Radar Resource Management for Multi-Target Tracking Using Model Predictive Control

de Boer, Thies ; Schöpe, Max Ian; Driessen, Hans

Publication date
2021

Document Version
Final published version

Published in
2021 IEEE 24th International Conference on Information Fusion (FUSION)

Radar Resource Management for Multi-Target Tracking Using Model Predictive Control

Thies de Boer, Max Ian Schöpe and Hans Driessen
Microwave Sensing, Signals and Systems (MS3)
Delft University of Technology
Delft, the Netherlands
T.deBoer-8@student.tudelft.nl, {m.i.schope, j.n.driessen}@tudelft.nl

***Abstract*—The radar resource management problem in a multi-target tracking scenario is considered. Partially observable Markov decision processes (POMDPs) are used to describe each tracking task. Model predictive control is applied to solve the POMDPs in a non-myopic way. As a result, the computational complexity compared to stochastic optimization methods such as policy rollout is dramatically reduced while the resource allocation results maintain similar. This is shown through simulations of dynamic multi-target tracking scenarios in which the cost and computational complexity of different approaches are compared.**

***Index Terms*—Radar Resource Management, Constrained Optimization, Partially Observable Markov Decision Process, Model Predictive Control**

I. Introduction

A. Introduction

In recent years, advances in the field of radar led to the development of phased array antennas that allow the application of so-called digital beamforming (DBF). Together with advanced signal processing and arbitrary waveform generation, modern multi-function radars (MFR) became increasingly flexible. They can quickly adapt to changes of the environment by automatically adjusting the measurement parameters during runtime. This adaptive process is often called radar resource management (RRM) and can be considered as a part of a cognitive radar (CR) [1]–[5]. Possible applications for RRM can be found in many domains, such as autonomous driving and traffic monitoring or maritime and air surveillance. This paper is based on the results of the master's thesis in [6] and builds upon the framework explained in [7], [8]. It focuses on implementing an alternative solution method for the assumed underlying partially observable Markov decision process (POMDP).

B. Radar Resource Management

Countless heuristic solutions have been presented, many of them focusing on scheduling tasks with a fixed resource need (as mentioned in the overviews in [9] and [10]). In an overload situation, this inevitably leads to tasks of lower priority being dropped. If tasks have the same priority, the decision which task needs to be dropped is taken potentially randomly. Therefore, the full potential of RRM can only be explored once the resource allocation fully depends on the tasks' impact on the mission. Some approaches try to quantify this impact by optimizing the expected information gain as shown, e.g., in [11] by Kreucher et al. In [12], Hintz gives an overview of different information-based sensor resource management strategies. Another common concept is to define a risk or threat that a certain task poses on the mission. Interesting approaches implementing this idea are shown, e.g., by Katsilieris et al. in [13] and by Martin in [14]. Additionally, task-related measures can be optimized directly, such as, e.g., the estimated target track position error as shown by White and Williams in [15]. To allow all tasks to be considered equally, it is generally considered that the resource allocation is adjustable and not restricted to specific predefined values.

Most of the available RRM approaches focus on single tasks. For example, within a tracking scenario, many approaches try to keep the track quality constant [16], [17]. When multiple tasks are considered, this problem becomes more difficult to solve. By design, an MFR system usually operates at its resource limit. Therefore, we consider resource allocation to be a balancing act. Consequently, increasing the resources for one task will automatically decrease the resources for all the other ones, which will deteriorate their sensing performance.

C. Markov Decision Processes

Many presented methods assume a Markov decision process (MDP) or a POMDP in their RRM solution method (e.g. [18], [19]). POMDPs can be used to describe a dynamic environment while assuming that its state can only be observed using noisy measurements. With growing state and action space, the exact solution of a POMDP becomes increasingly intractable. Therefore, real-time approaches inevitably require approximate POMDP solution methods to be applied. An overview of these kinds of techniques can be found in [19], [20]. Some recent RRM approaches apply policy rollout in order to solve the POMDP [8], [16], [21]. Since this is often a computationally expensive procedure, this paper focuses on solving the POMDP using model predictive control (MPC). This is a logical simplification step as it has been shown that MPC is a special case of policy rollout [22].

D. Novelty

The main contribution of this paper is the introduction of a practical RRM solution approach that solves the under-

lying POMDP with computational efficiency in mind. The simplifications of MPC are used to significantly decrease the computation time compared to stochastic optimization approaches, such as policy rollout. Additionally, it allows to calculate continuous actions whereas the policy rollout requires a previously discretized action space.

E. Structure of the Paper

The remainder of the paper is structured as follows. While Section II introduces the assumed RRM and optimization problem, Section III presents our proposed solution approach. Sections IV and V introduce the assumed simulation scenarios and the results of the simulations. Finally, Section VI contains the conclusions.

II. Problem Definition

This section introduces the general problem that is treated in the remainder of the paper. For illustration purposes, a two-dimensional tracking scenario assuming N targets is assumed. The problem description closely follows the ones shown in [7], [8].

A. Motion Model

There are N assumed targets in the scene that are moving in a two-dimensional plane. The state of each target at time t can be 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 in x and y direction of a Cartesian coordinate system, respectively. The new target state after a certain time $t + \Delta t$ can be calculated following a state transition function

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

where $s_{t+\Delta t}^n$ is the state of target n at time $t+\Delta t$ and $\boldsymbol{w}_t^n \in \mathbb{R}^4$ is its maneuverability noise at time t. Based on (2) a state transition probability density function can be defined as

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

B. Measurement Model

The assumed radar sensor takes noisy measurements of the state s_t^n by executing sensor actions a_t^n that are adjustable and influence the accuracy of the observations. At time t, the measurement of target n can be defined by

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

where $h(\cdot)$ is the measurement transformation function, $\boldsymbol{v}_t^n$ is the measurement noise and $\boldsymbol{a}_t^n$ is the chosen action for target n at time t. Based on (4) the measurement probability density function can be defined as

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

C. Tracking Algorithm

Since a tracking scenario is assumed, a tracking filter needs to be chosen. For purely linear scenarios, a Kalman filter is the optimal solution. When a non-linear measurement transformation function or state transition function is assumed, an extended Kalman filter (EKF) or a particle filter are possible solutions. Generally, the tracking algorithm aims at computing the posterior density $p(\boldsymbol{s}_t^n | \boldsymbol{z}_t^n)$ of the object state.

D. Budget Optimization Problem

The radar sensor is assumed to have a limited budget B_{max} (e.g. sensing time or energy). Each action $\boldsymbol{a}_t^n$ consumes a certain amount of budget. It is assumed that the sensor operates in some sort of overload situation without enough available resource budget for the perfect execution of all tasks. Therefore, the budget needs to be distributed over the different tasks based on some cost function $c(\cdot)$. At time t, the one-step-ahead optimization problem assuming N different tasks is defined as

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

where $B_t^n \in [0, 1]$ is the budget for task n at time t and $B_{max} \in [0, 1]$ is the maximum available budget.

III. Proposed approach

In this paper, the same POMDP formulation of the tracking tasks as in [8] is assumed. In that paper, the POMDP was solved using a policy rollout algorithm which allocates the resources by sampling the possible future in a Monte Carlo fashion. In order to receive accurate results, many of those rollouts have to be performed and averaged. It has been shown that this can lead to a huge computational complexity.

The proposed POMDP solution method in this paper is MPC. It has been shown that MPC is a special case of the policy rollout and replaces the random sampled future by a modeled approach [22]. It is therefore assumed to lead to a significant decrease in computational complexity.

MPC is a receding horizon approach, which means the prediction horizon is shifted forward after every iteration. The basic operating principle can be summarized as follows, where H indicates the prediction horizon, which is here always assumed to be equal to the control horizon:

At time-step k:

1) Minimize cost over prediction horizon: $\sum_{t=k}^{k+H} c(\boldsymbol{a}_t, \boldsymbol{s}_t)$ to get the optimal action sequence of length H over the prediction horizon.
2) Take only the action corresponding to next time-step.
3) Repeat 1 and 2 at next time-step $k + 1$.

A. Optimization Setup

At every budget update an optimization problem must be solved. For the considered case where N objects are being tracked by an MFR, at time-step k and with prediction and control horizon H, this optimization problem can be defined as follows:

$$\begin{aligned} \min_{\tau} \quad & \sum_{t=k}^{k+H-1} \Big(\sum_{n=1}^{N} c(\tau_t^n, T_t^n, s_t^n) \Big) \\ \text{s.t.} \quad & \sum_{n=1}^{N} \frac{\tau_t^n}{T_t^n} \leq B_{max} \text{ for } t = 1 \dots H-1 \end{aligned} \tag{7}$$

Solving this problem comes down to minimizing the cost of the N objects summed over the time horizon of N. The optimization variables that result from this are:

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1^1 & \tau_2^1 & \dots & \tau_H^1 \\ \tau_1^2 & \tau_2^2 & \dots & \tau_H^2 \\ \vdots & \vdots & \ddots & \vdots \\ \tau_1^N & \tau_2^N & \dots & \tau_H^N \end{bmatrix} \text{ and } \boldsymbol{T} = \begin{bmatrix} T_1^1 & T_2^1 & \dots & T_H^1 \\ T_1^2 & T_2^2 & \dots & T_H^2 \\ \vdots & \vdots & \ddots & \vdots \\ T_1^N & T_2^N & \dots & T_H^N \end{bmatrix} \tag{8}$$

where τ_j^i and T_j^i correspond to the dwell time and the revisit time of the i-th object and the $(k+j)$-th time-step respectively. As we are only using the actions corresponding to the next time-step as inputs to our system, we use only the first columns of both $\boldsymbol{\tau}$ and $\boldsymbol{T}$. At the next budget update, the optimization will be carried out again with the horizon window shifted and the dwell times and revisit times of the remaining columns can now be improved using the measurements taken in the time that passed since the previous budget update. The optimization problem is constrained such that at every time-step the sum of the budgets does not exceed the total available budget of the radar, which is indicated by B_{max}. The difference with regards to the optimized actions compared to policy rollout is that every time-step within the optimization horizon can have a distinct budget distribution, while in the case of policy rollout a base policy is assumed where at every time-step within the optimization horizon the same budget distribution is applied. For the remainder of this paper T is assumed to be fixed at 1 s and $\boldsymbol{\tau}$ will be the action that is optimized.

B. Lagrangian Relaxation

Using Lagrangian Relaxation the constraints of the optimization problem can be brought into the objective function. The optimization function now becomes:

$$\max_{\lambda} \quad \Big(\underbrace{\min_{\tau} \quad \sum_{t=k}^{k+H} \Big(\sum_{n=1}^{N} c(\tau_t^n, \boldsymbol{s}_t^n) + \lambda \tau_t^n \Big)}_{\text{sum of independent minimization problems}} - \lambda B_{max} \Big) \tag{9}$$

In this optimization problem it can be seen that the minimization problem consists of the sum of N minimization problems, which are now, for a fixed iteration of λ, independent of each other as they are no longer linked to each other by the constraint. As a result, the problem can be rewritten into N sub-problems for the N different objects that need to be tracked:

$$\min_{\tau, T} \quad \sum_{t=k}^{k+H} c(\tau_t, s_t) + \lambda \tau_t \tag{10}$$

These sub-problems are independent of each other and can be solved using parallel processing to reduce execution time. Finally, the Lagrange multiplier λ is updated iteratively at every time-step using golden section search.

C. Golden Section Search

The goal is to find the Lagrange multiplier λ such that:

$$\Big| \sum_{n=1}^{N} \tau_n - B_{max} \Big| \leq \varepsilon \tag{11}$$

where ε is some small number indicating the tolerance of the constraint. In this paper golden section search is employed to efficiently find this λ such that (11) is met. The standard golden section search method, as described in for example [23], is extended due to the fact that initially the lower and upper bounds of λ are unknown. To find these bounds, λ is increased after every time-step k. As it is known that the value of λ must ensure that (11) is met, our initial values for the upper and lower bounds are those values of λ_k and λ_{k+1} for which there is a change in sign when going from $f(\lambda_k)$ to $f(\lambda_{k+1})$. Here a function evaluation $f(\lambda)$ refers to the value of $\sum_{n=1}^{N} \tau_n - B_{max}$ resulting from the $\boldsymbol{\tau}$ that follows from solving the MPC problem using λ. These lower and upper bounds will be referred to as x_L and x_U. Once these bounds are found, the next step to find λ is to perform function evaluations of values of λ between x_U and x_L. An efficient way of choosing these intermediate points is using the golden ratio conjugate ($r = \frac{\sqrt{5}-1}{2} \approx 0.618$). Initially, the two intermediate points $x_1 = x_L + r(x_U - x_L)$ and $x_2 = x_U - r(x_U - x_L)$ are evaluated. Then, based on the values $f(x_1)$ and $f(x_2)$ at those points a new intermediate point is chosen and x_L or x_U is shifted to x_1 or x_2 respectively, after which the function is evaluated at the newly chosen point. This procedure is repeated until a point is found that meets the criterion set in (11). The whole search method is summarized in Algorithm 1.

Algorithm 1: Finding the Lagrange multiplier using golden section search

Step 1 Starting from $\lambda = 0$, increase λ and compute f(λ) until f(λ) becomes negative;
Step 2 This λ becomes x_U while the previous λ becomes x_L;
$x_1 = x_U - r(x_U - x_L)$;
$x_2 = x_L + r(x_U - x_L)$;
Compute f(x_1) and f(x_2);
Step 3;
while $f(x_1) \leq \varepsilon \cap f(x_2) \leq \varepsilon$ **do**
  if $f(x_1) \leq f(x_2)$ **then**
    $x_U = x_2$;
    $x_2 = x_1$;
    $x_1 = x_U - r(x_U - x_L)$;
    Compute f(x_1);
  else
    $x_L = x_1$;
    $x_1 = x_2$;
    $x_2 = x_L + r(x_U - x_L)$;
    Compute f(x_2);
  end
end

IV. Assumed Simulation Scenarios

In this paper the two-dimensional tracking of *N* moving objects using a single radar is considered. At every measurement interval the radar takes measurements in range and angle. It is assumed that there is a limited amount of budget available to track the objects and the algorithms that are compared are tasked with optimizing the dwell times τ at every budget update interval. The revisit time T is assumed to be constant for all tasks in this paper, which means that the measurements are always taken in the same regular time intervals k. To compare the results of this novel approach to the previously applied policy rollout method, the implementation of the MPC has been done as follows. The resource allocations are recalculated in regular budget update intervals. During these intervals, the algorithm assumes that the same action is executed. The algorithm calculates the corresponding actions for a certain horizon. The first action of this horizon will then be executed by the sensor.

A. Tracking

For the Extended Kalman Filter, the following motion model is assumed:

$$\boldsymbol{s}_{k+1}^n = F\boldsymbol{s}_k^n + \boldsymbol{w}_k^n \tag{12}$$

with

$$\boldsymbol{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

and w_k^n being the maneuverability noise at time k. The process noise covariance matrix for target n is defined as:

$$\boldsymbol{Q}_n = \begin{bmatrix} \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 \\ 0 & \frac{T^4}{4} & 0 & \frac{T^3}{2} \\ \frac{T^3}{2} & 0 & T^2 & 0 \\ 0 & \frac{T^3}{2} & 0 & T^2 \end{bmatrix} \sigma_{w,n}^2, \tag{14}$$

where $\sigma_{w,n}^2$ is the maneuverability noise variance. The observation matrix H is the Jacobian of a certain measurement transformation function $h(s_n)$:

$$\boldsymbol{H}_k^n = \left.\frac{\partial h}{\partial \boldsymbol{s}}\right|_{\boldsymbol{s}_k^n} \tag{15}$$

B. Cost Function

The assumed cost function is comprises the sum of the first two diagonal terms of the predicted error covariance matrix:

$$c(\tau_k^n, \boldsymbol{s}_k^n) = \text{trace } (\boldsymbol{E}\boldsymbol{P}_{k|k-1}\boldsymbol{E}^T) \tag{16}$$

where

$$E = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{17}$$

These terms correspond to the predicted variance in x and y position of the objects.

C. Measurement Model

In the examples in this paper sensor measurements in range (r) and azimuth (θ) are assumed. The following measurement model is introduced to convert the state from Carthesian coordinates into polar coordinates

$$\boldsymbol{z}_k^n = h(\boldsymbol{s}_k^n) + \boldsymbol{v}_k^n \tag{18}$$

where

$$h(\boldsymbol{s}_k^n) = \left[\sqrt{(x_k^n)^2 + (y_k^n)^2} \;\; \text{atan2}(\tfrac{y_k^n}{x_k^n}) \right]^T \tag{19}$$

and $\boldsymbol{v}_k^n$ is the measurement noise with variance $\boldsymbol{\sigma}_n^2 = [\sigma_{r,n}^2, \sigma_{\theta,n}^2]^T$ at time-step k.

Based on the range of an object, the signal-to-noise ratio (SNR) of a measurement is determined in the same way as was described in [8], following equations by Koch [24]. It is calculated using the values of a reference measurement which is summarized in Table I. Using these values, the measurement noise variance of the system can be calculated as

$$\sigma_{r,n}^2 = \frac{\sigma_{r,0}^2}{SNR_{n,k}} \tag{20}$$

and

$$\sigma_{\theta,n}^2 = \frac{\sigma_{\theta,0}^2}{SNR_{n,k}}. \tag{21}$$

V. Simulation Results and Evaluation

In this section the resulting budget distribution of the tracking scenario of Fig. 1 is discussed. Furthermore, the proposed algorithm is compared with the policy rollout algorithm based their respective execution times and realized costs. For all mentioned simulations, the revisit time T is assumed to be fixed at 1 s.

TABLE I
REFERENCE MEASUREMENT VALUES.

Reference parameter	Value
Measurement variance in range ($\sigma^2_{r,0}$)	25 m^2
Measurement variance in angle ($\sigma^2_{\theta,0}$)	2e-3 rad^2
Reference SNR (SNR_0)	1
Reference RCS (RCS_0)	10 m^2
Reference dwell time (τ_0)	1 s
Reference range (r_0)	50 km

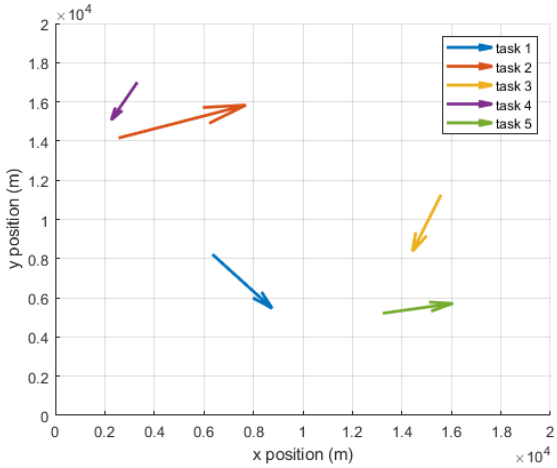

Fig. 1. Trajectories of the 5 objects to be tracked for a scenario of 100 seconds.

A. Simulation Scenario A

Simulations scenario A consists of 5 objects that can be seen in Fig. 1. In this example, some events took place during the simulation time so that their effect on the budget distribution could be evaluated:

- t = 20 s: A new object (task 5) needs to be tracked
- t = 60 s: Total available budget decreases from 1.0 to 0.9
- t = 90 s: Maneuverability of task 1 increases

For the simulation the parameters listed in Table II were used.

TABLE II
SIMULATION PARAMETERS SCENARIO A.

Reference parameter	Value
Maneuverability noise variance (σ^2_w)	2.5 $(\frac{m}{s^2})^2$
Radar cross section (RCS)	10 m^2
Prediction horizon length (H)	15
Number of rollouts	10
Simulation update interval	5 s
Budget precision (ε)	0.002

Fig. 2 and 3 show the budget distributions for the example scenario from Fig. 1 using the MPC and the policy rollout algorithm, respectively. It can be seen that objects located further away from the radar are allocated a larger amount of the budget than those closer to the radar. This is due to the way the cost function is constructed. Measurements of

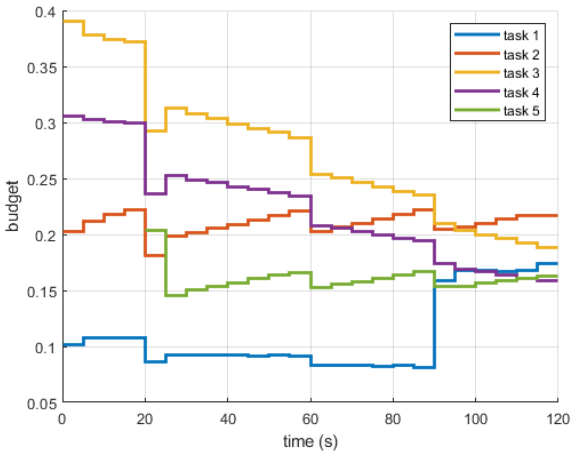

Fig. 2. Budget allocation of the 5 objects over the simulation time obtained using MPC.

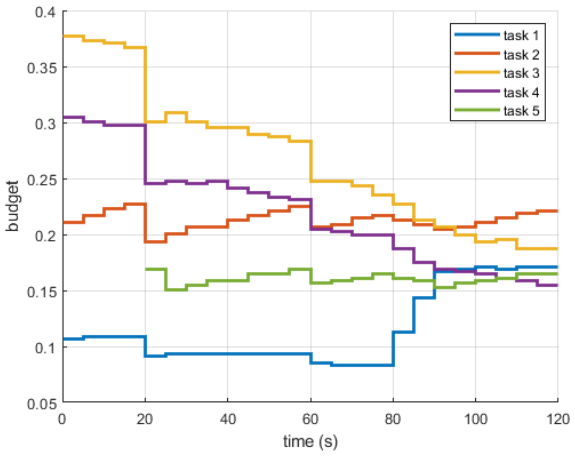

Fig. 3. Budget allocation of the 5 objects over the simulation time obtained using Policy Rollout.

objects further away will have a smaller signal-to-noise ratio and this is compensated for by increasing the budget available for tracking these objects. This behaviour is specific to the cost function choice and might not always be desired, so given the demands of the user a different cost function can be constructed. Furthermore, the figure reflects the changes to the system at the set time-steps. At t = 20 s, a new object needs to be tracked and some budget is made available for this task. At t = 60 s, the total available budget decreases and the budgets of all the tasks decrease. At t = 90 s, the maneuverability noise variance of the first task increases, resulting in the need to take better measurements of this object and therefore increasing the budget of task 1. When comparing the budget distributions from the policy rollout and MPC, it can be seen that the budgets are mostly the same. The main difference is seen at the change in maneuverability at t = 90 s. It can be seen that the effect of this change shows only after 90 s in the case of MPC, while in the policy rollout

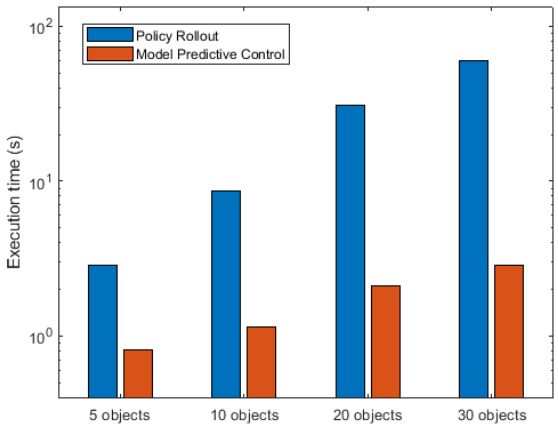

Fig. 4. Comparison between execution times of policy rollout and MPC algorithms. Note the logarithmic scale of the y-axis.

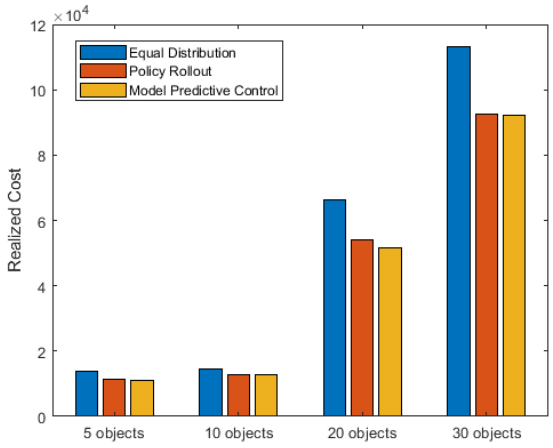

Fig. 5. Comparison between realized costs of equal distribution, policy rollout and MPC.

the effect of this change shows already at 80 s. This is due to the fact that in the policy rollout optimization the same action is chosen for the whole horizon, while for MPC this is not necessarily the case as was described in Section III. As a horizon of 15 s is used, at the budget update that takes place at 80 s this change in maneuverability already needs to be taken into account, and thus the actions are already slightly influenced by this change.

1) Execution Time Comparison: A goal of employing MPC was to lower the computational complexity compared to the policy rollout algorithm. This is evaluated by comparing the execution time of an average budget update of both algorithms. Here scenarios with linearly moving targets similar to scenario A are used with varying numbers of moving objects. In Fig. 4 the results from running each simulation 3 times and averaging the execution times are shown. It can be seen that the execution times when using MPC are significantly lower than when policy rollout is used.

2) Realized Cost Comparison: It is important that the performance of the resulting budget distribution is not degraded compared to the policy rollout case. To investigate this their realized costs were compared using the same scenarios as for the execution time comparison. The realized cost is defined as the sum of the evaluated cost function at every time-step during the simulation. In Fig. 5 the results from running each simulation 3 times and averaging the realised costs are shown. For comparison, the realised cost of using an equal distribution, i.e. allocating the same budget to all targets at every time-step, is also included. From the Fig. it can be seen that MPC and Policy Rollout have similar performances when looking at the realized costs.

B. Simulation Scenario B

Simulation scenario B consists of a static and a moving target. The moving target is first moving straight before making a turn towards the radar sensor and then continuing a linear trajectory. This trajectory is shown in Fig. 6. For the simulation the parameters listed in Table III were used.

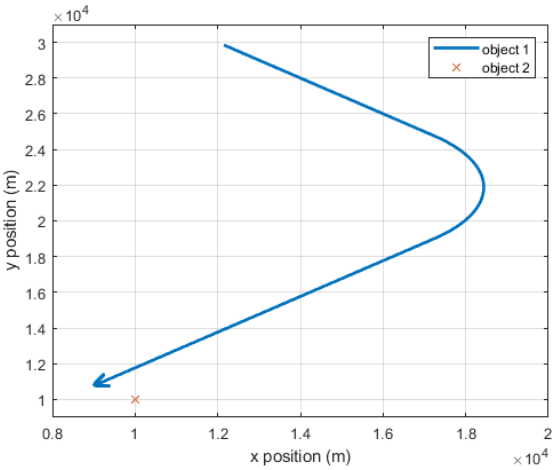

Fig. 6. Trajectories of the objects of scenario B.

TABLE III
SIMULATION PARAMETERS SCENARIO B.

Reference parameter	Value
Maneuverability noise variance object 1(σ^2_{w1})	2.5 $(\frac{m}{s^2})^2$
Maneuverability noise variance object 2(σ^2_{w2})	0.1 $(\frac{m}{s^2})^2$
Radar cross section (RCS)	10 m^2
Prediction horizon length (H)	5
Number of rollouts	10
Simulation update interval	5 s
Budget precision (ε)	0.002

Fig. 7 and 8 show the resulting budget distributions of this scenario for MPC and policy rollout respectively. In this case the budget distribution again are very similar. Furthermore, the resulting realised costs over the simulation time are again

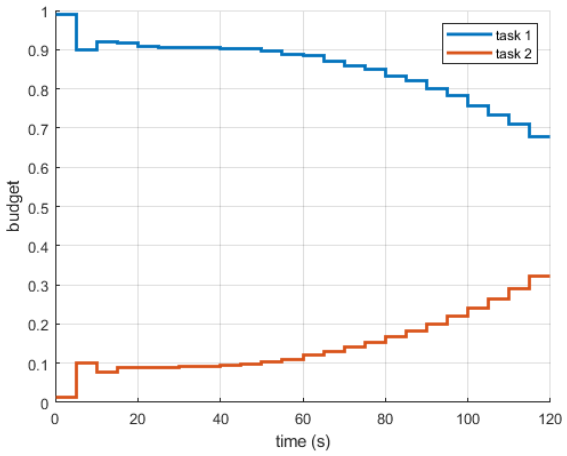

Fig. 7. Budget allocation of the 2 objects described in scenario B using MPC.

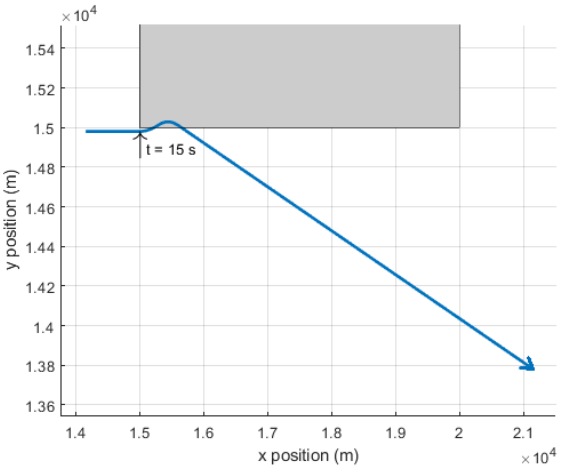

Fig. 9. Trajectory of object 1 of scenario C.

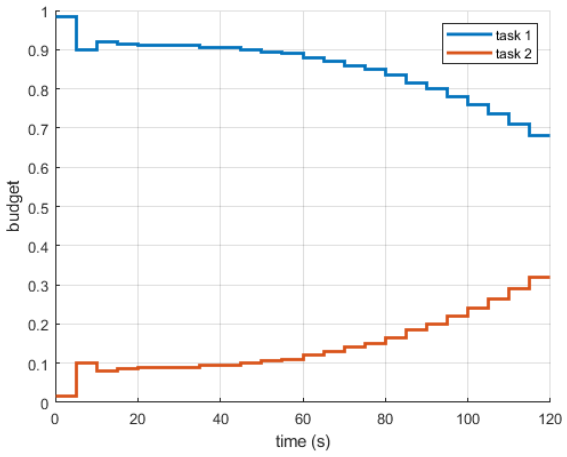

Fig. 8. Budget allocation of the 2 objects described in scenario B using policy rollout.

very close to each other, with the MPC approach having a 0.1% lower realised cost averaged over 10 runs.

C. Simulation Scenario C

Simulation scenario C consists again of two targets. The first target makes an unexpected maneuver at t = 15 s which can be seen in Fig. 9. In the scenario there is an area: $15000 \leq x \leq 20000$, $15000 \leq y \leq 20000$, in which the quality of the measurements are negatively affected due to e.g. weather conditions. If an object enters this area, at least 80% of the budget is needed or else the radar loses track of the target. The second target makes the same maneuver but in the negative x and y quadrant, where there is no such grey area. For the simulation the parameters listed in Table IV were used. The resulting budget distributions for MPC and policy rollout can be seen in Fig. 10 and 11. From these figures it can be seen that policy rollout adapts earlier to the unexpected maneuver, as in some of the rollouts the object maneuvers into the grey area between t = 15 s and t = 20 s, while the MPC approach at that point still assumes that the target continues moving in the same direction, avoiding the grey area. As a result, in the MPC case the track would be lost in this scenario. This downside of MPC only assuming a perfect system model can be negated by using a more robust MPC scheme, similar to [25], where different disturbances in the state are considered. However, this leads to a more computationally intense control law.

TABLE IV
SIMULATION PARAMETERS SCENARIO C.

Reference parameter	Value
Maneuverability noise variance object 1 (σ^2_{w1})	2.5 $(\frac{m}{s^2})^2$
Maneuverability noise variance object 2 (σ^2_{w2})	2.5 $(\frac{m}{s^2})^2$
Radar cross section (RCS)	10 m^2
Prediction horizon length (H)	5
Number of rollouts	10
Simulation update interval	5 s
Budget precision (ε)	0.002

VI. Conclusion

This paper introduces a novel solution approach for the RRM problem in multi-target tracking, which applies MPC on an underlying POMDP. The proposed algorithm solves the initial problem approximately optimally and has proven to be much more computationally efficient than previously applied stochastic optimization methods such as policy rollout.

The applicability of the MPC method has been shown in a simple dynamic tracking scenario with linearly moving targets, as well as in a scenario with a maneuvering target. The budget allocations and cost results of both the policy rollout implementation and the MPC implementation are very similar which highlights that both solutions are valid in the chosen simulation scenarios.

For the same type of scenario, it has been shown that the new MPC approach clearly outperforms the previously considered policy rollout method w.r.t. the computation time.

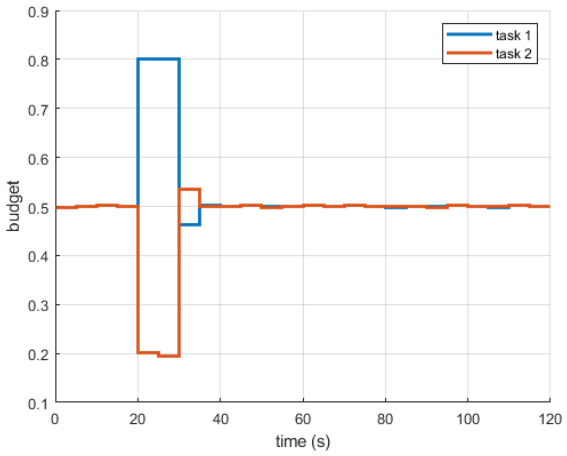

Fig. 10. Budget allocation of the 2 objects described in scenario C using MPC.

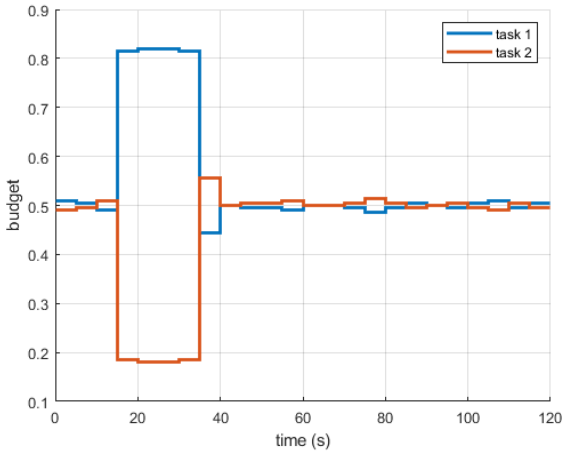

Fig. 11. Budget allocation of the 2 objects described in scenario C using policy rollout.

Nevertheless, it has also been shown that certain scenarios exist where the policy rollout leads to better tracking results than the MPC. This emphasizes that MPC is an approximation of a policy rollout that uses enough samples.

Due to this approximation, the usage of MPC might lead to a reduction in tracking accuracy. However, this was not investigated in this paper.

In future work, the limitations of assuming a single model in the tracking filter and the RRM algorithm need to be investigated. Furthermore, a combination of the current algorithm and an interacting multiple model approach will be explored, which allows the algorithm to switch between different motion models automatically. Finally, it is worthwhile investigating and comparing the tracking accuracy of both the policy rollout and the MPC approach.

REFERENCES

[1] A. Charlish, F. Hoffmann, and I. Schlangen, "The Development from Adaptive to Cognitive Radar Resource Management," *IEEE Aerospace and Electronic Systems Magazine*, vol. 35, 06 2020.

[2] S. Haykin, "Cognitive Radar: a Way of the Future," *IEEE Signal Processing Magazine*, vol. 23, pp. 30–40, Jan 2006.

[3] M. Bockmair, C. Fischer, M. Letsche-Nuesseler, C. Neumann, M. Schikorr, and M. Steck, "Cognitive Radar Principles for Defence and Security Applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 34, pp. 20–29, Dec 2019.

[4] 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*. Scitech Publishing, 2017.

[5] S. Brüggenwirth, M. Warnke, S. Wagner, and K. Barth, "Cognitive radar for classification,"

[6] T. de Boer, "Approximately Optimal Radar Resource Management Using Model Predictive Control," Master's thesis. TU Delft, Delft, The Netherlands, June 2021.

[7] M. I. Schöpe, H. Driessen, and A. Yarovoy, "Multi-Task Sensor Resource Balancing Using Lagrangian Relaxation and Policy Rollout," in *23rd International Conference on Information Fusion*, 2020.

[8] M. I. Schöpe, H. Driessen, and A. Yarovoy, "A Constrained POMDP Formulation and Algorithmic Solution for Radar Resource Management in Multi-Target Tracking," *ISIF Journal of Advances in Information Fusion*, vol. 16, pp. 31–47, June 2021.

[9] A. O. Hero and D. Cochran, "Sensor Management: Past, Present, and Future," *IEEE Sensors Journal*, vol. 11, pp. 3064–3075, Dec 2011.

[10] P. W. Moo and Z. Ding, *Adaptive Radar Resource Management*. London: Academic Press, 1st ed., 2015.

[11] C. M. Kreucher, K. D. Kastella, and A. O. H. III, "Information-based sensor management for multitarget tracking," in *Signal and Data Processing of Small Targets 2003* (O. E. Drummond, ed.), vol. 5204, pp. 480 – 489, International Society for Optics and Photonics, SPIE, 2004.

[12] K. Hintz, *Sensor Management in ISR*. Artech, 2020.

[13] F. Katsilieris, H. Driessen, and A. Yarovoy, "Threat-based sensor management for target tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, pp. 2772–2785, Oct 2015.

[14] S. Martin, "Risk-based sensor resource management for field of view constrained sensors," in *2015 18th International Conference on Information Fusion (Fusion)*, pp. 2041–2048, July 2015.

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

[16] A. Charlish and F. Hoffmann, "Anticipation in Cognitive Radar using Stochastic Control," in *2015 IEEE Radar Conference (RadarCon)*, pp. 1692–1697, May 2015.

[17] V. Krishnamurthy, "POMDP Sensor Scheduling with Adaptive Sampling," in *17th International Conference on Information Fusion (FUSION)*, pp. 1–7, July 2014.

[18] D. A. Castanon, "Approximate Dynamic Programming for Sensor Management," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 2, pp. 1202–1207 vol.2, Dec 1997.

[19] E. K. P. Chong, C. M. Kreucher, and A. O. Hero, "Partially Observable Markov Decision Process Approximations for Adaptive Sensing," *Discrete Event Dynamic Systems*, vol. 19, pp. 377–422, Sep 2009.

[20] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa, "Online Planning Algorithms for POMDPs," *J. Artif. Int. Res.*, vol. 32, pp. 663–704, July 2008.

[21] F. Hoffmann, A. Charlish, M. Ritchie, and H. Griffiths, "Policy Rollout Action Selection in Continuous Domains for Sensor Path Planning," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2021.

[22] D. P. Bertsekas, "Dynamic Programming and Suboptimal Control: A Survey from ADP to MPC*," *European Journal of Control*, vol. 11, no. 4, pp. 310–334, 2005.

[23] S. C. Chapra, *Applied Numerical Methods with MATLAB for Engineers and Scientists*. McGraw-Hill Science/Engineering/Math, 2006.

[24] W. Koch, "Adaptive parameter control for phased-array tracking," in *Signal and Data Processing of Small Targets 1999* (O. E. Drummond, ed.), vol. 3809, pp. 444 – 455, International Society for Optics and Photonics, SPIE, 1999.

[25] P. Scokaert and D. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Autom. Control.*, vol. 43, pp. 1136–1142, 1998.