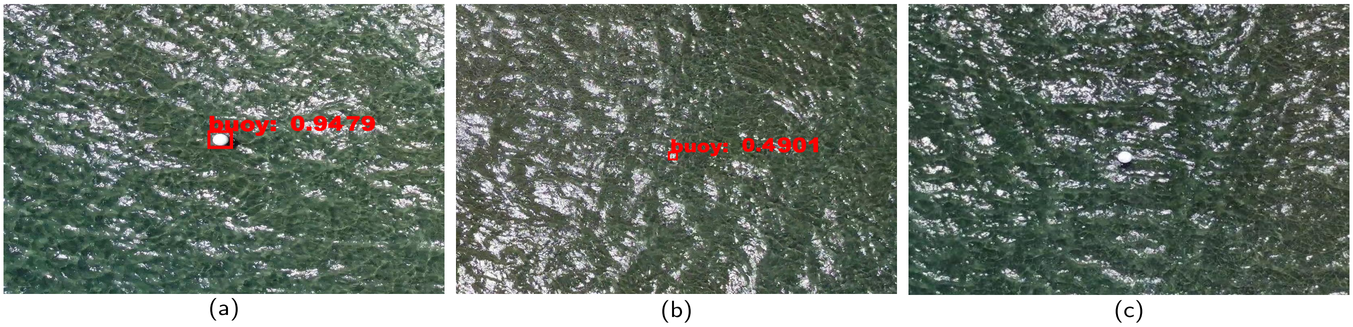

Fig. 5: Sample images of detected buoys at 30 meters altitude. Image (a) shows a successfully detected buoy. Image (b) shows a false positive detection in the absence of buoys. Image (c) shows a false negative detection in the presence of a buoy.

B. Model Predictive Control Formulation

To compute the vector u^a_k for controlling the UAV's movement, a receding horizon MPC is used. The MPC is formulated such that it does not infringe the UAV's capabilities and limitations such as horizontal and vertical speeds and accelerations. To achieve that, we formulate an optimization problem that minimizes the trace of the state covariance matrix $P^{c_i}_{k+\tau|k}$ for each castaway's predicted location by controlling the input of the agent as referred at Sec. III-B. In this work, the KF equations are employed to be able to predict the castaway's drift and provide estimates of its location and the accumulated uncertainty of the open-loop system (when no measurements are received over an extended period of time) over the planned horizon. After being initialized with noisy measurements about the castaway's state, the KF equations predict the castaway's next state and then correct the estimate using pseudo-measurements created by adding noise to the predicted positions, whenever the castaway is assumed to be within the FoV. The proposed model predictive controller, is shown in Eq, (8a)-(8t), as a non-linear mixed integer program.

The objective in Eq. (8a) is to minimize the trace of the covariance matrix $P^{c_i}_{k+\tau|k}$ of each castaway over all the time steps of the planning horizon where, N is the number of steps in the receding horizon and $\mathcal{C}$ is the number of castaways. Eq. (8b) defines the UAV's dynamical model as previously discussed in Sec. III-B. Vector $u^a_{k+\tau|k}$ is the control input of the UAV consisting of (u^a_x, u^a_y, u^a_z) for all steps in the horizon $\tau \in \{0, ..., N-1\}$. Constrain (8c) refers to the derivation of the horizontal and vertical size of the onboard camera's FoV.

$$\underset{u^a_{k+\tau|k}}{\arg\min} \sum_{i=1}^{\mathcal{C}} \sum_{\tau=0}^{N-1} tr(P^{c_i}_{k+\tau|k}) \tag{8a}$$

subject to $i \in \{1, ...\mathcal{C}\}, \tau \in \{0, ..., N-1\}:$

$$\chi^a_{k+\tau+1|k} = A^a\chi^a_{k+\tau|k} + Bu^a_{k+\tau|k} \qquad \forall\tau \tag{8b}$$

$$l^j_{k+\tau|k} = z^a_{k+\tau|k}\tan(\theta_j) \qquad \forall j,\tau \tag{8c}$$

$$\mathrm{d}^j_{k+\tau|k} = D_j\chi^a_{k+\tau|k} + (-1)^j l^j_{k+\tau|k} \qquad \forall j,\tau \tag{8d}$$

$$b^{c_i|j}_{k+\tau|k} = \begin{cases} 1, & \text{if } E_j\chi^{c_i}_{k+\tau|k} \le \mathrm{d}^j_{k+\tau|k} \\ 0, & \text{otherwise} \end{cases} \qquad \forall i,j,\tau \tag{8e}$$

$$\mathrm{s}^{c_i}_{k+\tau|k} = \sum_{j=1}^{4} b^{c_i|j}_{k+\tau|k} \qquad \forall i,\tau \tag{8f}$$

$$\mathrm{b}^{c_i}_{k+\tau|k} = \begin{cases} 1, & \text{if } \mathrm{s}^{c_i}_{k+\tau|k} = 4 \\ 0, & \text{otherwise} \end{cases} \qquad \forall i,\tau \tag{8g}$$

$$\chi^{c_i}_{k+\tau+1|k} = A^c\hat{\chi}^{c_i}_{k+\tau|k} \qquad \forall i,\tau \tag{8h}$$

$$P^{c_i}_{k+\tau+1|k} = A^c\hat{P}^{c_i}_{k+\tau|k}A^{cT} + Q \qquad \forall i,\tau \tag{8i}$$

$$K^{c_i}_{k+\tau+1|k} = \hat{P}^{c_i}_{k+\tau+1|k}C^T \left(C\hat{P}^{c_i}_{k+\tau+1|k}C^T + R_{k+\tau|k}\right)^{-1} \qquad \forall i,\tau \tag{8j}$$

$$\hat{\chi}^{c_i}_{k+\tau+1|k} = \chi^{c_i}_{k+\tau+1|k} + \mathrm{b}^{c_i}_{k+\tau|k}K^{c_i}_{k+\tau+1|k} \left(y^{c_i}_{k+\tau+1|k} - C\hat{\chi}^{c_i}_{k+\tau+1|k}\right) \qquad \forall i,\tau \tag{8k}$$

$$\hat{P}^{c_i}_{k+\tau+1|k} = P^{c_i}_{k+\tau+1|k} - \mathrm{b}^{c_i}_{k+\tau|k}K^{c_i}_{k+\tau+1|k} CP^{c_i}_{k+\tau+1|k} \qquad \forall i,\tau \tag{8l}$$

$$\sigma_{k+\tau|k} = \gamma r(z^a_{k+\tau|k}) \qquad \forall\tau \tag{8m}$$

$$R_{k+\tau|k} = \mathbf{I}_{2\times 2}\sigma_{k+\tau|k} \qquad \forall\tau \tag{8n}$$

$$y^{c_i}_{k+\tau+1|k} = C\hat{\chi}^{c_i}_{k+\tau+1|k} + n(\sigma_{k+\tau|k}) \qquad \forall i,\tau \tag{8o}$$

$$\delta u^a_{k+\tau+1|k} = (u^a_{k+\tau+1|k} - u^a_{k+\tau|k})^2 \qquad \forall\tau \tag{8p}$$

$$\chi^a_{k|k} \in \mathcal{X}, u^a_{k|k} \in \mathcal{U}, \delta u^a_{k|k} \in \delta\mathcal{U} \tag{8q}$$

$$b^{c_i|j}_{k+\tau|k}, \mathrm{b}^{c_i}_{k+\tau|k} \in \{0,1\} \qquad \forall j,i,q,\tau \tag{8r}$$

$$\mathrm{s}^{c_i}_{k+\tau|k} \in [0,..,4] \qquad \forall i,\tau \tag{8s}$$

$$j = [1,..,4], q = [1,2,3] \tag{8t}$$

For simplicity, the horizontal and vertical FoV have been split to halves. Thus, constrain (8d) can now be used to estimate the left, right, top and bottom limits of the FoV in the 2D Cartesian plane. Matrix $D_j \in \mathbb{R}^6$ is used to select the X or Y location from the agent's state vector. Using the FoV limits from (8d), we use constrains (8e), (8f) and (8g) to calculate whether a target is within the FoV or not. To do that, we check if a target is to the right of the left limit, to the left of the right limit, below the top limit and above the bottom limit of the FoV. Each check activates a binary variable (8e) using the big M technique, and the inclusion within the field of view is done by the summation of all these variables (8f). Matrix $E_j \in \mathbb{R}^3$ is used to select the X or Y location from the target's state vector. If the summation of the binaries is equal to 4 (8g), then another binary variable is activated, indicating that the target is within the FoV. In addition, constrains (8h) to (8l) describe the KF estimator. Constrain (8h) regards to the target's a priori state estimate using the transition matrix $A^c = \begin{bmatrix} \mathbf{I}_{2\times2} & \delta t\mathbf{I}_{2\times2} \\ \mathbf{0}_{2\times2} & \mathbf{I}_{2\times2} \end{bmatrix}$. The matrices $\mathbf{I}_{2\times2}$ and $\mathbf{0}_{2\times2}$ represent the identity and zero matrix respectively while δt defines the sampling interval. We use the KF only for the (X,Y) location of the target in the 2D Cartesian plane to reduce the complexity of the system. Estimating the target's position in the Z axis can be neglected due to its very small variations. Constrain (8i) computes the a priori covariance matrix. Matrix Q is constant and it represents the covariance of the process noise. Constrain (8j) computes the KF gain. Due to the inverse in this constrain, we needed to use non linear constrains in our formulation. Subsequently, constrain (8k) calculates the posteriori state estimate of the target using a psuedo-measurement derived in constrain (8o) and constrain (8l) calculates the posteriori covariance matrix of the target's state estimate. Similar to [30] we handle the intermittent observations of each target with the use of the technique described in [33]. However, we apply the technique within the proposed NMIP formulation with the planned horizon's observation binary variables $\mathrm{b}^{c_i}_{k+\tau|k}$. Further, constrain (8m) is used to create the measurement's standard deviation using the tuning parameter γ and function $r(z^a_{k+\tau|k})$ which returns a number in the range $[0,1]$ depending on the agent's altitude simulating the piece-wise function in Eq. (7) defined in Sec. IV-A. Thus, the measurement's standard deviation increases with the increase of altitude. The covariance of the observation noise is calculated using the aforementioned standard deviation as shown in constrain (8n). As explained above, in the planned horizon, we use a pseudo-measurement to estimate the target's location and covariance matrix in the future. The pseudo-measurement is derived in constrain (8o) and it is created by adding $n(\sigma_{k+\tau|k}) \sim \mathcal{N}(0, \sigma_{k+\tau|k})$, a zero mean Gaussian noise to the a priori state estimate of the castasway's location. Matrix $C \in \mathbb{R}^{2\times3}$ is the observation matrix and is used to get the X and Y position from the target's state vector. Constrain (8p) is employed to smooth the transition between the forces in the horizon by bounding the squared difference between each two consecutive forces in the horizon. Finally, Eq. (8q) ensures that the UAV is kept within the 3D space boundaries set by the $\mathcal{X}$ and the input vector is within the realistic capabilities $\mathcal{U}$ of the agent.

V. Simulation Experiments

This section elaborates on the simulation setup and procedure which was used to evaluate the proposed approach. Sec. V-A describes the setup used including the UAV's model parameters and the computation of the castaways ground truth trajectories. Using this setup, Sec. V-B provides simulation results and elaborates on key findings and insights.

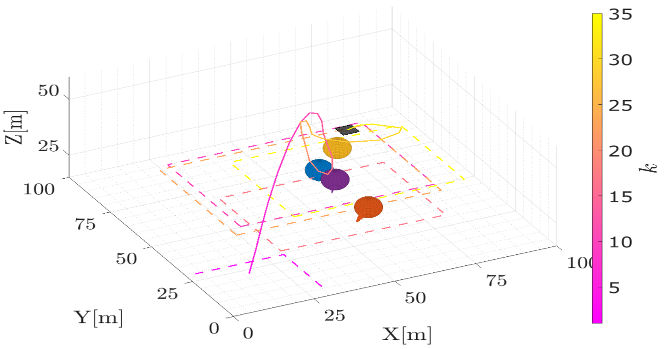

Fig. 6: Environment visualization during the first 35 time steps of the operation.

A. Simulation Setup

As described in Sec. III-D, we create ground truth castaway drifts that are subsequently used to evaluate our framework in simulation. The scenario considered the drift induced after one hour of simulation, with randomly generated wave sources of various amplitudes H, decay-rates c and wavelengths L. For all runs the water depth D was constant and both small amplitude ($q \cdot H \ll 1$) and deepwater ($K < 1$) conditions were satisfied. The ground truth drifts used in the simulation are plotted in Fig. 3 where each colored line corresponds to a castaway's path.

The measurements given by the stationary radar at the beginning of the mission and the measurements given by the onboard camera are generated by adding a zero mean Gaussian noise to the ground truth of each castaway. It should be noted that the variance of the added noise is different between the two sensors and in the case of the onboard camera it is proportional to the agent's altitude, with the radar sensor having the biggest variance. Also, the probability of detection is inversely proportional to the altitude of the UAV agent as described in Sec. III-C. In this work we do not investigate the problem of data association and assume that the agent is able to distinguish between each target.

For the agent's motion and input boundaries aforementioned in Sec. IV-B, we used realistic constrains that were taken from off-the-shelf UAVs readily available in the market. As such, the agent's velocity was limited to $\pm 11m/s$ in the horizontal axes and to $\pm 3m/s$ in the vertical axis. Acceleration was limited to $\pm 2m/s^2$ for all 3 dimensions while the FoV was set to $69°$ horizontally and $54°$ vertically. Finally, the number of steps in the rolling horizon was set to $N = 5$.

A 3D plot of the operation environment is shown in Fig. 6. There are four castaways represented by the yellow, blue, purple and orange spheres. The UAV agent is represented with a black square while the onboard camera's sensing area is visualized with the dashed rectangle. In this figure, the sensing area at each one of five sections is plotted as well as the executed path of the UAV is visualized with the colored line. The colormap represents the change in time.

TABLE I: Experimental Execution Times [s]

	$\mathcal{C}=2$	$\mathcal{C}=4$
$N=3$	0.1290	0.2898
$N=5$	0.3897	1.5880
$N=7$	1.3814	6.4169

B. Simulation Results

Several monte carlo simulations were conducted and in each simulation, we randomly placed the agent's initial position in close proximity to the castaways but not necessarily having the castaway's within the agent's FoV. Fig. 7 shows the first 35 time steps during the simulation which has been divided into 5 sections (a) to (e). In this figure, the behavior of the agent can be observed. The agent moves back and forth between the four targets with the aim of gathering measurements with the lowest altitude possible. This has the result of minimizing the covariance matrix of each target independently, since the measurement's noise is inversely proportional to the agent's altitude. If the agent cannot traverse between the targets in the given horizon due to physical limitations in its dynamical model, it increases its sensing area by increasing its altitude, thus achieving coverage of all the targets. Fig. 7 (f) we show both the variation of the agent's altitude along with the three control inputs. As mentioned in the NMIP objective (8a), our aim is to minimize the trace of the covariance matrix $P^{c_i}_{k+\tau|k}$ of each target. As such in Fig. 8 its shown that the covariance of each target is kept as low as possible during the duration of the simulation. The fluctuation on the value of each trace is caused by the back and forth behavior of the proposed controller. For ease of readability, only the first 35 time steps of the trace value were plotted and the trace values are shown in log scale across the y-axis.

Finally, we have investigated the computational complexity of the proposed approach. In order to do that, we have recorded the average time required to optimally solve the proposed NMIP problem, for various configurations of the planning horizon length (N) and number of targets ($\mathcal{C}$), as depicted in Table. I. Each setup was solved for 1800 times and the cumulative average execution time was recorded. Note that the numbers given in Table. I are the amount of seconds needed to solve the proposed NMIP once for the planned horizon. The simulations were conducted on a server machine with an Intel Xeon E5-2680 processor and 256GB of RAM using MATLAB R2020B and Gurobi solver 9.5.2. The results show that the computational complexity increases with the length of the planning horizon and the number of targets. This is attributed mainly to the increase of the number of binary variables that are required by the proposed controller, which increase linearly with the length of the planning horizon and the number of the targets. Although, it is well known that large mixed integer programs (MIPs) can be intractable, recent advances in optimization and machine learning have demonstrated that MIPs of moderate sizes can be solved in real-time [34]. A video animation of the whole MATLAB simulation can be accessed at *https://youtu.be/B5POJ7GV3-s*.

VI. Conclusion and Future Work

This work proposed a model predictive control approach for accurately tracking multiple castaways by computing the UAV's control inputs over a rolling planning horizon, which result in the minimization of the estimation error of the states of multiple moving castaway targets. It has been shown, that under the realistic scenario considered in this work, the proposed controller manages to track the multiple castaways reliably under various sea conditions (i.e. wave parameter configurations). Future work will investigate the extension of this approach to multiple UAV agents, and the design of a distributed multi-UAV system for this problem that can be applied in real-world settings. A non-linear motion model for the UAV agent will also be investigated to provide a more realistic behaviour.

Acknowledgment

This work was undertaken as part of the GLIMPSE project EXCELLENCE/0421/0586 which is co-financed by the European Regional Development Fund and the Republic of Cyprus through the Research and Innovation Foundation's RESTART 2016-2020 Programme for Research, Technological Development and Innovation and supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE), and from the Government of the Republic of Cyprus through the Cyprus Deputy Ministry of Research, Innovation and Digital Policy.

References

[1] I. Martinez-Alpiste, G. Golcarenarenji, Q. Wang, and J. M. Alcaraz-Calero, "Search and rescue operation using uavs: a case study," *Expert Systems with Applications*, vol. 178, p. 114937, 2021.

[2] S. Waharte and N. Trigoni, "Supporting search and rescue operations with uavs," in *2010 International Conference on Emerging Security Technologies*. IEEE, 2010, pp. 142–147.

[3] Y. Wu, K. H. Low, and C. Lv, "Cooperative path planning for heterogeneous unmanned vehicles in a search-and-track mission aiming at an underwater target," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6782–6787, 2020.

[4] A. Anastasiou, R. Makrigiorgis, P. Kolios, and C. Panayiotou, "Hyperion: A robust drone-based target tracking system," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2021, pp. 927–933.

[5] S. Papaioannou, P. Kolios, T. Theocharides, C. G. Panayiotou, and M. M. Polycarpou, "Uav-based receding horizon control for 3d inspection planning," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022, pp. 1121–1130.

[6] R. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, 2006.

[7] M. Terzi, A. Anastasiou, P. Kolios, C. Panayiotou, and T. Theocharides, "Swifters: A multi-uav platform for disaster management," in *2019 International Conference on Information and Communication Technologies for Disaster Management (ICT-DM)*. IEEE, 2019, pp. 1–7.

[8] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017.

[9] A. Anastasiou, P. Kolios, C. Panayiotou, and K. Papadaki, "Swarm path planning for the deployment of drones in emergency response missions," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 456–465.

[10] Allianz. (2022) Safety and shipping review 2021, page 12. [Online]. Available: https://www.agcs.allianz.com/content/dam/onemarketing/agcs/agcs/reports/AGCS-Safety-Shipping-Review-2021.pdf

[11] T. U. R. A. UNHCR. (2022) Mediterranean situation. [Online]. Available: https://data2.unhcr.org/en/situations/mediterranean

[12] A. Anastasiou, R. Makrigiorgis, and P. Kolios, "Aerial water buoys dataset," Oct. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7288444

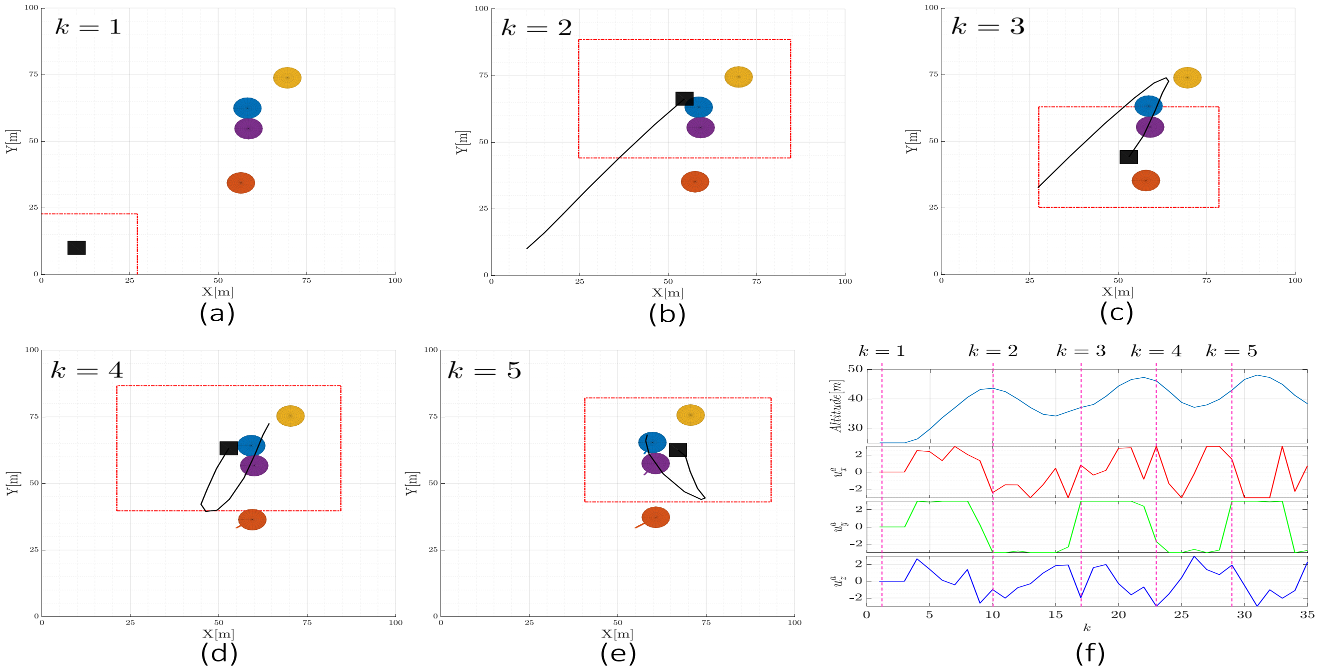

Fig. 7: Intermediate steps of the proposed solution. Red rectangle shows the onboard camera's sensing range. Blue, yellow, purple and orange spheres represent the targets. The black line shows the agent's trajectory for the the past 10 time steps.

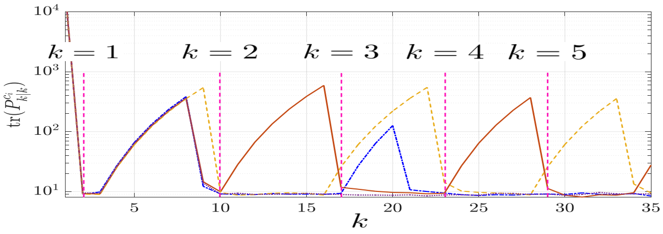

Fig. 8: Trace of each target's matrix $P^{c_i}_{k|k}$ for the first 35 time steps. Each line represents one of the targets trace covariance value in log scale.

[13] X. Hu, H. Fu, and X. Wu, "Research on technology of sea emergency rescue based on unmanned surface vehicle," in *Journal of Physics: Conference Series*, vol. 1575, no. 1. IOP Publishing, 2020, p. 012097.

[14] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[15] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[16] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "Uav assisted usv visual navigation for marine mass casualty incident response," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 6105–6110.

[17] A. Matos, A. Martins, A. Dias, B. Ferreira, J. M. Almeida, H. Ferreira, G. Amaral, A. Figueiredo, R. Almeida, and F. Silva, "Multiple robot operations for maritime search and rescue in eurathlon 2015 competition," in *OCEANS 2016-Shanghai*. IEEE, 2016, pp. 1–7.

[18] F. F. Ramírez, D. S. Benitez, E. B. Portas, and J. A. L. Orozco, "Coordinated sea rescue system based on unmanned air vehicles and surface vessels," in *OCEANS 2011 IEEE-Spain*. IEEE, 2011, pp. 1–10.

[19] F. Gustafsson, "Particle filter theory and practice with positioning applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 53–82, 2010.

[20] H. M. Jayaweera and S. Hanoun, "A dynamic artificial potential field (d-apf) uav path planning technique for following ground moving targets," *IEEE Access*, vol. 8, pp. 192 760–192 776, 2020.

[21] S. Yan, F. Pan, J. Xu, and L. Song, "Research on uav path planning based on improved artificial potential field method," in *2022 2nd International Conference on Computer, Control and Robotics (ICCCR)*, 2022, pp. 1–5.

[22] R. Dai and J. E. Cochran, "Path planning for multiple unmanned aerial vehicles by parameterized cornu-spirals," in *2009 American Control Conference*, 2009, pp. 2391–2396.

[23] N. Eslami and R. Amiadifard, "Moving target tracking and obstacle avoidance for a mobile robot using mpc," in *2019 27th Iranian Conference on Electrical Engineering (ICEE)*, 2019, pp. 1163–1169.

[24] K. Yang and S. Sukkarieh, "3d smooth path planning for a uav in cluttered natural environments," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 794–800.

[25] J. Tisdale, Z. Kim, and J. K. Hedrick, "Autonomous uav path planning and estimation," *IEEE Robotics Automation Magazine*, vol. 16, no. 2, pp. 35–42, 2009.

[26] S. Papaioannou, P. Kolios, C. G. Panayiotou, and M. M. Polycarpou, "Cooperative simultaneous tracking and jamming for disabling a rogue drone," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 7919–7926.

[27] 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, p. 2, 2021. [Online]. Available: https://doi.org/10.1007/s10846-021-01449-4

[28] ——, "3d trajectory planning for uav-based search missions: An integrated assessment and search planning approach," in *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 517–526.

[29] D. Agbissoh OTOTE, B. Li, B. Ai, S. Gao, J. Xu, X. Chen, and G. Lv, "A decision-making algorithm for maritime search and rescue plan," *Sustainability*, vol. 11, no. 7, p. 2084, 2019.

[30] C. Liu and J. K. Hedrick, "Model predictive control-based target search and tracking using autonomous mobile robot with limited sensing domain," in *2017 American Control Conference (ACC)*. IEEE, 2017, pp. 2937–2942.

[31] H. H. Shen and Y. Zhong, "Theoretical study of drift of small rigid floating objects in wave fields," *Journal of waterway, port, coastal, and ocean engineering*, vol. 127, no. 6, pp. 343–351, 2001.

[32] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," 2020.

[33] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, "Kalman filtering with intermittent observations," *IEEE transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[34] D. Bertsimas and B. Stellato, "Online mixed-integer optimization in milliseconds," *INFORMS Journal on Computing*, 2022.