Approximately Optimal Radar Resource Management for Multi-Sensor Multi-Target Tracking

van der Werk, Bas ; Schöpe, Max Ian; Driessen, Hans

Publication date
2021

Document Version
Final published version

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

Approximately Optimal Radar Resource Management for Multi-Sensor Multi-Target Tracking

Bas van der Werk, Max Ian Schöpe and Hans Driessen
Microwave Sensing, Signals and Systems (MS3)
Delft University of Technology
Delft, the Netherlands
h.b.vanderwerk@student.tudelft.nl, {m.i.schope, j.n.driessen}@tudelft.nl

***Abstract*—Radar Resource Management in a multi-sensor multi-target scenario is considered. A dynamic resource balancing algorithm is proposed which optimizes target task parameters assuming an underlying partially observable Markov decision process (POMDP). By applying stochastic optimization methods, such as policy rollout, the POMDP is solved non-myopically. The approximately optimal approach is formulated assuming a central processor. Subsequently, a distributed implementation is introduced that converges to the same results as given by the centralized implementation and requires less computational resources. The performance of the proposed approach for both centralized and distributed implementation is demonstrated through dynamic tracking scenarios.**

***Index Terms*—Radar Resource Management, Distributed Sensor Network, Constrained Optimization, POMDP**

I. Introduction

Recent developments in multi-function radars (MFR) led to an increasingly flexible usage of these systems. Phased array antennas, digital beamforming, and waveform agility led to new degrees of freedom. As a consequence, modern MFR systems are capable of adjusting automatically to new operational scenarios and tasks during runtime. The control of such automatic adaptation is often called radar resource management (RRM) and is often considered within the framework of cognitive radar (e.g. [1], [2]). By combining the measurements of multiple connected radar sensors placed at different locations, the accuracy, resolution and coverage of the radar system can be improved. An overview of sensor and data fusion algorithms can be found in [3], [4]. Possible applications of such a sensor fusion are for instance weather or space observation, where the information of multiple separate sensors is combined for more accurate results. Instead of optimizing each sensor individually, it would be beneficial to optimize the resources jointly as only this would lead to a globally optimal solution. Possible applications for such an RRM approach can be found, e.g., in automotive scenarios, as well as maritime or air surveillance scenarios. This paper is based on the results of the master's thesis in [5] and focuses on developing an approximately optimal solution approach for a multi-target tracking scenario assuming a sensor network. Although we illustrate our approach in a radar scenario, it is generally applicable to other kinds of adaptive sensors, as long as a limited resource has to be allocated.

A. Radar Resource Management

The majority of research on RRM focuses on single sensor and single target solutions (see e.g. the overviews in [6] and [7]). A possible application is keeping a constant track accuracy when tracking a single object. Since MFR systems can perform multiple tasks while having a limited resource budget available, increasing the resources for one task will automatically decrease them for the others. Therefore, the RRM problem for adaptive sensors needs to be defined as a resource balancing problem.

Many heuristic solutions have been presented, primarily focusing on scheduling tasks with certain fixed resource demand [8]. RRM will have its most significant impact when the resources are assigned to the tasks (such as the sensing time or the time between consecutive measurements) based on their impact on the mission objectives rather than fixed rules. This resource allocation can be achieved by applying optimization techniques, such as Lagrangian relaxation (LR) (see e.g. [9], [10]). In addition to that, POMDPs have been shown to be a good framework for RRM solution methods [11], [12]. POMDPs can be used to describe dynamic environments that can only be observed with noisy measurements. Furthermore, they allow to consider the possible future situation. In [13]–[15] the problem is formulated as a POMDP and solved by using LR and policy rollout. This paper extends this previous approach to a multi-sensor multi-task approach and presents a practical distributed implementation.

B. RRM for Sensor Networks

Many previous approaches to RRM for sensor networks have been focusing on sensor selection without resource balancing (see e.g. [16], [17]) which usually means that the sensor that results in the best measurement is chosen for the task in question. In [18], Charlish and Nadjiasngar implement sensor selection for radar networks and show that a centralized or distributed RRM approach leads to a better performance compared to a completely independent RRM for each sensor.

Bogdanović et al. show a game-theoretic approach in [19] but also do not formulate the problem as a budget balancing problem. Another RRM approach to a network scenario has been presented by Han et al. [20] and aims at decreasing the sensing time of the individual sensors while keeping the sensing performance at some desired level. This is done to free up sensor resources for extra communication tasks. For many applications, such an approach is not desirable since it does not lead to optimal measurement accuracy. In [21], Bell et al. developed an RRM solution that balances the resources for sensor networks. However, that approach does not exploit a non-myopic POMDP framework and only demonstrates results for a single target tracking scenario.

A simple sensor selection ignores a part of the potential of an adaptive sensor network. Therefore, this paper strives to optimize the actions for each sensor while taking into account the global mission, the local sensor constraints, and the expected future situation through the POMDP framework.

C. Novelty in this Paper

So far, a generic non-myopic solution for RRM using LR and policy rollout has not been presented for sensor networks. The strength of this approach lies in the approximately optimal solution of the problem and the practical implementation as a distributed algorithm. It allows each multi-sensor system to use the latest known information from the other sensors and solve a part of the problem independently without permanent communication to the other sensors.

D. Structure of the Paper

The remainder of the paper is structured as follows. Section II describes the considered RRM problem and defines the optimization problem, after which Section III introduces our proposed solution approach. Furthermore, Section IV defines the simulation scenarios and Section V presents the simulation results. Finally, the conclusions can be found in Section VI.

II. Problem Definition

In the remainder of this paper it is assumed that the sensor network consists of M sensors that are responsible for tracking N targets.

A. Motion Model

The state of every considered object is characterized by its position and velocity in x and y at moment k assuming a Cartesian coordinate system. This can be written as,

$$\mathbf{s}_k^n = [x_k^n \quad y_k^n \quad \dot{x}_k^n \quad \dot{y}_k^n]^T \tag{1}$$

where x_k^n, y_k^n, $\dot{x}_k^n$ and $\dot{y}_k^n$ are the position and velocity for target n in x and y respectively. The target state evolves following a certain state transition function,

$$\mathbf{s}_{k+1}^n = f(\mathbf{s}_k^n, \mathbf{w}_k^n) \tag{2}$$

where $\mathbf{w}_k^n$ is the processing noise for target n at time k. The function in (2) directly defines the probability density function of the state evolution:

$$p(\mathbf{s}_{k+1}^n | \mathbf{s}_k^n) \tag{3}$$

B. Measurement Model

It is assumed that the state of the targets cannot be observed directly but rather needs to be determined through noisy measurements using sensor m. A measurement of target n at time k can be defined as,

$$\mathbf{z}_k^{m,n} = h(\mathbf{s}_k^n, \mathbf{v}_k^{m,n}, \mathbf{a}_k^{m,n}) \tag{4}$$

where $\mathbf{v}_k^{m,n}$ is the measurement noise and $\mathbf{a}_k^{m,n}$ is the action that is executed at sensor m for target n at time k. Equivalently to (3), this measurement function directly defines a measurement probability density function given as,

$$p(\mathbf{z}_k^{m,n} | \mathbf{s}_k^n, \mathbf{a}_k^{m,n}) \tag{5}$$

C. Tracking Algorithm

Since this paper focuses on tracking scenarios, a tracking filter needs to be applied. In general, any filter that calculates the posterior density can be applied. For linear systems this can be a Kalman filter, while the extended Kalman filter (EKF) or a particle filter are applicable methods for non-linear systems.

The proposed solution for RRM in a sensor network is based on a centralized fusion approach. It is assumed that each sensor can broadcast information to other sensor nodes. A central processing node exploits the information received from other sensors to compute a fused global estimate.

The choice has been made to utilize measurement fusion instead of other available approaches since it is straightforward to implement and in general does not suffer from dependent estimation errors [22].

Measurement fusion can be defined using a recursive update scheme in which fused estimates of the state $\mathbf{s}_{k|k}^{f_n}$ and covariance $P_{k|k}^{f_n}$ are computed inside the central node. If it is assumed that the measurements of all sensors are available at the same time instant, the recursive update scheme is defined according to Algorithm 1.

Algorithm 1: Measurement fusion in central processor.

Input $\mathbf{P}_{k|k-1}^{f_n} \in \mathbb{R}^{4\times4}$, $\mathbf{s}_{k|k-1}^{f_n} \in \mathbb{R}^{4\times1}$, $\mathbf{R}$, $\mathbf{H}$, $\mathbf{h}$, M
$\mathcal{P} = \mathbf{P}_{k|k-1}^{f_n}$, $\mathcal{S} = \mathbf{s}_{k|k-1}^{f_n}$, $m = 1$
while $m < M$ **do**
  $\mathcal{P} = \text{update}_{covariance}(\mathbf{R}^m, \mathbf{H}^m, \mathcal{P})$
  $\mathcal{S} = \text{update}_{state}(\mathbf{R}^m, \mathbf{H}^m, \mathbf{h}^m, \mathbf{z}^m, \mathcal{P}, \mathcal{S})$
  $m = m + 1$
end
$\mathbf{P}_{k|k}^{f_n} = \mathcal{P}$, $\mathbf{s}_{k|k}^{f_n} = \mathcal{S}$
Return $\mathbf{P}_{k|k}^{f_n} \in \mathbb{R}^{4\times4}, \mathbf{s}_{k|k}^{f_n} \in \mathbb{R}^{4\times1}$

$\mathbf{H}$ is an observation matrix, $\mathbf{h}$ a measurement transformation function, $\mathbf{M}$ the number of sensors and $\mathbf{R}$ the covariance of the observation noise. Both $\mathbf{H}$ and $\mathbf{h}$ depend on the location of a sensor and a target. The superscript f_n indicates the fused data related to target n. Each iteration, the state estimate of target n is updated based on an observation ($\mathbf{z}_m$) made by sensor m and a corresponding estimated error covariance is computed. The resulting fused estimates are then used to compute a prediction of the error covariance and the state.

For the remainder of the paper, the tracking process is based on the sensing information of all sensors. It is assumed that all sensors produce independent measurements which are fused to a global estimate per target. Section III deals with the optimization of the sensing resources, and the use of the term measurement there solely refers to the internal simulation of the expected future in the POMDP.

D. Resource Allocation Optimization Problem

For the resource allocation optimization problem it is assumed that the sensors have a limited budget of any kind and operate at their resource limit. Therefore, the available resources may not be sufficient for all tasks and have to be balanced over them. The goal is to minimize a cost function $\mathbf{C}$ while each sensor m only has a limited resource budget b_m^{max} available. The optimization problem can be formulated as,

$$\begin{aligned} \min_{\mathbf{A}_k} \quad & \mathbf{1}^T\mathbf{C} \\ \text{s.t.} \quad & \underbrace{\begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ b_{21} & b_{22} & & \\ \vdots & & \ddots & \\ b_{M1} & & & b_{MN} \end{bmatrix}}_{\mathbf{B}(\mathbf{A})} \cdot \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \leq \underbrace{\begin{bmatrix} b_1^{max} \\ b_2^{max} \\ \vdots \\ b_M^{max} \end{bmatrix}}_{\mathbf{b}_{max}} \end{aligned} \quad (6)$$

where

$$\mathbf{C} = \begin{bmatrix} \mathcal{C}(\mathbf{A}_k, \mathbf{s}^1) & \mathcal{C}(\mathbf{A}_k, \mathbf{s}^2) & \cdots & \mathcal{C}(\mathbf{A}_k, \mathbf{s}^N) \end{bmatrix}^T \quad (7)$$

represents the cost related to all targets based on a basic cost function $\mathcal{C}$ and $\mathbf{1} \in \mathbb{R}^{p\times 1}$ a vector of all ones. The individual budgets $\mathrm{b}_{m,n}$ inside the resource budget matrix $\mathbf{B}(\mathbf{A})$ represent a percentage of the maximum budget $\mathbf{b}_m^{max}$ spent by sensor m on target n. The optimization variable $\mathbf{A}_k = [\mathbf{a}_k^{1,n}, \cdots, \mathbf{a}_k^{M,n}]$ is a stacked vector representing the actions of all sensors.

III. Proposed approach

This section introduces the proposed approach for solving the RRM problem in a sensor network as explained in Section II. In order to achieve this, the existing solution based on Lagrangian relaxation and policy rollout is extended to a multi-sensor case using a central processor. Finally, a distributed alternative is proposed as a practical implementation.

A. Distribution of Sensor Budgets Using LR

Lagrangian relaxation (LR) is used to include the constraints into the cost function (see e.g. [10]). This allows the original optimization problem to be decoupled into an optimization problem per task. Define the Lagrangian Dual Problem (LDP) as,

$$Z_D = \max_{\boldsymbol{\lambda}} \Big[\min_{\mathbf{A}_k} \textstyle\sum_{n=1}^N \Big(\mathcal{C}(\mathbf{A}_k, \mathbf{s}^n) + \sum_{m=1}^M \lambda_m \cdot b_{m,n} \Big) - \boldsymbol{\lambda}^T \cdot \mathbf{b}_{max} \Big] \quad (8)$$

with $\boldsymbol{\lambda} \in \mathbb{R}^{M\times 1}$ the Lagrangian multiplier and Z_D is the resulting cost. Due to the summation in the LDP, the problem can be solved myopically in parallel during each LR iteration. The separate sub-problems are connected through the Lagrangian multiplier, which is updated using the subgradient method. Generally, no assumptions are being made about the applied cost function which means strong duality cannot be guaranteed. Section III-B will introduce the policy rollout technique to indicate how the problem can be solved non-myopically.

B. Policy Rollout for POMDPs

The system under consideration is modeled according to a POMDP. Inside the POMDP, the tracking process as described in Section II-C is simulated. The POMDP describes a Markov Decision Process (MDP) in which the state cannot be observed directly. Instead an observation is generated that computes a probability distribution over the possible states called the belief state $\mathbf{b}$. Using the belief state and underlying knowledge of the MDP, the POMDP allows the problem to be solved non-myopically by taking into account the expected future.
Similarly as in [14] and [15], the actions for each sensor are found using policy rollout for POMDPs. The technique itself is based on taking Monte Carlo samples of the expected future. For each action $\mathbf{a} \in \mathcal{A}$ with $\mathcal{A}$ the action space, a rollout is evaluated starting from initial belief state $\mathbf{b}_0$ up until horizon $\mathcal{H}$. For a full derivation of the approach one can refer to [15]. Define the expected cost as the summed cost of each rollout averaged over L repetitions. The action corresponding to the lowest cost is selected as best possible action for the next time step. Mathematically, the evaluation for a policy rollout is expressed in terms of a Q-value given by,

$$Q_{H-k}^{\pi_{base}}(\mathbf{b}_k, \mathbf{a}_k) = \tfrac{1}{L} \textstyle\sum_{l=1}^L \Big[C_B(\mathbf{b}_k, \mathbf{a}_k) + \gamma \cdot E[V_{H-k-1}^{\pi_{base}}(\mathbf{b}_{k+1} | \mathbf{b}_k, \mathbf{a}_k)] \Big] \quad (9)$$

with $\gamma \in [0,1]$ the discount factor and π_{base} the applied base policy. C_B refers to the cost based on the believe state and the value function V_H is used to represent the expected future cost.

The best possible policy is found by minimizing the Q-value,

$$\pi_k(\mathbf{b}_k) = \arg\min_{\mathbf{a}_0 \in \mathbf{A}} (Q_{H-k}^{\pi_{base}}(\mathbf{b}_k, \mathbf{a}_k)) \quad (10)$$

Note that the policy rollout aims at improving the chosen base policy π_{base} but does not necessarily lead to the optimal policy. It has been shown that policy rollout leads to a policy that is at least as good as the base policy with a high probability [23].

C. Multi-Sensor implementation

The proposed solution is based on a centralized fusion approach, which is considered as the most optimal way of utilizing measurement fusion. A central processing node is used for allocating budgets over the sensors. Consequently, the representation of the cost C_n related to target n needs to incorporate the actions of multiple sensors i.e.,

$$C_n = \mathcal{C}(\mathbf{a}_k^{1,n}, \mathbf{a}_k^{2,n}, \cdots, \mathbf{a}_k^{M,n}, \mathbf{s}_k^n) \quad (11)$$

The problem can be decomposed into N parallel sub-optimization problems using LR and the optimal actions for M sensors related to the n^{th} target are computed in the central processing node using a global policy rollout.
The centralized implementation utilizes a global policy per task to explore the actions of multiple sensors. Hence, the action space will be $d^{ML} \times \mathbf{a}^n$ with d, the number of actions each sensor can take related to target n and L the number of optimization variables per sensor. Increasing the number of sensors will result in an exponential increase of the action space. Because the central processing node has access to all required information (e.g. measurements) the solution to this problem is regarded optimal, but becomes approximately optimal by applying the policy rollout.

D. Distributed Implementation

A practical distributed alternative is proposed in which the information of each sensor is shared amongst the sensors directly instead of transmitting it to a central node. Now each sensor functions as a processing node to compute fused estimates and to find the best possible actions $\mathbf{a}^{m,n}$ for that specific sensor. The cost C_n related to target n can be decomposed into a summation of costs from multiple sensors,

$$C_n = C_{1,n} + C_{2,n} + \cdots + C_{M,n} \tag{12}$$

where the cost for each sensor m related to task n is defined as,

$$C_{m,n} = \mathcal{C}(\mathbf{a}_k^{m,n}, I, \mathbf{s}_k^n) \tag{13}$$

with I a representation of the information received from all other sensors. Equivalently to the centralized approach, a cost can be calculated for each target track. As the sensor nodes are tracking the targets individually in the distributed approach, $N \times M$ different cost values can be calculated versus N cost values in the centralized approach. Since the centralized and distributed cost formulations are different, Section IV-C describes how they can still be compared.
The distributed optimization problem is defined as,

$$\begin{aligned} \min_{\mathbf{A}_k} \quad & \textstyle\sum_{n=1}^{N} \sum_{m=1}^{M} \mathcal{C}_{m,n} \\ \text{s.t.} \quad & \mathbf{B}(\mathbf{A}_k) \cdot \mathbf{1} \leq \mathbf{b}_{max} \end{aligned} \tag{14}$$

The problem is decomposed into a sub-optimization problem per task per sensor using LR. By doing so, each individual policy rollout only needs to explore the action space related to a single target and a single sensor.

As the policy rollout is making predictions of the expected future for a single sensor, it does not have access to the optimized actions of all other sensors during the resource allocation calculation. To maintain a similar performance as the centralized implementation, at the beginning of each policy rollout for a sensor, the last known actions of the other sensors are used as input. Hence, the information term I is defined as the last known actions of the other sensors.

Thus, during a policy rollout, the actions of a single sensor and a single target are explored while the other sensors are assumed to execute the same action. The additional communication overhead is assumed to be negligible compared to the reduction in computation time required for the policy rollout.

One would expect that over time the distributed implementation converges to similar results as the centralized approach described in the previous section. Nevertheless, the distributed approach could lead to a slightly decreased tracking performance, as it is an approximation of the centralized approach. A high level block diagram is presented in Fig. 1 showing the distributed approach.

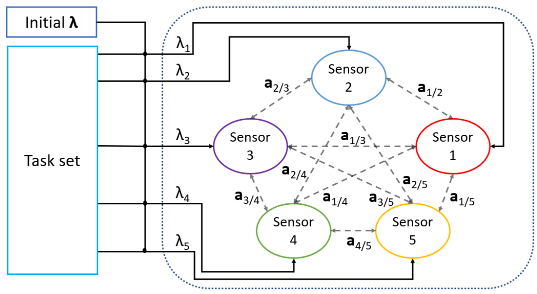


Fig. 1. High level block diagram of the distributed implementation. Each sensor computes their budget allocations via the policy rollout and then shares their last known actions with the other sensors.

IV. Assumed Simulation Scenario

From this point onwards a two-dimensional radar tracking scenario will be assumed. Measurements are taken in range and angle. For every sensor, the algorithm calculates the optimal dwell time i.e., the time spent on each target. Hence, the action vector related to target n is defined as $\mathbf{a}^n \in \mathbb{R}^{M \times 1}$ consisting of the dwell times of each sensor.

The revisit time T i.e., the time between consecutive measurements is assumed to be fixed at one second. In between budget allocation updates, the actions are assumed to remain unchanged.

For all tasks per sensor the budget allocation is computed such that the sum of the tasks fits in the time frame and the resource constraint is met. The rest of this section will present the assumptions made on the assumed radar scenario.

A. Assumed Radar System

Generated measurements in range and angle are picked randomly from a normal distributions $\mathcal{N}(r, \sigma_r^2)$ and $\mathcal{N}(\theta, \sigma_\theta^2)$ respectively. The measurement noise variances in range $\sigma_{r,0}^2$ and angle $\sigma_{\theta,0}^2$ with respect to the measurement of some reference target are given in Table I.

TABLE I
General System Parameters of the Assumed Radar System

Parameter	Value
Noise variance in range $\sigma_{r,0}^2$	$25\,\mathrm{m}^2$
Noise variance in angle $\sigma_{\theta,0}^2$	$4\mathrm{e}^{-4}\,\mathrm{rad}^2$

B. Modelling

1) Target Dynamical Model: It is assumed that each target moves according to a constant velocity model. Using 2, the transition from state k to $k+1$ can now be described according to

$$\mathbf{s}_{k+1}^n = \mathbf{F}\mathbf{s}_k^n + \mathbf{w}_k^n \tag{15}$$

with $\mathbf{w}_k^n$ defined as the process noise and state transition model $\mathbf{F}$ given by,

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

The corresponding process noise covariance $\mathbf{Q}_{w,k}$ for target n is in this case defined as

$$\mathbf{Q}_{w,k} = \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix} \cdot \sigma_{w,k}^2 \tag{17}$$

with $\sigma_{w,k}^2$ the process noise variance of a single target.

2) Observation Model: The sensor generates an observation that describes the range and angle with respect to the target. Due to the non-linear relationship between measurements and states, the EKF is applied to compute the system state.

The observation related to target n received at sensor m is given by,

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

with $\mathbf{v}_k^{m,n}$, the measurement noise for sensor m and target n. Note that the noise related to range and angle are assumed to be independent of each other, i.e.,

$$\mathbf{v}_k^{m,n} = [v_r^{m,n}, v_\theta^{m,n}]^T \tag{19}$$

with corresponding variances σ_r^2 and σ_θ^2. Here, $\mathbf{h}_k^{m,n}(\mathbf{s}_k^n)$ represents the measurement transformation function which transforms the Cartesian measurements into polar measurements defined as,

$$\mathbf{h}_k^{m,n}(\mathbf{s}_k^n) = \begin{bmatrix} \sqrt{(x_k^n - x'_m)^2 + (y_k - y'_m)^2} \\ \text{atan2}(y_k - y'_m, \, x_k^n - x'_m) \end{bmatrix} \tag{20}$$

with x'_m and y'_m the location of sensor m in Cartesian coordinates respectively.

The observation model $\mathbf{H}_k^{m,n} \in \mathbb{R}^{2\times 4}$ for target n and sensor m is defined as the Jacobian of the measurement transformation function $\mathbf{h}$ evaluated at the current state $\mathbf{s}_k^n$ of target n,

$$\mathbf{H}_k^{m,n} = \frac{\delta \mathbf{h}^m}{\delta \mathbf{s}}\Big|_{\mathbf{s}_k^n} \tag{21}$$

3) SNR Model: Computation of the Signal-to-Noise Ratio (SNR) is done according to 22, which is based on the theory provided by Koch in [24].

$$SNR_k = SNR_0 \cdot \frac{\zeta_n}{\zeta_0} \cdot \frac{\tau_n}{\tau_0} \cdot (\frac{r_n}{r_0})^4 \cdot \exp(-2\Delta\alpha) \tag{22}$$

with SNR_0, the SNR for a reference target, ζ_n the radar cross section (RCS) of target n, τ_n the dwell time and r_n the actual range of target n. The values in the denominator are corresponding to the RCS, dwell time and range of a reference target. The term in the exponential is called the relative beam positioning error, which is assumed to be zero.

The variance in range and angle related to the measurement noise can now be defined as,

$$(\sigma_{r/\theta}^2)^{m,n} = \frac{(\sigma_{r/\theta}^2)^{m,0}}{SNR_k} \tag{23}$$

The corresponding measurement covariance matrix is, due to the independent measurements, defined as the following diagonal matrix,

$$R_k^{m,n} = diag(\sigma_r^2, \sigma_\theta^2)^{m,n} \tag{24}$$

σ_r^2 and σ_θ^2 are computed based on a SNR value and measurement noise variances $(\sigma_{r,0}^2, \sigma_{\theta,0}^2)$ that are related to a reference target with parameters defined according to Table II.

TABLE II
PARAMETERS OF REFERENCE MEASUREMENT

Parameter	Value
Reference SNR SNR_0	1
Reference RCS ζ_0	$10\,\mathrm{m}^2$
Reference dwell time τ_0	1 s
Reference range r_0	50 km

C. Cost Function & Constraint

In both the centralized and distributed approach the cost $\mathbf{C}_n$ related to target n is based on the predicted error covariance at time step $k+1$. Define the current predicted error covariance at time step k as

$$\mathbf{P}_{k+1|k}^* = \mathbf{F}\mathbf{P}_{k|k}^*\mathbf{F}^T + \mathbf{Q}_{w,k} \tag{25}$$

Where $\mathbf{F}$ is the state transition matrix, $\mathbf{Q}$ the process noise covariance and $\mathbf{P}_{k|k}^{m,n}$ the estimated error covariance. The $*$ indicates that it can be either the fused predicted error covariance computed in the central processing node or the predicted error covariance computed locally in sensor m depending on the chosen approach.

The cost related to a single time step inside the policy rollout is defined to be the trace of the positional elements of the error covariance.

$$\mathbf{C}_n = [1 \quad 1 \quad 0 \quad 0] \cdot diag(\mathbf{P}_{k+1|k}^*) \tag{26}$$

Note that the formulation of the cost for the centralized and distributed approach are formulated slightly different. Based on the formulation of the optimization objective normalization factors of $\frac{1}{N}$ and $\frac{1}{MN}$ are added to the centralized and distributed approach respectively to be able to compare both methods with each other.

The individual budgets $b_{m,n}$ are defined as the ratio of the dwell time over the revisit time per sensor per target:

$$b_{m,n} = \frac{\tau_{m,n}}{T_{m,n}} \quad (27)$$

V. Simulation Results and Evaluation

In this section, the performance of the distributed implementation is evaluated a radar tracking scenario. In addition to that, a comparison is made based on the resulting cost and computation time.

A. Simulation Parameters

TABLE III
General Simulation Parameters

Parameter	Value
Maximum Budget (B_{max})	1
Budget Update (t_B)	20 s
Number of Budget Updates	40
Beam positional error ($\Delta\alpha$)	0
Probability of Detection	1
Precision of Lagrangian Relaxation (ϵ)	0.05
Action discretization $\mathcal{A}$ = [$\Delta\tau$]	0.01 s
Number of rollouts	4
Horizon Length ($\mathcal{H}$)	15

Table III shows the general simulation parameters used for the simulations. The maximum budget for each sensor is set to 1. The budget allocation is recalculated every 20 seconds, and there is a total of 40 budget updates. In between budget updates, measurements are taken using the current allocated budgets.

For simplicity, the applied base policy is defined as the evaluated action at each step in the policy rollout ($\pi_{base} = \mathbf{a}_k$). The policy rollout has a horizon length of 15 time steps. Each evaluation of the policy rollout is repeated four times and the resulting average is defined as the final result.

The actions are the dwell time allocations τ whi h are selected from a one-dimensional discrete actions space $\mathcal{A}$. The discretization for the dwell time ($\Delta\tau$) is defined to be 0.01 seconds.

B. Dynamic Radar Tracking Scenario

A dynamic radar tracking scenario is considered involving six linearly moving targets. The placement and trajectory of the targets is given in Fig. 2. The initial velocities, RCS and the maneuverability variances assumed in the EKF for targets 1 till 7 are given in Table IV.

A simulation consisting of 40 simulation steps is considered. During each simulation step, a budget allocation is computed for the sensors using the policy rollout based on the distributed solution.

The evolution of the cost of the distributed implementation during the first policy rollout is given in Fig. 3. Both the primal cost $\mathbf{1}^T\mathbf{C}$ and the dual cost Z_D are shown. Note how both the primal and dual cost converge in approximately 25 iterations

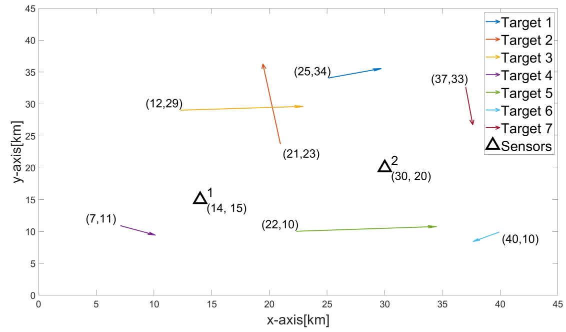


Fig. 2. Visualization of the dynamic radar tracking scenario involving six targets and two sensors. The initial starting locations are shown at the beginning of each trajectory.

TABLE IV
Target Parameters

Target	1	2	3	4	5	6	7
$\mathbf{V}_x$ (m/s)	30	-10	70	20	80	-15	4
$\mathbf{V}_y$ (m/s)	10	85	4	-10	5	-10	-40
ζ (m^2)	50	20	90	80	20	100	40
σ_w^2 (m/s^2)^2	13	24	15	22	17	11	9

to approximately equal values i.e., there is a small duality gap.

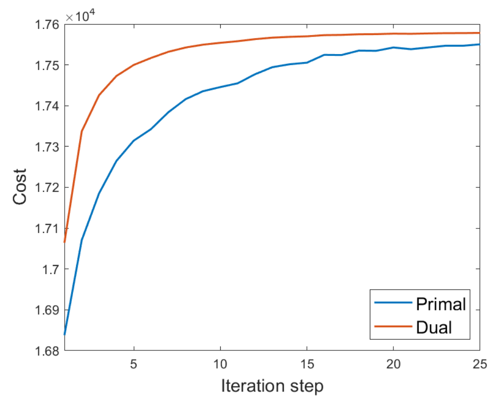
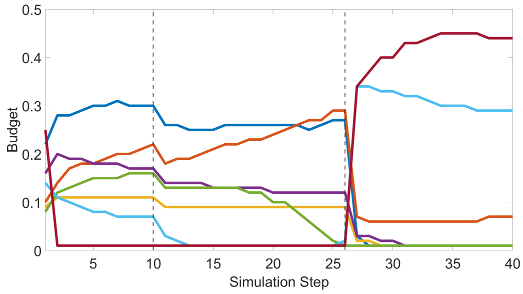


Fig. 3. Evolution of the primal and dual cost of the distributed implementation during the first budget allocation process

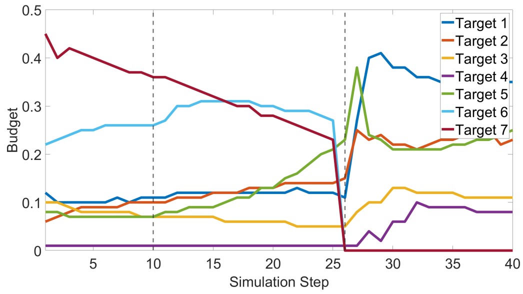
Fig. 4a and Fig. 4b show the resulting budget allocations over multiple time steps for sensor 1 and sensor 2 respectively. Initially, sensor 2 spends a significant amount of budget on target 6 and 7 whereas sensor 1 focuses mostly on targets 1 till 5. Furthermore, the figures indicate that the resources are allocated jointly to both sensors, based on the range-dependent measurement SNR.

At time step k = 10, the maximum budget of sensor 1 is decreased to 0.8 which is also reflected in the corresponding budget allocation plot for sensor 1.

At time step k = 26, sensor 2 cannot track targets 6 and 7 anymore due to e.g. a restriction on the scanning angle. Consequently, it spends more budget on the other targets, while sensor 1 compensates the overall budget by spending more time on target 6 and 7. Hence, allowing communication between sensors helps the network to cope with sudden changes in the number of targets that can be tracked by the sensors.



(a) Budget allocation for sensor 1



(b) Budget allocation for sensor 2

Fig. 4. Budget allocations over multiple simulation steps. At time step k = 10 the maximum budget for sensor 1 is reduced to 0.8. At time step k = 26 sensor 2 can no longer track targets 6 and 7.

C. Comparison Centralized and Distributed Approach

A comparison is made between the approximately optimal centralized approach, distributed approach and a third independent implementation. The independent approach applies multi-sensor tracking as explained in section II-C, but is applying the RRM algorithm from [13], [14] for each sensor individually. The resource allocation therefore does not depend on the presence of the other sensors. A similar comparison has been done in [18] and showed that a centralized or distributed RRM approach leads to improved tracking performance compared to an independent approach.

To compare the three implementations, four dynamic scenarios are considered. Each scenario consists of six linearly moving targets with arbitrary starting locations and two stationary sensors.

For each scenario, ten simulation steps are computed for all three solutions and each scenario is averaged over ten consecutive runs. Fig. 5 shows the average primal and dual cost over the entire simulation for the three implementations. Note that both the centralized and distributed implementation outperform the independent implementation with respect to the primal and dual cost. Table V shows the average runtime of the four scenarios. As expected, due to the exponential increase of the action space for an increasing number of sensors, the

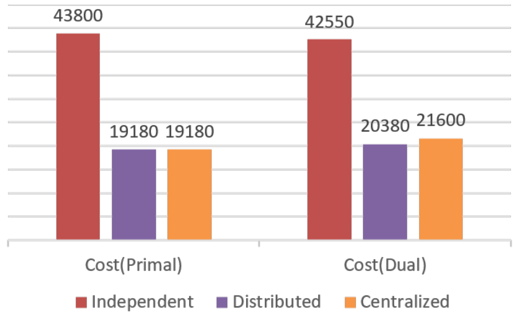


Fig. 5. Cost comparison between the implementations. The results are compared based on the average primal cost (left) and average dual cost (right).

average runtime of the centralized implementation is significantly larger with respect to the other two implementations. The individual tracking error of the distributed approach has not been studied for this paper. A detailed investigation could be an interesting addition for future work.

Interestingly, the average runtime of the distributed implementation is smaller than the independent implementation. This is probably due to the initial pick of the Lagrangian Multiplier.

TABLE V
RUNTIME COMPARISON OF THE THREE IMPLEMENTATIONS

Approach	Runtime in seconds
Independent	158
Distributed	113
Centralized	1187

The costs of the centralized and distributed implementation are approximately equal. This implies that both implementations computed more or less the same budget allocations. To verify this, the percentage difference in average budget allocation is computed for each considered target in one of the comparison scenarios (see Fig. 6). Since the targets will

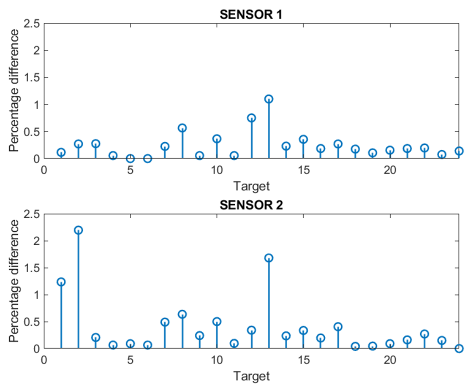


Fig. 6. Percentage difference in dwell times for 24 randomly placed targets between the centralized and distributed implementation in a dynamic scenario

be displaced in between budget allocation updates, the last known dwell times will not exactly correspond to the desired dwell times. Hence, one would expect to see some difference in dwell times between the centralized and distributed solution. However, the maximum difference in dwell times between both solutions are well within the predefined error margin ϵ of 5%.

Therefore, the centralized and distributed approach are approximately equal and converge to similar results over time, if the communication intervals of the sensors are chosen small enough.

VI. Conclusion

A recently developed RRM approach for a single multi-functional sensor has been extended to a novel framework for sensor networks. The different sensor tasks are modeled as a POMDP and the problem is solved non-myopically using a combination of Lagrangian relaxation and policy rollout. The multi-sensor implementation is achieved by exploiting communication between sensors.

Two novel multi-sensor implementations, namely an approximately optimal centralized and distributed implementation for network resource management, have been developed. The distributed implementation of the proposed RRM algorithm has been verified for a two-dimensional radar tracking scenario based on the convergence of the cost and the budget allocation. It was shown that the proposed distributed approach allows for each individual sensor to use the latest known information from the other sensors and solve a part of the problem independently without permanent communication. Moreover, it was shown that the implementation can deal with sudden changes in the tracking process (e.g. change in maximum budget or dropping targets).

Compared to a third approach which optimizes each sensor independently, the centralized and distributed implementation have been shown to perform significantly better.

Furthermore, the distributed implementation outperforms the centralized implementation based on the average runtime. This is due to the fact the distributed implementation optimizes the actions for each sensor separately which leads to a reduction in action space compared to the centralized approach.

By comparing the budget allocation of multiple randomly placed targets, it was shown that the distributed implementation leads to approximately the same resource allocations as the centralized implementation as long as the communication interval is chosen small enough.

In future work, real-world scenarios will be considered to test the derived algorithm. Furthermore, a joint optimization of dwell time and revisit time could be implemented. Additionally, other task types such as searching and classification will be considered. Finally, the selection of the cost function needs to be investigated further.

References

[1] A. Charlish, F. Hoffmann, and I. Schlangen, "The Development from Adaptive to Cognitive Radar Resource Management," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 35, 06 2020.

[2] 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, pp. 20–29, Dec 2019.

[3] M. E. Liggins, Chee-Yee Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proc. IEEE*, vol. 85, no. 1, pp. 95–107, 1997.

[4] C. Chong, "Forty years of distributed estimation: A review of noteworthy developments," in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–10, 2017.

[5] B. van der Werk, "Approximately Optimal Radar Resource Management for Multi-Sensor Multi-Target Tracking," Master's thesis. TU Delft, Delft, The Netherlands, May 2021.

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

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

[8] H. Zhang, J. Xie, B. Zong, W. Lu, and C. Sheng, "Dynamic priority scheduling method for the air-defence phased array radar," *IET Radar Sonar Navig.*, vol. 11, no. 7, pp. 1140–1146, 2017.

[9] J. Wintenby and V. Krishnamurthy, "Hierarchical Resource Management in Adaptive Airborne Surveillance Radars," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, pp. 401–420, April 2006.

[10] 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.

[11] D. A. Castanon, "Approximate Dynamic Programming for Sensor Management," in *36th Proc. IEEE Conf. Decis. Control*, vol. 2, pp. 1202–1207 vol.2, Dec 1997.

[12] 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.

[13] M. I. Schöpe, J. Driessen, and A. Yarovoy, "Optimal Balancing of Multi-Function Radar Budget for Multi-Target Tracking Using Lagrangian Relaxation," in *2019 22th International Conference on Information Fusion (FUSION)*, pp. 1–8, 2019.

[14] M. I. Schöpe, J. Driessen, and A. Yarovoy, "Multi-Task Sensor Resource Balancing Using Lagrangian Relaxation and Policy Rollout," in *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1–8, 2020.

[15] 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.

[16] J. L. Williams, J. W. Fisher, and A. S. Willsky, "Approximate Dynamic Programming for Communication-Constrained Sensor Network Management," *IEEE Trans. Signal Process.*, vol. 55, no. 8, pp. 4300–4311, 2007.

[17] S. Liu, S. P. Chepuri, M. Fardad, E. Maşazade, G. Leus, and P. K. Varshney, "Sensor Selection for Estimation with Correlated Measurement Noise," *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3509–3522, 2016.

[18] A. Charlish and R. Nadjiasngar, "Sensor Management for Radar Networks," in *Novel Radar Techniques and Applications, Volume 2 - Waveform Diversity and Cognitive Radar, and Target Tracking and Data Fusion* (R. Klemm, H. Griffiths, and W. Koch, eds.), pp. 157–193, Scitech Publishing.

[19] N. Bogdanović, J. Driessen, and A. Yarovoy, "Target Selection for Tracking in Multifunction Radar Networks: Nash and Correlated Equilibria," *IEEE Trans. Aerosp. Electron. Syst*, vol. 54, pp. 2448–2462, Oct 2018.

[20] Y. Han, E. Ekici, H. Kremo, and O. Altintas, "Optimal spectrum utilization in joint automotive radar and communication networks," in *2016 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, pp. 1–8, 2016.

[21] K. L. Bell, C. J. Baker, G. E. Smith, J. T. Johnson, and M. Rangaswamy, "Cognitive Radar Framework for Target Detection and Tracking," *IEEE J. Sel. Topics Signal Process.*, vol. 9, no. 8, pp. 1427–1439, 2015.

[22] C. Chong, K. Chang, and S. Mori, "Comparison of optimal distributed estimation and consensus filtering," in *2016 19th International Conference on Information Fusion (FUSION)*, pp. 1034–1041, 2016.

[23] D. Bertsekas and D. Castanon, "Rollout Algorithms for Stochastic Scheduling Problems," *J. Heuristics*, vol. 5, pp. 89–108, 04 1999.

[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.