Optimal Threat-Based Radar Resource Management for Multitarget Joint Tracking and Classification

Schöpe, Max Ian; Driessen, Hans; Yarovoy, Alexander

Publication date
2022

Document Version
Final published version

Published in
Journal of Advances in Information Fusion

Citation (APA)
Schöpe, M. I., Driessen, H., & Yarovoy, A. (2022). Optimal Threat-Based Radar Resource Management for Multitarget Joint Tracking and Classification. *Journal of Advances in Information Fusion*, *17*(1), 46-65.

Optimal Threat-Based Radar Resource Management for Multitarget Joint Tracking and Classification

MAX IAN SCHÖPE
HANS DRIESSEN
ALEXANDER YAROVOY

The Radar Resource Management problem in a multitarget joint tracking and classification scenario is considered. The problem is solved using a previously introduced dynamic budget-balancing algorithm that models the sensor tasks as Partially Observable Markov Decision Processes. It is shown that tracking and classification tasks can be considered as a single task type. Furthermore, it is shown how the task resource allocations can be jointly optimized using a carefully formulated cost function based on the task threat variance. Multiple two-dimensional radar scenarios demonstrate how sensor resources are allocated depending on the current knowledge of the target position and class. In contrast to previous approaches, the novelty of this paper lies in combining tracking and classification performance into a single cost function, preventing heuristic trade-offs.

Manuscript received September 29, 2021; revised May 25, 2022; released for publication July 7, 2022.

The authors are with the Microwave Sensing, Signals and Systems (MS3), Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: m.i.schope@tudelft.nl; j.n.driessen@tudelft.nl; a.yarovoy@tudelft.nl).

1557-6418/22/$17.00

I. INTRODUCTION

Due to the recent developments in multifunction radar (MFR), such systems have become more flexible and allow an automatic adjustment of many of their parameters during runtime [18]. Possible situations where such an adaptation is desirable are, e.g., quickly changing weather conditions, target maneuvers, or interference. This automatic control of the radar parameters or resources is often named Radar Resource Management (RRM). It is generally considered as a part of so-called cognitive radar (see, e.g, [11], [14], [18], [23], [32]). Possible applications can be found in many domains, such as traffic monitoring, autonomous driving, air traffic control, or (maritime) surveillance.

A. Radar Resource Management

Many different overviews of RRM approaches are available, for instance, by Moo and Ding in [45], Hero and Cochran in [25], or Hintz in [26]. Most RRM research focuses on a single task, e.g., guaranteeing consistent track quality even under target maneuvers. This commonly means that the available time budget has to be scheduled for a specific task. However, MFR systems are usually considered to operate at their resource limit (w.r.t., e.g., time or energy) and deal with a large number of different tasks. This means that increasing the resource budget for one task automatically reduces it for the others and inevitably deteriorates their performance, making the RRM problem a balancing problem.

As a solution for multitask RRM problems, several heuristic solutions have been presented in the past (see, e.g., the overview in [29]). Applying heuristics too early in the design makes it difficult to understand which problem is supposed to be solved. Additionally, it is challenging to judge the optimality of that solution. Moreover, the heuristic solutions frequently schedule tasks based on different priorities (as shown, e.g., in [42] and [53]). Such approaches usually assume that each task has a specific fixed resource need, which frequently leads to a situation where tasks need to be dropped. If different tasks have the same fixed priority, then this can potentially lead to tasks being dropped at random. Additionally, prioritizing tasks is usually tricky, and often it is not clear how many priority levels are necessary.

It should be noted that RRM algorithms are not identical to multitarget tracking algorithms. In this approach, a multitarget tracker is applied to process the radar measurements and provide the estimated tracks to the RRM algorithm, which then optimizes future radar transmissions. Although the proposed RRM approach comprises computing expected track accuracies, it does not represent the actual multitarget tracking and will not automatically lead to maximized track accuracies for every task.

This paper approaches the RRM problem as a resource balancing procedure and as an optimal stochastic

control problem. This strategy relies on an explicit formulation of

- the sensing problem that the radar needs to solve w.r.t. dynamic and measurement models,
- the control actions that the radar sensor has available, reflecting the degrees-of-freedom of the MFR mentioned earlier,
- a cost function that expresses the system performance that the user would like to optimize w.r.t. the sensing aim.

To the authors' best knowledge, an overall solution to the RRM problem based on such a problem solution approach has not been presented so far. A genuinely optimal solution could supposedly lead to a significant performance improvement of adaptive sensors [24], but that still needs to be illustrated. However, an optimality-based approach using a modular framework could significantly reduce the design effort for new MFR systems compared to heuristic solutions, even if the performance does not improve.

B. RRM for Tracking and Classification

For a successful radar application, it is often necessary to distinguish between different types of targets. Therefore, classification is a vital task for every modern radar system and needs to be considered in RRM. A general high-level overview of classification techniques in CR and RRM is shown by Brüggenwirth *et al.* in [14]. Furthermore, Kreucher and Hero presented a very generic framework that is potentially capable of doing joint detection, tracking, and classification [34]. The explanation of the approach stays at a very high level and is only demonstrated through a detection and tracking scenario.

Most RRM approaches for classification are myopic and focus on a simple waveform or sensor mode selection, often for a single object. In [55], Sowelam and Tewfik present such an approach where the Kullback–Leibler information is maximized for the subsequent measurement. Based on this, the algorithm can decide if another measurement is necessary and which waveform must be chosen from a predefined library. Another example has been shown by Bell *et al.* and considers both tracking and classification [2]. The system is assumed to have separate tracking and classification modes, which each have a predefined waveform library to choose from. The proposed algorithm decides the following sensor action to be executed. As for objective functions, both task-driven and information-driven possibilities are discussed. While the task-driven approach requires different objective functions for the two sensing modes, the information-driven approach can compare the two different task types through information gain.

A popular approach is to introduce a measure of risk or threat. The idea is to summarize the interesting task quantities into a single scalar number that is easy to compare. In [39], Martin introduces a risk-based approach where the risk depends on the probability of making a wrong classification and the possibility of track loss multiplied with predefined cost values. The approach finds a solution for both tracking and classification in a myopic fashion. The measurements are always taken the same way, but the algorithm decides which target will be sensed. In [22] and [47], similar approaches are presented. From the perspective of this paper, such a cost function definition is not preferred, as predefined cost values cannot easily represent the risk in all possible situations. Such a formulation leads to a lack of flexibility in the approach. Bolderhij *et al.* present an approach for military radar applications that relies on a large amount of expert knowledge to decide the risk level [12]. Although many different situations are considered, this approach does not automatically balance the resources and cannot flexibly adapt to different situations. A more interesting approach is shown by Katsilieris *et al.* in [30] where joint tracking and classification are performed by running a tracking filter per target class in parallel. The classification is done by comparing the likelihood of a measurement belonging to the different tracks. The next sensing action is then chosen by evaluating the threat's uncertainty based on the target state.

Additionally, some authors have introduced RRM approaches with a hierarchical structure. This is usually done for two main reasons. Firstly, such a structure can decrease the computational complexity and increase the efficiency of an algorithm. Two notable examples of such approaches are the ones by Wintenby and Krishnamurthy in [60] and Castãnón in [15], which both use a hierarchical structure to solve the RRM problem using Lagrangian relaxation (LR). Secondly, a hierarchical structure can also be used to emulate the cognitive behavior of the human brain in order to improve radar performance. An example for such an approach has been proposed by Mitchell *et al.* in [44] and Mitchell in [43].

This paper treats the RRM problem as an optimal control problem. It is not the intention to mimic the behavior of human or animal brains. Furthermore, the functional performance is the focus of the proposed approach rather than a computationally efficient implementation. However, the RRM approach in this paper can potentially be applied in a hierarchical fashion, which might lead to improvements that are beneficial for practical implementation.

C. Markov Decision Processes in RRM for Tracking and Classification

Many RRM approaches for tracking and classification of multiple targets assume a Markov Decision Process (MDP) or Partially Observable MDP (POMDP) framework.

Wintenby and Krishnamurthy have presented an interesting RRM approach for tracking scenarios in [60]

where a Markov chain consisting of performance states is applied for each tracking task. The problem is then solved using a combination of LR and approximate dynamic programming. Furthermore, White and Williams assume a discretized state space and solve a fully observable MDP by applying dynamic programming [59] in combination with LR. Similar to LR, some approaches also apply the Quality of Service resource allocation method (Q-RAM) with POMDPs to solve the RRM tracking problem. Two examples of such an approach can be found in [17], [19].

For simple classification problems, it is relatively easy to implement a solution assuming underlying MDPs or POMDPs, as the number of considered states is usually relatively low. An advantage of using such a framework is the possibility of taking the expected future into account. Chong *et al.* present a straightforward general example of how to use classification in RRM with POMDPs [20]. Castãnón applies a nonmyopic POMDP approach for a classification scenario of almost 10 000 objects where the algorithm chooses from a set of sensor modes [15]. This approach does not take the position and velocity of the objects into account. Another interesting approach has been presented by La Scala *et al.* in [37] and nonmyopically solves the underlying POMDP in a detection and classification scenario. The algorithm selects the best waveforms from a predefined library. The authors promise that an extension to tracking can be achieved without much effort, but do not demonstrate that explicitly. Two other classification approaches that assume an MDP or POMDP framework are shown in [22], [27]. Since the number of states is relatively low in most classification problems, also machine learning techniques have been suggested for solving the underlying (PO)MDPs, e.g., by Langford and Zadrozny in [36] and Blatt and Hero in [10].

D. The Cost Function

An optimization-based RRM approach is preferred over a heuristic approach. However, this requires an explicit definition of a cost function that determines the sensor system's performance. It has been suggested previously that generic measures, such as the Information Gain or the Renyi divergence, applied to the posterior density of the full state, could be applied (see, e.g., [35], [58]). Unfortunately, one single cost function will not meet the expectations of different users in different applications using different sensor systems in different environments and for different targets (see, e.g., [21]).

Developing a useful cost function is critical for the good performance of the RRM algorithm. The development of specific cost functions requires close cooperation with future users and is an intricate development process on its own. Since the focus of this paper is to illustrate how the underlying framework and algorithmic solution can be applied in an example scenario, the development of a user-specific cost function is out of scope. Therefore, it is not claimed that the presented cost function formulation is necessarily leading to the best performance possible.

An example of a more specific operationally relevant cost function can be found in the approach by Narykov *et al.* in [46] where the adversarial risk is introduced as a cost function in a military impact assessment scenario.

E. Proposed Approach

This paper is based on the framework and algorithmic solution presented in [51] and [50], which was previously mainly illustrated in multitarget tracking scenario, i.e., without classification. Here, the framework and approach are applied to a joint tracking and classification problem.

Most approaches that focus on RRM for classification are concrete and apply heuristic rules to compare different task types, such as tracking and classification tasks. In most proposed approaches, the different tasks are assumed to be independent or very weakly dependent. This paper specifically focuses on cases where the tasks are joint. In addition to that, most available approaches are myopic and do not consider MDP or POMDP frameworks. The introduction of risk or threat measures is widespread and seems promising as it enables the objective comparison of different task types. In this paper, it is shown that the generic algorithm introduced in previous publications can be used to address the shortcomings of previously published literature. It is explained how the approach can easily be adjusted to include joint tracking and classification, using a single cost function for both task types. The underlying POMDP is solved nonmyopically, and the resulting policy is achieved by balancing all the considered actions in the action space.

The purpose of this paper is to introduce as few assumptions and simplifications as possible and formulate the RRM problem as a single optimization problem. However, the techniques shown here could potentially be applied in a hierarchical algorithm as well, taking into account higher and lower levels of optimization.

F. Novelty

This paper shows that it is possible to solve the RRM problem for multiple task types by using only a single cost function based on a definition of a mission threat. Such an approach has been suggested previously but has never been fully developed and demonstrated with the help of practical simulation scenarios.

G. Structure of the Paper

The remainder of this paper is structured as follows: Section II defines the general RRM problem, and Section III introduces the proposed solution for a tracking and classification scenario. Furthermore, Section IV

introduces the applied threat and cost function, while Section V gives details about the assumed radar scenario for the simulations. The results of those simulations are discussed in Sections VI–IX. Finally, Section X contains the conclusions.

II. GENERAL RRM PROBLEM DEFINITION

A. Motion Model

Each target can be described by a state based on its position and velocity in x and y directions in a two-dimensional Cartesian coordinate system. For Target n at time t, this state is defined as

$$\boldsymbol{s}_t^n = [x_t^n \quad y_t^n \quad \dot{x}_t^n \quad \dot{y}_t^n]^T, \tag{1}$$

where x_t^n, y_t^n, and $\dot{x}_t^n, \dot{y}_t^n$ are the position and velocity of Target n in x and y, respectively. The future target state at time $t + \Delta t$ can be calculated following a function:

$$\boldsymbol{s}_{t+\Delta t}^n = f_{\Delta t}\left(\boldsymbol{s}_t^n, \boldsymbol{w}_t^n\right), \tag{2}$$

where $s_{t+\Delta t}^n$ is the next state at time $t + \Delta t$ and $\boldsymbol{w}_t^n \in \mathbb{R}^4$ is the maneuverability noise for Target n at time t. The state evolution equation (2) directly defines the evolution probability density function (PDF), which is given as

$$p\left(\boldsymbol{s}_{t+\Delta t}^n | \boldsymbol{s}_t^n\right). \tag{3}$$

B. Measurement Model

It is assumed that the considered sensor is taking noisy measurements of the state $\boldsymbol{s}_t^n$ by executing a sensor action $\boldsymbol{a}_t^n \in \mathbb{R}^m$, where m denotes the amount of adjustable action parameters. The measurement $\boldsymbol{z}_t^n$ of Target n at time t is expressed by the measurement function $\mathfrak{h}$ as

$$\boldsymbol{z}_t^n = \mathfrak{h}\left(\boldsymbol{s}_t^n, \boldsymbol{v}_t^n, \boldsymbol{a}_t^n\right), \tag{4}$$

where $\boldsymbol{v}_t^n \in \mathbb{R}^q$ is the measurement noise for Target n, and q is the amount of measurement parameters. The measurement equation (4) directly defines the measurement PDF, which can be written as

$$p\left(\boldsymbol{z}_t^n | \boldsymbol{s}_t^n, \boldsymbol{a}_t^n\right). \tag{5}$$

C. Tracking Algorithm

As this paper considers joint tracking and classification scenarios, a tracking filter needs to be chosen that aims at computing the posterior density. A simple Kalman filter (KF) would be the exact solution for a linear system. In contrast, nonlinear systems require approximate solutions given by, e.g., an extended KF (EKF) or a particle filter.

D. Budget Optimization Problem

The radar sensor has a limited maximum budget $\Theta_{\max}$ of any kind. For action $\boldsymbol{a}_t^n$ that is executed for each task n, a specific amount of budget (such as time or energy allocations) is required. An overload situation occurs when the current tasks require more of the total budget than is available. In such a situation, the available budget needs to be allocated to all tasks by minimizing the cost (e.g., related to the uncertainty of the current situation).

At time t, the optimization problem for N different tasks can be written as

$$\begin{aligned} \underset{\boldsymbol{a}_t}{\text{minimize}} \quad & \sum_{n=1}^{N} c(\boldsymbol{a}_t^n, \boldsymbol{s}_t^n) \\ \text{subject to} \quad & \sum_{n=1}^{N} \Theta_t^n(\boldsymbol{a}_t^n) \leq \Theta_{\max}, \end{aligned} \tag{6}$$

where $\Theta_t^n \in [0, 1]$ is the budget for task n at time t, $c(\cdot)$ is the applied cost function, and $\Theta_{\max} \in [0, 1]$ is the maximum available budget (0: no budget assigned, 1: all budget assigned). It is critical for the performance of the algorithm to define a relevant cost function. However, this is not the focus of this paper. An example of another operationally relevant cost function has been discussed by Katsilieris *et al.* [31].

III. PROPOSED APPROACH FOR THE RRM PROBLEM

A. Joint Tracking and Classification

This paper assumes that each target is of a specific predefined class that is initially unknown to the radar system. A Bayesian classifier will be applied to make the classification decision. Suppose a class feature could be measured directly and the features were independent of each other. In that case, the classification problem can be solved, e.g., by applying a naive Bayes classifier using these class measurements directly.

If the class features cannot be observed directly, then the behavior of the target often contains information about the underlying target type. In that case, joint tracking and classification can be applied. Similar approaches have been presented, e.g., in [1], [40]. Based on the measurements taken by the radar sensor, a track can be created with the help of a tracking filter (e.g., EKF or particle filter). The track then describes the movements of the observed objects. The problem that needs to be solved contains both discrete (class) and continuous variables (e.g., position and velocity) of the targets are considered. The following equations are based on Bayesian theory (see, for instance, [4], [49], [57]).

Taking into account the class of the target, the state evolution equation in (2) changes to

$$\boldsymbol{s}_{t+\Delta t}^n = f_{\Delta t}\left(\boldsymbol{s}_t^n, c^n, \boldsymbol{w}_t^n\right), \tag{7}$$

where $c^n \in \boldsymbol{C}$ is a scalar and denotes the class of Target n, which is not changing over time. The measurement function is defined similarly to (4):

$$\boldsymbol{z}_t^n = \mathfrak{h}\left(\boldsymbol{s}_t^n, \boldsymbol{v}_t^n, \boldsymbol{a}_t^n, f_c^n\right), \tag{8}$$

where f_c^n is a directly measurable feature of Target n, represented as a scalar value. The PDF function for feature f_c^n can explicitly be written as

$$p\left(z_{t,f}^n | \text{SNR}, c^n\right) = \frac{}{\sqrt{2\pi\sigma_f^2(\text{SNR})}} \exp\left(-\frac{(z_{t,f}^n - f_c^n)^2}{\sigma_f^2(\text{SNR})}\right), \tag{9}$$

where $z_{t,f}^n$ is the measurement of the feature of Target n at time t, and $\sigma_f^2(\text{SNR})$ is the feature measurement variance which depends on the signal-to-noise ratio (SNR). One could think, for instance, of the RCS or the micro-Doppler spectrum. For simplicity, the values used in this paper do not have any physical origin and are merely chosen for demonstration purposes. This feature is assumed to be directly connected to the object's class and not dependent on the state of the target. Therefore, the measurement $\boldsymbol{z}_t^n$ consists of a state (position and Doppler) and a class (feature) component. The PDFs of state, process, or maneuverability noise and measurement noise can then depend on the underlying target class:

$$\begin{aligned} &p\left(\boldsymbol{s}_{t+\Delta t}^n | c^n\right), \\ &p\left(\boldsymbol{w}_t^n | c^n\right), \\ &p\left(\boldsymbol{v}_t^n | c^n\right) \\ &p\left(f_c^n | c^n\right). \end{aligned} \tag{10}$$

The goal of this joint tracking and classification approach is to recursively calculate the posterior joint PDF

$$p\left(\boldsymbol{s}_t^n, c^n | \boldsymbol{Z}_t^n\right) = p\left(\boldsymbol{s}_t^n | c^n, \boldsymbol{Z}_t^n\right) P(c^n | \boldsymbol{Z}_t^n), \tag{11}$$

where $\boldsymbol{Z}_t^n = [\boldsymbol{z}_t^n, \boldsymbol{z}_{t-\Delta t}^n, \boldsymbol{z}_{t-2\Delta t}^n, \ldots, \boldsymbol{z}_0^n]$ are all measurements taken for Target n until time t and $P(c^n|\boldsymbol{Z}_t^n)$ are the prior class probabilities, which are known from the last iteration. Using the Bayesian evolution and update equations, the conditional posterior density can be written as

$$p\left(\boldsymbol{s}_{t+\Delta t}^n | c^n, \boldsymbol{Z}_t^n\right) = \int_S p\left(\boldsymbol{s}_{t+\Delta t}^n | \boldsymbol{s}_t^n, c^n\right) p\left(\boldsymbol{s}_t^n | c^n, \boldsymbol{Z}_t^n\right) d\boldsymbol{s}_t^n, \tag{12}$$

where

$$p\left(\boldsymbol{s}_t^n | c^n, \boldsymbol{Z}_t^n\right) = \frac{p\left(\boldsymbol{z}_t^n | \boldsymbol{s}_t^n, c^n\right) p\left(\boldsymbol{s}_t^n | c^n, \boldsymbol{Z}_{t-\Delta t}^n\right)}{p\left(\boldsymbol{z}_t^n | c^n, \boldsymbol{Z}_{t-\Delta t}^n\right)}. \tag{13}$$

The normalizing constant in the denominator is calculated with

$$p\left(\boldsymbol{z}_t^n | c^n, \boldsymbol{Z}_{t-\Delta t}^n\right) = \int_S p\left(\boldsymbol{z}_t^n | \boldsymbol{s}_t^n, c^n, \boldsymbol{Z}_{t-\Delta t}^n\right) p\left(\boldsymbol{s}_t^n | c^n, \boldsymbol{Z}_t^n\right) d\boldsymbol{s}_t^n. \tag{14}$$

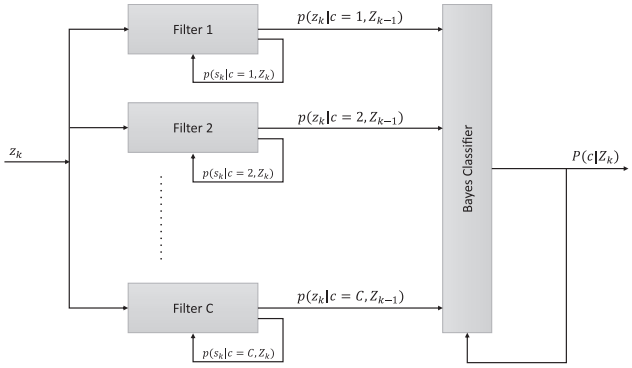


Figure 1. Joint tracking and classification process.

As the measurement consists of a state dependent and a state independent part, which is based only on the class, this expression can also be written as

$$p\left(\boldsymbol{z}_t^n | c^n, \boldsymbol{Z}_{t-\Delta t}^n\right) = p\left(\boldsymbol{z}_t^{n,s,c} | c^n, \boldsymbol{Z}_{t-\Delta t}^{n,s}\right) p\left(\boldsymbol{z}_t^{n,c} | c^n\right), \tag{15}$$

where $\boldsymbol{z}^{n,s,c}$ denotes the state and class dependent measurements and $\boldsymbol{z}^{n,c}$ the class dependent measurement of feature f_c^n for Target n at time t. The posterior class probability is calculated via

$$P\left(c^n | \boldsymbol{Z}_t^n\right) = \frac{p\left(\boldsymbol{z}_t^n | c^n, \boldsymbol{Z}_{t-\Delta t}^n\right) P\left(c^n | \boldsymbol{Z}_{t-\Delta t}^n\right)}{p\left(\boldsymbol{z}_t^n | \boldsymbol{Z}_{t-\Delta t}^n\right)}. \tag{16}$$

The likelihood of the current measurement given all the previous measurements is defined as

$$p\left(\boldsymbol{z}_t^n | \boldsymbol{Z}_{t-\Delta t}^n\right) = \sum_{c=1}^{C} p(\boldsymbol{z}_t^n | c^n, \boldsymbol{Z}_{t-\Delta t}^n) P\left(c^n | \boldsymbol{Z}_{t-\Delta t}^n\right), \tag{17}$$

where C is the number of assumed target classes. The recursive process that is described through (7)–(17) requires C different tracking filters, each conditioned to a specific class. Based on the likelihood of the current measurement being associated with one of the tracks, the class probability is updated. The process is summarized in Fig. 1.

B. Distribution of the Sensor Budgets Using LR

The approach presented here is based on the algorithmic solution presented in [51] and [50]. It applies LR to relax the problem by including the constraint into the cost function. This results in the so-called Lagrangian dual (LD). The original optimization problem is then decoupled into suboptimization problems, one for each task. This leads to the Lagrangian dual problem (LDP), which can be formulated as

$$Z_D = \max_{\lambda_t}\left(\min_{\boldsymbol{a}_t}\left(\sum_{n=1}^{N}\left(c(\boldsymbol{a}_t^n, \boldsymbol{s}_t^n) + \lambda_t \cdot \Theta_t^n\right)\right) - \lambda_t \cdot \Theta_{\max}\right), \tag{18}$$

where $\lambda_t \in \mathbb{R}$ is the Lagrange multiplier for the resource budget constraint. The sum in the LDP allows the algorithm to solve the minimization problem in parallel for each Target n before updating the Lagrangian multiplier

in an iterative process using the subgradient method. This process is explained following, where an internal index l is used for the iterations within the LR process:

1) $l = 0$: Set the initial Lagrange multiplier ($\lambda = \lambda_0$).
2) For every task n, minimize the LD with respect to the actions. Keep the resulting $\boldsymbol{a}_l^n$ and Θ_l^n.
3) Choose the subgradient for the Lagrange multiplier as $\mu_l^\lambda = \sum_{n=1}^{N} \Theta_l^n - \Theta_{\max}$.
4) Is $\mu_l^\lambda \approx 0$ reached with desired precision? If yes, then stop the process. The current λ_l, $\boldsymbol{a}_l^n$ and Θ_l^n are the LR solution for λ_t at time t.
5) Set $\lambda_{l+1} = \max\{0, \lambda_l + \gamma_l \mu_l^\lambda\}$, where γ_l is the LR step size at time l. In this step, the LD is iteratively maximized with respect to λ.
6) Go to step 2 and set $l = l + 1$.

Further information regarding LR can be found in Appendix A, as well as in [50]–[52].

C. Definition of a POMDP

A POMDP is defined as an MDP whose state cannot be observed directly. Instead, the state can be observed through noisy measurements, leading to a probability distribution over the possible states called the belief state. Knowing the structure of the underlying MDP and having noisy measurements available, the POMDP framework allows solving optimization problems nonmyopically, which means calculating the expected cost in future time steps. For the following equations, the time is assumed to be discretized in intervals k with length T, the time between two consecutive measurement operations.

A POMDP is commonly defined by the following parameters (see, e.g., [48] and [20]):

State space $\boldsymbol{S}$: All possible states that can be reached within the process, see (1). At time step k the state is defined as $\boldsymbol{s}_k$. The belief-state defines a probability distribution over all possible states based on the previous measurements and is defined as $\boldsymbol{b}_k$.

Action space $\boldsymbol{A}$: All possible actions within the process. Each executed action leads to a certain cost defined by the cost function. The action at time step k is written as $\boldsymbol{a}_k$.

Observation space $\boldsymbol{Z}$: All possible observations that can be made within the process. An observation at time step k it is defined as $\boldsymbol{z}_k$.

Transition probability $\Psi(\boldsymbol{s}_k, \boldsymbol{s}_{k+1}, \boldsymbol{a}_k)$: The probability function $p(\boldsymbol{s}_{k+1}|\boldsymbol{s}_k, \boldsymbol{a}_k)$ that defines the probability of transitioning from state $\boldsymbol{s}_k$ to state $\boldsymbol{s}_{k+1}$ given action $\boldsymbol{a}_k$. Note: In this paper, the transition probability does not depend on the action.

Probability of observation $\boldsymbol{O}(\boldsymbol{z}_k, \boldsymbol{s}_{k+1}, \boldsymbol{a}_k)$: The probability function $p(\boldsymbol{z}_k|\boldsymbol{s}_{k+1}, \boldsymbol{a}_k)$ that defines the probability to make a certain observation $\boldsymbol{z}_k$ when action $\boldsymbol{a}_k$ is executed with the resulting state being $\boldsymbol{s}_{k+1}$.

Cost function $c(\boldsymbol{s}_k, \boldsymbol{a}_k)$: The immediate cost of executing action $\boldsymbol{a}_k$ in state $\boldsymbol{s}_k$.

Discount factor γ: A possible factor that discounts future costs. Note: in this paper, the discount factor is always set to $\gamma = 1$.

D. Policy Rollout for POMDPs

A variety of different POMDP solution methods exist. A short general discussion of possible approaches can be found in Appendix B, or the overviews by Ross *et al.* in [48] and Chong *et al.* in [20].

In this paper, the policy rollout (PR) technique is applied, which takes Monte Carlo samples of the expected future. This means that it stochastically explores the possible future actions and their related costs. Per possible action $\boldsymbol{a}$ in the action space $\boldsymbol{A}$, a so-called rollout is used to evaluate the expected cost. Expected observations and belief states are generated from a given initial belief state and a given candidate action within such a rollout. The candidate action is executed first, while a so-called base policy (BP) π_{base} is used for every following step in the rollout, until the horizon $\mathscr{H}$ is reached. The cost of all steps within a rollout is summed up. This procedure is repeated M times, and finally, the cost of all rollouts is averaged. The resulting number is then called the expected cost of the evaluated action. The candidate action with the lowest expected cost is chosen to be executed in the next time step. It has been shown that PR leads to a policy that is at least as good as the BP with a very high probability if enough samples are provided [6]. Choosing a good BP and a large enough number of samples is therefore crucial to the algorithm's performance. In this case, the number of samples is equivalent to the number of rollouts M per action. Therefore, one sample is the evaluation of one possible future. It is no trivial task to find a good BP for the considered scenario. As an example, previously experienced situations could be used for it, e.g., as a lookup table. Additionally, the BP could also be improved with new information over time, which could be considered in RL, for instance. Unfortunately, it is not very likely to experience the exact same situation multiple times if a huge state space is assumed, so the usefulness of RL techniques is questionable for typical radar scenarios. Another straightforward choice of the BP could be an equal resource allocation to all the tasks. PR has been covered extensively, e.g., by Bertsekas in [5]–[7].

The PR can be expressed as shown in (19) and (21). The Q-value is defined as

$$Q^{\pi_{\text{base}}}(\boldsymbol{b}_k, \boldsymbol{a}_k) = C_B(\boldsymbol{b}_k, \boldsymbol{a}_k) + E\left[V^{\pi_{\text{base}}}(\boldsymbol{b}_{k+1})|\boldsymbol{b}_k, \boldsymbol{a}_k\right], \tag{19}$$

where $C_B(\boldsymbol{b}_k, \boldsymbol{a}_k) = \sum_{s \in S} \boldsymbol{b}_k(\boldsymbol{s}) c(\boldsymbol{s}, \boldsymbol{a}_k)$ being the expected cost given belief state $\boldsymbol{b}_k$, $E[\cdot]$ is the expectation and $V^{\pi_{\text{base}}}(\cdot)$ is the so-called value function assuming the

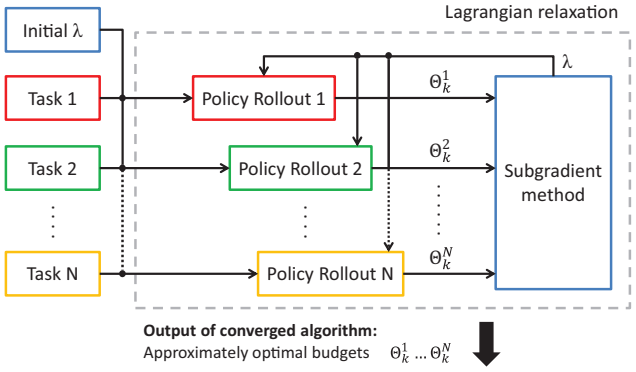


Figure 2. High-level block scheme of the AODB algorithm [50].

chosen BP. The value function can be expressed as

$$V^{\pi_{\text{base}}}(\boldsymbol{b}_{k_0}) = E\left[\sum_{k=k_0}^{k_0+\mathscr{H}} C_B(\boldsymbol{b}_k, \boldsymbol{a}_k)|\boldsymbol{b}_{k_0}\right]. \tag{20}$$

The best policy can then be found by applying

$$\pi_k(\boldsymbol{b}_k) = \arg\min_{a_k \in A}(Q^{\pi_{\text{base}}}(\boldsymbol{b}_k, \boldsymbol{a}_k)). \tag{21}$$

PR does not necessarily lead to the optimal policy. It instead aims at improving the chosen BP π_{base}.

E. Approximately Optimal Dynamic Budget Balancing

This paper uses the Approximately Optimal Dynamic Budget Balancing (AODB) algorithm as introduced in [51] and [50], which applies a combination of PR and LR. The general structure of our proposed algorithm is illustrated in Fig. 2. The outputs of the algorithm are the converged budgets for each task. The PR is applied per task, which in this paper means that for each observed object, the expected cost for each action is calculated, taking into account the current class probabilities for all possible classes.

IV. FORMULATION OF THE COST FUNCTION

The assumed cost function in this paper is based on a definition of threat. This definition depends heavily on the considered scenario and the wishes and expectations of the user. There are practically an unlimited amount of possibilities for constructing such a function. In this section, it is considered that the threat $\phi(c, \boldsymbol{s})$ depends on the class and the state of a target. The cost function will be defined by the variance in the threat knowledge of a target. This means that the cost will be very high for unclassified targets, as all class-dependent threat values are equally likely. Once the knowledge of the target class increases, this variance in threat will decrease also. An explicit example formulation of the threat and the cost function will be introduced later, together with the simulation scenarios. First, the focus is on transforming the PDF from the state domain to the threat domain. As the cost calculation is done for each target separately, the target-related superscript n is dropped to simplify the notations in the following subsections.

A. Unscented Transform

The running target tracks supply a PDF of the target state. Since the transformation of the state PDF to the threat PDF is nonlinear, a sampling approach is chosen. A possible implementation is to sample the threat PDF with a certain number of random samples in the state PDF. For an accurate result, many samples are necessary, which can make this approach very slow. Therefore, in this paper, the samples in the state space of the target are chosen with the help of the unscented transform that is also applied in the unscented KF [28]. For a D-dimensional PDF, $2D + 1$ sigma points are necessary. The procedure for calculating the current threat at a certain moment in time is as follows:

1) Calculate the Cholesky decomposition of the belief state covariance matrix of the target:
$$\boldsymbol{L}\boldsymbol{L}^T = \boldsymbol{P}, \tag{22}$$
where $\boldsymbol{P}$ is the belief state covariance matrix of the target.
2) Calculate the so-called sigma points:
$$\begin{aligned} \mathbb{X}^0 &= \hat{\boldsymbol{s}}, \\ \mathbb{X}^i &= \hat{\boldsymbol{s}} + \sqrt{D+\kappa}\ \text{col}_i\boldsymbol{L} \quad i = 1, \ldots, D, \\ \mathbb{X}^{i+D} &= \hat{\boldsymbol{s}} - \sqrt{D+\kappa}\ \text{col}_i\boldsymbol{L} \quad i = 1, \ldots, D, \end{aligned} \tag{23}$$
where $\hat{\boldsymbol{s}}$ is the belief state mean of the target, $\kappa = 3-D$ and $\text{col}_i\boldsymbol{L}$ denotes the i-th column of matrix $\boldsymbol{L}$.
3) Now, each of these samples has to be transformed into the threat domain by using the threat function $\phi(c, \boldsymbol{s})$.
$$\mathbb{Y}_c^i = \phi(\hat{c}, \mathbb{X}^i)) \quad i = 0, \ldots, 2D, \tag{24}$$
where $\hat{c}$ is the believed class of the target.
4) From the samples in the threat domain, the threat PDF is defined by the mean and covariance:
$$\begin{aligned} \hat{\phi}_c &= \sum_{i=0}^{2D} w^i \mathbb{Y}_c^i, \\ \Sigma_{\phi,c} &= \sum_{i=0}^{2D} w^i (\mathbb{Y}_c^i - \hat{\phi}_c)(\mathbb{Y}_c^i - \hat{\phi}_c)^T, \end{aligned} \tag{25}$$
where $\hat{\phi}_c$ is the mean and $\Sigma_{\phi,c}$ the covariance of the threat PDF based on class c and w^i are weights for the samples given as
$$w^i = \begin{cases} \frac{\kappa}{D+\kappa}, & \text{if } i = 0 \\ \frac{}{2(D+\kappa)}, & \text{otherwise} \end{cases}. \tag{26}$$

B. Combination of Threat PDFs

Since the threat PDF depends on the class c, is has to be calculated for each class separately. Based on the resulting PDFs for C different classes, a total PDF can be constructed. The total mean of the threat $\hat{\phi}_{\text{tot}}$ for the target is defined as

$$\begin{aligned}\hat{\phi}_{\text{tot}} &= \int_{\Phi} \phi p(\phi|\boldsymbol{z}) d\phi \\ &= \int_{\Phi} \phi \sum_{c=1}^{C} \mathcal{N}(\phi; \hat{\phi}_c, \Sigma_{\phi,c}) P(c|\boldsymbol{Z}) d\phi \\ &= \sum_{c=1}^{C} P(c|\boldsymbol{Z}) \int_{\Phi} \phi \mathcal{N}(\phi; \hat{\phi}_c, \Sigma_{\phi,c}) d\phi \\ &= \sum_{c=1}^{C} P(c|\boldsymbol{Z}) \hat{\phi}_c,\end{aligned} \tag{27}$$

where $\boldsymbol{z}$ is a recent measurement of the target state, $\mathcal{N}(\phi; \hat{\phi}_c, \Sigma_{\phi,c})$ denotes a normal distribution with mean $\hat{\phi}_c$ and variance $\Sigma_{\phi,c}$, and $P(c, \boldsymbol{Z})$ is the posterior class probability based on all previous measurements $\boldsymbol{Z}$. The variance can be calculated using

$$\begin{aligned}\Sigma_{\phi,\text{tot}} &= \int_{\Phi} (\phi - \hat{\phi}_{\text{tot}})^2 p(\phi|\boldsymbol{z}) d\phi \\ &= \int_{\Phi} (\phi^2 - 2\phi\hat{\phi}_{\text{tot}} + \hat{\phi}_c)^2) p(\phi|\boldsymbol{z}) d\phi \\ &= \int_{\Phi} \phi^2 p(\phi|\boldsymbol{z}) d\phi - \hat{\phi}_{\text{tot}}^2.\end{aligned} \tag{28}$$

Using

$$\begin{aligned}\int_{\Phi} \phi^2 p(\phi|\boldsymbol{z}) d\phi &= \int_{\Phi} \phi^2 \sum_{c=1}^{C} \mathcal{N}(\phi; \hat{\phi}_c, \Sigma_{\phi,c}) P(c|\boldsymbol{Z}) d\phi \\ &= \sum_{c=1}^{C} P(c|\boldsymbol{Z}) \int_{\Phi} \phi^2 \mathcal{N}(\phi; \hat{\phi}_c, \Sigma_{\phi,c}) d\phi \\ &= \sum_{c=1}^{C} P(c|\boldsymbol{Z}) (\Sigma_{\phi,c} + \hat{\phi}_c^2),\end{aligned} \tag{29}$$

it can also be written as

$$\Sigma_{\phi,\text{tot}} = \sum_{c=1}^{C} P(c|\boldsymbol{Z}) (\Sigma_{\phi,c} + \hat{\phi}_c^2) - \hat{\phi}_{\text{tot}}^2. \tag{30}$$

C. Variance of Threat

The previous subsection described transforming the PDF from the state and class domain to the threat domain by considering multiple possible target classes. Given this threat PDF, a different cost function could be constructed. A simple and unambiguous choice is to simply evaluate the total threat variance $\Sigma_{\phi,\text{tot}}$. The underlying assumption is that the radar system cannot influence the target state but only the uncertainty about the knowledge of the target state by adjusting its sensing actions. Following this cost function, the most resources will be assigned to the targets where the biggest decrease in uncertainty (decrease in threat variance) is expected.

The hypothesis is that this will lead to more resources being assigned to objects of an uncertain class. Once all the objects are classified, the uncertainty in the threat will drop significantly and only depend on the uncertainty in the track. This emphasizes the jointness of the proposed tracking and classification approach, as the uncertainty in both the tracking and classification classes is directly taken into account through this cost function. For the remainder of this paper, the cost function will thus be defined as

$$\mathcal{C}(\boldsymbol{a}, \boldsymbol{s}_{k|k-1}, \boldsymbol{P}_{k|k-1}, c) = \Sigma_{\phi,\text{tot}}, \tag{31}$$

where $\boldsymbol{s}_{k|k-1}$ is the predicted state and $\boldsymbol{P}_{k|k-1}$ is the predicted error-covariance for the considered target given by the tracking filter. Therefore, the predicted belief state is used as input for the cost calculation.

V. ASSUMED RADAR SCENARIO

For the following simulation sections, a simplified radar scenario for tracking and classification is assumed. An EKF is applied as a tracking algorithm and similar definitions are used as already shown in [50]. As mentioned in (1), the targets move in a two-dimensional Cartesian coordinate system. The algorithm is jointly optimizing the revisit interval T and the dwell time τ. The former is the time between two consecutive measurements, and the latter is the time the radar sensor spends on a target. For a Target n, a pair of T_n and τ_n defines a budget allocation, also called action $\boldsymbol{a}_n \in \mathbb{R}^2$. The actions influence both the classification and the tracking performance. The outcome of the algorithm is budget allocations per target that theoretically fit into the time frame.

Furthermore, as the situation changes over time, the resource allocation needs to be adjusted to it. In this paper, this is done in regular predefined update intervals.

A. Assumed Radar System

The assumed radar system is able to take measurements in range r and angle θ. Additionally, it can take measurements of a certain target feature f. There exists a measurement noise with variances $\sigma_{r,0}^2$, $\sigma_{\theta,0}^2$, and $\sigma_{f,0}^2$, which refer to a reference measurement. The parameters of that reference measurement are shown in Table I.

B. Target Dynamics

The targets are assumed to move with a constant velocity in x and y direction with an added maneuverability noise, which is class-dependent. Instead of (2), the next

TABLE I
Parameters of the Reference Measurement

Parameter	Value
SNR (SNR_0)	1
RCS (ς_0)	10 m^2
Dwell time (τ_0)	1s
Range (r_0)	50 km
$\sigma^2_{r,0}$	625 m^2
$\sigma^2_{\theta,0}$	4×10^{-4} rad^2
$\sigma^2_{f,0}$	4

state can therefore be written as

$$\boldsymbol{s}^n_{k_n+1} = \boldsymbol{F}_n \boldsymbol{s}^n_{k_n} + \boldsymbol{w}^{n,c}_{k_n}, \tag{32}$$

with $\boldsymbol{F}_n \in \mathbb{R}^{4\times4}$ defined as

$$\boldsymbol{F}_n = \begin{bmatrix} 1 & 0 & T_n & 0 \\ 0 & 1 & 0 & T_n \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{33}$$

and the maneuverability noise $\boldsymbol{w}^{n,c}$ with covariance

$$\boldsymbol{Q}_{n,c} = \begin{bmatrix} T_n^3/3 & 0 & T_n^2/2 & 0 \\ 0 & T_n^3/3 & 0 & T_n^2/2 \\ T_n^2/2 & 0 & T_n & 0 \\ 0 & T_n^2/2 & 0 & T_n \end{bmatrix} \sigma^2_{w,c}, \tag{34}$$

where $\sigma^2_{w,c}$ is the maneuverability noise variance for class c.

C. SNR Model

As previously mentioned, measurements are taken in range r and angle θ. Due to the nonlinearity between measurements and target states, at $\boldsymbol{s}^n_{k_n}$ a measurement transformation function $\boldsymbol{h}(\boldsymbol{s}^n_{k_n}) \in \mathbb{R}^3$ is defined. The measurement equation in (4) therefore becomes

$$\boldsymbol{z}^n_{k_n} = \boldsymbol{h}(\boldsymbol{s}^n_{k_n}, c^n) + \boldsymbol{v}^n_{k_n}, \tag{35}$$

with

$$\boldsymbol{h}(\boldsymbol{s}^n_{k_n}) = \left[\sqrt{(x^n_{k_n})^2 + (y^n_{k_n})^2}\,,\ \text{atan2}\left(y^n_{k_n}, x^n_{k_n}\right)\,,\ f(c^n)\right]^T, \tag{36}$$

and $\boldsymbol{v}^n_{k_n} \in \mathbb{R}^3$ being the measurement noise for Target n. The feature of class c^n is denoted as $f(c^n)$ and atan2$(\cdot)$ denotes the two-argument arctangent as commonly used in programming languages.

The range and azimuth components of $\boldsymbol{v}^n_{k_n}$ are considered to be independent:

$$\boldsymbol{v}^n_{k_n} = [v^{r,n}_{k_n} \quad v^{\theta,n}_{k_n} \quad v^{f,n}_{k_n}]^T, \tag{37}$$

with variances $\sigma^2_{r,n}, \sigma^2_{\theta,n}$, and $\sigma^2_{f,n}$.

Since the relationship between the measurements and the states is nonlinear, an EKF is applied in the following simulations. The observation matrix $\boldsymbol{H}^n_{k_n}$ is defined as the Jacobian of the measurement transformation function $\boldsymbol{h}$:

$$\boldsymbol{H}^n_{k_n} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{s} \partial f}\right|_{\boldsymbol{s}^n_{k_n}, f^n}. \tag{38}$$

For the assumed radar systems, it has dimensions $\boldsymbol{H}^n_{k_n} \in \mathbb{R}^{3\times5}$.

In line with the radar scenario described [50], the SNR is calculated by using (39), which is based a paper by Koch [33]:

$$\text{SNR}_{k_n}(\varsigma_n, \tau_n, r^n_{k_n}) = \text{SNR}_0 \cdot \left(\frac{\varsigma_n}{\varsigma_0}\right) \cdot \left(\frac{\tau_n}{\tau_0}\right) \cdot \left(\frac{r^n_{k_n}}{r_0}\right)^{-4} \cdot e^{-2\Delta\alpha}. \tag{39}$$

where $\Delta\alpha$ is the relative beam positioning error, ς_n is the constant radar cross section (RCS) of the Target n, $r^n_{k_n}$ is the distance of Target n at time step k_n and ς_0, τ_0 and r_0 are the corresponding values for a reference target. The dwell time is used equivalently to the transmitted energy mentioned by Koch. The relative beam positioning error is calculated as

$$\Delta\alpha = \frac{\left(\theta_{k_n} - \hat{\theta}_{k_n}\right)^2}{\Gamma^2}, \tag{40}$$

where θ_{k_n} is the real target angle and $\hat{\theta}_{k_n}$ is the predicted target angle in azimuth at time k_n and Γ is the one-sided beam-width in azimuth.

The variance of the range and azimuth measurement noise for Target n can then be defined as (see, e.g., [41])

$$\sigma^2_{\bullet,n} = \frac{\sigma^2_{\bullet,0}}{\text{SNR}_{k_n}(\varsigma_n, \tau_n, r^n_{k_n})}, \tag{41}$$

where $\bullet \in (r, \theta, f)$ and $\sigma^2_{\bullet,0}$ is the measurement noise variance for a reference target 0 as defined in Table I.

Assuming independent measurements, the measurement covariance can be written as

$$\boldsymbol{R}^n_{k_n} = \begin{bmatrix} \sigma^2_{r,n} & 0 & 0 \\ 0 & \sigma^2_{\theta,n} & 0 \\ 0 & 0 & \sigma^2_{f,n} \end{bmatrix}. \tag{42}$$

D. Target Classes

Each target is assumed to belong to a specific class. The class is defined before the simulation scenario starts and cannot be changed. Therefore, it stays the same during the entire scenario. The measurement variance regarding the class feature f^n of object n is calculated as shown in (41). The corresponding variance value for the reference measurement f_0 is shown together with the other simulation parameters in the specific subsection. For the simulations discussed below, different target classes are considered that influence the maneuverability of the targets. These maneuverabilities are implemented in the trajectory simulations of the targets and are also considered in the resource optimization algorithm. As discussed earlier, one tracking filter per target class is applied, each tuned to one of the classes.

D. Optimization Problem

There are N tracked targets in the environment. Based on the general definition in (6), the RRM problem can thus be expressed as

$$\begin{aligned} \min_{T,\tau} \quad & \sum_{n=1}^{N} E\left[\mathcal{C}\left(\boldsymbol{s}^n_{k_n|k_n-1}(T_n,\tau_n), \boldsymbol{P}^n_{k_n|k_n-1}(T_n,\tau_n), c^n\right)\right] \\ \text{s. t.} \quad & \sum_{n=1}^{N} \frac{\tau_n}{T_n} \leq \Theta_{\max}, \end{aligned} \tag{43}$$

where $E[\cdot]$ denotes the expected value. The cost that is optimized is therefore based on the current prediction of the tracking filter, which is based on the measurement actions T_n and τ_n for Target n. Both the revisit times T_n, as well as the dwell times τ_n are optimized. The state measurements are influenced by both T and τ, while the state independent feature measurement is only influenced by the dwell time.

For all shown simulations, the implemented BP is simply to apply the evaluated action in every step of the PR. Therefore, $\pi_{\text{base}} = \boldsymbol{a}$.

F. Threat Definition

Since a two-dimensional scenario is assumed, the dimension parameter in the unscented transform is $D = 2$. As mentioned before, the choice of the "correct" threat definition depends on the scenario and the user's wishes. As an example, in the following, the threat ϕ is defined as

$$\phi(c, \boldsymbol{s}) = \frac{\rho_c \cdot \left(0.1 + \exp\left(-\frac{r-r'}{\eta}\right)\right)}{1 + \exp\left(-\frac{r-r'}{\eta}\right)}, \tag{44}$$

where ρ_c is a scalar factor unique for each class, $r = \sqrt{x^2 + y^2}$ is the range of the target from the sensor, $r' = 18$ km is a reference range and $\eta = 5000$ is a parameter to fine-tune the threat function slope. A possible example of such a threat is shown in Fig. 3. This formulation assumes that targets at a long distance pose a very low threat, while the threat increases the closer the target advances toward the sensor location. At a certain distance, the maximum threat value is reached. In addition to that, some classes generally pose a higher threat than others. One could think of an automotive scenario where a vehicle is moving toward the sensor location. When it is far away, the threat would be low as it would probably turn away at some point. However, once it comes closer, the threat increases until it is not very likely to turn away anymore, which means that the maximum threat level is reached, and a collision is almost inevitable. For instance, regarding the different classes, one could think of a truck having a higher base threat level than a cyclist.

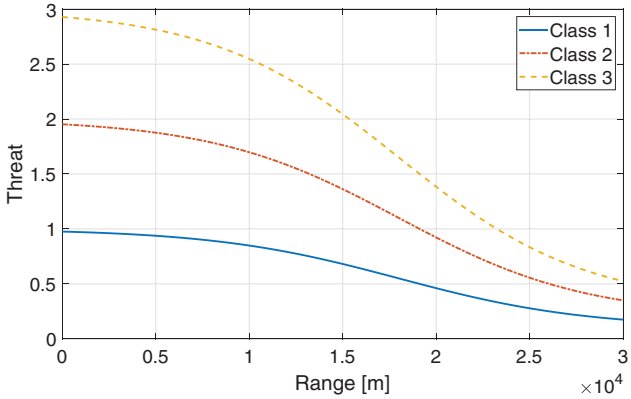


Figure 3. Example threat function for three different targets. The reference range r' is 18 km and the tuning parameter η is 5000. The class parameters ρ are set to values 1, 2, and 3, respectively.

TABLE II
Simulation Parameters for Simulation Scenario A

Parameter	Value
Precision of solution (δ)	0.01
Action space discretization steps ($\Delta T, \Delta\tau$)	Adaptive
Action space limits revisit interval ($T_{\min}, T_{\max}$)	$T \in [0.1\text{s} \ldots 5\text{ s}]$
Action space limits dwell time ($\tau_{\min}, \tau_{\max}$)	$\tau \in [0.1\text{ s} \ldots \infty]$
Number of rollouts (M)	10
Rollout horizon ($\mathscr{H}$)	10
Maximum available budget ($\Theta_{\max}$)	1
Budget update interval (t_B)	5 s
Beam positioning error ($\Delta\alpha$)	0
Probability of detection (P_D)	1
Threat reference range (r')	18 km
Threat slope parameter (η)	5000

VI. SIMULATION SCENARIO A

In this section, the dynamic tracking example as presented in [50] is used to show the impact of the chosen cost function based on the threat. Essentially, the AODB algorithm from [50] is applied with the cost function as defined in (31). The radar sensor is placed at the origin of the coordinate system. Initially, there are four targets in the scene. After 25 s, a fifth target is detected, and a new track is started. All targets move with constant velocities. Here, it is assumed that the class of the targets is not of interest, so no classification is applied during the simulation scenario. The simulation parameters are summarized in Table II, while the target parameters are shown in Table III. The trajectories of the targets during the simulation scenario are shown in Fig. 4, and the resulting budget allocation of the simulations is shown in Fig. 5.

Since classification is not considered in this example, the uncertainty in threat comes directly from the tracking accuracy. This is reflected in Fig. 5 by the fact that the budgets overall show very similar behavior to the dashed lines. Those lines indicate the results from [50], where the error-covariance of the tracking filter was used directly to optimize the resource allocation. For example,

TABLE III
Initial Target Parameters for Simulation Scenario A

Parameter \ Target n	1	2	3	4	5
x_0^n [km]	12	12	7	19	7.9
y_0^n [km]	10	15	11	2	8.3
$\dot{x}_0^n$ [m s^{-1}]	9	−30	45	−35	−20
$\dot{y}_0^n$ [m s^{-1}]	−15	15	30	0	−25
ς^n [m^2]	25	25	64	64	64

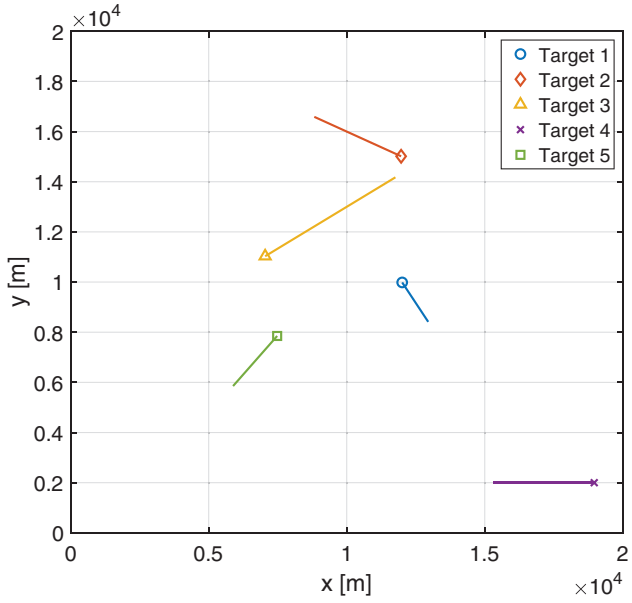


Figure 4. Trajectories of the targets for Simulation Scenario A. The symbols mark the starting positions.

Target 4 receives the largest budget allocation during the first 70% of the scenario, while Target 5 always receives the smallest, which is in line with the previous results. It should be noted that the algorithm decides the resource allocations on the expected threat variance reduction rather than the actual threat variance values. Nevertheless, the target with the highest threat variance will offer

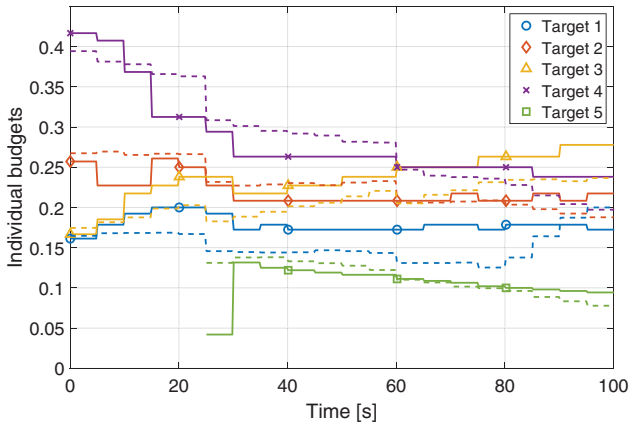


Figure 5. Resulting budget distribution for Simulation Scenario A. The dashed lines denote the results for tracking without classification as shown in [50].

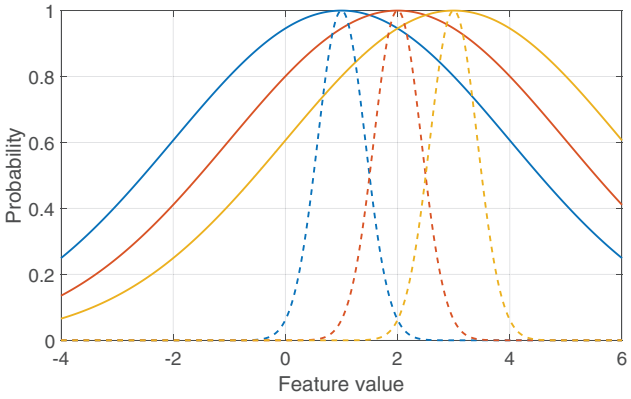


Figure 6. The PDFs of the class features given different SNR values. The solid lines correspond to SNR = 1 and the dashed lines to SNR = 50.

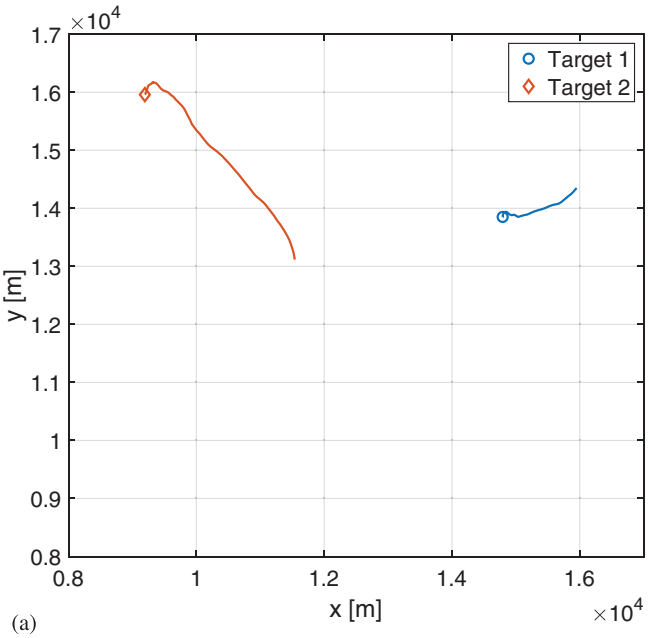
the biggest opportunity to reduce this threat variance in most cases.

VII. SIMULATION SCENARIO B

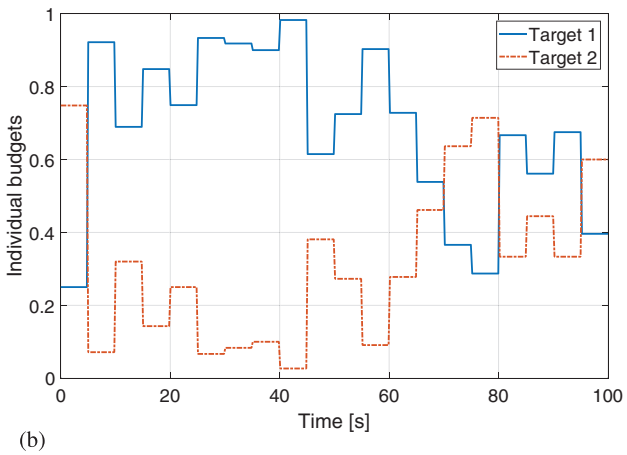
In this section, a dynamic joint tracking and classification scenario are presented. The radar sensor is placed at the origin of the coordinate system. There are two possible classes and two observed targets. The first target is of Class 1, and the second one of Class 2. The radar sensor is aware of all possible target classes, but it does not know which target is of which class. Therefore, the initial class probabilities are equal for both classes for each target. The available budget is set to 1, implying that the radar system fully focuses on these two tracking tasks. Aspects like false alarms are not taken into account. However, even if ambiguities in the measurement-to-track assignment were considered, it would still be necessary to compute the joint posterior of the state variables that are used for the optimization. Therefore, the approach would not be significantly different from what is presented here. For future use in a real radar system, those aspects certainly need to be taken into account.

The targets move with a class-typical maneuverability noise, which can be seen from trajectories in Fig. 7(a). The simulation parameters are identical with the ones for simulation A as mentioned in Table II. The target and class parameters are shown in Tables IV and V, respectively. As mentioned before, the values used in this section do not have any physical origin and are merely chosen for demonstration purposes. Figure 6 shows the PDFs of the features given two different SNR values. The simulation results are presented in Fig. 7(b)–(e).

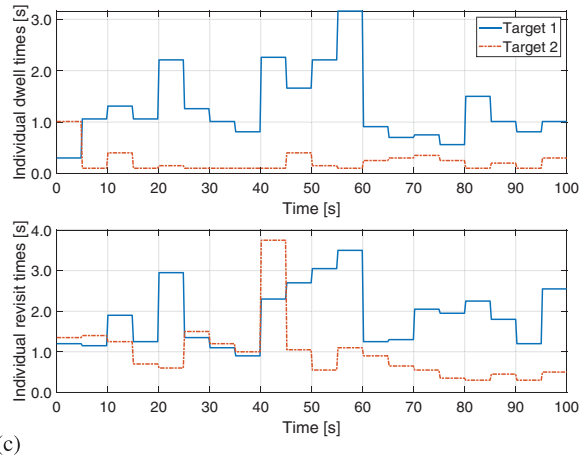
In the beginning, Target 2 gets a larger amount of dwell time assigned than Target 1, which leads to a quick classification. Target 2 is closer to the radar sensor than Target 1, which means that not knowing the class leads to a higher threat variance. Additionally, the feature measurements for Target 2 are more accurate than for Target 1 due to the smaller distance and, therefore, higher SNR. Subsequently, after 5 s, the dwell time and



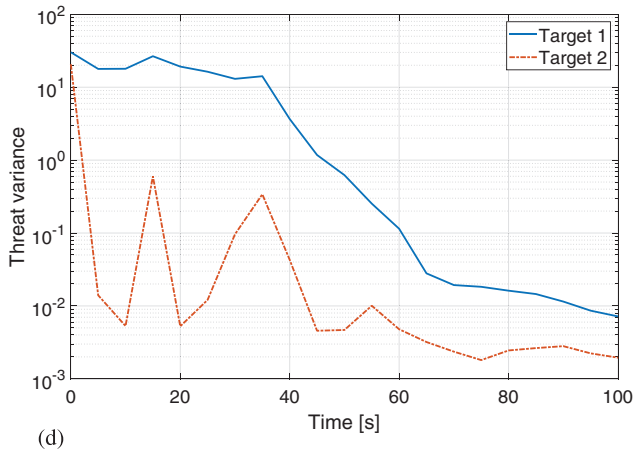
(a)



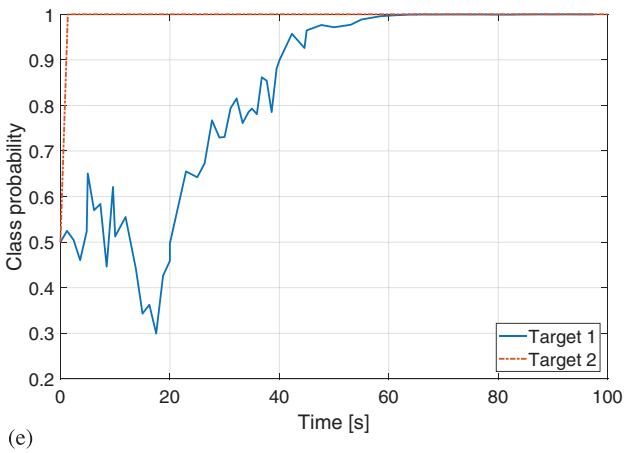
(b)



(c)



(d)



(e)

Figure 7. Simulation results for Simulation Scenario B. (a) Trajectories of the targets for Simulation Scenario B. The symbols mark the starting positions. (b) Resulting budget distribution. (c) Resulting dwell time and revisit time distribution. (d) Resulting optimized cost (threat variance). (e) Resulting class probabilities. A value of 1 means that a target was correctly classified.

TABLE IV
Initial Target Parameters for Simulation Scenario B

Parameter \ Target n	1	2
x_0^n [km]	14.8	9.2
y_0^n [km]	13.9	15.9
$\dot{x}_0^n$ [m s^{-1}]	2	2
$\dot{y}_0^n$ [m s^{-1}]	−1	1
ς_n [m^2]	5	5
c^n	1	2

TABLE VI
Initial Target Parameters for Simulation Scenario C

Parameter \ Target n	1	2	3
x_0^n [km]	10.1	12.3	12.1
y_0^n [km]	17.1	17.5	15.3
$\dot{x}_0^n$ [m s^{-1}]	1	−2	1
$\dot{y}_0^n$ [m s^{-1}]	2	2	−2
ς_n [m^2]	5	5	5
c^n	1	2	3

with it the relative budget for Target 2 drops, while Target 1 gets significantly more dwell time and budget assigned. During the bigger part of the scenario, the sensor focuses on classifying Target 1, which is more difficult due to its larger distance from the sensor. While Target 1 gets slowly classified, its budget starts to decrease after about 45 s. The budget for Target 2 increases at the same time. After both targets are successfully classified at about 70 s, the assigned budgets for both targets stay around 0.5, although Target 1 still gets a higher dwell time. The targets are both classified and theoretically deserve a similar amount of attention. However, as Target 1 was classified later, there is still slightly more uncertainty left about its class, leading to a higher dwell time allocation. This behavior emphasizes that the resource allocation is based on the expected threat variance reduction. Figure 7(d) shows how the early classification of Target 2 leads to a significant direct decrease in threat variance, while the classification of Target 1 takes longer, and the cost therefore also drops slower.

VIII. SIMULATION SCENARIO C

This simulation scenario is similar to Scenario B. The radar sensor is again placed at the origin of the coordinate system, and the available maximum budget is set to $\Theta_{\max} = 0.5$. The reason for a lower maximum budget could be, e.g., that an operator of the radar system manually assigned some of the total budget to other tasks. Additionally, the length of the simulation is 500 s, which is longer than in Scenario B. All other general simulation parameters are the same as shown in Table II. The class parameters are summarized in Table VII, and the initial target parameters are shown in Table VI. This time, there are three targets in the environment, and the targets can be of three possible classes. Figure 8(a) shows the trajectories of the simulated targets. The simulation results are shown in Fig. 8(b)–(e).

At the beginning of the scenario, it can be seen that Target 3 gets the largest relative budget assigned. Subsequently, it gets classified very quickly. It can be seen that the algorithm makes a wrong decision about the class of Targets 1 and 2. Figure 7(d) shows that making the first classification decisions leads to a large reduction of the calculated threat variance for all targets within the first 100 s. It can be seen that while the algorithm slowly classifies Target 2 between about 100 s and 300 s, the threat variance increases and then drops again. The reason is that the class probabilities are shifting during that phase, and there has no clear decision been made yet. The same happens to Target 1, as its class probability values are also changing at that time. Between about 320 s and 420 s Target 1 is classified correctly, which also leads to an increased threat variance. After 400 s, all targets are correctly classified, and the threat variances decrease rapidly.

In Fig. 8(b), it can be seen that the budget allocations roughly follow the threat variances. The target with the highest threat variance usually receives the largest budget. Similarly, the dwell times are assigned approximately proportional to the threat variance.

IX. SIMULATION SCENARIO D

This simulation is similar to Simulation C, except for all targets being of the same class. The available budget is set to $\Theta_{\max} = 0.7$. The length of the simulation is 100 s. All other simulation parameters are identical with Simulation A as shown in Table II. The target and class parameters for this simulation scenario are shown in Tables VIII and IX. The simulation results are shown in Fig. 9(a)–(d).

TABLE V
Class Parameters for Simulation Scenario B

Parameter \ Class	1	2
Class feature f_c	1	2
Threat parameter ρ_c	1	9
Maneuverability $\sigma_{w,c}$ [m s^{-2}]	2	5

Table VII
Class Parameters for Simulation Scenario C

Parameter \ Class	1	2	3
Class feature f_c	1	2	3
Threat parameter ρ_c	1	9	19
Maneuverability $\sigma_{w,c}$ [m s^{-2}]	2	5	9

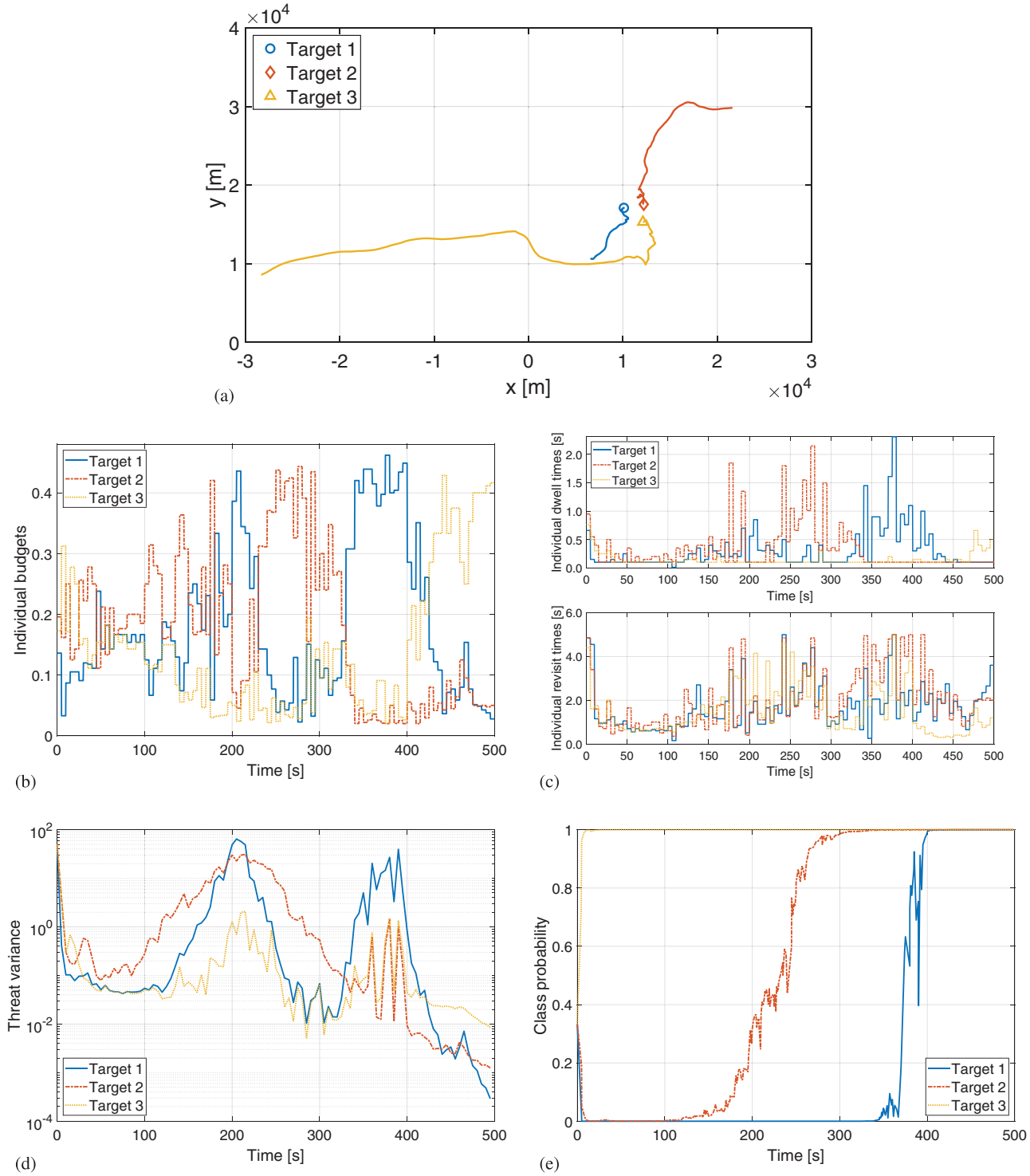


Figure 8. Simulation results for Simulation Scenario C. (a) Trajectories of the targets for Simulation Scenario C. The symbols mark the starting positions. (b) Resulting budget distribution. (c) Resulting dwell time and revisit time distribution. (d) Resulting optimized cost (threat variance). (e) Resulting class probabilities. A value of 1 means that a target was correctly classified.

Figure 9(c) shows that the closer the targets, the earlier they get classified. This has to do with the higher SNR at shorter ranges and the fact that the proposed algorithm assigns larger budgets to "dangerous" targets to classify them quickly. This can be seen in Fig. 9(b), where Target 2 receives the largest budget between 5 and 10 s. After the classification of the targets is completed, the budget distribution in Fig. 9(b) shows a clear influence from the distance of the targets. The reason is that there is no significant uncertainty in the class

TABLE VIII
Initial Target Parameters for Simulation Scenario D

Parameter \ Target n	1	2	3
x_0^n [km]	4	8	12
y_0^n [km]	4	8	12
$\dot{x}_0^n$ [m s^{-1}]	−1	−1	−2
$\dot{y}_0^n$ [m s^{-1}]	1	−1	−2
ς_n [m^2]	5	5	5
c^n	3	3	3

TABLE IX
Class Parameters for Simulation Scenario D

Parameter \ Class	1	2	3
Class feature f_c	1	2	3
Threat parameter ρ_c	1	9	19
Maneuverability $\sigma_{w,c}$ [m s^{-2}]	2	6	10

anymore, so the position uncertainty becomes more important.

Additionally, Fig. 9(d) shows that the proposed approach delivers the lowest expected cost compared to three other resource allocation methods. It can also be seen that once the target classes are determined, the costs of the different approaches are relatively small. The reason, therefore, is that the cost function heavily depends on the assumed class probabilities, especially when the target class is not determined yet. The differences between the approaches depend on the cost function and feature definition.

X. CONCLUSION

This paper introduced a novel RRM approach for joint tracking and classification using a previously introduced framework. In contrast to most available approaches, two different task types are combined into one. It is shown that it is possible to solve the RRM problem for multiple task types by using only a single cost function based on a definition of mission threat. Such approaches have been suggested previously but have never been fully developed and demonstrated with the help of fully worked out practical simulation scenarios.

Firstly, the joint tracking and classification framework have been introduced, which builds on the previous RRM framework as shown in [50].

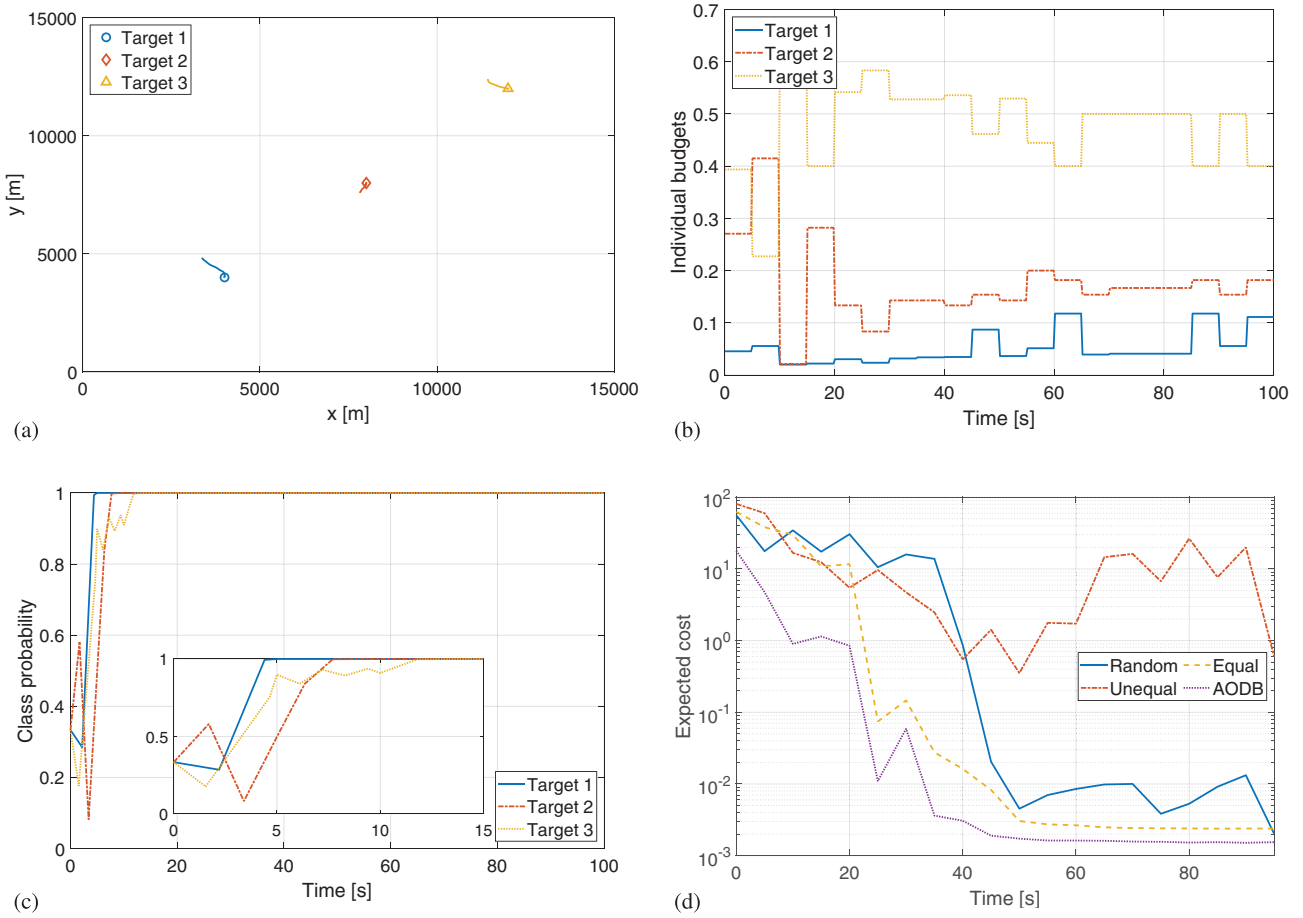


Figure 9. Simulation results for Simulation Scenario D. (a) Trajectories of the targets. The symbols mark the starting positions. (b) Resulting budget distribution. (c) Resulting class probabilities. A value of 1 means that a target was correctly classified. The box on the bottom middle shows the class probabilities zoomed in on the first 15 s. (d) Comparison of the expected optimized cost for different resource allocation methods. The values are averaged over 10 different runs.

Secondly, it has been explained how to move from the state to the threat domain and combine the costs of different target classes. The idea of threat is to transform the state of each task into an easily comparable scalar number.

Finally, an explicit definition of a possible mission threat has been introduced. The presented threat definition is based on the position and a class-dependent parameter. It has been shown how the threat looks like for multiple classes in a two-dimensional environment.

Through an analysis of the dynamic tracking Scenarios A–D, it has been shown that the algorithm works in different situations. It calculates the resource allocations based on the class probabilities and the tracking state accuracy. The algorithm tries to classify targets of unknown classes faster, especially when they are close to the radar sensor. On the other hand, the classification is done over time while tracking if the targets are further away and have a smaller threat variance. Once the targets are classified, the resource allocations depend primarily on the track uncertainty. This means that the target tracks get the resources assigned based on the expected decrease in uncertainty. Compared to other approaches, the presented approach leads to a quicker classification of dangerous objects. Within the presented simulations, a single measurable feature was chosen for demonstration purposes. In a practical implementation of the approach, multiple features (e.g., RCS and velocity) could be taken into account to further accelerate the classification procedure. Nevertheless, the presented simulations confirm the applicability of the proposed algorithm.

The proposed algorithm fills the timeline on average, leading to some overlap of tasks. Specifically, this can happen when predefined start or end times are required, or the tasks cannot be split up into multiple parts. A possible defensive solution to this problem is to assume a lower available budget for the tasks to keep a part of the timeline free. However, this would lead to a less optimal result. For practical implementation, the impact of these overlaps would need to be investigated, and an explicit scheduler would need to be implemented at a lower level, but this is out of the scope of this paper.

Future research has to be conducted w.r.t. the definition of the threat and cost function. Additionally, it has to be investigated further how these different threat and cost formulations influence the budget allocations. It will be especially interesting to look at how the classification process exactly depends on the different input parameters, such as the measurement variance. Finally, a practical implementation of the algorithm requires the investigation of an explicit scheduler and its impact on the results.

APPENDIX A
LAGRANGIAN RELAXATION PRINCIPLE

LR simplifies complicated constrained optimization problems by removing constraints and adding them as penalty terms into the original problem multiplied by so-called Lagrange multipliers. Consequently, a new optimization problem is created with fewer constraints than the original problem. LR maximizes the minimum of the cost function by adjusting the Lagrange multipliers. This problem is called the LDP, which is usually a lower estimate of the original problem if the initial Lagrange multipliers are chosen properly (see, for example, [8]).

LR and Lagrange multipliers have been extensively covered in literature (for example, in [5], [8], [9], [13], or [38]). As an example of how LR is applied, we consider the general optimization problem with N input variables that is shown below:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{A} \\ & \boldsymbol{h}(\boldsymbol{x}) \geq \boldsymbol{B}, \end{aligned} \tag{45}$$

where

$$\begin{aligned} \boldsymbol{x} &= [x_1, \ldots, x_N]^T \in \mathbb{R}^N, \\ \boldsymbol{g}(\boldsymbol{x}) &= [g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})]^T \in \mathbb{R}^m, \\ \boldsymbol{h}(\boldsymbol{x}) &= [h_1(\boldsymbol{x}), \ldots, h_p(\boldsymbol{x})]^T \in \mathbb{R}^p, \\ \boldsymbol{A} &= [A_1, \ldots, A_m]^T \in \mathbb{R}^m, \\ \boldsymbol{B} &= [B_1, \ldots, B_p]^T \in \mathbb{R}^p. \end{aligned}$$

Including the constraints into the optimization problem is done by adding a penalty term for each removed constraint, multiplied by Lagrange multipliers, which are defined as $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_m]^T \in \mathbb{R}^m$ and $\boldsymbol{\mu} = [\mu_1, \ldots, \mu_N]^T \in \mathbb{R}^p$. The Lagrangian is defined as

$$\begin{aligned} L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) &+ \sum_{i=1}^{m} \lambda_i (g_i(\boldsymbol{x}) - A_i) \\ &+ \sum_{j=1}^{p} \mu_j (B_j - h_j(\boldsymbol{x})). \end{aligned} \tag{46}$$

The relaxed problem is then called Lagrangian dual function and can be expressed as

$$d(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \underset{x}{\text{minimize}}\, L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{47}$$

The meaning of this expression is to find the maximum of the Lagrangian dual function with respect to the Lagrange multipliers, as

$$Z_D = \underset{\lambda, \mu}{\text{maximize}}\, d(\boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{48}$$

Therefore, the objective function is minimized over $\boldsymbol{x}$ while also being maximized over the Lagrange multipliers. The goal is to come as close to the original problem as possible. Iterative approaches are typically used to find the optimal Lagrange multipliers and, consequently, the tightest lower bound to the original problem. One of those techniques, the subgradient method, will be explained in the following subsection.

A. Subgradient Method

The subgradient method is an iterative process, which starts with an initial value for the Lagrange multipliers (e.g., 1). At each iteration k, the minimum of the relaxed problem is calculated using that value (Lagrangian dual function, see (46)). Then, subgradients are chosen for each of the constraint as $\boldsymbol{s}_{\lambda}^{k} = [s_{\lambda,1}^{k}, \ldots, s_{\lambda,m}^{k}]^{T} \in \mathbb{R}^{m}$ and $\boldsymbol{s}_{\mu}^{k} = [s_{\mu,1}^{k}, \ldots, s_{\mu,p}^{k}]^{T} \in \mathbb{R}^{p}$. Assuming the constraints mentioned in (45), the subgradients are defined as

$$\begin{aligned} \boldsymbol{s}_{\lambda}^{k} &= (\boldsymbol{g}(\boldsymbol{x}^{k}) - \boldsymbol{A}) \\ \boldsymbol{s}_{\mu}^{k} &= (\boldsymbol{B} - \boldsymbol{h}(\boldsymbol{x}^{k})). \end{aligned} \tag{49}$$

Next, the Lagrange multipliers are updated with a step size γ^{k}. If an inequality constraint is given, then the penalty term may not become negative. The updated Lagrange multipliers are therefore calculated as

$$\begin{aligned} \boldsymbol{\lambda}^{k+1} &= \max\{0, \boldsymbol{\lambda}^{k} + \gamma^{k}\boldsymbol{s}_{\lambda}^{k}\} \\ \boldsymbol{\mu}^{k+1} &= \max\{0, \boldsymbol{\mu}^{k} + \gamma^{k}\boldsymbol{s}_{\mu}^{k}\}. \end{aligned} \tag{50}$$

There are many possible step size approaches, such as constant or decreasing step sizes like γ_0/k or $/\gamma^{k}$, for example. The procedure started again with the new Lagrange multipliers. A new Lagrangian dual function is found, and afterward, new subgradients are again calculated. The exact result is found when the gradients $\boldsymbol{s}_{\lambda}^{k}$ and $\boldsymbol{s}_{\mu}^{k}$ reach $\mathbf{0}$. Since this value can never be reached exactly using this method, the process is repeated until the gradient reaches 0 with the desired precision.

To summarize, a short overview of the subgradient algorithm for the above mentioned optimization problem is given here:

1) $k = 0$: Set the Lagrangian multipliers to initial value $(\boldsymbol{\lambda}^{\mathbf{0}} = \boldsymbol{\lambda}_{\mathbf{0}}, \boldsymbol{\mu}^{\mathbf{0}} = \boldsymbol{\mu}_{\mathbf{0}})$.
2) Calculate solution for $d(\boldsymbol{\lambda}, \boldsymbol{\mu})$ and save $\boldsymbol{x}^{k}$.
3) Choose subgradients for Lagrangian multipliers $\boldsymbol{s}_{\lambda}^{k}$ and $\boldsymbol{s}_{\mu}^{k}$ (see (49)).
4) Check if $\boldsymbol{s}_{\lambda}^{k} \approx \mathbf{0}$ and $\boldsymbol{s}_{\mu}^{k} \approx \mathbf{0}$ with desired precision. If it is, then stop the process.
5) Adjust Lagrangian multipliers as shown in (50).
6) Go to step 2 and set $k = k + 1$.

APPENDIX B
SOLUTION METHODS FOR POMDPS

POMDPs can be solved for finite or infinite horizons. In order to reduce complexity, limited horizons $\mathscr{H}$ are often considered. The value of $\mathscr{H}$ represents the number of measurement time steps into the future that are considered in the optimization. Once a new budget allocation is calculated, the horizon $\mathscr{H}$ will be moved to the current moment in time. This approach is therefore also called a receding horizon.

In [16], Charlish and Hoffmann have presented an excellent summary of the general solution of a POMDP. The following equations are based on their explanations. The goal is to find the actions that minimize the total cost (value $V_{\mathscr{H}}$ over horizon $\mathscr{H}$). Starting at time step k_0, this can be expressed as

$$V_{\mathscr{H}} = E\left[\sum_{k=k_0}^{k_0+\mathscr{H}} c(\boldsymbol{s}_k, \boldsymbol{a}_k)\right]. \tag{51}$$

Using $C_B(\boldsymbol{b}_k, \boldsymbol{a}_k) = \sum_{s\in S} \boldsymbol{b}_k(\boldsymbol{s})c(\boldsymbol{s}, \boldsymbol{a}_k)$ being the expected cost given belief state $\boldsymbol{b}_k$, $V_{\mathscr{H}}$ can be written as a so-called value function of the belief state $\boldsymbol{b}_{k_0}$ at time step k_0:

$$V_{\mathscr{H}}(\boldsymbol{b}_{k_0}) = E\left[\sum_{k=k_0}^{k_0+\mathscr{H}} C_B(\boldsymbol{b}_k, \boldsymbol{a}_k)|\boldsymbol{b}_{k_0}\right]. \tag{52}$$

For belief state $\boldsymbol{b}_0$ and taking action $\boldsymbol{a}_0$, the optimal value function is defined according to Bellman's equation [3] as

$$V_{\mathscr{H}}^{*}(\boldsymbol{b}_0) = \min_{a_0\in A}\left(C_B(\boldsymbol{b}_0, \boldsymbol{a}_0) + \gamma \cdot E\left[V_{\mathscr{H}-1}^{*}(\boldsymbol{b}_1)|\boldsymbol{b}_0, \boldsymbol{a}_0\right]\right). \tag{53}$$

For very long or infinite horizons, the discount factor can be set to $\gamma < 1$. Using this equation, the optimal policy can be expressed as

$$\pi_0^{*}(\boldsymbol{b}_0) = \arg\min_{a_0\in A}\left(C_B(\boldsymbol{b}_0, \boldsymbol{a}_0) + \gamma \cdot E\left[V_{\mathscr{H}-1}^{*}(\boldsymbol{b}_1)|\boldsymbol{b}_0, \boldsymbol{a}_0\right]\right). \tag{54}$$

For each $\boldsymbol{b}_k$ and $\boldsymbol{a}_k$, the optimal so-called Q-value is then defined as

$$Q_{\mathscr{H}-k}(\boldsymbol{b}_k, \boldsymbol{a}_k) = C_B(\boldsymbol{b}_k, \boldsymbol{a}_k) + \gamma \cdot E\left[V_{\mathscr{H}-k-1}^{*}(\boldsymbol{b}_{k+1})|\boldsymbol{b}_k, \boldsymbol{a}_k\right]. \tag{55}$$

Another way to find the optimal policy is to find the action $\boldsymbol{a}_k$ that minimizes the optimal Q-value:

$$\pi_k^{*}(\boldsymbol{b}_k) = \arg\min_{a_k\in A}(Q_{\mathscr{H}-k}(\boldsymbol{b}_k, \boldsymbol{a}_k)). \tag{56}$$

Therefore, it is necessary to calculate the Q-value for all possible actions, which is generally infeasible.

Generally, POMDPs can be solved both online as well as offline. Which type of solution is applied depends on the size of the state space. The so-called state-space explosion limits the usefulness of exact offline techniques.

Many offline methods are based on the so-called value iteration (VI). This technique iteratively calculates the cost/reward values of all possible states. An exact approach is, e.g., the One-Pass algorithm [54]. Examples for approximate point-based algorithms are PBVI, or Perseus [56]. Exact methods often lead to highly complex optimization problems, while approximate point-based methods require many grid points within the state space to converge toward the exact solution. The advantage of offline solutions is that the POMDP is fully solved only once. Unfortunately, this type of method is already infeasible for a very small state space.

Contrary to that, online algorithms only solve a small currently relevant part of the POMDP. This makes them less accurate but much easier and faster to compute. Some of the online approaches involve approximate tree methods (see, for example, the overview in [48]) or Monte Carlo sampling (e.g., PR).

Acknowledgments

This research was partially funded by Nederlandse Organisatie voor Wetenschappelijk Onderzoek (NWO) through the "Integrated Cooperative Automated Vehicles" project (i-CAVE).

REFERENCES

[1] D. Angelova and L. Mihaylova
"Joint target tracking and classification with particle filtering and mixture Kalman filtering using kinematic radar information,"
Digit. Signal Process., vol. 16, no. 2, pp. 180–204, 2006.

[2] K. Bell, C. Kreucher, and M. Rangaswamy
"An evaluation of task and information driven approaches for radar resource allocation,"
in *Proc. IEEE Radar Conf.*, 2021, pp. 1–6.

[3] R. E. Bellman
Dynamic Programming. Princeton, NJ, USA: Princeton Univ. Press, 1957.

[4] J. M. Bernardo and A. F. M. Smith
Bayesian Theory. Hoboken, NJ, USA: Wiley, 1994.

[5] D. P. Bertsekas
Constrained Optimization and Lagrange Multiplier Methods. Nashua, NH, USA: Athena Scientific, 1996.

[6] D. P. Bertsekas and D. A. Castanon
"Rollout algorithms for stochastic scheduling problems,"
J. Heuristics, vol. 5, no. 1, pp. 89–108, Apr. 1999.

[7] D. P. Bertsekas, J. N. Tsitsiklis, and C. Wu
"Rollout algorithms for combinatorial optimization,"
J. Heuristics, vol. 3, no. 3, pp. 245–262, Dec. 1997.

[8] D. Bertsimas and J. N. Tsitsiklis
Introduction to Linear Optimization. Nashua, NH, USA: Athena Scientific, 1997.

[9] S. S. Blackman and R. Popoli
Design and Analysis of Modern Tracking Systems. London, UK: Artech House, 1999.

[10] D. Blatt and A. O. Hero
"From weighted classification to policy search,"
in *Proc. 18th Int. Conf. Neural Inf. Process. Syst.*, 2005, pp. 139–146.

[11] M. Bockmair, C. Fischer, M. Letsche-Nuesseler, C. Neumann, M. Schikorr, and M. Steck
"Cognitive radar principles for defence and security applications,"
IEEE Aerosp. Electron. Syst. Mag., vol. 34, no. 12, pp. 20–29, Dec. 2019.

[12] F. Bolderheij, F. G. J. Absil, and P. van Genderen
"A risk-based object-oriented approach to sensor management,"
in *Proc. 8th Int. Conf. Inf. Fusion*, 2005, pp. 1–8.

[13] S. Boyd and L. Vandenberghe
Convex Optimization. Cambridge, UK: Cambridge Univ. Press, 2004.

[14] S. Brüggenwirth, M. Warnke, S. Wagner, and K. Barth
"Cognitive radar for classification,"
IEEE Aerosp. Electron. Syst. Mag., vol. 34, no. 12, pp. 30–38, Dec. 2019.

[15] D. A. Castañón
"Approximate dynamic programming for sensor management,"
in *Proc. 36th IEEE Conf. Decis. Control*, 1997, pp. 1202–1207.

[16] A. Charlish and F. Hoffmann
"Anticipation in cognitive radar using stochastic control,"
in *Proc. IEEE Radar Conf.*, 2015, pp. 1692–1697.

[17] A. Charlish and F. Hoffmann
"Cognitive radar management,"
in *Novel Radar Techniques and Applications: Waveform Diversity and Cognitive Radar, and Target Tracking and Data Fusion* Vol. 2. London, UK: SciTech Publishing, pp. 157–193, 2017.

[18] A. Charlish, F. Hoffmann, and I. Schlangen
"The development from adaptive to cognitive radar resource management,"
IEEE Aerosp. Electron. Syst. Mag., vol. 35, no. 6, pp. 8–19, Jun. 2020.

[19] A. Charlish, K. Woodbridge, and H. Griffiths
"Phased array radar resource management using continuous double auction,"
IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 3, pp. 2212–2224, Jul. 2015.

[20] E. K. P. Chong, C. M. Kreucher, and A. O. Hero
"Partially observable Markov decision process approximations for adaptive sensing,"
Discrete Event Dyn. Syst., vol. 19, no. 3, pp. 377–422, Sep. 2009.

[21] T. H. de Groot
"Mission-driven resource management for reconfigurable sensing systems," Ph.D. thesis, Delft Univ. Technol., Delft, Netherlands, 2015.

[22] M. E. Gomes-Borges, D. Maltese, P. Vanheeghe, and E. Duflos
"A Risk-based sensor management using random finite sets and POMDP,"
in *Proc. 20th Int. Conf. Inf. Fusion*, 2017, pp. 1–9.

[23] S. Haykin
"Cognitive radar: A way of the future,"
IEEE Signal Process. Mag., vol. 23, no. 1, pp. 30–40, Jan. 2006.

[24] A. O. Hero, D. A. Castañón, D. Cochran, and K. Kastella
Foundations and Applications of Sensor Management, New York, NY, USA: Springer, 2008.

[25] A. O. Hero and D. Cochran
"Sensor management: Past, present, and future,"
IEEE Sensors J., vol. 11, no. 12, pp. 3064–3075, Dec. 2011.

[26] K. Hintz
Sensor Management in ISR, Boston, MA, USA: Artech House, 2020.

[27] D. Hitchings and D. A. Castañón
"Receding horizon stochastic control algorithms for sensor management,"
in *Proc. Amer. Control Conf.*, pp. 6809–6815, 2010.

[28] S. Julier and J. Uhlmann
"Unscented filtering and nonlinear estimation,"
Proc. IEEE, vol. 92, pp. 401–422, Mar. 2004.

[29] F. Katsilieris
"Sensor management for surveillance and tracking: An operational perspective," Ph.D. thesis, Delft Univ. Technol., Delft, Netherlands, 2015.

[30] F. Katsilieris, H. Driessen, and A. Yarovoy
"Threat-based sensor management for joint target tracking and classification,"
in *Proc. 18th Int. Conf. Inf. Fusion*, 2015, pp. 435–442.

[31] F. Katsilieris, H. Driessen, and A. Yarovoy
"Threat-based sensor management for target tracking,"
IEEE Trans. Aerosp. Electron. Syst., vol. 51, no. 4, pp. 2772–2785, Oct. 2015.

[32] R. Klemm, H. Griffiths, and W. Koch
Novel Radar Techniques and Applications, Volume 2 - Waveform Diversity and Cognitive Radar, and Target Tracking and Data Fusion. London, UK: Scitech Publishing, 2017.

[33] W. Koch
"On adaptive parameter control for phased-array tracking," *Signal Data Process. Small Targets*, vol. 3809, pp. 444–455, 1999.

[34] C. Kreucher and A. O. Hero
"Non-myopic approaches to scheduling agile sensors for multistage detection, tracking and identification," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2005, pp. 885–888.

[35] C. Kreucher, K. Kastella, and A. O. Hero
"Sensor management using an active sensing approach," *Signal Process.*, vol. 85, no. 3, pp. 607–624, Mar. 2005.

[36] J. Langford and B. Zadrozny
"Reducing T-step reinforcement learning to classification. Unpublished, 2003. [Online]. Available: https://hunch.net/jl/projects/reductions/RL to class/colt submission.ps.

[37] B. La Scala, W. Moran, and R. Evans
"Optimal adaptive waveform selection for target detection," in *Proc. Int. Conf. Radar*, 2003, pp. 492–496.

[38] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos
Optimal Control, Hoboken, NJ, USA:Wiley, 2012.

[39] S. Martin
"Risk-based sensor resource management for field of view constrained sensors," in *Proc. 18th Int. Conf. Inf. Fusion*, 2015, pp. 2041–2048.

[40] S. Maskell
"Joint tracking of manoeuvring targets and classification of their manoeuvrability," *EURASIP J. Adv. Signal Process.*, vol. 2004, no. 15, 2004, Art. no. 613289.

[41] H. Meikle
Modern Radar Systems, Norwood, MA, USA: Artech House, 2008.

[42] H. S. Mir and A. Guitouni
"Variable dwell time task scheduling for multifunction radar," *IEEE Trans. Automat. Sci. Eng.*, vol. 11, no. 2, pp. 463–472, Apr. 2014.

[43] A. E. Mitchell
"Advancements and applications of the fully adaptive radar framework," Ph.D. thesis, Elect. Comput. Eng., Ohio State Univ., Columbus, OH, USA, 2018.

[44] A. E. Mitchell, G. E. Smith, K. L. Bell, A. J. Duly, and M. Rangaswamy
"Hierarchical fully adaptive radar," *IET Radar, Sonar Navigation*, vol. 12, pp. 1371–1379, Dec. 2018.

[45] P. W. Moo and Z. Ding
Adaptive Radar Resource Management, London, UK: Academic Press, 2015.

[46] A. Narykov, E. Delande, and D. E. Clark
"A formulation of the adversarial risk for multiobject filtering," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, pp. 2082–2092, Aug. 2021.

[47] D. Papageorgiou and M. Raykin
"A risk-based approach to sensor resource management," in *Advances in Cooperative Control and Optimization*. Heidelberg, Germany: Springer, 2007, pp. 129–144.

[48] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa
"Online planning algorithms for POMDPs," *J. Artif. Intell. Res.*, vol. 32, no. 1, pp. 663–704, Jul. 2008.

[49] V. Savchuk and C. P. Tsokos
"Bayesian theory and methods with applications," in *Atlantis Studies in Probability and Statistics*. Dordrecht, Netherlands: Atlantis Press, 2011.

[50] M. I. Schöpe, H. Driessen, and A. Yarovoy
"A constrained POMDP formulation and algorithmic solution for radar resource management in multi-target tracking," *ISIF J. Adv. Inf. Fusion*, vol. 16, no. 1, pp. 31–47, Jun. 2021.

[51] M. I. Schöpe, H. Driessen, and A. Yarovoy
"Multi-task sensor resource balancing using Lagrangian relaxation and policy rollout," in *Proc. 23rd Int. Conf. Inf. Fusion*, 2020, pp. 1–8.

[52] M. I. Schöpe, H. Driessen, and A. Yarovoy
"Optimal balancing of multi-function radar budget for multi-target tracking using Lagrangian relaxation," in "*Proc. 22nd Int. Conf. Inf. Fusion*," 2019, pp. 1–8.

[53] M. Shaghaghi and R. S. Adve
"Task selection and scheduling in multifunction multichannel radars," in *Proc. IEEE Radar Conf.*, 2017, pp. 969–974.

[54] R. D. Smallwood and E. J. Sondik
"The optimal control of partially observable Markov processes over a finite horizon," *Operations Res.*, vol. 21, no. 5, pp. 1071–1088, 1973.

[55] S. Sowelam and A. Tewfik
"Optimal waveform selection for radar target classification," in *Proc. Int. Conf. Image Process.*, 1997, pp. 476–479.

[56] M. T. J. Spaan and N. Vlassis
"Perseus: Randomized point-based value iteration for POMDPs," *J. Artif. Intell. Res.*, vol. 24, pp. 195–220, 2005.

[57] L. D. Stone, R. L. Streit, T. L. Corwin, and K. L. Bell
Bayesian Multiple Target Tracking, Boston, MA, USA: Artech House, 2013.

[58] K. A. B. White, J. L. Williams, and P. Hoffensetz
"Radar sensor management for detection and tracking," in "*Proc. 11th Int. Conf. Inf. Fusion*," 2008, pp. 1–8.

[59] K. A. B. White and J. L. Williams
"Lagrangian relaxation approaches to closed loop scheduling of track updates," *Signal Data Process. Small Targets*, vol. 8393, pp. 8393–8393, 2012.

[60] J. Wintenby and V. Krishnamurthy
"Hierarchical resource management in adaptive airborne surveillance radars," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 2, pp. 401–420, Apr. 2006.

Max Ian Schöpe received the B.Eng. degree in Electrical Engineering from the University of Applied Sciences, Hamburg, Germany in February 2015. Subsequently, he started the M.Sc. program in Telecommunication and Sensing Systems at Delft University of Technology, Delft, The Netherlands, which he finished in October 2017. From 2017 to 2021, he was a Ph.D. candidate with the Microwave Sensing, Signals and Systems (MS3) group of TU Delft. His main research interests are in the area of radar resource management using a partially observable Markov decision process framework.

Hans Driessen received the M.Sc. and Ph.D. degrees from the Department of Electrical Engineering, Delft University of Technology, Delft, The Netherlands, in 1987 and 1992, respectively. Since then, he has been employed with Thales Nederland BV, Hengelo, The Netherlands. He has been and is still involved in various (international) research projects and radar development programs in the areas of signal/data processing and radar management. Since January 2015, he additionally holds a part-time position as an Associate Professor of EEMCS faculty at the TU Delft in the Microwave Signals Sensor and Systems group in the field of radar systems, waveforms, and processing. His research interests are in the practical application of detection, estimation, information, and control theory to various problems in sensor systems.

Alexander G. Yarovoy (FIEEE' 2015) graduated from Kharkov State University, Ukraine, in 1984 with a Diploma with honors in radiophysics and electronics. He received the Candidate Phys. & Math. Sci. and Doctor Phys. & Math. Sci. degrees in radiophysics in 1987 and 1994, respectively. In 1987, he joined the Department of Radiophysics, Kharkov State University, Kharkiv, Ukraine as a Researcher and became a Full Professor there in 1997. From September 1994 to 1996, he was with the Technical University of Ilmenau, Ilmenau, Germany as a Visiting Researcher. Since 1999, he has been with the Delft University of Technology, Delft, The Netherlands. Since 2009, he has been leading there a chair of Microwave Sensing, Systems, and Signals. His main research interests are in high-resolution radar, microwave imaging, and applied electromagnetics (in particular, UWB antennas). He has authored and coauthored more than 450 scientific or technical papers, 6 patents, and 14 book chapters. He is the recipient of the European Microwave Week Radar Award for the paper that best advances the state-of-the-art in radar technology in 2001 (together with L. P. Ligthart and P. van Genderen) and in 2012 (together with T. Savelyev). In 2010, together with D. Caratelli, Prof. Yarovoy got the best paper award from the Applied Computational Electromagnetic Society (ACES). He served as the General TPC Chair of the 2020 European Microwave Week (EuMW'20), as the Chair and TPC Chair of the 5th European Radar Conference (EuRAD'08), as well as the Secretary of the 1st European Radar Conference (EuRAD'04). He also served as the cochair and TPC Chair of the 10th International Conference on GPR (GPR2004). He served as an Associated Editor of the International Journal of Microwave and Wireless Technologies from 2011 to 2018 and as a Guest Editor of five special issues of the IEEE Transactions and other journals. In the period 2008–2017, Prof. Yarovoy served as Director of the European Microwave Association (EuMA).