22nd International Conference on Information Fusion
Ottawa, Canada - July 2-5, 2019

Optimal Balancing of Multi-Function Radar Budget for Multi-Target Tracking Using Lagrangian Relaxation

Max Ian Schöpe, Hans Driessen and Alexander Yarovoy
Microwave Sensing, Signals and Systems (MS3)
Delft University of Technology
Delft, the Netherlands
{m.i.schope, j.n.driessen, a.yarovoy}@tudelft.nl

***Abstract*—The radar resource management problem in a multi-target tracking scenario for multi-function radar is considered. To solve it, an optimal balancing of the sensor budget by applying Lagrangian relaxation and the subgradient method is proposed. In a time-invariant scenario it is shown that the proposed method will lead to balanced budgets based on track parameters like maneuverability and measurement uncertainty. Moreover, since real world applications quickly lead to time-varying scenarios, it is demonstrated how the approach can be extended to such cases. Furthermore the proposed method is compared with other budget assignment strategies. This paper is the first step into exploring optimal non-myopic solutions using a POMDP framework for surveillance radar applications involving detection, tracking and classification.**

***Index Terms*—Radar Resource Management, Lagrangian Relaxation, Steady-State Kalman Filter, Subgradient Method**

I. Introduction

Due to various technological improvements, the degrees of freedom of radar systems have increased significantly in recent decades [1]. The most notable examples of such improvements are the rise of phased-array antennas, digital beamforming (DBF) on transmit and receive, as well as digital waveform generation. This has led to a shift in radar systems from highly specialized systems that focus mostly on a single application towards so-called multi-function radar (MFR) systems that are able to execute multiple functions jointly [2]. Among those functions are surveillance related functions, such as object detection, tracking and classification.

One of the major problems faced by MFR systems during operation are limited resources, especially the sensor time budget. If the radar parameters are being determined online and independently of the other tasks, the total requested time budget often exceeds the available budget. In such a case, drastic load limiting measures will be invoked that may potentially have dramatic effects on system performance. In order to avoid such overload situations, radar resource management (RRM) has to distribute the available budget over all radar tasks in an optimal, operationally relevant manner. In [3], it has been suggested that such an optimal approach could significantly improve the radar performance compared to ad-hoc approaches.

Various approaches to optimally solve RRM problems have already been suggested in the past, but current management schemes still include a lot of heuristics and a truly optimal solution is not yet available. Especially having phased array MFR applications like for instance air/surface surveillance radars in mind, we would like to develop more sound approaches to tackle this problem. Our main interest lies in exploring optimal non-myopic solutions to the RRM problem, taking into account expected future situation changes. Other researchers have already identified that a partially observable Markov decision process (POMDP) might be an appropriate framework for doing so. Some examples can be found in the work of Castañon in [4] and Charlish and Hoffman in [5]. A good overview of possible solution approaches to a POMDP for RRM has been published in [6] by Chong, Kreucher and Hero. Two specifically notable approaches have been published by Wintenby and Krishnamurthy in [7] and White and Williams in [8] which both decouple the main optimization problems into sub-problems per targets by the use of Lagrangian relaxation (LR). Subsequently, dynamic programming is applied to solve a POMDP in a non-myopic fashion. Since the solution of POMDPs can become computationally complex, decoupling the problem into smaller sub-problems promises to significantly reduce the computational load, for instance due to parallelization.

Our long term objective is to develop and evaluate RRM approaches for surveillance, such as object detection, tracking and classification based on the POMDP framework presented in [7] and [8]. We will explore the use of LR to decouple the overall problem into sub-optimization problems and to jointly optimize the radar parameters. The algorithm will automatically balance the individual budgets according to individual target parameters, like its maneuverability for example. Since the objective is quite ambitious, this paper will be a first step into that direction. It will be used to numerically illustrate the potential advantages of such an approach using simulations involving a relatively simple (though definitely not trivial) tracking problem setup. To start with, a one-dimensional scenario is considered. This allows us to exploit the optimal steady-state solution of the Kalman filter, which is an explicit

978-0-9964527-8-6©2019 ISIF

solution of the corresponding POMDP. For now, our attention is concentrated on the problem of balancing radar budget for tracking multiple targets (i.e. excluding target detection and classification).

The remaining paper is structured as follows. The assumed tracking problem is defined in section II as a constrained optimization problem. Subsequently, section III introduces our proposed solution of the RRM problem, using LR and the subgradient method. Furthermore, the results are illustrated by time-invariant and time-variant tracking scenarios in section IV, using the position uncertainty from the predicted steady-state Kalman error covariance matrix as cost function. Finally, the paper is completed with general conclusions in section V.

II. RRM Tracking Problem Definition

In this section, the tracking problem under consideration is described in detail.

The chosen parameters to be optimized are the revisit time T and the dwell time τ. We assume that a certain number of targets are in the observable area around our radar system and are already being tracked. The state of a target in a Cartesian coordinate system is defined as

$$\boldsymbol{x} = [p_x, p_y, v_x, v_y]^T, \tag{1}$$

where p_x and p_y are the target position in x and y respectively. Analogous to that, v_x and v_y define the target velocities in two dimensions.

We assume that the objects to be tracked are moving according to a linear dynamical system. When we assume discrete time steps k, the next state for target i can be predicted as

$$\boldsymbol{x}_i^{k+1} = \boldsymbol{F}(T_i^k) \cdot \boldsymbol{x}_i^k + \boldsymbol{w}_i^k, \tag{2}$$

where $\boldsymbol{x}_i$ is the state of the target and $\boldsymbol{F}(T_i^k) \in \mathbb{R}^{4\times 4}$ is the according state transition matrix which is based on the revisit time T_i^k. Moreover, $\boldsymbol{w}_i^k \in \mathbb{R}^4$ is the maneuverability noise for target i, whose covariance is defined as

$$\begin{aligned} E(\boldsymbol{w}_i^k \boldsymbol{w}_i^{kT}) &= \begin{bmatrix} \frac{(T_i^k)^2}{2} & 0 \\ 0 & \frac{(T_i^k)^2}{2} \\ T_i^k & 0 \\ 0 & T_i^k \end{bmatrix} \begin{bmatrix} \frac{(T_i^k)^2}{2} & 0 & T_i^k & 0 \\ 0 & \frac{(T_i^k)^2}{2} & 0 & T_i^k \end{bmatrix} \sigma_{w,i}^2 \\ &= \begin{bmatrix} \frac{(T_i^k)^4}{4} & 0 & \frac{(T_i^k)^3}{2} & 0 \\ 0 & \frac{(T_i^k)^4}{4} & 0 & \frac{(T_i^k)^3}{2} \\ \frac{(T_i^k)^3}{2} & 0 & (T_i^k)^2 & 0 \\ 0 & \frac{(T_i^k)^3}{2} & 0 & (T_i^k)^2 \end{bmatrix} \sigma_{w,i}^2 \end{aligned} \tag{3}$$

with $\sigma_{w,i}^2$ being the maneuverability noise variance.

The assumed radar system is able to measure the range and the angle of each target. All measurements are considered to be well separated, so there are no association problems. From each of those polar measurements, we can easily obtain the Cartesian measurement $\boldsymbol{z} = [z_x, z_y]^T$. For target i at time k it can therefore be described as

$$\boldsymbol{z}_i^k = \boldsymbol{H} \cdot \boldsymbol{x}_i^k + \boldsymbol{v}_i^k, \tag{4}$$

where $\boldsymbol{H} \in \mathbb{R}^{2\times 4}$ is the measurement matrix and $\boldsymbol{v}_i^k \in \mathbb{R}^2$ the measurement noise. It is assumed that the standard deviation of the latter depends on the dwell time τ as

$$\sigma_{n,i} = \frac{\sigma_{0,i}}{\tau_i}, \tag{5}$$

where $\sigma_{0,i}$ is a predefined basic measurement noise standard deviation for target i.

For the actual tracking, a Kalman filter can be applied, for instance. The problem that we want to solve in this paper is how to optimally assign dwell times and revisit intervals to the different tracks, in order to achieve the best result according to some cost. We formulate the problem as an optimization problem to find the minimum of a cost function $c(T_i, \tau_i)$, constrained by a maximum available budget for all tracking tasks.

The general optimization problem can therefore be described as

$$\begin{aligned} \underset{\boldsymbol{T},\boldsymbol{\tau}}{\text{minimize}} \quad & \sum_{i=1}^{N} c(T_i, \tau_i) \\ \text{subject to} \quad & \sum_{i=1}^{N} \frac{\tau_i}{T_i} \leq B_{max}, \end{aligned} \tag{6}$$

where $N \in \mathbb{Z}^+$ is the amount of tasks (or the amount of targets to be tracked), $\boldsymbol{T} = [T_1, ..., T_N]^T \in \mathbb{R}^N$ are the revisit intervals and $\boldsymbol{\tau} = [\tau_1, ..., \tau_N]^T \in \mathbb{R}^N$ the dwell times for all N targets, $c(T, \tau)$ is the chosen cost function and $B_{max} \in [0, 1]$ is the maximum time budget for all tasks combined.

We define the time budget as the ratio of dwell time and revisit interval. Therefore, this number represents the fraction of the revisit interval that is being used by the dwell time per task. The idea of the global constraint B_{max} is to limit the total time budget of all tasks to a value between 0 (no sensor time used) and 1 (all sensor time used).

III. Proposed Solution of the RRM Tracking Problem

Our solution approach uses the LR technique, which is generally described in appendix A. Following this approach, the original (or primal) optimization problem can be relaxed by adding the constraints as penalty terms to the cost function, which results in the so-called Lagrangian dual. The optimization problem of finding the maximum of the Lagrangian dual over the Lagrange multiplier (also referred to as dual variable), is called Lagrangian dual problem and can be expressed as

$$Z_D = \max_{\lambda} \left(\min_{\boldsymbol{T},\boldsymbol{\tau}} \left(\sum_{i=1}^{N} \left(c(T_i, \tau_i) + \lambda \cdot \frac{\tau_i}{T_i} \right) \right) - \lambda \cdot B_{max} \right). \tag{7}$$

where $\lambda \in \mathbb{R}$ is the Lagrange multiplier for the budget constraint.

It can easily be seen that the resulting Lagrangian dual problem in (7) is just a sum of N sub-optimization problems, one for each task. Therefore, this problem does not have to be solved for all tasks jointly, but can be decoupled into the individual tasks i. Accordingly, we can split up (7) into N easier to solve optimization problems. It is important to realize that λ is a single multiplier for the sum of all sub-optimization problems. We therefore consider it as outer optimization problem which is solved after optimizing the parameters T_i and τ_i per task i. Since those sub-optimization problems can still be quite complicated, they are going to be solved in different stages through an iterative process with steps $l \in \mathbb{Z}^+$. First, an initial value for the Lagrange multiplier λ^l is chosen.

The Lagrangian dual function for each target i is solved with the current Lagrange multiplier value, as shown in (8):

$$\{T_i^l, \tau_i^l\} = \underset{T_i,\tau_i}{\arg\min} \left(c(T_i, \tau_i) + \lambda^l \left(\frac{\tau_i}{T_i} \right) \right). \tag{8}$$

The total budget B_{max} is omitted here, because it is a constant with respect to $\boldsymbol{T}^l$ and $\boldsymbol{\tau}^l$ and does therefore not change the position of the minimum in the Lagrangian. The current optimal values $\boldsymbol{T}^l = [T_1^l, ..., T_N^l]^T$ and $\boldsymbol{\tau}^l = [\tau_1^l, ..., \tau_N^l]^T$ are then used to find the next Lagrange multiplier λ^{l+1}. This is done by the use of the subgradient method, as explained in detail in appendix A. The subgradient for Lagrange multiplier λ^l is chosen as

$$s_\lambda^l = \sum_{j=1}^{N} \frac{\tau_j}{T_j} - B_{max}. \tag{9}$$

The Lagrangian multiplier is then updated with a chosen step size γ^l. Therefore the new Lagrangian multipliers for the next iteration are calculated as shown in (10):

$$\lambda^{l+1} = max\{0, \lambda^l + \gamma^l s_\lambda^l\}. \tag{10}$$

The initial multiplier value λ^0 has to be suitably chosen. Since the budget constraint is an inequality constraint, the value of its Lagrange multiplier can only be positive. With the new value λ^{l+1} the process is started again, until the desired precision of the solution is reached.

IV. One-Dimensional Tracking Scenario

In this paper, a one-dimensional tracking scenario is considered as an example. To conduct the tracking, the Kalman filter is used according to the state and measurement definitions mentioned in (1), (2) and (4). In this section, a possible solution to such a scenario is discussed.

A. Steady-State of the Kalman Filter

If the targets are known already and the Kalman filters are assumed to be staying in a steady-state, the predicted error-covariance matrix of the Kalman filter could be used as the cost function. Being in steady-state means that the measurement uncertainties and maneuverabilities per target are constant and therefore an optimal time-invariant Kalman filter error-covariance can be computed. It can be shown that the α-β-filter is an example for a steady-state Kalman filter [9]. The predicted error-covariance matrix is minimized in order to find the revisit intervals $\boldsymbol{T}$ and the dwell times $\boldsymbol{\tau}$.

For this scenario, we assume that the objects follow a linear dynamical system as described in (1), (2) and (4). The state matrix in the one-dimensional case is therefore defined as

$$\boldsymbol{x} = [p, v]^T, \tag{11}$$

where p is the position of the target and v is the velocity of the target.

A method to calculate the error covariance for a steady-state Kalman filter has been introduced by Kalata in [10] with an extension by Gray and Murray in [11]. In their work, the steady-state equations for an α-β-filter are calculated based on a tracking index Λ, which for a target i is defined as

$$\begin{aligned} \Lambda_i &\propto \frac{\text{position maneuverability uncertainty of target i}}{\text{position measurement uncertainty of target i}} \\ &\hat{=} \frac{T_i^2 \sigma_{w,i}}{\sigma_{n,i}}, \end{aligned} \tag{12}$$

where $\sigma_{n,i}$ is the standard deviation of the measurement noise (see (5)) and $\sigma_{w,i}$ is the standard deviation of the maneuverability noise for target i. Through the tracking index, the filter parameters α and β can be calculated. To simplify the calculations, an extra damping parameter has been introduced in [11], which is defined as

$$\begin{aligned} r_i &= \sqrt{1 - \alpha_i} \\ &= \frac{4 + \Lambda_i - \sqrt{8\Lambda_i + \Lambda_i^2}}{4}. \end{aligned} \tag{13}$$

From this, we can calculate α as

$$\begin{aligned} \alpha_i &= 1 - r_i^2 \\ &= 1 - \left(\frac{4 + \Lambda_i - \sqrt{8\Lambda_i + \Lambda_i^2}}{4} \right) \end{aligned} \tag{14}$$

and β as

$$\beta_i = 2(2 - \alpha_i) - 4\sqrt{1 - \alpha_i}. \tag{15}$$

Based on (14) and (15), the filtered covariance matrix can be formed, as shown by Kalata in [10]. The corresponding matrix is shown in (16).

$$\begin{aligned} \boldsymbol{P}_{k|k}(T_i, \tau_i, \sigma_{0,i}, \sigma_{w,i}) &= \begin{bmatrix} \sigma_{x,i}^2 & \sigma_{xv,i}^2 \\ \sigma_{xv,i}^2 & \sigma_{v,i}^2 \end{bmatrix} \\ &= \begin{bmatrix} \alpha_i \sigma_{n,i}^2 & \frac{\beta_i}{T_i} \sigma_{n,i}^2 \\ \frac{\beta_i}{T_i} \sigma_{n,i}^2 & \frac{(2\alpha_i - \beta_i)\beta_i}{2(1-\alpha_i)T_i^2} \sigma_{n,i}^2 \end{bmatrix}. \end{aligned} \tag{16}$$

Since we are interested in creating a cost function based on the prediction of the error covariance matrix, we follow the approach in [10] and define it for target i according to the Kalman filter prediction equations as

$$\begin{aligned} &\boldsymbol{P}_{k|k-1}(T_i, \tau_i, \sigma_{0,i}, \sigma_{w,i}) \\ &\quad = \boldsymbol{F}(T_i) \boldsymbol{P}_{k|k,i}(T_i, \tau_i, \sigma_{0,i}, \sigma_{w,i}) \boldsymbol{F}^T(T_i) + \boldsymbol{\Psi}_i \boldsymbol{\Psi}_i^T \sigma_{w,i}^2 \\ &\quad = \begin{bmatrix} 1 & T_i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha_i \sigma_{n,i}^2 & \frac{\beta_i}{T_i} \sigma_{n,i}^2 \\ \frac{\beta_i}{T_i} \sigma_{n,i}^2 & \frac{(2\alpha_i - \beta_i)\beta_i}{2(1-\alpha_i)T_i^2} \sigma_{n,i}^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ T_i & 1 \end{bmatrix} \\ &\qquad + \begin{bmatrix} \frac{T_i^2}{2} \\ T \end{bmatrix} \begin{bmatrix} \frac{T_i^2}{2} & T_i \end{bmatrix} \sigma_{w,i}^2, \end{aligned} \tag{17}$$

where $\boldsymbol{F}(T_i)$ is the dynamic matrix for target i.

As a simple cost function for target i, we will use the first element of the predicted error-covariance matrix, which corresponds to the error covariance in range:

$$c_{1,i}(T_i, \tau_i) = \begin{bmatrix} 1 & 0 \end{bmatrix} \boldsymbol{P}_{k|k-1}(T_i, \tau_i, \sigma_{0,i}, \sigma_{w,i}) \begin{bmatrix} 1 \\ 0 \end{bmatrix}. \quad (18)$$

Cost function c_1 will lead to a budget distribution only based on the maneuverability and measurement uncertainty of the different targets.

B. Time-invariant problem

In a time-invariant scenario, the Kalman filters for all objects are assumed to be in a steady-state, which means that both the measurement uncertainties as well as the maneuverabilities are constant. This leads to a constant predicted error-covariance matrix. A single solution can therefore be calculated that is valid for every moment in time.

To illustrate that the LR approach leads to an automatic budget distribution, a simulation is conducted. The budgets are calculated according to the method mentioned in section III, without creating an explicit schedule of the task. Three targets are assumed, starting at different positions from the radar, which is positioned at the origin of the coordinate system. The state of the objects is defined as $\boldsymbol{x} = [p, v]^T$ in a Cartesian coordinate system where p is the position of the target in meters while v is its velocity in meters per second. For the Kalman filter, the dynamic matrix for target i is defined as $\boldsymbol{F}(T_i) = [1, T_i; 0, 1]$, while the measurement matrix is defined as $\boldsymbol{H} = [1, 0]$. As cost function, c_1 is used as defined in (18). The targets have different measurement uncertainties and maneuverabilities to point out that our approach leads to a balancing of the budgets according to those uncertainties, while still taking the constraints into account. A total budget B_{max} of 1 is assumed, which corresponds to using all available sensor time for tracking. The LR step size is set to a constant value. The features of the tracked targets are summarized in Table I, while the general simulation parameters are shown in Table II. The mentioned values have been carefully selected in order to lead to a convergence of the Lagrange multiplier. Other selections can potentially lead to a very slow convergence or even divergence.

TABLE I
INITIAL TARGET PARAMETERS FOR TIME-INVARIANT SCENARIO SIMULATION.

Target	**Position** $[m]$	**Velocity** $[m/s]$	**Measurement variance** $[m^2]$	**Maneuverability** $[m^2]$
1	-1000	10	25	25
2	2000	20	25	250
3	1000	-30	300	25

The simulation results are shown in Figure 1 and in Table III. It can be seen that the LR indeed converges to constant budget values which add up to a total of 1. In every time step of the simulation, the Lagrange multipliers are adjusted, which leads to the Lagrangian approaching the value of the cost function. The process stops after 411 iterations, when the subgradient of the constraint reaches 0 with the desired precision of 0.01. Since the cost function is only based on the measurement uncertainty and the maneuverability, the actual state of the targets has no direct impact on the result. It is obvious that both the differences in maneuverability (see budget difference between tasks 1 and 2), as well as the differences in measurement uncertainty (see budget difference between tasks 1 and 3) lead to quite different sensor budgets for the different tasks.

TABLE II
SIMULATION PARAMETERS FOR TIME-INVARIANT SCENARIO.

Parameter	**Value**
Amount of targets N:	3
Maximum budget B_{max}:	1
Initial Lagrangian multiplier λ_0:	10000
Step size for LR at time step k γ^k:	100
Precision of subgradient solution:	0.01
Cost function:	c_1

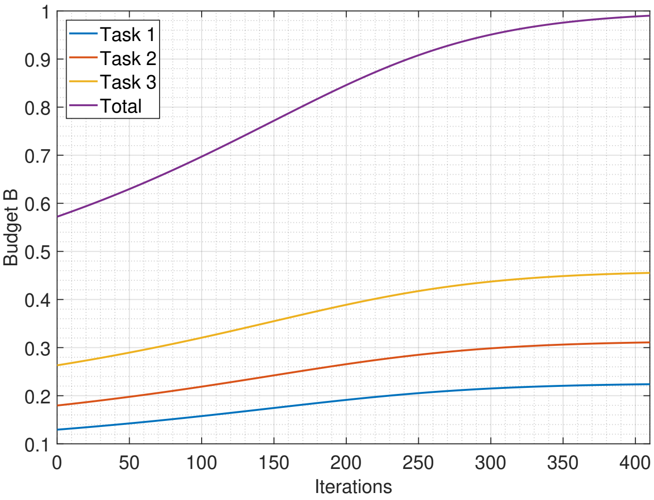

Fig. 1. Simulation results of time-invariant budget allocation using cost function c_1 for three tracked targets.

TABLE III
SIMULATION RESULTS OF TIME-INVARIANT BUDGET ALLOCATION AFTER CONVERGENCE USING COST FUNCTION c_1 FOR THREE TRACKED TASKS.

Target	**Revisit interval** T **[s]**	**Dwell time** τ **[s]**	**Budget**
1	1.01	0.23	0.22
2	0.62	0.19	0.31
3	1.21	0.55	0.46

C. Time-variant problem

In this time-variant scenario, it is assumed that the state of the targets has an influence on the cost. The position and velocity will influence the solutions which will therefore not be valid for the whole future, as assumed in the time-invariant scenario. For that reason, it needs to be updated in certain intervals in which the filters can be assumed to be nearly in a steady-state.

The formulation of the state vector $\boldsymbol{x}$, the dynamic matrix $\boldsymbol{F}$ and the measurement matrix $\boldsymbol{H}$ are the same as in the time-invariant scenario. Since cost function c_1 is only depending on the uncertainty of the measurement and the maneuverability, it will assign resources to targets only according to the uncertainty of their states. In a real application, this will for instance lead to paying more attention to targets that are far away than to closer ones. This is not a very useful cost function formulation, because the threat of an object is directly related to its state. It is therefore obvious that is is very important to carefully formulate the cost function according to the mission needs. For illustration purposes, we extend the above mentioned cost function by a heuristic threat factor $\theta_t(\boldsymbol{x})$. This threat factor is based on the threat formulation as used for example by Katsilieris, Driessen and Yarovoy in [12] and is related to the closest point of approach (CPA). Since the examples presented in this paper are one-dimensional, we limit ourselves to the time to reach the CPA. The CPA is equivalent to the radar location in our case. To convert the time into threat, the same sigmoid function as suggested in [12] is used, with an additional offset of $+0.1$, to avoid a factor of 0. The following parameters for the sigmoid function are applied: $t_1 = 10s$, $t_{0.5} = 20s$ and $t_0 = 30s$. We do not claim that this is the best cost function. Its purpose is to point out how important it is to define a proper cost function and to illustrate the impact of an extra heuristic factor. The cost function c_2 for target i is therefore defined as

$$c_{2,i}(T_i, \tau_i) = \begin{bmatrix}1 & 0\end{bmatrix} \boldsymbol{P}_{k|k-1}(T_i, \tau_i, \sigma_{0,i}, \sigma_{w,i}) \begin{bmatrix}1\\0\end{bmatrix} \theta_t(\boldsymbol{x}_i). \tag{19}$$

The budgets for all targets are updated in a fixed time interval β_t. During this time interval, measurements are conducted according to the calculated dwell times $\boldsymbol{\tau}$ and revisit intervals $\boldsymbol{T}$. Separate Kalman filters are used to track the objects accordingly. All $\boldsymbol{T}$ and $\boldsymbol{\tau}$ stay constant until a new update of the budgets is performed by the use of LR. Two separate simulations with $\beta_t = 5s$ and $\beta_t = 10s$ are conducted. It is assumed that the targets are constantly tracked without track drops or reinitializations.

The LR budget algorithm is fed with the predicted positions given by the Kalman filter based on noisy measurements. The features of the simulated targets are the same as in the previous simulation, see Table I. All other simulation-related values are shown in Table IV and the trajectories of the targets are shown in Figure 2. As in the previous example, the mentioned values in table IV have been carefully selected in order to lead to a convergence of the Lagrange multiplier. Other selections can potentially lead to a very slow convergence or even divergence.

The simulation results are shown in Figures 3 and 4. It can be seen that our LR approach leads to changing budgets over time according to the uncertainty and the threat. When an object is expected to reach the radar position in a comparatively short time, it gets much more attention than the other targets, while the total sum of budgets stays within the constraint.

TABLE IV
SIMULATION PARAMETERS FOR TIME-VARIANT SCENARIO.

Parameter	Value
Amount of targets N:	3
Maximum budget B_{max}:	1
Total simulation time t_{max}:	100s
Simulation step size t_{step}:	0.1s
Budget update interval β_t:	5s and 10s
Initial Lagrangian multiplier λ_0:	10000
Step size for LR at time step k γ^k:	1
Precision of subgradient solution:	0.01
Cost function:	c_2

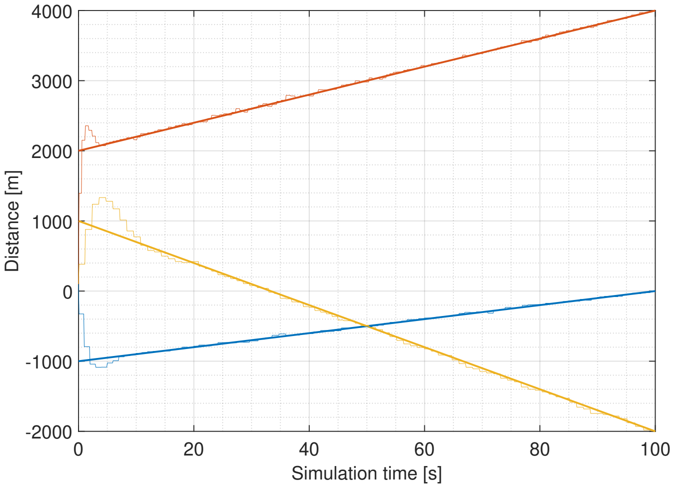

Fig. 2. Target trajectory of the three simulated targets mentioned in Table I. The thin lines show the predicted positions by the Kalman filters. Target 1: blue, target 2: red, target 3: yellow.

Of course, the budget update interval β_t has an impact on the resulting budgets, which can be seen when comparing Figures 3 and 4. If the budgets are updated fast enough, the Kalman filters can be assumed to a stay in a steady-state. If the target states change very quickly, the chosen budget update interval needs to be reduced. It is therefore important to choose this interval properly. Figure 5 shows the resulting cost differences when different fixed budget update periods are compared to a budget update period of $\beta_t = 1s$. Contrary to the previous simulations, the LR budget algorithm is fed with the exact target position in order to remove any uncertainty and to be able to properly compare the cost. It can be seen that the smallest budget update interval always leads to the smallest cost differences. The longer the update interval, the higher the cost compared to the optimal solution.

D. Comparison of time-variant solution with other approaches

To illustrate that this approach leads to improved results, it is compared with different budget allocation techniques according to the cost given by cost function c_2. The simulation parameters are identical with those in Tables III and IV, but the LR budget algorithm is again fed with the exact target positions.

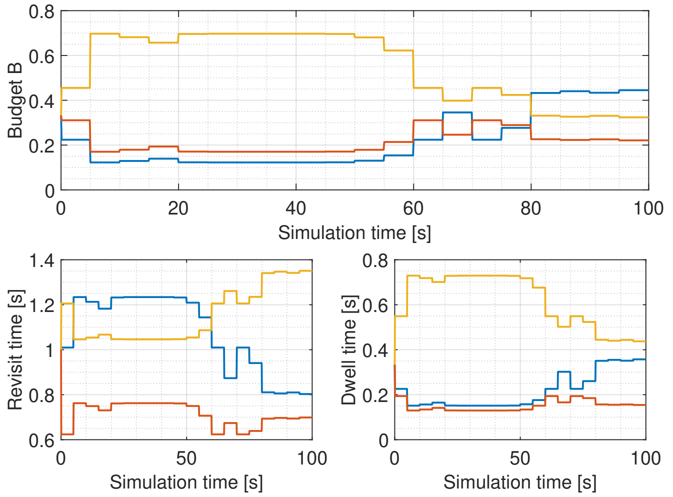

Fig. 3. Simulation results of time-variant budget allocation using cost function c_2 for three tracked targets with a budget update interval of $\beta_t = 5s$. Target 1: blue, target 2: red, target 3: yellow.

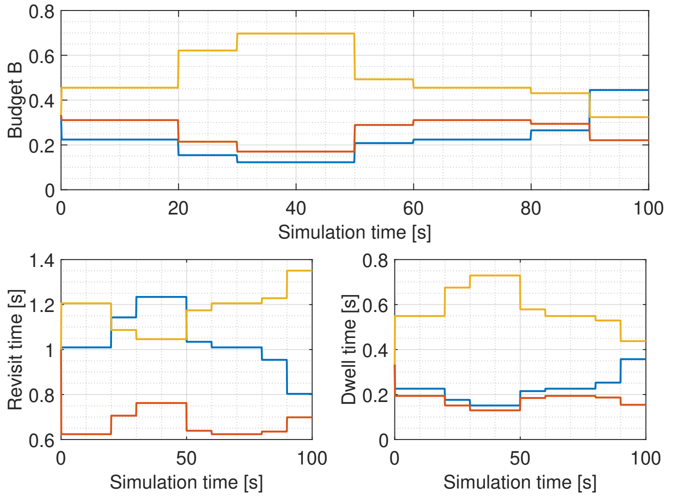

Fig. 4. Simulation results of time-variant budget allocation using cost function c_2 for three tracked targets with a budget update interval of $\beta_t = 10s$. Target 1: blue, target 2: red, target 3: yellow.

The different strategies are

- LR approach with a budget update interval of $\beta_t = 1s$
- Random budget distribution
- Equal budget distribution ($^1/_N$)

A comparison of the results from different budget assignment approaches is shown Figure 6. It can be seen that the LR approach always delivers the lowest cost.

V. Conclusions

In this paper, we have explored advantages of an optimal approach for solving the RRM problem of tracking multiple existing targets. The optimization problem has been decoupled into sub-optimization problems by the use of LR and has then been solved by using the subgradient method. The presented approach is a first step towards the development and evaluation of an optimal approach for solving the RRM problem for

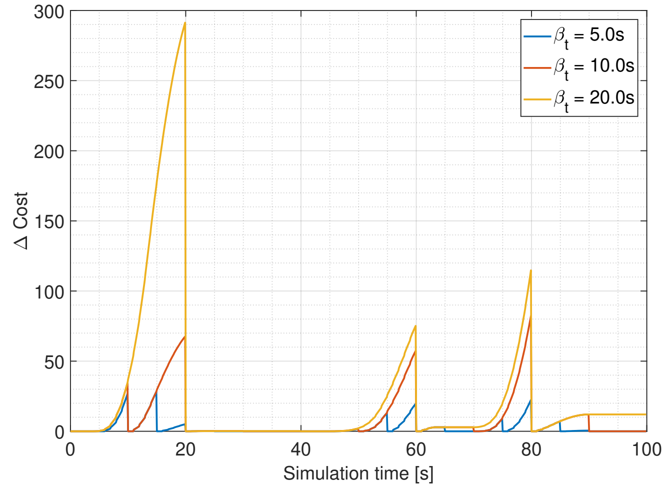

Fig. 5. Simulation results of time-variant budget allocation using cost function c_2 for three tracked targets with different budget update intervals β_t. The difference in cost is with respect to the cost result for $\beta_t = 1s$.

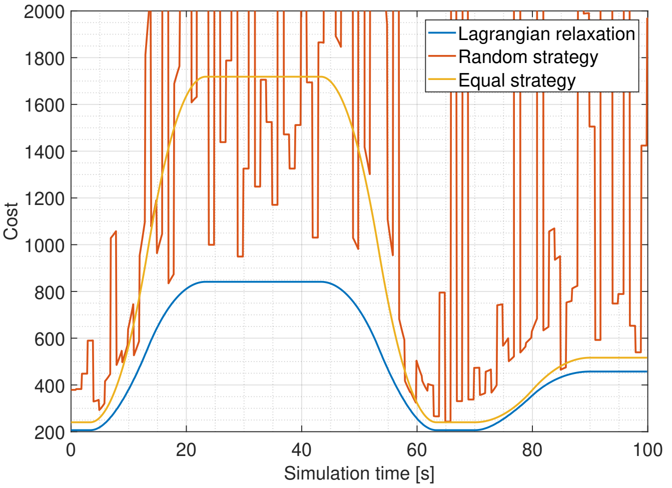

Fig. 6. Comparison of the cost of three different budget assignment strategies. It can be seen that the LR strategy always leads to the lowest total cost.

general surveillance tasks such as target detection, tracking and classification. We applied the proposed solution to a simple one-dimensional tracking scenario with three objects to be tracked, leading to three tasks to be executed by the radar. We have selected the revisit intervals and dwell times as the radar parameters that have to be tuned, where a constraint on the total budget guarantees that the total available budget will not be exceeded. Two different cost functions based on the steady-state Kalman filter error covariance have been taken into account.

The first considered scenario is time-invariant and allows a steady-state analysis. It has been shown that our algorithm leads to a balanced distribution of the available sensor budget over the tasks according to the different maneuverabilities and measurement uncertainties of the targets. We have illustrated that the total budget stays within the predefined budget constraint. In the case of a cost function based on the one-step-ahead prediction error variances, this leads to a larger relative budget for targets with higher maneuverability or larger measurement error variances. The time budget is therefore based on the uncertainty, which is generally desirable. In an operational radar scenario, this would imply that objects at long range will receive more radar budget than objects close-by. This usually is not very useful from an operational point of view and makes it clear that for a future practical implementation, the explicit formulation of operationally relevant cost functions is required.

It has been illustrated that slightly more useful solutions are already generated when applying a heuristic operationally inspired adaptation to the cost function. Since this adapted cost function introduces a time-varying setting, the LR algorithm based on steady-state analysis has been applied to predefined time intervals.

It has been shown that the budget update intervals lead to approximately optimal solutions, as long as they are chosen small enough. In that case, the tracking filters can be assumed to stay in a steady-state during those intervals.

Finally, a comparison of our LR approach with other budget distribution strategies has been presented. It can be seen that our technique indeed always delivers the lowest cost.

In the future, we are going to extend our current approach by improving and accelerating the subgradient method algorithm and extending our simulations to more realistic examples (using two-dimensional position and velocity, as well as range dependent SNR, for instance). Furthermore, we would like to investigate operationally more relevant and time-dependent cost functions and finally extend the approach to other surveillance tasks, such as classification.

Appendix
Lagrangian Relaxation for RRM

By using LR, one can decouple big constrained optimization problems into smaller ones that can be solved independently of each other. This appendix section introduces how this technique can be used in RRM.

A. Lagrangian relaxation principle

LR is an approach to simplify a complicated constrained optimization problem. In this process, constraints can be removed by adding them as penalty terms into the original problem in combination with so-called Lagrange multipliers. As a consequence, a new optimization problem is created that has less constraints than the original problem. The optimization procedure consists of maximizing the minimum of the cost function by adjusting the Lagrange multipliers. This is called the Lagrangian dual problem which is usually a lower estimate of the original problem if the initial Lagrange multipliers are chosen correctly (see for example [13]).

LR and Lagrange multipliers have been extensively covered in literature (for example in [13], [14], [15], [16] or [17]). As an example of how LR is applied, we consider the general optimization problem with N input variables that is shown in (20).

$$\begin{aligned} \underset{\boldsymbol{x}}{\text{minimize}} \quad & f(\boldsymbol{x}) \\ \text{subject to} \quad & \boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{A} \\ & \boldsymbol{h}(\boldsymbol{x}) \geq \boldsymbol{B}, \end{aligned} \tag{20}$$

where

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

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

$$L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\boldsymbol{x}) + \sum_{i=1}^{m} \lambda_i (g_i(\boldsymbol{x}) - A_i) + \sum_{j=1}^{p} \mu_j (B_j - h_j(\boldsymbol{x})). \tag{21}$$

The relaxed problem is called Lagrangian dual function and is defined as

$$d(\boldsymbol{\lambda}, \boldsymbol{\mu}) = \underset{\boldsymbol{x}}{\text{minimize}} \, L(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{22}$$

The Lagrangian dual problem is then characterized as finding the maximum of the Lagrangian dual function with respect to the Lagrange multipliers, as shown in (23).

$$Z_D = \underset{\boldsymbol{\lambda}, \boldsymbol{\mu}}{\text{maximize}} \, d(\boldsymbol{\lambda}, \boldsymbol{\mu}). \tag{23}$$

To summarize, the objective function is minimized over $\boldsymbol{x}$, while also being maximized over the Lagrange multipliers, in order to come as close to the original problem as possible. To find the optimal Lagrange multipliers and therefore the tightest lower bound to the original problem, iterative approaches can be used. There are many techniques available to calculate the Lagrange multipliers iteratively, like the commonly used subgradient method. It will be explained in the next subsection.

B. Subgradient method

The subgradient method (see for example [13]) is an iterative process that starts with a chosen initial value for the Lagrange multipliers (e.g. 1). At each iteration k, first the minimum of the relaxed problem is calculated (Lagrangian dual function, see (21)). Next, the subgradients are chosen for each constraint as $\boldsymbol{s}_\lambda^k = [s_{\lambda,1}^k, ..., s_{\lambda,m}^k]^T \in \mathbb{R}^m$ and $\boldsymbol{s}_\mu^k = [s_{\mu,1}^k, ..., s_{\mu,p}^k]^T \in \mathbb{R}^p$. Following the notation that has been used above for the constraints, the subgradients are defined as

$$\begin{aligned} \boldsymbol{s}_\lambda^k &= (\boldsymbol{g}(\boldsymbol{x}^k) - \boldsymbol{A}) \\ \boldsymbol{s}_\mu^k &= (\boldsymbol{B} - \boldsymbol{h}(\boldsymbol{x}^k)). \end{aligned} \tag{24}$$

The Lagrange multipliers are then updated with a specific step size γ^k. For inequality constraints, the penalty terms are not allowed to become negative. Therefore the new Lagrange multipliers for the next iteration are calculated as shown in (25):

$$\begin{aligned}\boldsymbol{\lambda}^{k+1} &= max\{0, \boldsymbol{\lambda}^k + \gamma^k \boldsymbol{s}_\lambda^k\} \\ \boldsymbol{\mu}^{k+1} &= max\{0, \boldsymbol{\mu}^k + \gamma^k \boldsymbol{s}_\mu^k\}.\end{aligned} \tag{25}$$

The step size can be chosen freely. A possibility are constant or decreasing step sizes like γ_0/k or $1/\gamma^k$, for example. The process is then started again with the new Lagrange multipliers. A new Lagrangian dual function is found and afterwards, new subgradients are calculated again. Theoretically, the exact result has been found when the gradients $\boldsymbol{s}_\lambda^k$ and $\boldsymbol{s}_\mu^k$ reach $\mathbf{0}$. Since this value will never be reached exactly, the process is repeated until the gradient reaches 0 with a 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 (24)).
4) Check if $\boldsymbol{s}_\lambda^k \approx \mathbf{0}$ and $\boldsymbol{s}_\mu^k \approx \mathbf{0}$ with desired precision. If it is, stop the process.
5) Adjust Lagrangian multipliers as shown in (25).
6) Go to step 2 and set $k = k + 1$.

References

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

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

[3] A. O. Hero, D. A. Castan, D. Cochran, and K. Kastella, *Foundations and Applications of Sensor Management*. New York, NY: Springer Publishing Company, Incorporated, 1st ed., 2008.

[4] 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.

[5] A. Charlish and F. Hoffmann, "Anticipation in cognitive radar using stochastic control," in *2015 IEEE Radar Conference (RadarCon)*, pp. 1692–1697, May 2015.

[6] 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.

[7] J. Wintenby and V. Krishnamurthy, "Hierarchical resource management in adaptive airborne surveillance radars," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, pp. 401–420, April 2006.

[8] K. A. B. White and J. L. Williams, "Lagrangian relaxation approaches to closed loop scheduling of track updates," 2012.

[9] C. K. Chui and G. Chen, *Kalman Filtering with Real-Time Applications*. Springer-Verlag Berlin Heidelberg, 4th ed., 2009.

[10] P. R. Kalata, "The Tracking Index: A Generalized Parameter for alpha-beta and alpha-beta-gamma Target Trackers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, pp. 174–182, March 1984.

[11] J. E. Gray and W. Murray, "A derivation of an analytic expression for the tracking index for the alpha-beta-gamma filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 29, pp. 1064–1065, Jul 1993.

[12] F. Katsilieris, H. Driessen, and A. Yarovoy, "Threat-based sensor management for joint target tracking and classification," in *2015 18th International Conference on Information Fusion (Fusion)*, pp. 435–442, July 2015.

[13] D. Bertsimas and J. N. Tsitsiklis, *Introduction to Linear Optimization*. Athena Scientific, 1st ed., 1997.

[14] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1st ed., 1996.

[15] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 1st ed., 2004.

[16] S. S. Blackman and R. Popoli, *Design and analysis of modern tracking systems*. London :: Artech House,, 1999.

[17] F. L. Lewis, D. L. Vrabie, and V. L. Syrmos, *Optimal Control*. John Wiley & Sons, Inc., 3rd ed., 2012.